جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH DE FES

كلية العلوم والتقنيات فاس
FACULTÉ DES SCIENCES ET TECHNIQUES DE FÈS

Ordre N°: 80/2023

Center of Doctoral Studies: Sciences and Techniques and Medical Sciences
Doctoral program: Engineering Sciences, Physical Sciences, Mathematics, and Computer Science
Discipline: Computer Science
Specialization: Computer Science
Laboratory: Intelligent Systems and Applications

Thesis for DOCTORATE degree

Segmentation and Recognition of Handwritten Arabic Text

Prepared by : Mr ZOIZOU ABDELHAY

Defended on November 04, 2023

Jury Members:

Full Name	Rank	Establishment	Quality
Pr. Azeddine ZAHI	PES	FST FES	President
Pr. Mohamed EL ANSARI	PES	FS Meknès	Reviewer & examiner
Pr. Aicha MAJDA	PES	FSJES Meknès	Reviewer & examiner
Pr. Khalid ZENKOUAR	PES	FST FES	Reviewer & examiner
Pr. Khalid ABBAD	PH	FST FES	Examiner
Pr. Ilham CHAKER	PH	FST FES	Examiner
Pr. Arsalane ZARGHILI	PES	FST FES	Thesis Director

Résumé de la thèse

Un système de reconnaissance optique de caractères (OCR) est un système qui permet aux ordinateurs de lire et de traiter le texte à partir d'images. Il convertit le texte en format éditable et consultable en analysant les formes et les motifs des caractères. L'OCR trouve une large application dans la numérisation de documents et permet la reconnaissance de texte dans divers systèmes pour différentes langues.

L'OCR pour le texte cursif arabe est d'une importance significative car il permet la numérisation et la recherche de contenu en arabe, facilitant ainsi la compréhension et l'analyse du texte. L'OCR de l'arabe implique plusieurs étapes, dont la segmentation qui est l'une des plus cruciales. Le processus de segmentation passe par trois étapes principales, commençant par l'extraction des lignes, puis leur décomposition en mots ou sous-mots, qui sont finalement segmentés en caractères. De nombreuses techniques ont été développées pour relever ces défis. Cependant, les travaux trouvés dans la littérature ont du mal à traiter les textes comportant différentes polices. De plus, l'évaluation de ces systèmes manquent de clarté et de transparence, car chacun est évalué sur un ensemble de données différents. Dans cette thèse, nous relevons ces défis en proposant une méthode de segmentation du texte arabe imprimé multi-polices. Nous proposons également un corpus standard pour l'évaluation et la comparaison des systèmes de segmentation du texte arabe imprimé.

Dans un autre contexte, étant donné le grand nombre de documents historiques marocains stockés dans les archives du Maroc et d'autres pays, nous avons identifié un besoin urgent d'un OCR conçu spécifiquement pour les textes arabes anciens. En effet, l'OCR pour les textes arabes anciens revêt une importance cruciale pour explorer plus efficacement le patrimoine culturel en numérisant et en rendant ces textes accessibles. Cependant, le développement d'un tel OCR est également confronté à de sérieux défis, allant de la complexité du prétraitement et de la segmentation au manque de sources de données. Dans le processus d'OCR, l'extraction des lignes et des mots joue un rôle crucial. Cette opération devient difficile en raison des variations dans l'écriture. Un autre obstacle qui entrave la recherche sur l'OCR de l'arabe historique est le manque de bases de données contenant des mots arabes anciens pour l'entraînement des modèles de reconnaissance. La création d'une telle base de données est vitale, car elle permet d'entraîner les modèles d'OCR sur une large gamme de mots arabes anciens. Dans cette thèse, nous proposons des méthodes d'extraction des lignes et des sous-mots à partir de documents historiques rédigés dans le style marocain Al-Mojawhar, ainsi qu'une grande base de données de sous-mots extraits de documents anciens. Nous étudions également la reconnaissance des caractères Al-Mojawhar, en appliquant des différents descripteurs et classifieurs. Au cours de cette thèse, nous avons mené des expériences approfondies pour démontrer la robustesse de nos méthodes et de nos bases de données par rapport aux méthodes de pointe. Il est indéniable que nos méthodes ont surpassé les autres en termes de précision exceptionnelle.

Mots-clés : Reconnaissance optique de caractères, texte Arabe, segmentation du texte, numérisation de documents historiques, construction de bases de données.

Abstract

Optical Character Recognition (OCR), is a system that allows computers read and process text from images. It converts the text into an editable and searchable format by analyzing the shapes and patterns of characters. OCR finds wide application in digitizing documents, and enabling text recognition in various systems for different languages.

OCR for the Arabic cursive text is of a significant importance as it enables the digitization and searchability of Arabic content, facilitating text understanding and text analysis. Arabic OCR involves multiple steps, with segmentation being a crucial one. The process of segmentation goes through three main steps, starting by extracting lines, then decomposing them into words or subwords which are finally segmented into characters. The segmentation of Arabic text presents many challenges due to the script cursiveness, and variations in writing styles. Many techniques have been developed to address these challenges. However, the works found in literatures find it difficult to handle multi-font text. Moreover, the evaluation and comparison of these systems lacks clarity and transparency, as each one is evaluated on different datasets. In this thesis we address these challenges by proposing a segmentation method of multi-font printed Arabic text. We also propose a standard corpus for the evaluation and comparison of Arabic segmentation systems.

In other context, acknowledging the high numbers of Moroccan historical documents stored in archives of Morocco and other countries, we realized the urgent need for an OCR designed for ancient Arabic text. In fact, OCR for ancient Arabic text is essential for exploring the cultural heritage more efficiently by digitizing and making these texts available. However, developing such an OCR also faces serious challenges, from the complexity of pre-processing and segmentation to the lack of data sources. Certainly, within the OCR process, line and word extraction plays a crucial role before recognition. Due to the writing complexity, it becomes difficult to extract classifiable words. Another impediment causing the delay of research on historical Arabic OCR is the lack of databases containing ancient Arabic words for training recognition models. Creating such a database is vital as it allows OCR models to be trained on a wide range of ancient Arabic word. In this thesis we propose new methods of line and subwords extraction from historical documents written with the Moroccan Al-Mojawhar style, and a database of subwords extracted from ancient documents. We also investigate the recognition of Al-Mojawhar characters by studying the feature extraction and classification of these characters. We conducted intensive experiments to prove the robustness of our methods and databases against the state-of-the-art ones. Certainly, our methods outperformed the other ones, showing exceptional accuracies.

Keywords: Optical Character Recognition, Arabic text, Text segmentation, Historical document digitization, Database construction

وَقُل رَّبِّ زِدْنِي عِلْمًا (طه: 114)

Dedication

To My Father

Acknowledgements

In reaching the culmination of this academic endeavor, I am humbled to express my heartfelt appreciation to the individuals who have played an indispensable role in making this journey possible. Their contributions, both seen and unseen, have left an indelible mark on my academic and personal growth.

To my esteemed supervisors, Arsalane Zarghili and Ilham Chaker, thank you for your mentorship, expertise, and continuous encouragement throughout this research journey. Your valuable insights and dedication to my growth as a researcher have been truly invaluable.

I am deeply indebted to the members of the thesis committee, professors Pr. Azeddine Zahi, Pr. Khalid Zenkouar, Pr. Khalid Abbad, Pr. Aicha Majda, and Pr. Mohamed El Ansari, for their invaluable feedback. Your meticulous evaluation has significantly contributed to the refinement of this work.

To my family, my gratitude knows no bounds. Thank you for your constant love, understanding, and sacrifices. Dad, your unchanged belief in me and your constant encouragement have been the driving force behind my perseverance. I owe this achievement to you in a way words cannot express. Mom, your boundless love and support have been my guiding light throughout this journey. This achievement is a testament to your love and sacrifices, which have shaped me into the person I am today.

To my steadfast companion on this journey, many years have woven a beautiful tapestry of shared experiences. Your unwavering presence has been my shelter, a beacon of strength and understanding that has seen me through the darkest nights.

I extend my sincere appreciation to all the professors and administration staff at Faculty of Science and Technology Fez for their dedication to education and for imparting their knowledge to countless students, including myself.

To my friends, colleagues and the family of HopeClub: Saliha, Ilham, Saad, Hicham, Moulay and the others, thank you for being my pillars of support. Your camaraderie, encouragement, and shared experiences have made this academic pursuit not only successful but also memorable and enjoyable.

Lastly, I want to express my gratitude to all those whose names may not be mentioned here but have been instrumental in my academic and personal development. Your presence in my life has made a significant impact, and I am deeply thankful for your support.

Remerciements

En atteignant le terme de cet effort académique, je suis honoré(e) d'exprimer ma profonde reconnaissance envers les personnes remarquables qui ont joué un rôle indispensable dans la réalisation de ce parcours. Leurs contributions, à la fois visibles et invisibles, ont laissé une empreinte indélébile sur ma croissance académique et personnelle.

Je dédie ce travail à mes estimés superviseurs, Arsalane Zarghili et Ilham Chaker, je vous remercie pour votre mentorat, votre expertise et votre encouragement constant tout au long de ce parcours de recherche. Vos précieuses idées et votre dévouement à mon développement en tant que chercheur ont été vraiment inestimables.

Je suis profondément redevable aux membres du comité de thèse, les professeurs Pr. Azeddine Zahi, Pr. Khalid Zenkouar, Pr. Khalid Abbad, Pr. Aicha Majda et Pr. Mohamed El Ansari, pour leurs précieux commentaires et leurs critiques constructives. Votre expertise et votre évaluation minutieuse ont largement contribué à l'amélioration de ce travail.

À la famille de Zoizou, ma gratitude n'a pas de limites. Je vous remercie pour votre amour, votre compréhension et vos sacrifices. Papa, votre croyance en moi et votre encouragement constant ont été la force derrière ma persévérance. Je vous dois cette réussite d'une manière que les mots ne peuvent exprimer. Maman, ton soutien infinis ont été ma lumière tout au long de mon parcours. Cette réussite témoigne de ton amour et de tes sacrifices qui m'ont façonné(e) en la personne que je suis aujourd'hui.

J'exprime ma sincère appréciation à tous les professeurs de la Faculté des Sciences et Techniques de Fès pour leur dévouement à l'éducation et pour avoir transmis leur savoir à d'innombrables étudiants, dont moi-même. Votre engagement à favoriser une culture de l'apprentissage a été une source d'inspiration.

À mes amis, mes collègues, Ilham, Saad, Moulay, Saliha et la famille du Club Espoir merci d'avoir été mes piliers de soutien. Votre camaraderie, votre encouragement et vos expériences partagées ont rendu cette poursuite académique non seulement réussie, mais aussi mémorable et agréable.

Enfin, je tiens à exprimer ma gratitude à toutes les personnes dont les noms ne sont peut-être pas mentionnés ici, mais qui ont joué un rôle déterminant dans ma croissance académique et personnelle. Votre présence dans ma vie a eu un impact significatif, et je vous suis profondément reconnaissant pour votre soutien. Cette thèse témoigne des efforts collectifs et de l'encouragement de ces personnes remarquables. Chaque mot écrit et chaque idée explorée est le reflet du réseau inspirant de personnes qui ont fait partie de ce voyage transformatif.

Table of content

List of Figures ii
List of tables iv
List of abbreviations v
General Introduction 6
1. Optical character recognition 7
2. Arabic Optical character recognition 8
2.1. Arabic writing 8
2.2. Arabic writing characteristics 8
2.3. Arabic Optical Character Recognition 11
2.3.1. Pre-processing 11
2.3.2. Segmentation 12
2.3.3. Feature extraction 12
2.3.4. Classification 12
2.3.5. Post-processing 13
3. Printed Arabic text segmentation 13
4. Historical documents digitization 14
4.1. Motivations 14
4.2. Al-Mojawhar writing style 14
4.3. Historical text recognition 15
5. Thesis Presentation 16
5.1. Problematics 17
5.2. Contributions 17
5.3. Publications 18
5.4. Thesis organization 18
Chapter 1. Printed Arabic text segmentation 20
Introduction 21
1. State of the art of Arabic text segmentation 22
1.1. Segmentation of pages into lines 22
1.2. Segmentation of lines into words and subwords 23
1.3. Word segmentation 24
1.3.1. Baseline detection 24
1.3.2. Projection-based word segmentation 25
1.3.3. Word segmentation based on Skeletonization 25
1.3.4. Contour-based segmentation 26
1.3.5. Word Segmentation based on Template-Matching 28

1.4. Evaluation of segmentation methods 29
2. Study of printed word segmentation methods 29
2.1. Template-Matching 30
2.2. Projection-based segmentation 31
2.3. Contour based segmentation 32
3. Contribution to the Segmentation of Arabic printed text 33
3.1. Pre-Processing 34
3.2. Segmentation of printed Arabic words 35
3.2.1. Segmentation of descendant characters 35
3.2.2. Contour-based segmentation 36
3.3. Corpus construction 41
4. Experimental Results 43
Conclusion 47
Chapter 2. Text line extraction of Arabic historical documents 48
Introduction 49
1. State of the art of line extraction from Handwritten Arabic text 49
2. The proposed method of Skew Correction and Text Line Extraction from historical documents 51
2.1. Pre-Processing 52
2.2. Skew correction 54
2.3. Image transformations 56
2.4. Line extraction 57
3. Experimental study 59
3.1. Skew Correction 60
3.2. Text line segmentation 61
Conclusion 62
Chapter 3. Building MOJ-DB database with new approach of Subword extraction from historical document. 64
Introduction 65
1. Literature review of subword extraction and Handwritten Arabic Databases 65
1.1. Arabic handwriting text line segmentation 65
1.2. Arabic handwriting databases 67
2. The proposed method of subwords extraction from ancient Arabic documents 69
3. Database construction 74
3.1. Data collection 74
3.2. Data pre-processing 75
3.3. Subword extraction 75
3.4. Database Ground truth 76

3.5. Data balancing 77
4. Experimental study 77
4.1. Subwords extraction 77
4.2. Database validation 79
4.2.1. Handcrafted features 79
4.2.2. Learned features 79
4.2.3. Database comparison 80
4.3. Discussion 81
4.3.1. Subword extraction 81
4.3.2. Database construction 81
Conclusion 82
Chapter 4 . Comparison of feature extraction method for Al-Mojawhar characters recognition 83
Introduction 84
1. State of the art of feature extraction and classification methods of Handwritten Arabic Character 85
2. Study of feature extraction and classification methods for Al-Mojawhar characters 87
2.1. Feature extraction methods 88
2.2. Classification methods 89
3. Experimental study 92
3.1. Handcrafted features classification 93
3.2. Learned features classification 94
4. Discussion of results 95
Conclusion 96
Conclusions and Perspectives 98
1. Conclusions 99
2. Perspectives and future works 100
References 101

List of Figures

Figure 1 : Types of Optical character recognition systems 7
Figure 2 : (a) Word composed of 3 subwords (b) word composed of isolated characters 9
Figure 3 : (a) a word without ligature (b) the same word with ligature 10
Figure 4 : (a) Example of characters with dots (b): the same letter with different diacritics 10
Figure 5 : The baseline of an Arabic text 10
Figure 6 : The same text in : a) handwritten, b) printed 11
Figure 7 : The complete process of an Arabic analytic OCR 11
Figure 8 : Example of the raw document 15
Figure 9 : Overlapping and touching lines 16
Figure 10 : Levels of text segmentation 22
Figure 11 : Detection of Horizontal Projection profile 23
Figure 12 : Example of the segmentation of a line of text into subwords by the vertical projection 23
Figure 13 : Example of word Skeletonization 26
Figure 14 : Example of the location of segmentation points (arrow) 27
Figure 15 : The angles of junction between the characters 28
Figure 16 : Connection of the characters and the baseline 30
Figure 17 : The template used for segmentation 30
Figure 18 : Result of the segmentation by Template-Matching 30
Figure 19 : Special cases of Template Matching failure 31
Figure 20 : Word segmentation using vertical projection 32
Figure 21 : The proposed system of printed Arabic character segmentation 34
Figure 22 : (a) the original image. (b) the binary image. (c) the image after the morphological closing 35
Figure 23 : Descendant characters 36
Figure 24 : (a) Template used, (b) segmentation result of descendant characters 36
Figure 25 : Freeman's eight directions 37
Figure 26 : The upper and lower contour of an Arabic word 38
Figure 27 : Extraction of the upper contour 38
Figure 28 : The defined vertical parts of the word image 38
Figure 29 : (a) Segmentation areas. (b) Segmentation result 39
Figure 30 : Vertical overlap of characters 39
Figure 31 : (a) contour, (b) separation of overlapping subwords 40
Figure 32 : Special cases of additional signs 40
Figure 33 : Elimination of repeats 41
Figure 34 : Improving the segmentation 41
Figure 35: Different challenges for segmentation systems of multi-font printed Arabic text 42
Figure 36 : A sample of the corpus 43
Figure 37 : Samples of correct segmentations 45
Figure 38 : a) projection method1; b) projection method2; c) template matching; d) contour labelling; e) contour local minima; f) our proposed method 45
Figure 39 : Individual segmented characters 46
Figure 40 : Cases of failed Segmentations 46
Figure 41 : Raw ancient Arabic skewed and noisy document 51
Figure 42 : The proposed process of segmentation 52
Figure 43 : A page of a historical document 53

Figure 44 : a) original Image, b) grayscale, c)Otsu's thresholding, d) Niblack's thresholding 54
Figure 45 : Fake angle of skew correction 55
Figure 46 : Skew correction. a: original. b: transformation. c: projection profile 55
Figure 47 : Detection of the skew correction angle 56
Figure 48 : The applied transformations. a) original image, b) RLSA, c) transitions, d) BGF 57
Figure 49 : Segmentation area between two consecutive lines 57
Figure 50 : The proposed algorithm of lines segmentation 58
Figure 51 : Overlapping resolving 58
Figure 52 : Touching line segmentation 59
Figure 53 : Example of Line Segmentation 59
Figure 54: a) & c) : Successful segmentations. b) & d): Failed segmentations 60
Figure 55 : Summary of skew correction tests 61
Figure 56 : Segmentation rates and time cost 61
Figure 57 : Intra-word and inter-word spaces in historical writing 66
Figure 58 : Local baseline detection 69
Figure 59: Relations of contours: C1 is a Brother of c2. C2 is a Parent of C3. C3 is a Child of C2 70
Figure 60 : Hole example: C2 is a hole of C1 70
Figure 61 : Association of additions to the main body 71
Figure 62 : Complete overlap of subwords 71
Figure 63 : Examples of touching components 72
Figure 64 : A descending character "rae - ر" 73
Figure 65 : Detection of Segmentation point 73
Figure 66 : Example of the raw document 74
Figure 67 : The process of MOJ-DB database construction 75
Figure 68 : Data augmentation with distortion, erosion and dilation 77
Figure 69 : Examples of text lines used for the segmentation test 78
Figure 70 : The used CNN model 80
Figure 71 : Examples from MOJ-DB and VML-HD databases 81
Figure 72 : Different shapes for the same letter (kaf - ك) 87
Figure 73 : SIFT Keypoint detection 88
Figure 74 : HOG descriptor simulation 89
Figure 75 : Multi-Layer perceptron 90
Figure 76 : Support Vector Machine simulation 90
Figure 77 : Random Forest simulation 91
Figure 78 : K-nearest Neighbor simulation 91
Figure 79 : Convolutional Neural network architecture 92
Figure 80 : Samples from AHCD Database 92
Figure 81 : CNN architecture 95

List of tables

Table 1 : All possible forms of Arabic characters 9
Table 2 : Detection of the next step according to the Freeman codes 37
Table 3 : The fonts used for the corpus construction 43
Table 4 : The results of segmentation of the different methods 44
Table 5 : Results of segmentation using the four image representations 62
Table 6 : Arabic to Latin Annotation 76
Table 7 : The ground truth of the subword "يعْتَقِد" 77
Table 8 :total number of subwords extracted with each method 78
Table 9 : Results of classification of subwords extracted with each method 78
Table 10 : The results of Classification of handcrafted features 79
Table 11 : Results of subwords classification with CNN 80
Table 12 : Classification of learned and handcrafted features for different databases 80
Table 13 : The test accuracy (%) of classification for the different models' combinations 93
Table 14 : Time cost for each features extractor 93
Table 15 : The results of classification using SVM-HOG on the augmented databases 94
Table 16 : The results of classification using the CNN model 95

List of abbreviations

AHCD	: Arabic Handwritten Characters Database
AHDB	: Arabic Handwriting Data Base
ANN	: Artificial Neural Network
BGF	: Binary Gaussian Filter
BL	: Baseline
CA	: Classic Arabic
CNN	: Convolutional Neural Network
EHD	: Edge Histogram Descriptor
EM	: Expectation Maximization
EMDC	: Euclidean Minimum Distance Classifier
FC	: Fully Connected
HACDB	:Handwritten Arabic Characters Database
HMM	: Hidden Markov Models
HOG	: Histogram of Oriented Gradients
KHATT	: KFUPM Handwritten Arabic TexT
KNN	: K-Nearest Neighbor
MLP	: Multi-Layer Perceptron
MOJ-DB	: Mojawhar Database
MP	: Middle Point
MSA	: Modern Standard Arabic
OCR	: Optical Character Recognition
PCA	: Principal Component Analysis
PNG	: Portable Network Graphics
PP	: Projection Profile
RBF	: Radial Basis Function
RCATSS	: Reference Corpus for Arabic Text Segmentation
RF	: Random Forest
RLSA	: Run Length Smoothing Algorithm
SHL	: Shared-Hidden-Layer
SIFT	: Scale-Invariant Feature Transform
SOM	: Self-Organizing Map
SP	: Segmentation Point
SVM	: Support Vector Machines
TC	: Touching Components
UNESCO	: United Nations Educational, Scientific and Cultural Organization
XML	: Extensible Markup Language

General Introduction

1. Optical character recognition

Optical character recognition, known for OCR, is a research subfield of pattern recognition, and artificial intelligence. It refers to a computer's ability to distinguish individual characters and words. OCR is built to convert text-based images into a text format that can be recognized, manipulated and used by text-processing software, eliminating the need for manual retyping and increasing human productivity. This technology finds applications in diverse areas, including automated postal mail sorting, check processing, and editing of historical documents. Character recognition can be categorized into two main types: online recognition and off-line recognition. Online recognition involves identifying characters in real-time as they are being written, utilizing the digital representation of the pen's movement. On the other hand, off-line recognition focuses on analyzing scanned images of documents that were written beforehand. **Figure 1.** An Offline recognition system can be either analytic, relying on words segmentation into characters, or global segmentation-free, based on the recognition of words as input units.

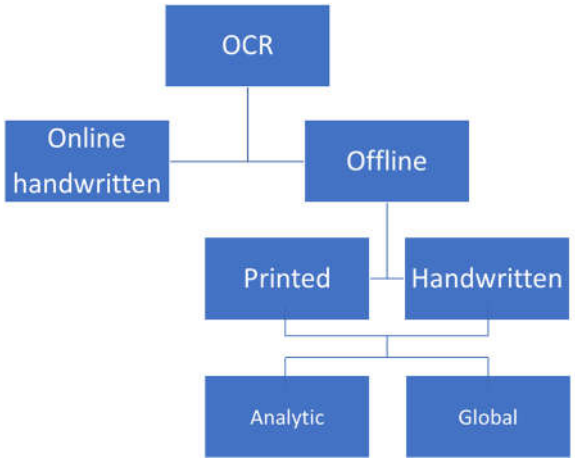

Figure 1 : Types of Optical character recognition systems

While Latin and Chinese handwriting recognition has seen significant advancements, research on Arabic character recognition either for printed or handwritten text has been relatively limited, despite Arabic characters being utilized in various languages like Persian, Urdu, Jawi, and others.

In fact, the development of commercial OCR systems dates back to the 1950s, leading to the availability of many commercial text recognition products. These systems have witnessed notable progress in character and document recognition, enabling the recognition of complex layouts and multilingual scripts in both machine-printed and handwritten modes. While early OCR efforts focused on numerals, they have now expanded to encompass other alphabets like Latin, Chinese, Japanese syllabic characters, Hangul characters, and more.

During the 1940s, when digital computers were being developed, initial efforts in Latin character recognition were undertaken. Similarly, Chinese character recognition saw its early attempts in 1966, while work on Arabic Optical Character Recognition (OCR) began in the 1970s, as referenced in [1], [2]. The first research work on Arabic OCR can be traced back to 1975 [3], while the availability of the first Arabic OCR system occurred in the 1990s, as highlighted in [4], and mentioned in a comprehensive survey conducted in 2013 [5].

2. Arabic Optical character recognition

2.1. Arabic writing

Arabic, spoken by over 350 million people worldwide, is a Semitic language with its own distinct grammar, spelling, punctuation rules, and pronunciation. It exhibits notable characteristics, including its numerous varieties and written forms. Moreover, Arabic writing can be classified into two types: Modern Standard Arabic (MSA) and Classical Arabic (CA). MSA holds more significance for business purposes in Arabic-speaking markets, as it is commonly used across all Arab countries in newspapers, magazines, books, and official documents. On the other hand, Classical Arabic is primarily employed in religious books, ancient documents, and classical literature. Most Arabic speakers consider these two varieties as different registers of the same language, referred to as "فصحى العصر" (MSA) and "فصحى التراث" (CA) in Arabic.

Modern Standard Arabic (MSA) serves as the official language in Arab nations and is the predominant form of Arabic instruction across educational institutions. During the early Islamic era, Classical Arabic writing experienced notable transformations. These changes included the incorporation of dots to distinguish similar letters and the utilization of diacritical marks to aid in pronunciation. MSA diverges from Classical Arabic in three significant aspects: vocabulary, stylistic choices, and peripheral innovations that do not adhere strictly to classical regulations. MSA exhibits some degree of variation, as certain writers emulate classical models closely, while others endeavor to establish novel stylistic patterns.

2.2. Arabic writing characteristics

In the Arabic writing system, text is written in a right-to-left direction. There are 28 characters in the Arabic alphabet, and the shape of each character changes depending on its position within a word. Consequently, a single character can have up to four different shapes. As a result, the number of distinct shapes to be identified increases from 28 to over 120. **Table 1** shows all the possible shapes of Arabic characters. Note that the words are separated by spaces. Obviously, if one or more of the six characters (ا-د-و-ز-ذ-ر) appear in the middle of the word, the word will be divided into connected component blocks called subwords. Thus, a word can be composed of one or more subwords. As they are also separated by spaces, these are usually shorter than the spaces between words: **Figure 2**.

Table 1 : All possible forms of Arabic characters

Name	isolated	Beginning	Middle	end		Name	isolated	Beginning	Middle	end
Alif	ا			ـا		Fa	ف	فـ	ـفـ	ـف
Ba	ب	بـ	ـبـ	ـب		Qua	ق	قـ	ـقـ	ـق
Ta	ت	تـ	ـتـ	ـت		Kaf	ك	كـ	ـكـ	ـك
Tha	ث	ثـ	ـثـ	ـث		Lam	ل	لـ	ـلـ	ـل
Jeem	ج	جـ	ـجـ	ـج		Mim	م	مـ	ـمـ	ـم
Hha	ح	حـ	ـحـ	ـح		Noun	ن	نـ	ـنـ	ـن
Kha	خ	خـ	ـخـ	ـخ		Haa	ه	هـ	ـهـ	ـه
Dal	د			ـد		Waw	و			ـو
Thal	ذ			ـذ		Yaa	ي	يـ	ـيـ	ـي
Ra	ر			ـر		Alif	آ			ـا
Zay	ز			ـز		Alif	أ			ـأ
Seen	س	سـ	ـسـ	ـس		Alif	إ			ـإ
Sheen	ش	شـ	ـشـ	ـش		Waw	ؤ			ـؤ
Sad	ص	صـ	ـصـ	ـص		Ya	ئ	ئـ	ـئـ	ـئ
Dhad	ض	ضـ	ـضـ	ـض		LamAlif	لا			ـلا
Ta	ط	طـ	ـطـ	ـط		LamAlif	لأ			ـلأ
Dza	ظ	ظـ	ـظـ	ـظ		LamAlif	لإ			ـلإ
Ain	ع	عـ	ـعـ	ـع		LamAlif	لآ			ـلآ
Ghain	غ	غـ	ـغـ	ـغ		Hamza	ء			

(a) عبد الحي (b) ا ز و ا ز و

Figure 2 : (a) Word composed of 3 subwords (b) word composed of isolated characters

Arabic characters vary in length and height e.g. (ت، ا). Furthermore, it is worth noting that the width and height of Arabic characters also vary based on their different positions within a word, e.g. (ح، حـ). This means that segmentation based on a fixed character size is not applicable.

In specific fonts, when certain letters are combined, they can form a new symbol referred to as a ligature. **Figure 3**. These multiple letters' shapes and combinations significantly increase the number of different forms that a classifier must recognize to well over 140, in addition to punctuation marks and numbers. Segmenting this kind of letters may appear

impossible; therefore, these ligatures are considered new forms to be recognized as units.

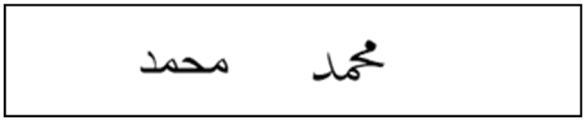

Figure 3 : (a) a word without ligature (b) the same word with ligature

Many Arabic letters are distinguished from each other only by dots below or above the letter. **Figure 4.a**. Misclassification of only one of these letters can lead to a completely different word. Moreover, diacritics representing vowels can also be found below or above the word. The diacritics -when used- determine the pronunciation of the character and the meaning of the word. For example, the word "كتب" can mean depending on the diacritics associated <to write = كَتَبَ> or <books = كُتُبٌ> or <was written = كُتِبَ>, among other meanings. But these signs are often left out, and the meaning of a word is identified by the context. **Figure 4.b** summarizes all possible diacritic signs. Although, segmentation systems should consider the presence of these signs.

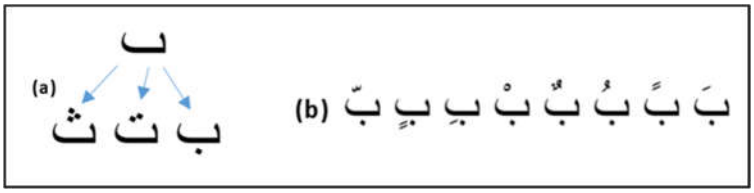

Figure 4 : (a) Example of characters with dots (b): the same letter with different diacritics

The Arabic characters are connected by a baseline forming words or subwords. The space is used to separate words within the line, or to separate subwords within the same word. The baseline is horizontal and crosses connected parts of the text. The baseline contains the maximum number of the textual image's pixels; **Figure 5** shows the baseline of an Arabic text sentence. Considering the different characteristics of the baseline, such as position and thickness, improves the ability of the segmentation system.



Figure 5 : The baseline of an Arabic text

Printed and handwritten Arabic texts possess notable distinctions in their characteristics and present distinct challenges in terms of processing and recognition. **Figure 6.** On the one hand, Printed Arabic text encompasses professionally produced material, such as books or official documents, wherein the text is generated using printing methods. It adheres to typographic rules about spacing, and consistent shapes and proportions, resulting in standardized text. Moreover, printed text often adheres to strict alignment and baseline rules, facilitating easier segmentation. The regularity of Printed Arabic text contributes to enhanced recognition accuracy and enables automated text analysis.

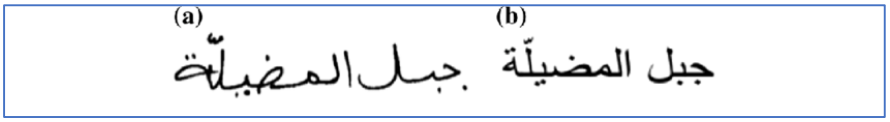

Figure 6 : The same text in : a) handwritten, b) printed

On the other hand, Arabic handwritten text refers to text produced manually by individuals, including personal notes or informal documents. Handwritten Arabic text introduces greater variability due to the unique handwriting styles of individuals. Handwritten characters are frequently interconnected, posing challenges for segmentation. Additionally, Arabic handwriting displays variation in terms of character shapes, sizes, slants, and spacing. The presence of inconsistencies in handwriting, such as overlapping strokes and varying pen pressure, further amplifies the complexity of recognizing and interpreting handwritten Arabic text.

Therefore, the recognition of Printed Arabic text benefits from its standardized and well-defined structure, facilitating efficient processing and yielding higher accuracy. Conversely, Handwritten Arabic text presents various challenges due to its individuality, cursive nature, and greater variation in character forms and connections. Effective processing and recognition techniques must be employed to address the distinct characteristics and challenges associated with each type of Arabic text.

2.3. Arabic Optical Character Recognition

The identification of Arabic text poses distinctive challenges and advantages for researchers. In contrast to the recognition of text in other writing systems, the recognition of Arabic text has received comparatively limited attention and effort over the past three decades. The absence of a commercially available Handwritten Arabic OCR product that exhibits acceptable accuracy remains a significant issue. The process of offline recognizing Arabic text comprises five stages: pre-processing, segmentation, feature extraction, classification, and post-processing. **Figure 7.**

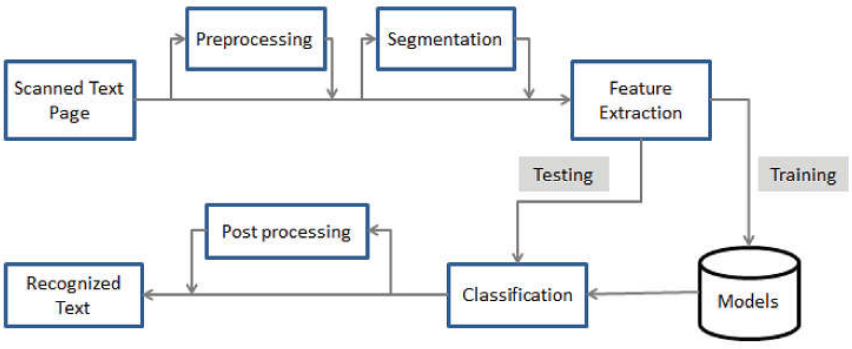

Figure 7 : The complete process of an Arabic analytic OCR

2.3.1. Pre-processing

The pre-processing stage serves as an initial step in which the data is transformed into a format that can be effectively processed. It involves enhancing the raw input images by

reducing noise and distortion and enhancing the overall quality of the image. This stage encompasses various techniques such as thinning, binarization, smoothing, alignment, normalization, and baseline detection.

2.3.2. Segmentation

Segmentation is a crucial step in Arabic text recognition due to the cursive nature of Arabic writing. It involves dividing a page of text into logical components, including page decomposition and word segmentation. Page decomposition separates different elements, such as text, graphics, and lines, while word segmentation breaks down words into individual characters. Segmentation plays a vital role as it influences the recognition accuracy. Different types of segmentation techniques are employed, including segmenting pages into paragraphs, separating text from other objects, segmenting text into lines, segmenting lines into words and sub-words, and further for analytical systems, the segmentation of sub-words into isolated characters. This latter is particularly significant but also challenging and resource-intensive. Various techniques such as vertical projection, thinning, contour tracing, and others are utilized for segmentation.

2.3.3. Feature extraction

The feature extraction stage involves capturing relevant information extracted from the text image. This information should encompass the essential characteristics that differentiate one character or word from another. The selected attributes and properties are then passed on to the classifier to aid in the classification process. Feature extraction techniques vary depending on the specific application objectives. A technique that proves successful in one application may not yield the same precision in another. Therefore, the careful selection of a feature extraction method remains crucial for achieving high recognition accuracy. According to [6] Handwritten text features can be categorized into structural or geometrical features, statistical features, and transformations-based features.

2.3.4. Classification

The classification stage is the primary decision-making phase of an OCR system. During this phase, the system carefully examines individual elements, such as lines, words, and characters, and assigns them specific classifications based on their visual attributes. This process involves comparing the extracted features of these elements with pre-defined templates or statistical models. The classification phase plays a vital role in accurately identifying the correct characters or symbols for each element. In the classification stage, the previously extracted features are used to identify different parts of the text. This is done by comparing them to pre-defined models that describe the structure or statistics of the text. To perform this task, machine learning techniques like Artificial Neural Networks (ANN), support vector machines (SVM), k-nearest neighbors (k-NN), and Hidden Markov Models (HMM) are commonly used.

2.3.5. Post-processing

The post-processing stage aims to improve recognition accuracy by refining decisions made in the previous stage and utilizing contextual information. It is responsible for generating the best possible solution and often involves techniques based on character occurrences, lexicons, and other contextual information. This stage further enhances word recognition by considering the surrounding context. [2].

3. Printed Arabic text segmentation

Segmentation is a critical step in Optical Character Recognition systems for printed and handwritten Arabic text, with specific emphasis on the segmentation of words into individual characters. This process holds significant importance as it enables accurate recognition and analysis of Arabic text at the character level. Proper segmentation facilitates the extraction of distinct characters, which are then passed through subsequent stages of feature extraction, classification.

Segmentation of printed Arabic text for OCR involves distinct levels. The first level encompasses the segmentation of the entire page into lines, establishing the overall structure and organization of the document. This level ensures that text is appropriately separated into individual lines, providing a foundation for subsequent processing. Moving to the second level, the lines are further segmented into words. Word extraction plays a critical role as it defines the boundaries between individual words within the text. Techniques such as horizontal projection, contour tracing, or connected component analysis are utilized to accurately identify and isolate words.

Finally, the third level of segmentation focuses on segmenting words into individual characters. This level is vital in recognizing printed Arabic text accurately. The segmentation of words into characters entails precisely identifying and separating the constituent characters within each word. Techniques such as contour tracing, connected component analysis, vertical projection, or stroke analysis are employed to detect the transitions, gaps, and strokes that define the boundaries of each character. This level of segmentation enables precise feature extraction, facilitating more accurate character recognition and subsequent analysis.

However, segmenting printed Arabic words into characters presents several challenges. One challenge arises from the cursive nature of Arabic script, where characters are typically connected within a word. This connectivity requires careful identification of characters to avoid misinterpretation or merging during segmentation. Additionally, the use of different fonts further complicates the segmentation process. Thus, multi-font text is quite difficult to process due to the variation of shapes of the characters. Inaccurate segmentation can lead to errors in character recognition and subsequent analysis, affecting the overall OCR accuracy.

Regarding the importance of segmentation, developing robust systems of word-to-character segmentation requires a lot of effort in studying different techniques and approaches. However, the evaluation and comparison of different segmentation techniques should be done on unified

data sources. In literature, researchers usually use their own private datasets to evaluate their segmentation system, which does not allow a clear comparison of their capabilities. Therefore, comes the need for standard corpora for evaluating the segmentation systems.

4. Historical documents digitization

4.1. Motivations

As stated previously, processing handwritten Arabic presents additional challenges compared to printed text. Furthermore, the processing of historical text creates a higher level of complexity for segmentation and recognition systems. In fact, it may require specialized approaches and expert knowledge due to variations in script styles, archaic vocabulary, and different conventions used in different historical periods.

The application of Optical Character Recognition (OCR) to historical documents holds significant importance for various reasons. Firstly, historical documents contain valuable information about the past, including cultural, social, and historical insights. However, these documents are often handwritten or printed in older fonts, making them difficult to read and interpret. By implementing OCR on historical documents, they can be converted into machine-readable formats, allowing for easier access, analysis, and preservation. This facilitates the exploration of historical knowledge, enabling researchers, historians, and the general public to uncover hidden insights.

Secondly, OCR for historical documents contributes to the digitization and preservation of delicate and aging materials. Many historical manuscripts, papers, and records are prone to deterioration over time. By digitizing these documents through OCR, we minimize physical handling and reduce the risk of further damage. Digital copies also enable the creation of backups and ensure the long-term preservation of historical materials for future generations.

Lastly, OCR assists in language transcription and translation of historical documents, which may be written in outdated or regional dialects, posing challenges to understanding and interpretation. By utilizing OCR, these texts can be transcribed into modern languages, making them more accessible to a wider audience.

4.2. Al-Mojawhar writing style

The complexity of ancient Arabic text can be seen in **Figure 8**, which provides an example of an ancient document dating back to the 17th century and written in the Moroccan Al-Mojawhar writing style. This latter is one of several old writing styles. It is an artistic style of Arabic calligraphy known for its intricate and elaborate designs resembling jewelry. This style is characterized by harmonious geometric shapes and intricate details. For centuries, Al-Mojawhar was one of the few Moroccan styles that were used to write books and official letters and records. Therefore, thousands of books written in Al-Mojawhar are today preserved and stored in libraries and archives around the world. However, nowadays Al-Mojawhar writing style is commonly used for decorative purposes in designs such as jewelry, traditional

buildings, and handmade artworks. Different variations of the Moroccan Al-Mojawhar style can be found, reflecting the local culture and traditions across different regions of Morocco. Al-Mojawhar style is considered part of Morocco's cultural and artistic heritage and is popular in the fields of design and traditional arts. Writing in Al-Mojawhar style requires special skills and knowledge of the associated rules and techniques. Thus, learning this style is quite challenging, and only a few institutes in Morocco still teach it.



Figure 8 : Example of the raw document

4.3. Historical text recognition

Despite the importance of building an OCR for ancient text and the abundance of historical documents in libraries, only a few researches can be found in the literature addressing the segmentation and recognition of Arabic historical text. In fact, developing such a recognition system may face various difficulties coming from the nature of the ancient script. The analytical approach relying on the segmentation of text into characters is not an option for building an OCR for historical text. The reason behind this is the complexity of text segmentation into characters. In fact, segmenting a text line into characters may be impossible due to the cursiveness of writing and variations in characters' shapes, in addition to the intensive presence of diacritic and additional signs. Therefore, the only possibility to build such an OCR is to follow the global approach, which intends to recognize the words without decomposing them into characters.

Nevertheless, the main reason behind the research's scarcity is the lack of public and standard databases of Arabic historical text that offer a sufficient amount of data to build and

test the recognition models. This lack of resources can be explained on the one hand by the complexity of collecting, organizing, and creating such databases. On the other hand, because of the wide diversity of scripts and the difficulty of accessing the historical materials or finding collaborative calligraphy writers.

Clearly, Arabic historical script differs from modern handwriting in several ways. Firstly, it was written on blank pages without line guides, and additional signs frequently occur between lines. Additionally, the tools and ink used were very distinct from modern pens, which caused having different pen thicknesses within the same text line.

Additionally, Arabic historical manuscripts may include highly close lines and words of text, and the spacing between them might often vary within a single document. As seen in **Figure 9**, when the ascenders and descenders of two subsequent lines are at the same horizontal position, the effect of lines overlapping occurs. Moreover, when a descender from the first line crosses an ascender from the second line, it creates the touching lines effect. These two effects contribute significantly to the complexity of extracting and segmenting the lines and subsequently extracting words and subwords.

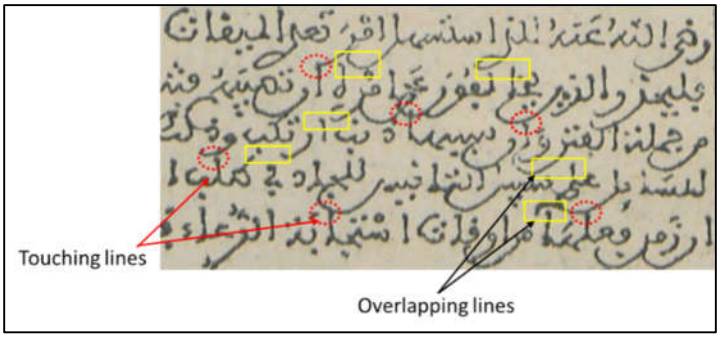

Figure 9 : Overlapping and touching lines

5. Thesis Presentation

In the end of 2017, we started this thesis with a lot of enthusiasm and motivation to contribute to the recognition of Arabic text segmentation and recognition. Acknowledging the difficulties and problematics from which suffers the optical character recognition of Arabic text, we identified the causes of delay in this field and lack of modern researches. Then we have worked on different aspects of Arabic OCR, by publishing several contributions for printed and ancient text segmentation. After our first work on printed Arabic text segmentation, we realized that Morocco, being one of the oldest countries of the world, needs a different kind of OCR; an OCR for historical documents written in Moroccan Al-Mojawhar style. In fact, this country possesses thousands of historical documents, which need to be exploited and digitally preserved. Moreover, the Moroccan historical documents are spread around the world, which means that such an OCR, designed for this kind of documents will have a global impact. Therefore, we shifted our focus to this field of research, by studying the state-of-the-art and proposing new methods of segmentation and data sources. We hope that this thesis will help

pushing forward the research on Arabic optical character recognition field and help researchers interested in developing OCRs for historical text recognition.

5.1. Problematics

This thesis came to address several problematics in the field of Arabic text recognition:

1. The segmentation of Printed Arabic is a challenging task in all its three levels. Existing methods often struggle to accurately identify the segmentation points, resulting in errors and delays in subsequent analysis and recognition tasks. This issue hinders the overall efficiency and accuracy of Arabic text recognition systems.
2. Evaluating and comparing different multi-font word segmentation systems is crucial for advancements in the field. However, the lack of comprehensive resources delays the progress. Without a reference corpus of printed Arabic text covering various fonts and cases, researchers face challenges in objectively assessing and comparing different segmentation techniques. This scarcity inhibits the development of robust and effective multi-font word segmentation algorithms.
3. Historical text documents often suffer from issues such as skewed text and inconsistent text lines. Correcting the skew and extracting coherent text lines from historical texts are essential steps for accurate analysis and interpretation. However, the inherent challenges of historical texts like touching components demand robust techniques capable of handling variations in skew angles and accurately extracting text lines for further processing.
4. Extracting words or subwords is essential for recognizing historical documents. However, historical text poses challenges due to variations in writing styles, ligatures, and overlapping. Especially the Moroccan writing style Al-Mojawhar. Accurately extracting subwords from historical text lines written in such style requires a specialized approach.
5. The absence of databases specifically focused on historical words presents a significant obstacle for researchers in the field. Without access to comprehensive datasets, training and evaluating algorithms and models for historical text analysis become difficult.

5.2. Contributions

To address the aforementioned problematics, this thesis came with several contributions, published in well-recognized and indexed journals:

1. To address the problematic segmentation of printed Arabic words into characters, a novel approach is proposed which enhances the accuracy of word-to-character segmentation of printed Arabic text. The approach combines contour analysis and template matching techniques. to locate segmentation points.
2. To overcome the lack of comprehensive resources for comparing and evaluating multi-font word segmentation systems, a corpus of printed Arabic text with 34 diverse fonts is created. This corpus encompasses a wide range of font styles commonly used in Arabic typography. Researchers can utilize this corpus as a standardized benchmark to evaluate

the performance of their segmentation techniques across different font styles. This resource facilitates objective assessments and comparisons, leading to the advancement of research on multi-font word segmentation and recognition.

3. To tackle the challenges associated with historical text segmentation, a novel method is proposed, aiming to address the issues of skew correction and text line extraction. This approach uses horizontal projection to effectively correct skew angles and extract coherent text lines from historical texts.
4. To address the line-to-subword segmentation problem in historical text, we propose a novel contour-based approach. By analyzing the contours and features of characters within a text line, this method accurately extracts subwords. This method enables researchers to delve deeper into the segmentation and recognition of historical textual content.
5. Recognizing the scarcity of databases specifically catering to historical words, we propose MOJ-DB. This comprehensive database contains a vast collection of historical subwords sourced from diverse historical documents written in Al-Mojawhar Moroccan style. MOJ-DB can serve as a valuable asset for researchers, offering 560000 words images for training and evaluating algorithms and models. By utilizing this database, researchers can improve the accuracy and performance of their systems in historical text analysis tasks.

5.3. Publications

The works presented in this thesis are referenced to the following scientific publications:

- Zoizou Abdelhay, Zarghili Arsalane and Chaker Ilham (2018): A new hybrid method for Arabic multi-font text segmentation, and a reference corpus construction [7].

- Zoizou Abdelhay, Zarghili Arsalane and Chaker Ilham (2019). Skew Correction and Text Line Extraction of Arabic Historical Documents [8].

- Zoizou Abdelhay, Zarghili Arsalane and Chaker Ilham (2022), “MOJ-DB: A new database of Arabic historical handwriting and a novel approach for subwords extraction [9].

- Zoizou Abdelhay, Errebiai Chaimae., Zarghili Arsalane and Chaker Ilham (2023). : Feature extraction for character recognition in historical Arabic documents A comparative study.

5.4. Thesis organization

This introductive chapter introduces the reader to the field of optical character recognition for Arabic text. In this chapter, we mentioned our objectives to work on Arabic OCR and the problems facing researchers willing to develop segmentation systems for Arabic words into characters. We also shared our motivations to contribute to the Arabic historical documents’ digitization and resolve the different problems causing the delay of research on Arabic historical document digitization. **In Chapter 2**, we present our contributions to the segmentation of printed Arabic text with a hybrid method of word-to-character segmentation

based on contour and template matching. Then, we present our corpus of printed Arabic text that allows the evaluation and comparison of segmentation systems. **In Chapter 3**, we address the problem of text line extraction by proposing a method of skew correction and text line extraction from historical text written in Al-Mojawhar style. **In Chapter 4**, we present our approach of subword extraction from Al-Mojawhar text lines. We also propose MOJ-DB, which is a new database of words and subwords written in al-Mojawhar ancient style, providing a solution for the lack of resources in the field of OCR for historical text. In the final **Chapter 5**, we investigate character recognition using the proposed database. We study the most famous feature extraction and classification methods used in OCR, by forming multiple combinations and testing them against MOJ-DB.

Chapter 1. Printed Arabic text segmentation

Introduction

In a typical analytic offline Arabic OCR system for the cursive script, the input text is acquired by an optical scanner, and each character is then located and segmented. The segmented characters are classified against a predefined database. Following this approach, the word is segmented into smaller classifiable units (characters). It is clear that this approach is a bit difficult, especially in the case of the handwritten script. The success of an Arabic OCR system in recognizing characters depends heavily on the quality of the segmentation of the text into individual characters. Segmentation errors will surely evolve into false recognition or character rejection in the best of cases, which cannot be supported in a critical domain such as mailing, healthcare, or banking. To avoid the difficulties of segmentation during the recognition process, researchers have proposed another approach known as the holistic (or global) approach. Where recognition is performed globally on the complete representation of words without resorting to segmentation, which is why it is also called the free-segmentation approach. However, this approach is quite difficult to establish because the time cost is very high, and the fact that global recognition systems require a lot of training data regarding the huge amount of Arabic words. Instead, the analytical recognition approach based on segmentation is much faster, and input training data is limited to the number of characters.

The nature of Arabic writing creates serious challenges for the segmentation systems. The cursiveness and overlapping are the biggest ones. An Arabic word is written from right to left, and characters are connected with a line crossing the bottom of the word called the baseline. By creating cursiveness, the presence of this line justifies the big difference between Arabic and Latin-based writings when it comes to text segmentation and character extraction. On the other side, the Arabic alphabet contains some ascender and descender characters whose shapes go under and above the baseline. For some writing styles, the presence of these two types of characters creates overlapping, which makes the segmentation task a bit difficult.

In this chapter, we present a brief overview of Arabic text segmentation, the main steps, and the best segmentation methods found in literature, as well as the works on datasets used for the evaluation of the different systems of segmentation. Giving that most of the methods in the literature are limited to the segmentation of the mono-font and mono-size text, we introduce our contribution to the segmentation of Arabic multi-font and multi-size printed text. Our contribution consists of a hybrid and robust method, which uses contour analysis and Template Matching to split Arabic words into individual characters. We also present in this chapter a description of a reference corpus, which we have proposed to be used by all Arabic printed text segmentation systems for evaluation and comparison purposes.

This chapter consists of three sections organized as follows: The first section presents the state of the art of Arabic text segmentation into lines, then into subwords, and especially segmentation of words characters. We give a description of the most recent works on character segmentation and a brief preview of the different datasets and corpuses used for evaluating the segmentation systems. The second section is a study of some word-to-character segmentation

methods, where we discuss four methods that we have studied and implemented. In the third section, we present our contribution to multi-font and multi-size Arabic text segmentation, proposing improvements for the methods studied in the second section. In section 4, we present the experimental aspect of this study, highlighting the different obtained results. Finally, a conclusion summarizes our contribution to the printed Arabic text segmentation.

1. State of the art of Arabic text segmentation

Segmentation is the process of isolating individual characters to be passed to the recognition system. Segmentation of text into characters is the most crucial and difficult step for an Arabic OCR system. In fact, a weak segmentation process will surely produce poor recognition results. As stated before, the text segmentation process for character recognition can be divided into three levels: page-to-line segmentation, where the text lines are extracted from the page image. Line-to-word/subword segmentation is where the lines are decomposed into words and subwords. Finally, character segmentation is where the previously extracted words are split into classifiable characters. **Figure 10** illustrates the segmentation levels of an OCR system.



Figure 10 : Levels of text segmentation

Several efforts have been devoted to the segmentation of cursive texts, but so far, it is still an unsolved problem. Whatever algorithm is used, it has to deal with the problem of cursiveness and the effects of other Arabic features that influence the segmentation. Besides the complex nature of Arabic writing, other challenges can face segmentation systems like overlapping, skewness, and touching characters.

1.1. Segmentation of pages into lines

The goal of line segmentation is the extraction of text lines from the text image. To this end, many Arabic OCR systems use the Horizontal Projection method. This technique is most

used for printed text segmentation, where text is straight and no overlapping or touching occurs between lines. This method calculates the horizontal histogram of the image, and if the number of successive blank lines encountered -from top to bottom- is greater than a threshold to be set experimentally, the number of lines is increased. Then the system extracts the beginning and end of each line. **Figure 11** shows an example of the method used. Other techniques have been adopted for line extraction. The projection approach has been enhanced by [10]–[12]. Other works like [13], [14] adopted Hough transform for line extraction. Run Length Smoothing Algorithm RLSA was used by [15], [16].

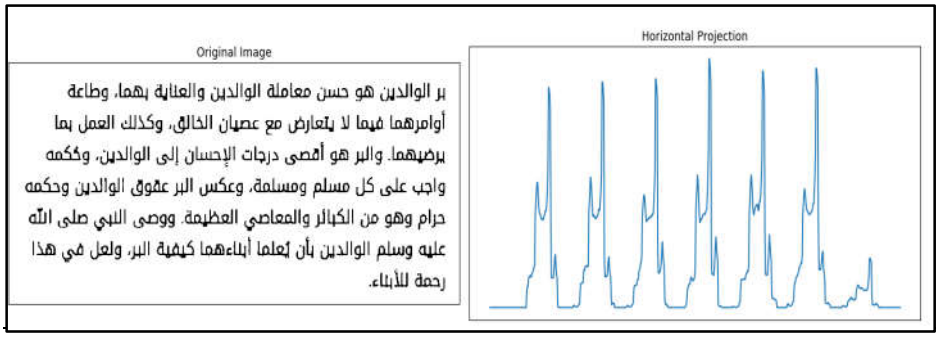

Figure 11 : Detection of Horizontal Projection profile

1.2. Segmentation of lines into words and subwords

The second phase of segmentation is extracting words or subwords from lines. To do that, one of the most used techniques is vertical projection. In contrast to the horizontal projection, the vertical projection counts the number of black pixels in each column of the image. If the total is not zero, it indicates a connected part, and therefore the line is not segmented. In contrast, if the number of successive white columns encountered is greater than the threshold to be set, the current part is segmented. **Figure 12** shows the profile of the vertical projection for a given image.

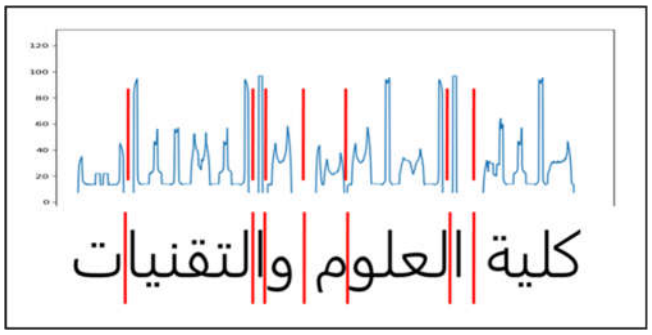

Figure 12 : Example of the segmentation of a line of text into subwords by the vertical projection

The projection method is generally suitable for segmenting printed texts that are written with large spaces between words and lines. Texts that have overlapping or touching components will certainly cause difficulties for projection-based segmentations. Other works have proposed different techniques for line extraction, especially for handwritten text. [17] Used a spanning tree graph to locate the subwords and the Otsu algorithm to regroup the

subwords of the same word. [18] Proposed a topological feature-based algorithm to extract the line components after overlapping resolving. [19] used an enhanced version of vertical projection. Considering the distances between all the connected components of the line and classifying the distances into inter-word or intra-word. A similar approach was followed by [20] using clustering with Self-Organizing Map (SOM). [21] Proposed using bounding boxes and classifying them into primary and secondary components to resolve overlapping.

1.3. Word segmentation

The stage of segmenting words into characters is extremely important in analytical OCR systems. It is based on detecting segmentation points. Which are located between consecutive characters on the baseline. The work of [22], followed by the work of [23], were the first attempts to segment Arabic characters. In analytical OCR systems, several techniques were proposed for the segmentation of words into characters; in the first attempts, the vertical projection was used for this purpose. Later, the tendency was to obtain the skeleton of the word and analyze it to find the appropriate segmentation points[24]–[26]. This was followed by attempts to segment words by tracing the word contour [27]–[31]. Many other works were based on the Template-Matching technique alone or combined with the other techniques mentioned above[32], [33]. Global OCR systems use the holistic or segmentation-free approach in which words are recognized without segmentation. In the following, segmentation works are grouped according to the technique used.

1.3.1. Baseline detection

The baseline is the horizontal line containing the maximum number of black points in the binary image. Baseline detection is critical for all segmentation approaches. It consists of finding the vertical start and end indices of the baseline that connects the characters of the word. Baseline estimation is not only necessary for subword segmentation, but it is also important for other document processing, such as skew correction, text line extraction, and feature extraction.

Baseline detection, as it is very important, it faces different challenges well described in [34]. Considering the importance of baseline position, many works contributed to detecting the baseline by proposing different approaches, particularly for handwriting processing. authors in [35] proposed a method based on geometric features like the alignment of neighborhoods and topologic features that connect the baseline to the characters. Authors of [36] published a comparative study of several baseline detection methods, including Horizontal projection, Skeletonization, contour, and Principal Component Analysis (PCA). The study came to the conclusion that methods based on the word skeleton produce the best precision of baseline detection. Later in [37], the same authors used the Voronoi diagram based on contour tracing and sampling. The authors of [38] proposed a method for baseline detection of multilingual handwriting. The proposed work uses contour curvatures and invariant features of the orientation to train an SVM, which decides if the input is a probable baseline or not.

1.3.2. Projection-based word segmentation

As described before, projection-based methods reduce 2D information to 1D. It works best with printed documents, especially with fonts that do not form ligatures, such as "transparent Arabic" and "simplified Arabic". However, for fonts such as "traditional Arabic" which contains many ligature forms, the projection method presents unavoidable weaknesses.

These methods are based on the fact that the connecting lines are always less thick than other parts of the words. The principle of this approach is calculating the vertical profile of the image with formula 1:

$$Vi=\sum P(i,j) \quad (1)$$

Where *P(i,j)* is either equal to 0 (white) or equal to 1 (black), representing the pixel value at the ith column and j^{th} row. The vector V forms the histogram of the image, which can be analyzed to determine the segmentation points.

The authors in [39] proposed an Arabic character segmentation algorithm, which is based on vertical histogram analysis and some other rules based on the structural features of the background regions and the character components. The characteristics of Arabic characters are also taken into consideration to check whether the subword has one or more characters and to find potential segmentation points. Then, the vertical histogram is scanned and analyzed to find the actual segmentation points. The reported experimental results show that the algorithm achieved about 94% correct segmentation rate.

In [40], the researchers used the vertical projection of the center area instead of the whole word. The word is split into three distinct vertical regions: the baseline, the center area, and the bottom area. In the case where the vertical projection value of the center area is less than two-thirds of the baseline thickness, it is considered a connector region between two characters. Additionally, any area with a higher value after the connector region is considered the starting point of a new character as long as its profile is greater than one-third of the baseline. This technique was used with the Naskh font, and over-segmentation was observed with some characters such as (س ش).

It is evident that projection-based techniques depend heavily on a predefined threshold value related to the width of the characters and the font used. It should also be noted that secondary elements, such as dots and diacritical marks, influence the segmentation. It is concluded that segmentation methods that use the vertical projection histogram are highly dependent on baseline determination. These methods are independent of the shape, size, or font of the characters when the latter contains no overlap. They are best suited for printed type while being insufficient for segmenting characters in overlap or handwritten script.

1.3.3. Word segmentation based on Skeletonization

The researchers of image processing agree that the essential information of a shape is stored in its skeleton. Many algorithms have been proposed to extract the skeleton, such as

the distance transform and homotypic thinning, with two approaches: sequential and parallel, like the algorithm of [41]. In fact, Skeletonization is an essential technique that could help solve the character segmentation problem. However, combined with other techniques would be essential to ensure better segmentation quality. **Figure 13** shows an example of an Arabic word and its skeleton.

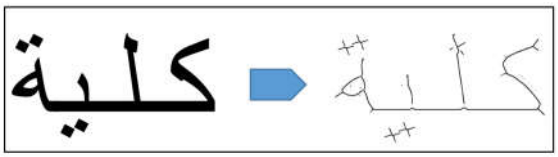

Figure 13 : Example of word Skeletonization

In the method of [24], the baseline of the thinned word (skeleton) is found. It is the line that contains the maximum number of black points. Then, only the columns that have no pixels above or below the baseline are considered connection segments that potentially contain segmentation points. The segmentation points are, according to the authors, in the middle of the connection segments.

The segmentation procedure presented in [25] is similar to that of [42]. Essentially, the proposed algorithm works by first obtaining the skeleton of the word using the [43] algorithm and locating the starting point and ending points of the character.

Authors in [26] followed a specific procedure for analyzing Arabic text. Firstly, they plotted the skeleton from right to left using a 3x3 window. Then, a binary tree was constructed to represent the skeleton using Freeman's code [44]. Each node in the binary tree corresponds to a specific part of the subword and describes its shape. To optimize the binary tree, the authors applied smoothing techniques to reduce the number of nodes. This involved eliminating empty nodes, minimizing the Freeman chain string, and reducing any noise present in the thinned image. Finally, the binary tree was segmented into subtrees that represent individual characters. The authors claim that their algorithm can be applied to Arabic text of any font and size.

Analyzing this literature, we see that the methods based skeletonization are very old. One of their disadvantages is that different thinning algorithms may produce different thinned word. Furthermore, the thinning process may change the shape of the character, especially in the case of poor-quality images. Some common problems encountered during the skeletonization process include the removal or erosion of secondary characters. These changes make skeleton image recognition a difficult task, even for human visual processing. More details about skeletonization can be found in [45].

1.3.4. Contour-based segmentation

Detecting edges in an image can greatly reduce the data volume and filter out less significant information while retaining crucial structural characteristics of the image. This technique offers numerous benefits compared to using the skeleton method. In [33], the top contour is examined for candidate segmentation points. This is done by analyzing the upper

contour and detecting its peaks. Between every two consecutive peaks, a segmentation point is found at the minimum point of the contour. **Figure 14.**

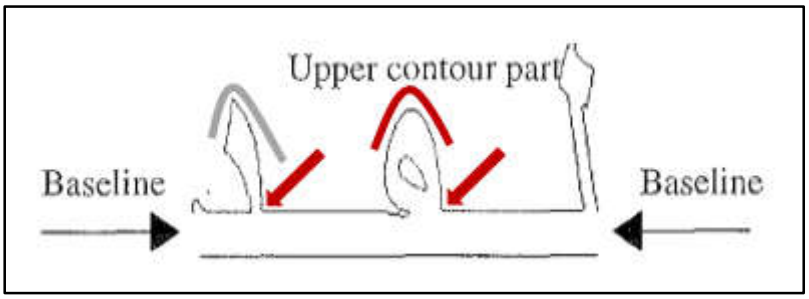

Figure 14 : Example of the location of segmentation points (arrow)

The segmentation technique proposed in [28] is based on analyzing the shape of the main body of words. Initially, the starting and ending points of the shape are determined, specifically the lower-right and lower-left points. Next, the shape is divided into segments based on changes in curvature. When the curvature changes from positive to negative, a segment ends and a new one begins. To minimize the impact of noise, a low-pass filter is applied to the shape data. Sometimes, there may be a significant difference in the horizontal line between characters, but this line doesn't carry any relevant information. To address this, a horizontal line detector is utilized to identify and disregard these lines during the recognition process. Although the authors did not mention the segmentation rate, they reported a recognition rate of 96.9%.

In their study [29], Peng et al. introduced an approach for segmenting Arabic printed characters by analyzing the contour of each character. The first step involves estimating the baseline position of a given word, followed by calculating a function D(x) that represents the distance between the contour and the baseline. The D(x) curve is then analyzed to identify potential segmentation points. However, the specific method for detecting these segmentation points was not mentioned by the researchers. Subsequently, structural rules are proposed to merge characters that were initially segmented into smaller parts. The researchers concluded that it is challenging to differentiate between similar characters from different languages. Although specific recognition rates for Arabic text were not provided, the study reported results for other languages, including Uyghur, Chinese, and Sindhi. Since Uyghur characters bear resemblance to Arabic characters, the estimated recognition rate for Uyghur was around 98%.

Olivier and Miled proposed a method in their study [30] that utilizes the top contour for segmenting Handwritten Arabic text. The process begins by detecting the contour of each subword using the Freeman code [44]. The contour is analyzed in a counterclockwise direction, and the top contour is identified based on the directions F3, F1, and F5. A filter is then applied to retain only the points of the upper contour located in the upper part of the subword. Subsequently, an automaton is employed to detect the local minima of the upper contour, which are considered as segmentation points for the subword. The authors evaluated their system on a database consisting of 6000 Arabic words written in 20 different fonts, and reported an achieved error rate of 2.59%.

The technique introduced by [31] employed the contour of subwords to identify segmentation points by applying specific rules to the local minima of the lower contour. In cases where characters were vertically overlapped, typically due to font characteristics or writing style (such as italic), additional processing steps were implemented. The method achieved a success rate of 86% but was evaluated on a limited dataset of only 100 words. Consequently, this approach highlights the necessity of combining contour features with other characteristics to improve character segmentation.

In conclusion, the contour remains the best choice to isolate connected and overlapping characters. The results of this technique (contour) are very remarkable compared to the other techniques seen previously. However, the segmentation based on the detection of the contour has its weak points as it has its strong ones. Among the problems that can be encountered in the analysis of the contour for the segmentation is its discontinuity. This can be due mainly to noise or the low resolution. Another problem with using the contour for segmentation is that there are several contour detection methods, such as the Sobel filter, Prewitt filter, Canny filter [46], or Laplacian cancellation, and each method may give a different contour. And it should also be noted that contour-based segmentation systems show a very poor performance against ligatures.

1.3.5. Word Segmentation based on Template-Matching

Template matching is a method used in digital image processing to locate specific portions of an image that match a predefined template image. This technique has been employed in character recognition applications to identify characters or words based on a stored representation of the characters. While this technique can be viewed as a segmentation method, it requires manual selection of the specific part of the character within the image that we are interested in locating.

In the study [33], a technique was introduced to detect potential segmentation points based on the occurrence of an angle formed by the junction of two characters at the baseline. **Figure 15** illustrates the utilization of a 7 × 7 window to analyze the character's neighborhood. If such an angle is identified, it is considered a potential segmentation point. To confirm the accuracy of these identified points, the extracted character is compared with a pre-existing manually stored model.

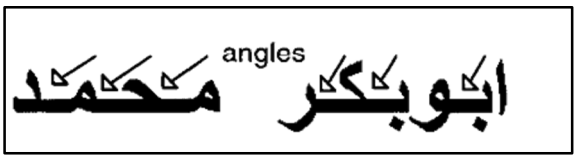

Figure 15 : The angles of junction between the characters

In [32], the authors utilized the Template-Matching technique to create an automatic segmentation system for video scenes. The goal was to separate the script from the remaining content in the videos. They conducted experiments on a dataset of 35 videos, with each video comprising 700 frames. The system achieved an accuracy rate of 91.52% in segmenting the script.

Although Template-Matching was able to achieve good results, its success in finding the appropriate angle depends heavily on the noise in the image. Template-Matching is not a suitable technique for the segmentation of Arabic printed or handwritten text due to the diversity of fonts and writing sizes. Although, it may be helpful in detecting one specific shape.

1.4. Evaluation of segmentation methods

In the literature on printed Arabic text segmentation, we noticed that researchers either used non-standard corpora or developed their own test corpora by scanning books, magazines, newspapers, and other documents: authors of [47] utilized 40 daily newspaper pages printed in 20 Naskh font extensions. Unknown datasets in Uighur (Arabic alphabet), Chinese, and English were used by [48]. In other work, [49] tested their proposed system using randomly chosen text written with a single type of Naskh font. The authors of [33] used an undefined dataset of a single font. A dataset of 15 pages, chosen from a magazine and written in the Naskh font, was utilized by [50]. The authors of [51] and [52] utilized a large dataset consisting of 264 pages but did not specify the font used. [53] employed an unconventional dataset made up of 40,000 sub-words printed in 20 different fonts. [54] used a dataset of 25 pages written in both conventional and simplified Arabic fonts. Five hundred (500) samples of Arabic text produced in two fonts—"simplified Arabic" and "Arabic transparent"—were utilized by [39]. While in [55], authors did not even mention the dataset utilized in their research. In conclusion, these corpora were written in a limited number of fonts that do not provide many variations. Especially Naskh, which is very simple and does not present many challenges for segmentation systems. Therefore, in order to have a clear view of segmentation systems' capabilities, a standard and robust corpus of printed Arabic text is needed for evaluation and comparison.

2. Study of printed word segmentation methods

After discovering the different approaches used for the segmentation of printed Arabic words into characters in the previous section, in this one, we have chosen three approaches to implement and compare: Vertical projection, the Template-Matching approach, and the contour approach. This comparison study is conducted to look up for any combination to improve the capabilities of the selected methods. In addition, we implemented the chosen methods and tested them against one corpus to prove that using different data sources for evaluating different systems does not give clear insights on their capabilities. It is worth noting that all the mentioned methods depend highly on accurate detection of the word's baseline. The simplest way to detect a given word's baseline is to use vertical projection by finding the first and the last rows of this line (in the vertical direction). This is done by calculating the maximum value of the horizontal projection of the image (peak value), then we set a threshold of 70% of this value. The first row of the baseline is the first row encountered (from the top

to the bottom of the entire image) whose number of black dots satisfies the threshold. Thus, we determine the end of the baseline, which is - starting from the previously detected beginning - the first row with the number of black dots below the threshold. As a result, we obtain the beginning and the end of the baseline. In the following, we discuss several segmentation methods for character extraction.

2.1. Template-Matching

In this part, we have studied and implemented the Template-matching method described in [33]. The method consists in looking for the connection point between the character and the baseline. **Figure 16**.



Figure 16 : Connection of the characters and the baseline

To identify possible points where a segmentation can occur, the row just above the baseline is scanned from left to right in this method. Each point in this row is examined in relation to its surrounding using the Template shown in **Figure 17**. In this template, the current point is represented by P, white points are represented by 0, black points by 1, and x is not relevant. When both Y's in the template are 1, it signifies the presence of <ain - عـ>. If the surrounding neighborhood of point P satisfies this condition, then P is identified as a segmentation point. At this point, the character can be divided vertically along the baseline's height. In an ideal scenario where the image quality is high and noise is minimal, the result resembles to **Figure 18**.

X	X	X	Y	0	0	0
1	X	X	Y	0	0	0
X	1	X	0	0	0	0
X	1	1	P	0	0	0
1	1	1	1	1	1	1
X	X	X	X	X	X	X
X	X	X	X	X	X	X

Figure 17 : The template used for segmentation

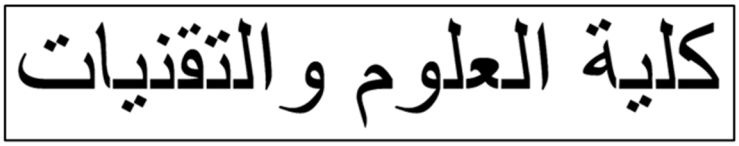

Figure 18 : Result of the segmentation by Template-Matching

For some Arabic descender characters written in specific fonts, the angle that joins the character to the baseline is below the character itself, which makes the system unable to find the segmentation point according to the template **Figure 17**. As well as characters with holes

on the baseline, such as (ط - ظ – ص - ض - مـ - و- هـ), present an additional problem for this segmentation method. For a few fonts, the system considers the inside of the hole as a segmentation point. **Figure 19** illustrates these situations. Although these points satisfy the Template condition, they should not be taken as segmentation points because the cut in these points will split the character into two or more parts.

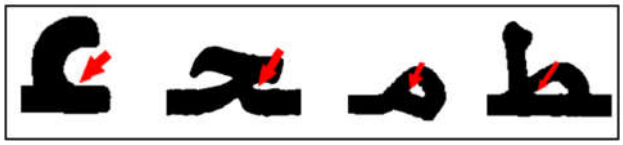

Figure 19 : Special cases of Template Matching failure

2.2. Projection-based segmentation

Previously mentioned, when an image is projected onto the standard X-Y axis, the accumulation of non-background pixel values along parallel lines is measured in a histogram known as a projection profile. For line segmentation, horizontal projection on the Y-axis is employed, while character segmentation primarily relies on vertical projection along the X-axis. To explore the application of projection, we have implemented the following two methods:

Method 1

The developers of the technique [56], suggested two algorithms to find segmentation points (SP) between connected characters. Vertical projection is obtained after eliminating satellite components like dots and diacritical marks. The first algorithm uses the average stroke thickness to determine potential cutting positions. As a threshold, the average stroke thickness is determined. If the profile in a given column is less than or equal to the threshold, a cutting point is identified. The statistical mode of stroke thickness forms the basis of the second algorithm. The vertical profile's statistical mode, or most frequent value, serves as the threshold. The outputs of the two suggested algorithms are combined, and the over-segmentation impact for the two proposed methods is reduced using the following four guidelines.

Rule 1: If the SP opens a closed loop, it shall be removed (the loop is part of a single character).

Rule 2: If the SP cuts a branch point, it shall be removed (in such cases of ص ض ط ظ).

Rule 3: If the distance between two consecutive SPs is less than a threshold, the one to the right shall be removed. The threshold is simply considered to be greater than or equal to twice the stroke width.

Rule 4: The segmentation point is retained otherwise.

Method 2

In this technique, proposed by [52]. The segmentation point between two letters is taken into consideration in the center of constant amplitude in the vertical projection profile after the subword's

vertical projection profile has been computed without dots **Figure 20**.

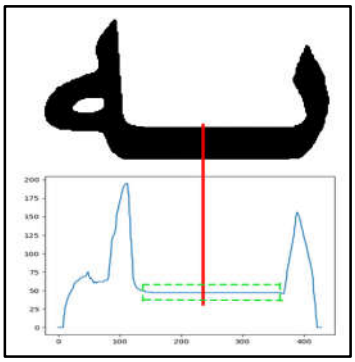

Figure 20 : Word segmentation using vertical projection

The techniques based on projection are known to be sensitive to condensed space between letters within the same word, overlapping characters, and text that is slanted. These methods are not considered for segmenting word containing ligatures.

2.3. Contour based segmentation

One of the various pre-processing methods used on digital photos to get the best data about the shape and structure is contour extraction. Contour is frequently utilized in many OCR systems to represent word shapes and aid in segmentation point detection. To explore the use of contour for segmenting printed Arabic text, we implemented the following methods:

Contour Labeling

The segmentation procedure in this approach is inspired by the work of [53], it consists of:

1- The words and subwords within are extracted during the pre-processing stage by identifying the empty spaces indicated by minimum values in the vertical projection profiles.

2- Using a convolution kernel and the Laplace edge detection, the contour of each sub-word is obtained. The upper contour is retrieved by going clockwise across the contour from the most right top black pixel to the most left down black one, while the lower contour is extracted from the left down black pixel to the right up one.

3- The segmentation stage is based on the conditional labeling of the up and down contours of each sub-word; each point is labeled 1, 0, and -1, denoting up, middle, and down, respectively. Cutting points are found when the label changes from 0 to 1 or 0 to -1.

Local minima method

Olivier et al. proposed in [30] to extract the contour of the sub-word, then extract the upper contour using the Freeman chain code. Then, browse it to locate the segmentation points at its local minima in order to segment the sub-words into individual characters.

The primary drawbacks of these two contour-based approaches are their reliance on noise density and picture quality for contour extraction. Moreover, erroneous pre-processing directly affects segmentation.

We have seen in this section a study of four methods of Arabic text segmentation. We implemented them against one corpus and exposed the advantages and drawbacks of each one of them. We have seen that projection-based methods suffer mainly from overlapping and touching characters, in addition to their reliance on precise baseline detection, just like the techniques based on Template matching. Contour-based approaches suffer mainly from noise and poor pre-processing. Therefore, Arabic printed text segmentation needs more effort in developing new techniques with fewer drawbacks and more efficiency.

3. Contribution to the Segmentation of Arabic printed text

In this section, we present our contribution to the segmentation of Arabic printed text into characters, which is a complete hybrid segmentation system based on Contour analysis and template matching. We also propose a reference corpus of printed Arabic text, written with multiple fonts, for evaluation and comparison of segmentation systems.

The acquired digital image undergoes pre-processing operations (binarization, noise reduction...). Then it is passed to the segmentation system. The text in the image is first segmented into several lines, and each line is segmented into several subwords using horizontal and vertical projections. Each subword is segmented using Template matching to extract the descendant characters. The rest of the subword is segmented using a contour-based method. The following **Figure 21** summarizes the diagram's steps of the proposed segmentation process.

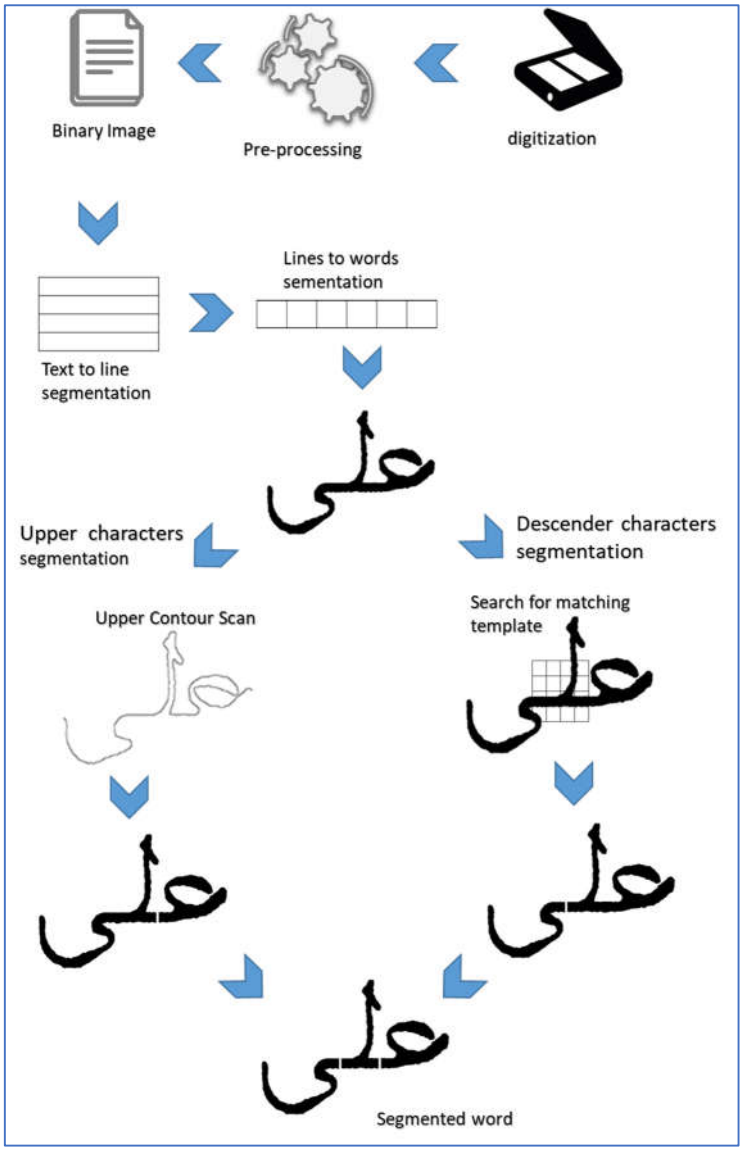

Figure 21 : The proposed system of printed Arabic character segmentation

3.1. Pre-Processing

The document containing the printed Arabic text is first scanned with a resolution of 700 dpi, loaded and then converted to grayscale. In some cases, the acquired image is of low resolution, which prevents the distinction of the continuity of the pixels. Hence, we propose to increase the resolution of the initial image two times. It should be noted that this phase must precede the binarization step, which consists of converting the image into black and white. For this purpose, we propose to use a simple binarization technique (thresholding) which consists in comparing the gray levels of each point of the image with a threshold fixed at 127.

Morphological operations

The morphological transformation modifies the value of a pixel of the image according to the value of its neighbors. For this, we use a structuring element, which is a binary mask.

It allows taking into account the neighborhood of the pixel. The Morphological operations are defined from two basic operations, which are erosion and dilation. The following definitions are given, considering the image is binary with white background (pixels = 0) and black foreground (pixels = 1).

Erosion thins the borders of the foreground object. The kernel slides onto the image. A pixel in the original image will only be given 1 (black) if all the pixels under the kernel, of which it is the center, are black. Otherwise, it is eroded to white, i.e., set to zero. On the other hand, dilation is the opposite of erosion. A pixel element is given 1 if at least one pixel below the kernel is 1. So, it increases the size of the foreground object. Normally, dilation should only be applied after erosion so as not to enlarge the noise.

Opening is just another name for erosion followed by dilation. It's very useful for reducing noise in images, because erosion removes the small noise points, but it also thins the main object. Dilation restores it to its original shape. As the noise points are eroded, they will not be restored. The Closing is the opposite of Opening, i.e., dilation followed by erosion.

Often, noise appears in the acquired image, and this is due to the quality of the digitization. This noise is in the form of additional points that are not part of the text. To eliminate these points and keep only the text on the image, we propose to use a morphological Opening with a structuring element of 8-connexity. **Figure 22** shows the result of the Opening operation.

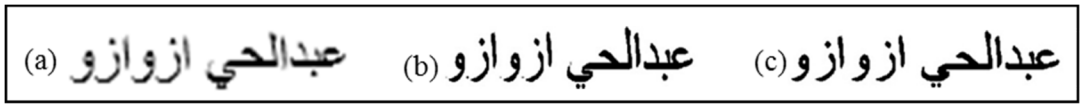

Figure 22 : (a) the original image. (b) the binary image. (c) the image after the morphological Opening

3.2. Segmentation of printed Arabic words

3.2.1. Segmentation of descendant characters

One of the main weaknesses of the segmentation methods in literature is the low accuracy of segmenting descending characters when they appear at the end of the word. Characters like (ن - ي – ز – ر) pose serious challenges if they are written in complex fonts. **Figure 23**. Our proposed method uses mainly contour analysis for character segmentation by locating segmentation points at the local minima of the upper contour of the word. However, for some fonts descender characters can’t be segmented using this approach.

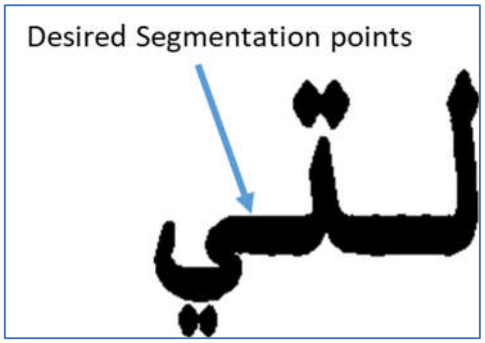

Figure 23 : Descendant characters

As a solution for this situation, we propose to use the Template-Matching method described earlier in section 2 to segment the descendant characters at the end of the subword. For this, we propose the Template of **Figure 24**.a This method can give similar results to those of **Figure 24**.b.

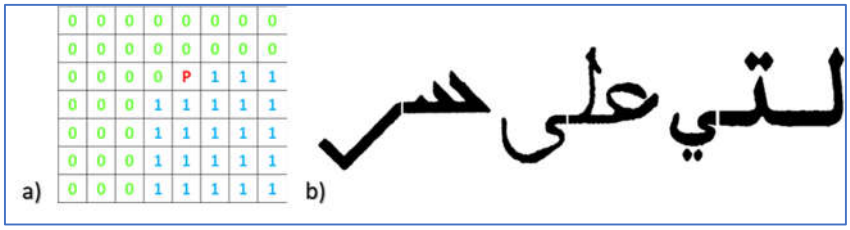

Figure 24 : (a) Template used, (b) segmentation result of descendant characters

First, the subword is scanned with the template. If there are any descendant characters, they are segmented and extracted. The rest of the subword body is then segmented using the contour-based method described in the following part.

3.2.2. Contour-based segmentation

Edges represent the defining characteristics of an object, providing crucial information about its shape. The human visual system heavily relies on edges to differentiate between objects, without having to use other features like colors and brightness. In the field of computer vision, these edges are referred to as contours. As a research area, it is specifically focused on extracting features. Its objective is to identify the edges' points of an image that correspond to sudden changes in light intensity. Edge detection plays a significant role in computer vision applications by reducing data volume and eliminating less important information, while preserving the essential structural properties of the image.

In this work, our proposal for segmenting Arabic characters involves employing the Canny edge detector [46] to identify the contours of word images. Subsequently, we utilize the Freeman chain code [44] to extract the upper section of the contour, and analyze it to locate segmentation points.

Tools

Canny detector : The Canny detector is an analytical method of contour detection. It is based

on the search for local maximums of the gradient. The Canny detector (or Canny filter) is the most widely used today for edge detection. The algorithm was designed by John Canny in 1986 to satisfy three clearly stated criteria:

- Good detection: Low error rate in contour detection.
- Good localization: Minimization of distances between detected and real contours.
- Clarity of response: Only one response per contour and no false positives.

Freeman's code : To extract the upper contour, we employed the Freeman chain code with a 8-connectivity. The success of this method relies on two factors. Firstly, it considers the direction of movement, whether it is clockwise or counterclockwise. Secondly, it takes into account the starting point. The Freeman chain code is a representation comprising numbers ranging from 0 to 7. Each number indicates the direction of the next connected pixel within the 3x3 window, as illustrated in **Figure 25**. By adding or subtracting 1 from the columns and rows, we can calculate the coordinates of the subsequent pixel. **Table 2** provides the necessary information to determine the position of the next pixel based on the corresponding code among the seven available codes.

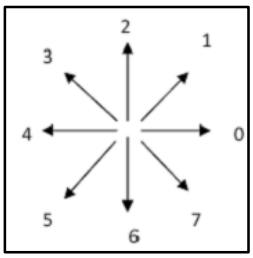

Figure 25 : Freeman's eight directions

Table 2 : Detection of the next step according to the Freeman codes

Code	0	1	2	3	4	5	6	7
Next column	X+1	X+1	X	X-1	X-1	X-1	X	X+1
next row	Y	Y-1	Y-1	Y-1	Y	Y+1	Y+1	Y+1

Word Segmentation

To segment subwords into individual characters, we propose the following process: Extract the subword contour using Canny edge detector, extract the upper contour using Freeman chain code, and analyze it to locate the segmentation points in its local minima that belong to a specific area. Finally, split the word into characters. The choice of using the upper contour came from the fact that All characters of almost all fonts are connected to the baseline at the top side. We then post-process the segmented word to improve the detection of segmentation points to reduce over and under-segmentation.

Extraction of the upper contour

After extracting the contour using the Canny detector, we proceed to browse it with Freeman's code. The contour comprises two distinct parts: An upper part and a lower part. In the case of Arabic characters, which are connected to the baseline, we focus on extracting the contour specifically from the upper side of the baseline, as illustrated in **Figure 26**.

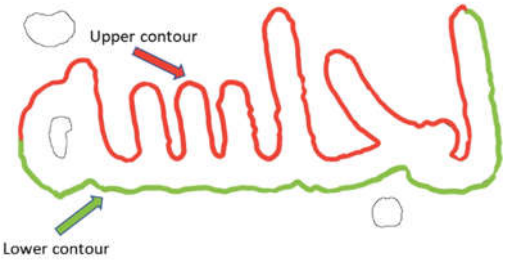

Figure 26 : The upper and lower contour of an Arabic word

First, we choose two points on the contour; the rightmost point and the leftmost one. Starting with one of them, we build a Freeman chain to the other point. Note that the first step must be upwards, i.e., one of the directions (1, 2, 3), to avoid extracting the lower contour. **Figure 27**.

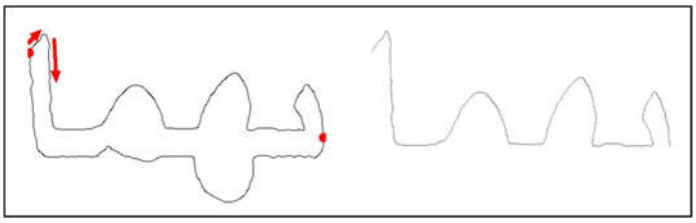

Figure 27 : Extraction of the upper contour

Detection of segmentation points

The next step involves identifying segmentation points along the extracted contour. To achieve this, we partition the word vertically into three areas: an upper area, a lower area, and a median area that encompasses the baseline and potential segmentation points. **Figure 28**. This division aims to limit the search space for segmentation points within the middle zone, allowing us to address issues related to the baseline and enable independent segmentation. The middle zone comprises the baseline and extends one-third of the baseline's height above and below it.

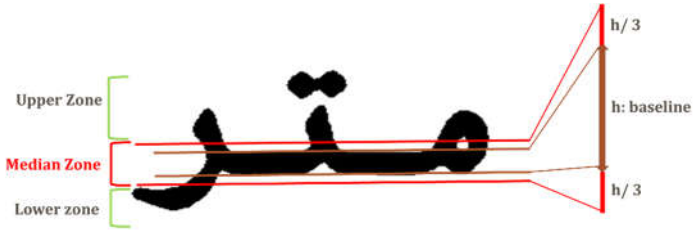

Figure 28 : The defined vertical parts of the word image

To locate potential segmentation points, the top contour is analyzed, and its local minimums are extracted, **Figure 29**.a. A segmentation area is defined as a part of the contour, belonging to the median zone, continuous horizontally, and containing a local minimum. The areas A and B in **Figure 29.a** form local minima for contour, yet they should not be considered segmentation areas. After detecting the segmentation areas, the middle of each area is chosen as the segmentation point. **Figure 29**.b.

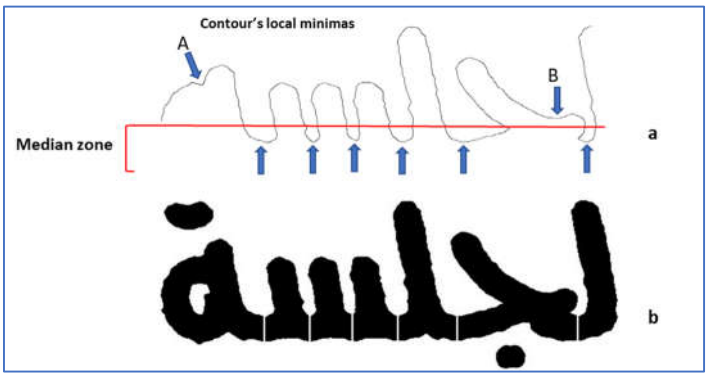

Figure 29 : (a) Segmentation areas. (b) Segmentation result

As seen in the previous **Figure 29,** the proposed segmentation system is unable to find optimized segmentation points for some characters like (ـشـ - ـسـ). It produces over-segmentation by cutting characters into several parts. This effect is due to the nature of these characters, as they contain multiple peaks. In such cases, the segmentation is enhanced in the post-processing stage by considering these segments as parts of the same character.

Overlapping resolving

For the methods described in section 2, character overlap is one of the problems that make the segmentation of Arabic words into characters a complex task or even impossible for some fonts. Overlap is detected when parts of two characters share the same vertical position. They are seen as one character. **Figure 30**. Tracing the word contour results in a closed contour for each of the subwords. The segmentation task is therefore reduced to the segmentation of each subword if it contains more than one character.

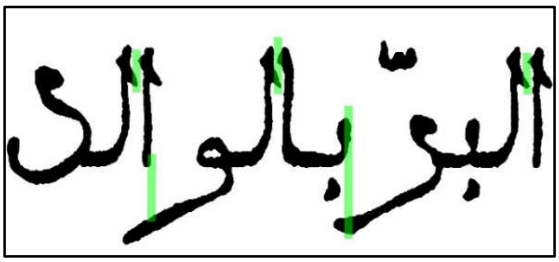

Figure 30 : Vertical overlap of characters

Figure 30 shows a sentence presenting five instances of character overlap. To segment this sentence into subwords, it is sufficient to draw the contour of the sentence. This process gives nine distinct closed contours, as shown in **Figure 31**.a. For each contour that intersects with the middle area (excluding those encompassing diacritical points), a binary image is

created. These binary images represent the respective subwords along with their associated diacritical points, as illustrated in **Figure 31**.b.

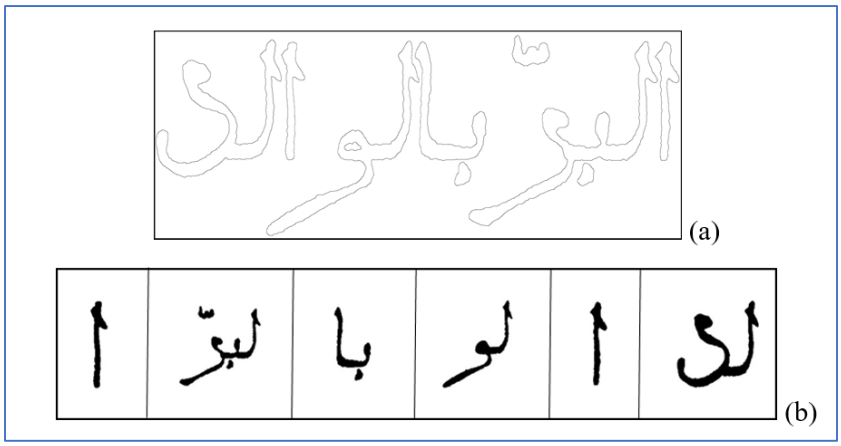

Figure 31 : (a) contour, (b) separation of overlapping subwords

Special cases

In certain fonts, the dots or vowel marks of a character might be positioned either before or after the character (in a right-to-left manner), resulting in a diacritic- that does not entirely align with the character, as shown in **Figure 32**. This presents a challenge when attempting to detect the starting and ending points of the upper contour. Consequently, it becomes impossible to establish a path between these points.



Figure 32 : Special cases of additional signs

Since dots and diacritical marks are not part of the segmentation process, the idea is to eliminate them and work only on the body of the word. To do this, we drew the contours of the word and took only the largest contour that intersects with the middle area of the word, which corresponds to the body of the word.

In the ideal scenario, there should be a single segmentation point between two characters. However, factors such as noise or poor image quality can often result in the presence of multiple segmentation points. This gives rise to the challenge of over-segmentation, where the baseline gets divided into several disconnected segments that contains no information. To eliminate the repetition of the segmentation points, we propose to validate only one segmentation point between two characters defined by their limits **Figure 33**. The others are ignored. Since Arabic text is written from right to left, the first point on the right is validated. However, this choice does not affect the segmentation.

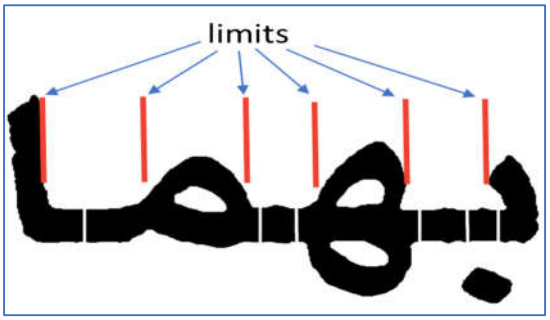

Figure 33 : Elimination of repeats

For some characters (ـكـ ـغـ ـعـ ـغـ ـعـ ـخـ ـحـ ـجـ), the segmentation point appears below the character's body. In other situations, it appears above or below a dot or diacritic sign. As a result, the character is not fully segmented. To remedy this problem, the segmentation point is shifted left or right on the silent baseline until it has no part of the character above or below. **Figure 34** shows some results of the segmentation:



Figure 34 : Improving the segmentation

3.3. Corpus construction

As seen in the state-of-the-art section, comparing the results of multiple segmentation systems is inaccurate when each system is evaluated using different datasets. To tackle this issue, it is crucial to develop a standardized and consistent corpus that can be utilized for evaluating Arabic multi-font text segmentation systems. This dataset should consider various significant font variations, including: characters that overlap (**Figure 35**.a), noisy images (**Figure 35**.b), determining the pen size and the baseline (**Figure 35**.d, c). Additionally, it should account for slanted text (**Figure 35**.f) and different gaps between characters within the same word (**Figure 35**.g).



Figure 35: Different challenges for segmentation systems of multi-font printed Arabic text

To ensure that the created corpus can serve as a reference for all segmentation techniques, the corpus text was selected at random from a variety of books and publications. It was then written in 34 different fonts with the following restrictions: 1) It contains all of the Arabic alphabet's letters. 2) All characters' shapes, 3) Characters with several peaks (**Figure 35**.e) such as (س, ش, ص, ض, ب, ت, ث, ن). 4) Letters in the start, end, middle, and isolated. 5) Character overlap. 6) Vertical ligatures of characters. 7) Various font types. 8) Various font sizes. 9) Average picture quality (Average image size, noise), as well as all the other issues listed in the previous paragraph and shown in **Figure 35**.

We proposed a corpus of printed Arabic text that satisfies all the aforementioned criteria. It has been given the name "RCATSS". To build it, we used 34 different fonts to write 83 text lines with a total of 984 words and 4518 characters. We mention that the fonts used are publicly available, and they listed in **Table 3**. The corpus was made freely accessible to public research community at https://github.com/Abdelhay-Z/RCATSS. **Figure 36** shows a sample of the proposed corpus.

Table 3 : The fonts used for the corpus construction

Courier New	AL-Battar	AdvertisingLight	Mohammad Laha
AL-Sayf Bold	AF_Jeddah	MCS Masahif S_U normal.	khalaad al-arabeh
A Mosalas	FS_Cairo	Times New Roman (En-têtes)	AF-Al Nassrah
Yakout Linotype Light	Al-Hadith2	SC_AMEEN	Aldhabi
ACS Almass	DecoType Naskh Extensions	Arial (Corps)	Andalous
ACS Morgan	FS_Ahram	Traffic	Sakkal Majalla
AF_Buryidah	MCS Jeddah S U normal.	Segoe UI Semilight	Almarai
AGA Nada Regular	Hesham Free	arabswell_2	Aqmar
Cairo		Moslas	



Figure 36 : A sample of the corpus

In this section, we have presented our contribution to the segmentation of Arabic printed text into characters. We have proposed a hybrid segmentation system based on text contour and sliding window scanning (Template-matching), and we have dealt with the different problems that a segmentation system may encounter. We have also proposed a reference corpus to allow the evaluation and comparison of segmentation systems of printed Arabic text.

4. Experimental Results

The practical part of this work is presented in this section; The development environment, the different parameters, and the results of different segmentation systems. The segmentation

methods described in section 2 and our proposed method described in section 3 are all implemented and tested against our proposed corpus of multi-font printed Arabic text.

All the experiments were implemented in the Python-OpenCV environment (Python 2.7 with OpenCV 3.0 library). This environment is known for its simplicity as it eases the implementation of different ideas and helps researchers by providing a lot of predefined functions of image processing such as binarization, morphological operations, canny edge detection, and many others.

The input to all of the studied systems is a multi-line text image taken from the proposed corpus. The expected output is the separate individual characters of each line. For the sake of simplicity, we suppose that the input image is binary with a correct skew angle, well pre-processed, and noise-clean.

For line and subword segmentation, the basic horizontal and vertical projections were used. The given success rate is 100% since the lines are well separated by enough spaces with no overlapping.

The following results describe the correct character segmentation rates of each method. **Table 4** summarizes the segmentation results of each method. The segmentation is evaluated manually, character by character, and the rate of segmentation is given by:

$$R=CC/TC$$

Where CC is the number of correctly segmented characters, and TC is the total number of characters in the corpus, which is 4518 characters. To consider a character being correctly segmented, it has to contain the global shape of the character with no additional parts of other characters. Otherwise, it is counted as a failed segmentation.

Table 4 : The results of segmentation of the different methods

Method	Character correctly segmented	Rate of characters correctly segmented
Template-Matching	3608	79.84 %
Projection method 1	3766	83.33 %
Projection method 2	3924	86.83%
Contour labeling	4042	89.44 %
Contour's local minima	3894	86.16 %
The proposed method	**4281**	**94.74 %**

Because our proposed method benefits from the advantages of the contour-based and template-matching approaches, the rate of correctly segmented characters has increased to 94.74%. Our approach, which was discussed in the previous section, has the best capacity to separate multi-font and multi-size Arabic text into individual characters when compared to the other methods. **Figure 37** shows some of the successful segmentations. Also, this comparison demonstrates how having a single reference corpus provides a clear view of how strong various segmentation techniques are. **Figure 38** shows the segmentation result of the same

word by using the six different methods.



Figure 37 : Samples of correct segmentations

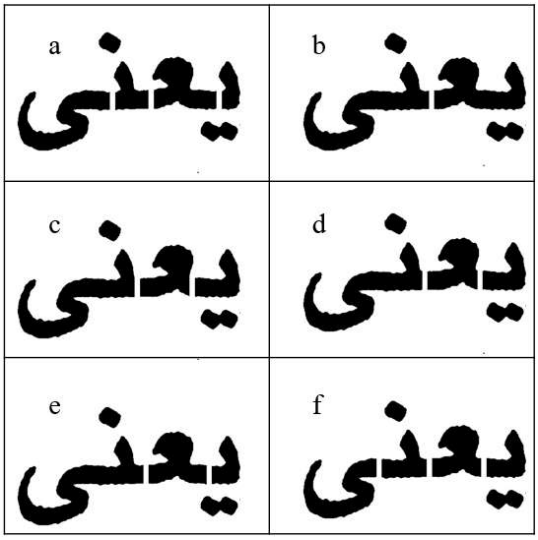

Figure 38 : a) projection method1; b) projection method2; c) template matching; d) contour labelling; e) contour local minima; f) our proposed method

The segmentation operation so far consists of creating the gap between the characters in the segmentation point. The extraction of individual characters is done by drawing the contours of the segmented word. Each closed contour belonging to the middle area represents a character. It is extracted and saved in separate image as shown in the **Figure 39**.

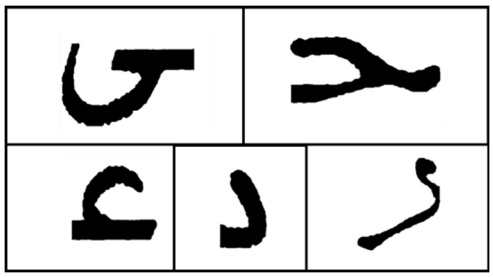

Figure 39 : Individual segmented characters

The calculated success rate of word-to-character segmentation, using the proposed hybrid method, is 94.74%. however, **Figure 40** shows some cases of failed segmentation:

- Characters (ـه) connected below the baseline cannot be segmented because, for the contour, there is no local minimum between (ـهـ) and(ـه).
- b, d: Due to poor scan or binarization quality, characters may touch above or below the baseline, and they are seen as one character by the machine. The problem of touching components in still a non-solved problem for many segmentation systems.
- c: A wrong estimation of the baseline can also give a wrong segmentation. Here the extended part of the character (ن) is considered the baseline for the subword.
- g, f, e: Poor digitization quality can give rise to noise that involves contour clipping, especially in the corners. This results in an open contour of the subwords, and consequently, segmentation becomes impossible.

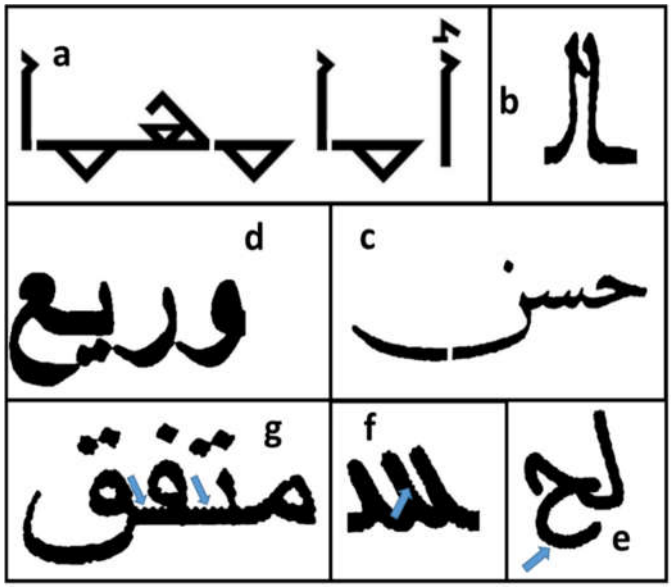

Figure 40 : Cases of failed Segmentations

Conclusion

In this chapter, we have explored the field of Printed Arabic text segmentation for recognition purposes. We reviewed the main Arabic printed text properties and the different challenges that may arise in a segmentation system. We then discovered the different approaches used, and we analyzed several works done in the field of segmentation of printed Arabic text. In order to compare the different segmentation methods, we have implemented and tested them against a standard corpus of 4518 characters written in 34 different fonts. This corpus is built taking into account all the cases that may pose difficulties for a segmentation system.

In our main contribution, we focused on the most challenging phase in an Arabic OCR system, that of segmenting the text into characters, which remains a vast area of research. The segmentation of Arabic text presents several difficulties. This particular stage is prone to a higher occurrence of mistakes, and any errors in segmentation during this phase can subsequently result in classification errors. We have proposed a new hybrid method for segmenting Arabic text into characters based on the contour and Template-Matching approaches. This new method is studied and developed in such a way that the segmentation is performed with minimal errors and a maximum success rate for Arabic text printed in multi-font and multi-size. With a decent rate of characters accurately segmented at 94.74%, the results have demonstrated the effectiveness of our hybrid technique in comparison to the other methods.

The optical processing of the Arabic printed text is indeed a complex operation, it requires rigorous attention especially for segmentation task. However, processing handwriting is more challenging as it encompasses dealing with variation in writing style and the absence of structural writing rules. Sequentially, processing ancient writing present another level of difficulty. In such texts, text lines are written very close to each other with many additional signs in between. Therefore, the projection-based methods are unable to segment lines as for the printed text. in the following chapter, we study line extraction from historical Arabic documents, written with the ancient Al-Mojawhar style.

Chapter 2. Text line extraction of Arabic historical documents

Introduction

As mentioned in the previous chapters, Optical character recognition (OCR) for Arabic text is currently not as advanced as it is for Latin text. Furthermore, when it comes to historical documents digitization, progress is even slower, and there is a scarcity of relevant research in the existing literature, despite the existence of thousands of ancient documents in libraries and archives. These ancient documents contain valuable records, either artistic, scientific, or related to historical events. However, the processing of ancient text is quite different and more complex than current printed or handwritten text, especially for Arabic text. As a result, the ancient Arabic documents have not yet been digitally exploited. Due to the nature of Arabic writing and the variety of letters' shapes, the tasks of segmentation and recognition are much more challenging compared to Latin writing. The majority of ancient Arabic text has several additional signs and diacritics between lines.

An analytical character recognition system relies on text segmentation, starting with line extraction. In this operation, the text must be broken down into its constituent lines, then into words, and finally into characters. Clearly, no errors are allowed in the first step because they will affect the following levels of segmentation and consequently will result in false recognitions. The extraction of lines from printed text is considered the easiest, as enough space well separates the text lines. While for ancient handwriting, text lines are often written randomly, and the spacing between them might often vary within a single document. All of these facts result in the overlapping and touching of lines. When the ascending and descending characters of two subsequent lines are at the same horizontal position, the effect of lines overlapping occurs. Alternatively, if a descender from the first line crosses an ascender from the second line, it will create the touching lines' effect.

In this chapter, we address line extraction by first presenting a pre-processing method for skew correction, before introducing our method of text line extraction from Arabic historical text. This chapter is organized as follows: section 2 presents some earlier efforts of text line extraction. Section 3 discusses the contributions of this work, where we first go through the pre-processing procedures that will be used on the text image. Then, we propose a projection-based technique to resolve the skew of the text and extract text lines by determining the separation path. Section 4 presents the experimental study of this work, presenting the different results of segmentation and skew correction. Finally, Section 5 presents the conclusions.

1. State of the art of line extraction from Handwritten Arabic text

The segmentation of overlapping and touching lines is one of the most challenging obstacles for any text line extraction system, especially when working with skewed and

condensed text. Finding the ideal segmentation path between lines of handwriting, whether for Arabic or other languages, was a constant challenge for segmentation systems. A variety of strategies have been adopted to address the line segmentation problem.

A method for text line segmentation was developed in 2001 [57]. It is used on modern handwritten text. The approach is based on a fusion of partial projection and partial contour analysis. According to the authors, the overall success segmentation rate is 82%. However, touching lines and the presence of diacritical marks cause the procedure to fail in some cases.

Since then, a variety of strategies have been created to address the unique case of text line extraction from historical texts, whether for Latin [58]–[62], Chinese [63], [64], or Bangla [65]. However, there is still a significant lack of focus on the work on Arabic text.

The authors in [66] divide a page into columns and use partial projection to divide each column into blocks in order to segment Arabic text lines. Finally, based on the size of each block, they utilize a KNN algorithm to divide each block into three classes: a class of big blocks indicating overlapping components, a class for diacritical marks, and a third class for words. The technique has been tried on printed and handwritten Latin text as well as printed Arabic text; however, the final success segmentation rate was not provided.

The work in [67] describes a technique for text line extraction from old Arabic. The initial step of this approach involves employing automated paving and the Winger-Ville Distribution to find multi-skewed zones. After that, lines are extracted using an analysis of the direction of each zone and the baseline. According to the authors, the segmentation rate is 98.6%, which may be considered very high, taking into account the privacy of old Arabic handwriting. Due to the diacritics that appear at the beginning of the line, this approach is still susceptible to false maxima and text inclinations.

Line extraction has recently also used machine learning techniques. The authors of [68] utilized a convolutional neural network model to extract Latin text lines from old manuscripts. Initially, a patch size is determined based on the length of the entire text line. Many patches are sent to CNN for training. Next, line segmentation is performed using the CNN model. Two sets of Latin datasets have been used to examine this strategy, but no recorded results have been made public. Because of the lack of a standardized database with sufficient data on ancient Arabic text, CNNs cannot yet be used to analyze historical Arabic texts.

In [69], a method of text line segmentation and binarization of historical manuscripts is proposed. The authors used fast and adaptive bidimensional empirical mode decomposition (FABEMD) [70]. The extraction method starts with a thresholding algorithm that decomposes the image into two components using the EMD decomposition method [71]. Moreover, it uses a combination of Niblack’s and Savaula's algorithms [72], [73]. The main goal of the binarization algorithm is to prepare the document for text line extraction using the horizontal projection profile method, which is widely used for text line extraction for all script languages. The main drawback of this method and all methods based on projection is the weakness of dealing with overlapping or touching lines, especially in cases where the document contains the skewed text.

All the mentioned works may encounter issues of weak pre-processing, especially from low binarization and skew correction. Due to storage conditions of usage accidents, the backgrounds of ancient documents often lack color normalization, potentially resulting in variations in color across different parts of the document. Consequently, the simple binarization algorithms may fail to find the optimal threshold. Additionally, an important challenge facing segmentation systems is the incorrect skew of text. Given that ancient documents were written randomly without guidelines, the whole text or parts of it may be skewed. **Figure 41.** Skew correction has been the subject of much research over the years. However, it is still a non-resolved problem, at least for an Arabic document. Therefore, we firmly believe that, for accurate segmentation of Arabic historical text lines, an advanced pre-processing must be done to correct the skew angle , find the best binarization threshold, and reduce noise.

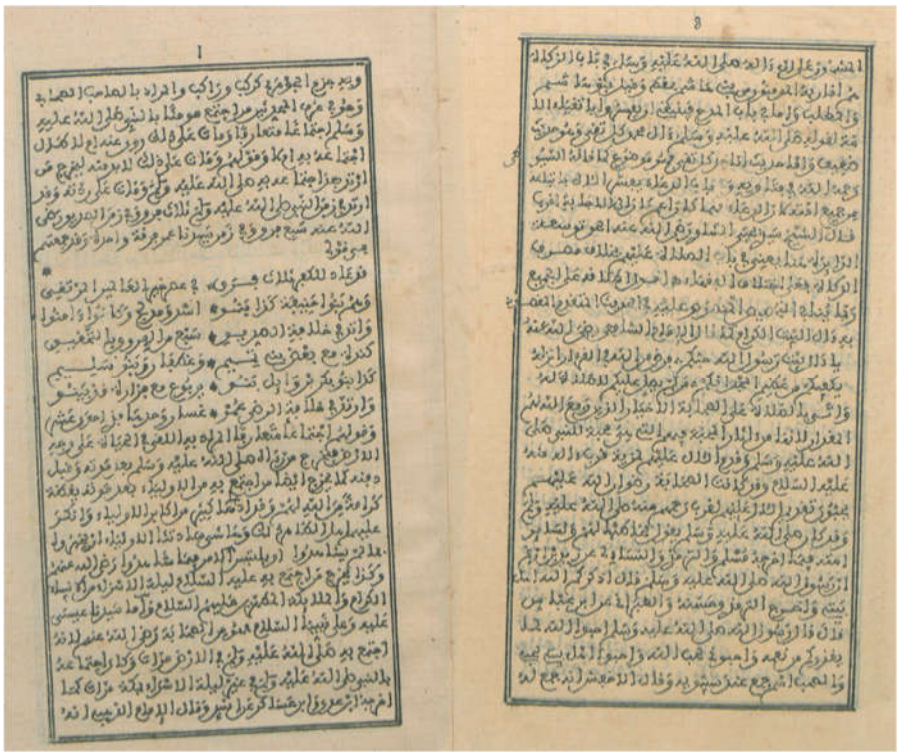

Figure 41 : Raw ancient Arabic skewed and noisy document

2. The proposed method of Skew Correction and Text Line Extraction from historical documents

Text line segmentation is the process of decomposing a single text page into individual lines and extracting them with all the composing words, subwords, and signs. This work uses a projection profile-based approach for segmentation. The projection profile as stated before,

refers to the running count of the black pixels in a text image's particular direction [74]. In the case of text line extraction, the horizontal direction is followed.

Because ancient records were typically handwritten and the equipment available did not aid in writing in a straight line, the text is frequently skewed. Arabic writing was more challenging because lines were written too closely together with several ascenders, descenders, and other additional signs. This frequently results in the issue of multiple lines overlapping and touching one another. The issue has always been the fundamental obstacle for Latin or Arabic text segmentation systems. Despite the best efforts of the most accurate systems, the issue of overlapping and touching lines still has not been completely solved.

In order to solve the line segmentation problem of historical Arabic manuscripts, we propose in this chapter a method based on projection, first to correct the skew angle of the entire text page all at once, then to separate and extract lines. Before segmentation and skew correction are conducted, the text image is initially subjected to several pre-processing procedures, such as binarization and noise reduction. Four alternative image formats of the document are used to test the approach. The process of this study is summarized in **Figure 42**. Three transformations were used to produce these representations, including smearing with the Run length Smoothing Algorithm (RLSA) technique [75], conversion to up-to-down binary transitions, and Gaussian smoothing. The separation operation is done by finding the best separation path that goes between lines.

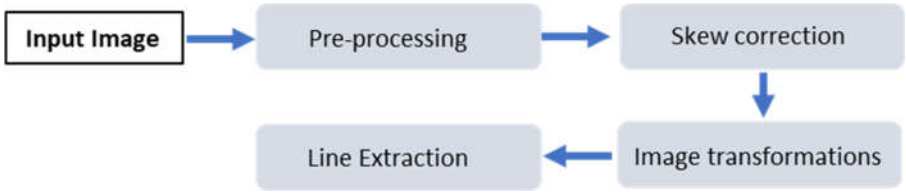

Figure 42 : The proposed process of segmentation

2.1. Pre-Processing

To our knowledge, there was no standard pre-processing algorithm dedicated to preparing historical documents for segmentation. This is due to the variety of operations needed and the fact that each document may present different challenges. However, noise, borders, and additional objects are common problems found in all documents of our dataset. Some document images may have non-uniform illuminations in the document background due to the use of multiple paper types, users' accidents, long storage conditions, or it is simply because of false scanning. **Figure 43**. Therefore, each document was prepared manually to split each double-page into individual pages, and to eliminate borders and objects such as paginations, stamps, ink bleeds, and side notes.

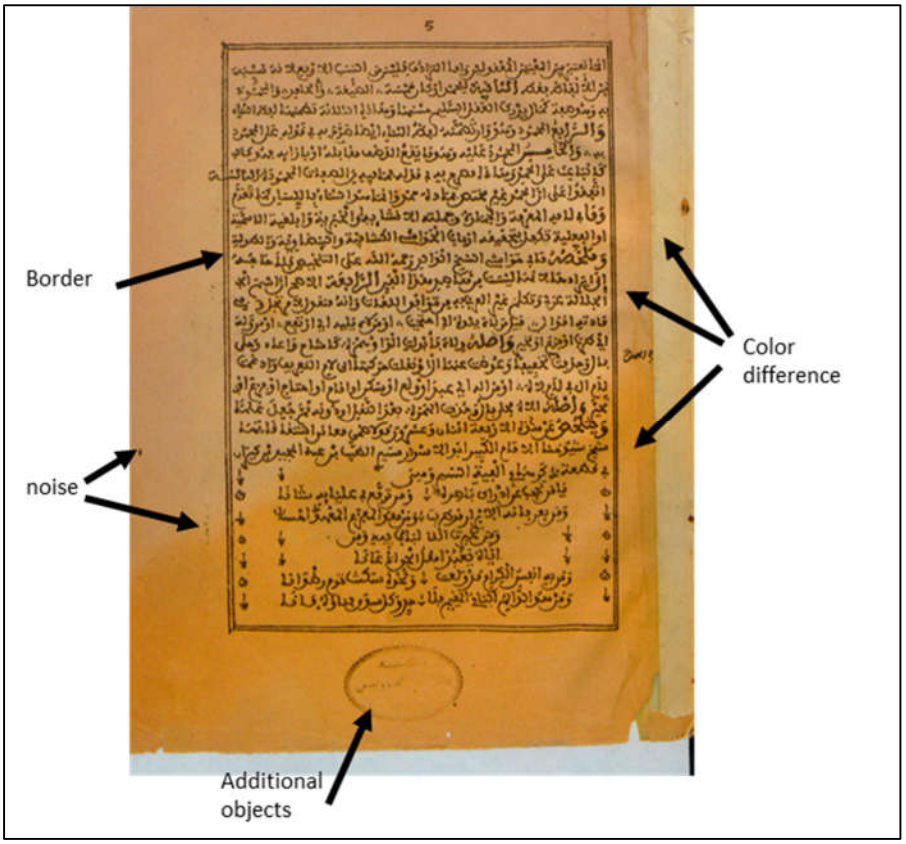

Figure 43 : A page of a historical document

For binarization, we studied the use of two different approaches; Global thresholding, represented by Otsu's algorithm [76], and local thresholding, represented by Niblack's algorithm [72]. Otsu's thresholding is a popular image-processing algorithm that automatically determines the best threshold for image segmentation. The algorithm partitions pixel intensities into two classes, calculates the variance between them, and determines the threshold that maximizes the variance. This results in a distinct separation between foreground and background regions in the image. Due to its effectiveness, Otsu thresholding is widely used in object recognition, medical imaging, and document analysis. Conversely, Niblack's algorithm is a local adaptive thresholding method used for image segmentation. It determines the threshold value for each pixel by analyzing the local mean and standard deviation of its neighboring pixels. To this end, the algorithm first calculates the mean and standard deviation of the pixel intensities in a small neighborhood around each pixel. Then, it subtracts a constant factor (typically between 0.1 and 0.3) times the standard deviation from the local mean to obtain the threshold value. This approach enables Niblack's algorithm to handle variations in lighting and contrast within the image, making it effective for segmenting images with uneven illuminations. This algorithm is frequently employed in tasks such as document image analysis and handwriting recognition. **Figure 44** shows the results of binarization using the two algorithms.



Figure 44 : a) original Image, b) grayscale, c)Otsu's thresholding, d) Niblack's thresholding

As seen in **Figure 44**, adaptive thresholding with Niblack's algorithm gives better results for images with non-uniform illumination. It is also good with noise removal, compared to Otsu's thresholding.

To reduce noise, we first applied morphological Opening with a kernel of 3x3, which is nothing but an erosion followed by dilation. Then we used a median filter based on a kernel of 7x7 regarding the high resolution of the image.

2.2. Skew correction

The issue of text skew correction has been addressed for decades. Since then, several methods have been developed. However, the most popular and effective are still the ones based on Hough Transform [13] and projection [74]. These two approaches have known many improvements. However, our skew correction method is based on horizontal projection, which was chosen for its simplicity, effectiveness, and low processing time cost. The approach is

based on computing the highest peak of the text image's projection profile across a number of clockwise and counterclockwise rotations around the vertical center of the text page. Nevertheless, this approach is acknowledged to have shortcomings due to overlapping and condensed lines. Because the orthogonal of a condensed text image frequently contains more pixels than lines; as a result, when rotating the image and calculating the projection profile, the maximum peak corresponds to this angle. So, it is possible to find a fake angle when applying the algorithm of skew correction to the original binary image. **Figure 45**. To overcome this issue, we propose transforming the text image into its up-to-down binary transitions **Figure 46.b** . The binary image's columns are only preserved with their black-to-white transitions as the up-to-down transitions reflect the baseline position and can thus be processed alone to determine the skew angle. The major objectives of this transformation are first to overcome the issue of condensed lines. Secondly, to minimize the processed information to speed up the segmentation process.

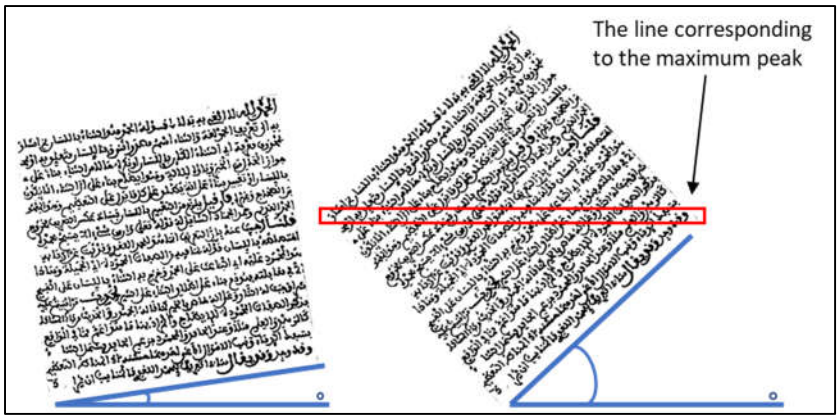

Figure 45 : Fake angle of skew correction

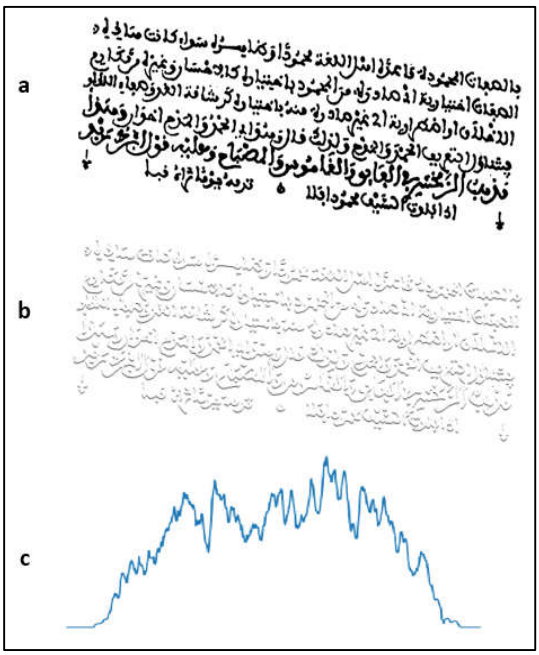

Figure 46 : Skew correction. a: original. b: transformation. c: projection profile

The rotational angle range for the transition image is [-a, a], where "a" is a fixed angle value. As illustrated in **Figure 47**, the projection profile is determined for each rotation, and its highest value is stored. The underlying idea is that the optimal rotation angle, which produces straight lines since horizontal projection is utilized, will result in the highest peak.

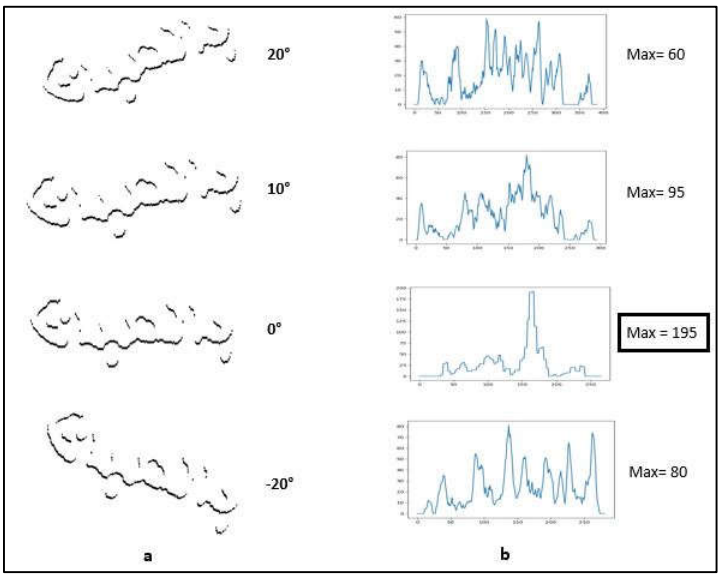

Figure 47 : Detection of the skew correction angle

2.3. Image transformations

The separation of lines from the input image is a complex task. Most algorithms fail to find the best separation path because of the condensed spacing and variations of ascender and descender characters. Therefore, we propose first to transform the image into a more compact format to reduce the character variations. To this end, and based on the literature study, besides the original binary version of the text image, we propose three transformations for evaluation. We applied the segmentation procedure on all four alternative versions of the text image. **Figure 48** shows the different versions of the word. While segmenting an image, the results are compared to the original binary image using each representation. The Run length Smoothing Algorithm (RLSA) is used to produce the first transformation [75]. The technique smooths the area between two black pixels in a single row or column. **Figure 48.b**. If the white space is smaller than a certain threshold, it would be blackened. It may be used in either vertical or horizontal directions. The second transformation involves converting the original binary image into its black-to-white transitions from left to right and up to down. **Figure 48.c**. The decision to maintain right transitions was inspired by the fact that Arabic is written from right to left. Bottom transitions, on the other hand, are justified because they represent the line's location relative to the baseline of the word. A binary Gaussian filter BGF is used for the third and final transformation. Similar to RLSA, the BGF transformation involves adding gray pixels between black pixels using a Gaussian blur filter. The blurring method known as Gaussian blur estimates the changes to be applied to each pixel based on a Gaussian function. The additional gray pixel is then turned into a black pixel via thresholding. **Figure 48.d**.

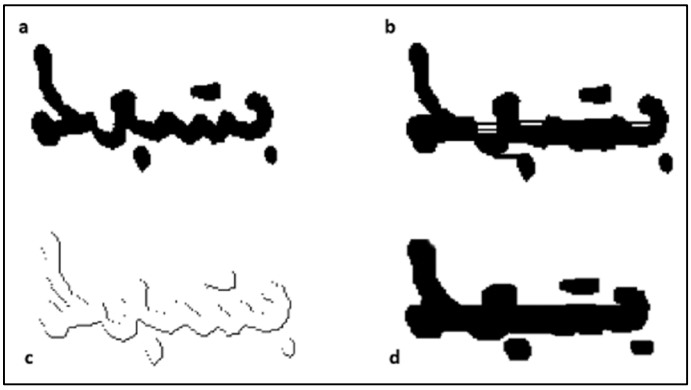

Figure 48 : The applied transformations. a) original image, b) RLSA, c) transitions, d) BGF

2.4. Line extraction

Historical documents often have text lines that are written too closely together. As a result, the produced horizontal profile is noisy, and the locations of the lines are difficult to locate. Therefore, we use the SavGol filter [77] to smooth the projection vector in order to get around this problem and locate the global lines' locations as the local maxima of the projection profile.

The segmentation area for every two consecutive lines is next located, represented by the region between the two global baselines. **Figure 49**. Next, local maxima representing the global line locations are found. We proposed the algorithm of **Figure 50** to draw the segmentation path and separate lines from right to left, starting with the right index of the local minima of the segmentation region. Let us note that *Du* (*Dd*) is the distance between the current pixel and its nearest up (down) black pixel, PP is the projection profile, and up and down thresholds are set to 20% of the segmentation area height. Starting from the right-most point at the center of the area, the cursor follows the white space ahead to the left. When a black pixel is met, the distances *Du* and *Dd* are calculated to check whether the path goes up or down. So, the next jump follows the direction of the highest distance to avoid touching the closest threshold.



Figure 49 : Segmentation area between two consecutive lines

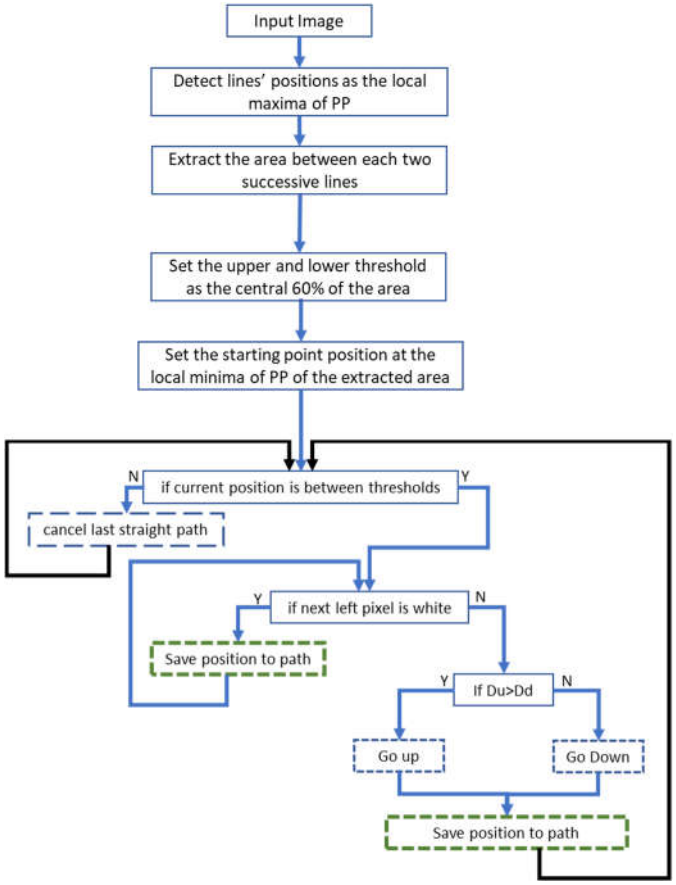

Figure 50 : The proposed algorithm of lines segmentation

As shown in **Figure 51**, if the path is blocked by a black pixel in all forward directions (path number 1), it means that overlapping may have occurred. In this particular case, the path cancels the last move and tries again in another direction. The process is repeated until it finds its way through the white space again (path number 2).

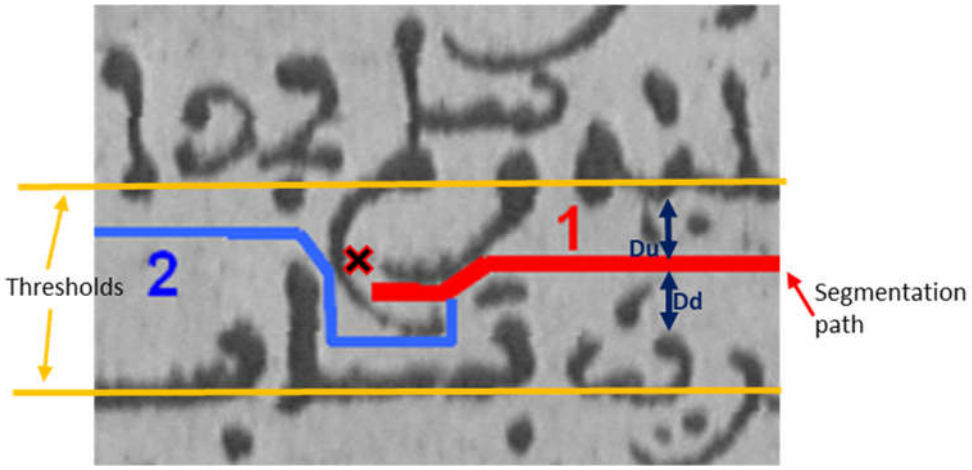

Figure 51 : Overlapping resolving

Alternatively, if following the segmentation procedure leads the path to touch one of the thresholds set previously, the last straight path will be canceled, and the cursor will go back

to the last vertical jump. It follows again the white space in another jump direction other than the one that caused the threshold touching. If the path runs out of all the directions' possibilities, it means the two lines are touching, and the path does not find any white space between lines. In this very special case, the algorithm calculates all the cut possibilities and selects the one with the least amount of crossed black pixels in order to minimize data loss. **Figure 52**. We believe that with this overlapping and touching resolving process, the text line would be segmented with minimal data loss, preserving most of the characters' shapes.

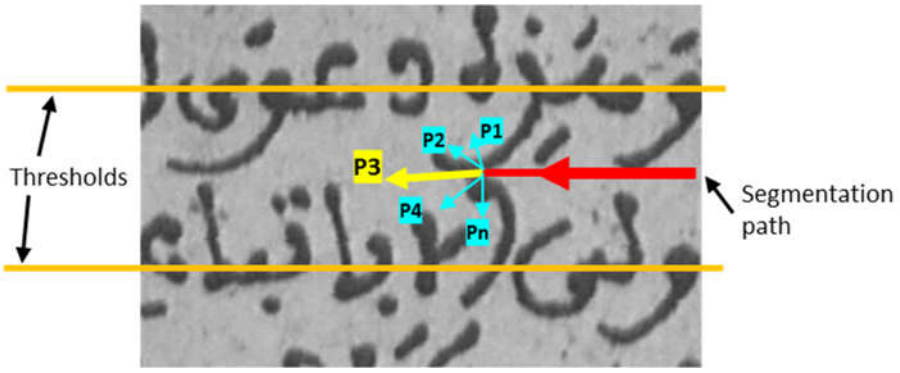

Figure 52 : Touching line segmentation

In **Figure 53**, a segmentation region is shown together with the segmentation thresholds and corresponding segmentation path.

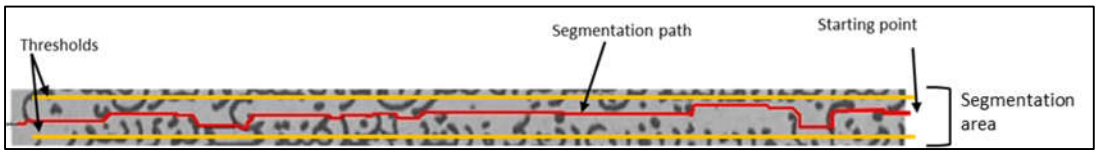

Figure 53 : Example of Line Segmentation

3. Experimental study

Environment

To our knowledge, there is no publicly accessible and standardized database of Arabic historical documents to test text line extraction systems. Therefore, we used a private set of Arabic historical documents to test the current segmentation method. The Al-Qaraouiyine Library provided the documents digitized with a 300-dpi optical scanner. The collection has a total of 129 pages and 3068 text lines. These were manually prepared to remove borders, large black spots, and other effects that users had left behind during manual use. The experiments were conducted using Python-OpenCV development environment installed on a Windows 10 computer with a quad-core processor of 2.8 GHz and 8 Gb of access memory.

Rules

Finding a path through the white space between the lines is the foundation of the segmentation technique discussed in this work. We observed that there is no established method in the literature for determining if a line has been properly segmented. Because of this, we set

the following tight criteria for segmentation success: A line segmentation is manually reviewed. If the segmentation path crosses (cuts) one or more of the line's characters under the condition that a crossed character loses its global form, the segmentation is deemed unsuccessful. Several successful and unsuccessful segmentations are shown in **Figure 54**.

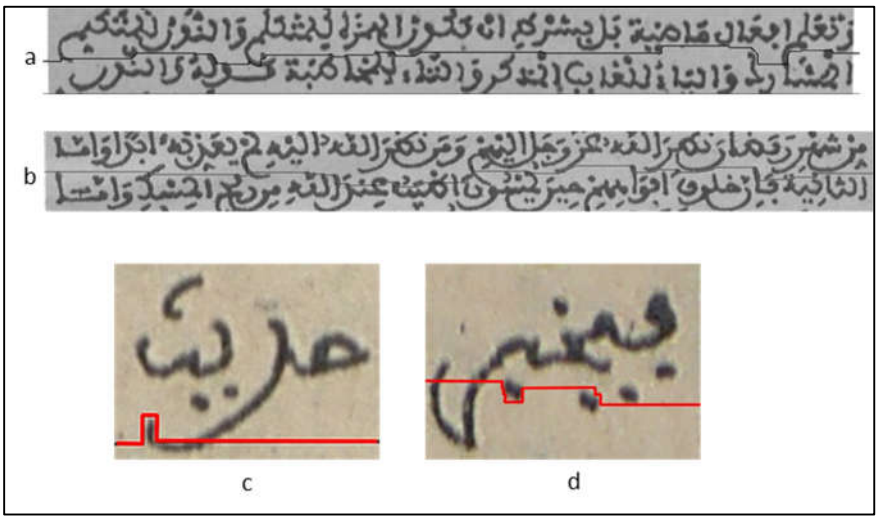

Figure 54: a) & c) : Successful segmentations. b) & d): Failed segmentations

3.1. Skew Correction

The search range for the angle of skew correction is empirically determined as [-40°:40°] in order to consider any rotational deviations that could be present in ancient texts. The best step angle of rotation is calculated after a number of experiments summarized in **Figure 55**. We divided the rotation range into multiple divisions, which gave several potential rotation angles. Then, for each skew correction iteration, we calculated the time cost, as well as the angle precision, as the difference between the calculated angle *(ca)* and the exact angle *(ce)*. The calculated angle is the angle that gives the highest peak of the projection profile, and the exact angle is determined manually. We found that time increases as the number of steps rises as there are constantly new iterations to perform. The rotation angle of 0.55° gave the best results; however, considering time cost, the rotation angle of 1.14° gave the best *(ca)* results with the optimized time cost.

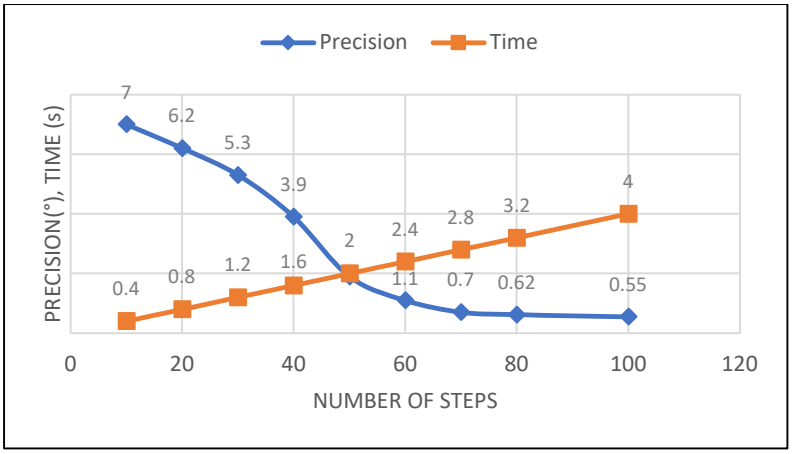

Figure 55 : Summary of skew correction tests

3.2. Text line segmentation

We applied the proposed segmentation algorithm on the three mentioned versions of the text image along with the original binary image. In **Figure 56**, we plot the segmentation results regarding error rate and time cost. The formula (2) is used to determine the error rate. A global binarization technique is utilized for the first representation, in which the image has only been binarized, and the noise has been reduced by applying a morphological 3-iteration opening with a 2x2 kernel. The threshold in the horizontal-RLSA representation was empirically determined after several tests in 15 pixels. The same kernel is used for the Gaussian filter.

$$e = \frac{|m-n|}{n} * 100 \qquad (2)$$

Where 'm' is the number of lines well segmented, and 'n' is the total of lines in the database.

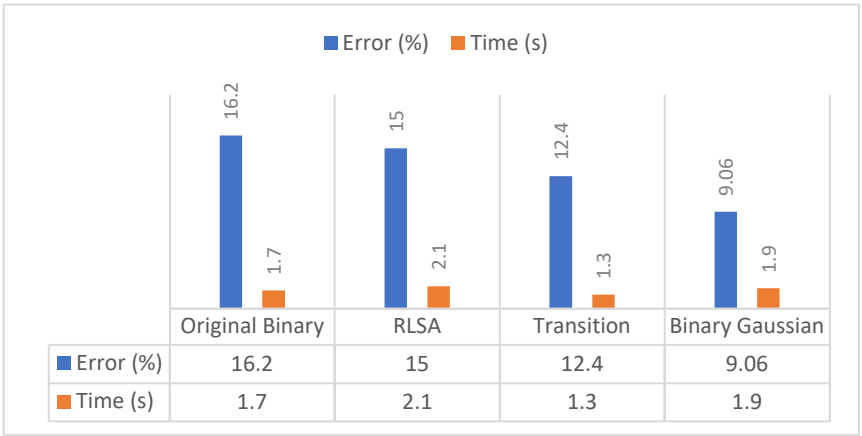

Figure 56 : Segmentation rates and time cost

Processing the image of the transition produces the fastest segmentation since it requires less data than the other images. However, the segmentation with binary Gaussian transformation takes longer than the segmentation with original binary and the transitional image, while still producing the highest rates in terms of segmentation accuracy. This is because

the binary Gaussian transformation produces a more compact version of the words than the RLSA transformation by reducing noise, blending, and smearing the edges. In contrast, the RLSA method blurs the text image without removing noise, which may result in the segmentation path being lost. While for the transitions' transformation, more space and route options for the segmentation path are created. The segmentation results are summarized in **Table 5**.

Table 5 : Results of segmentation using the four image representations

Representation	Segmentation rate
Original Binary	83.8%
RLSA	85%
Transitions	87.6
Binary Gaussian	**90.94%**

Conclusion

The focus of this chapter is on presenting a projection-based method for text line segmentation of historical documents and addressing common issues such as skew and noise. We performed a pre-processing step that involved binarization and noise removal. To provide straight lines for the segmentation process, we introduce a skew correction method based on the horizontal projection of the transformed text image into transitions, by estimating the best rotation angle required to correct the skew of each document. The idea of the segmentation algorithm is to find the best separation path between lines; on one side, it provides an optimized segmentation, while on the other side, it resolves the overlapping and touching lines' effect by minimizing data loss.

We tested the segmentation algorithm on various versions of the text image. Four versions of the image were investigated. Besides the original binary image, the RLSA algorithm and Gaussian smoothing filter were used to obtain more compact text shapes to minimize variations. Also, the vertical black-to-white transitions were extracted and used as the fourth image format. For each image in the dataset, we evaluated the segmentation algorithm on each transformation. We found that applying a Gaussian blur followed by binary thresholding gives the best compact image, and consequently yields the best results of text line separation, with a segmentation rate of up to 90.94%.

We believe that this study helps the researchers to develop more robust systems, able to extract text lines from ancient documents. However, the line extraction is only the first level of document segmentation. In spite of the importance of this segmentation stage, and despite the efforts being devoted to the digitization of ancient document, the biggest problem facing such systems is the segmentation of lines and the extraction of the words and subwords. In addition, the lack of a standard database of ancient Arabic poses a substantial impediment for conducting

research on text recognition. In the next chapter, we introduce a new method of subword extraction, besides a novel and big database of ancient Arabic words and subwords, written in Al-Mojawhar ancient style.

Chapter 3. Building MOJ-DB database with new approach of Subword extraction from historical document.

Introduction

The Moroccan manuscripts written in Al-Mojawhar are invaluable with their cultural and historical content. They can be found scattered throughout some of the oldest countries in the world. Preserving them is of utmost importance as they hold vital records in various fields. To protect their contents, it is crucial to digitize these manuscripts. However, despite their significance, the research on Optical Character Recognition (OCR) for Arabic historical documents is severely lacking, with minimal progress in recent years. This lack of research limits our ability to effectively study and understand these manuscripts. In fact, the research on Arabic historical handwriting recognition faces several challenges, mainly due to the lack of accurate systems of segmentation and standard accessible databases of optical text. We have seen in the previous chapter, how difficult is the extraction of text lines from ancient documents. Nevertheless, segmenting lines into word and subword is much more complex and requires additional effort in all aspects.

On the other hand, constructing a comprehensive database for Arabic historical writing is not as simple as for modern handwriting, involving challenges in text segmentation and database annotation. Overcoming these challenges is necessary to push the research on OCR for ancient Arabic documents and advance our understanding of these valuable manuscripts.

This chapter presents the complementary segmentation of historical text. We propose an approach for text line segmentation into subwords, which helps improve the recognition of Arabic historical handwriting. The proposed method is then used in the process of creating a new database of Arabic historical words and subwords. This database is primarily designed for offline recognition and word-spotting systems for Arabic historical script.

The next section of this chapter presents a literature review of text line segmentation and Handwritten Arabic databases. Section 3 describes our method of text line segmentation and subwords extraction. Section 4 presents the process of building the MOJ-DB database, which includes. Section 5 provides details on the experimental study and validation process, and compares the results with other works. Finally, in section 6, a conclusion is presented.

1. Literature review of subword extraction and Handwritten Arabic Databases

1.1. Arabic handwriting text line segmentation

The length of spaces in Arabic printed and handwritten text holds essential significance, as they serve to separate words from each other within a single line of text. These spaces, whether inter-word or intra-word, play a crucial role in the readability of the written text.

Interestingly, historical Arabic writing sometimes features spaces of equal length between subwords and words, as shown in **Figure 57**. The similarity of blank spaces creates the effects of overlapping and touching components, making the segmentation of lines and word extraction based on projection kind of impossible. Therefore, other techniques must be employed for word extraction.



Figure 57 : Intra-word and inter-word spaces in historical writing

Numerous studies have been conducted to extract words and subwords from written text in both Arabic and Latin writing. In 2019, [19] presented a method for text line segmentation and word extraction. Their approach to word extraction involves detecting the boundaries of connected components (CCs) by analyzing the vertical projection profile of the image. After identifying the CCs, the distances between them are computed, and an average distance is determined. This average distance is then used as a threshold to classify the spaces between CCs as either intra-word or inter-word.

In 2016, [78] proposed a method for extracting words from text lines of old manuscripts, which involves curve convexity analysis to resolve touching words. After extracting subwords through bounding boxes, the baseline is estimated, and the word image is thinned to obtain the Skelton. At the ending letters, segmentation points are detected to enable word extraction. While the subwords are extracted through bounding boxes, the authors do not provide much detail on this process. For word extraction, the authors combine subwords based on their geometrical relationship, considering the white space (distance) between subwords and classifying it as inter-words or intra-words.

Later In 2017, [79] further enhanced their previous work by proposing a method for touching component segmentation. They built a dictionary of touching components (TC) with their corresponding correct segmentation based on the approach reported by [80]. To segment a given TC, the method first identifies a similar model in the dictionary using a shape context descriptor. Then, the TC is segmented using the segmentation parameters of the equivalent model.

In one study by [17], a method was proposed for extracting subwords from Latin cursive handwriting text lines. The process begins with skew correction and proceeds to build a graph from the convex hulls of the subwords. This graph includes convex hulls as nodes and the distances between words as edges. To cluster subwords, the Otsu algorithm is used to determine a distance threshold (t) between subwords. This threshold is used to determine whether two subwords should be considered part of the same subword or not. The authors report that their method achieved 94.9% correct segmentation of words when tested on 541 text lines.

In their study, [20] developed a technique to extract words from handwritten lines based on the length of the connected components and the space between them. To achieve this, they used two Self-Organizing Map (SOM) clustering algorithms. The first algorithm classifies the components by size into three categories, namely separate letters, words, and subwords. The second algorithm classifies the gaps between the components into two classes, namely inter-words and intra-words. The authors reported an 86.3% success rate in their experiments on 35 pages of Handwritten Arabic text.

A segmentation methodology for Handwritten Arabic words into subwords was proposed by [18]. The process begins with pre-processing to eliminate noise and smooth edges, followed by converting the images into a skeleton format to reduce the image size to the minimum required for processing. The skew is then corrected, and the baseline position is estimated using a combination of Hough transform [81] and Local minima regression [35]. A dual-phase process is then employed for segmentation. Firstly, overlapping among the line components (subwords) is resolved by considering the distance between primary and auxiliary components as a metric. Secondly, a topological feature-based algorithm is used to extract the characters of each subword in the line image. The proposed segmentation method was evaluated on a dataset of 600 words and validated by successful classification.

The authors of [21] introduced a technique for extracting subwords from text lines. The method involves five main steps: It begins with the extraction of connected components (CCs) using bounding boxes. Subsequently, the baseline is estimated through an expectation-maximization (EM) process utilizing linear regression of the CC centroids. The CCs are then categorized into primary or secondary components based on their proximity to the estimated baseline. Finally, the secondary components are matched with their respective primary components by distance of overlap.

1.2. Arabic handwriting databases

To construct a database of words and subwords, researchers can generally choose one of three approaches. The first approach does not involve segmentation but instead relies on labeling the bounding boxes of subwords' coordinates within a database of text pages, as in the work of [82]. The second approach, which we employed in this work, involves segmenting pages into lines, then further segmenting lines into words and subwords. The database is built then using the resulting subwords and corresponding ground truth, as followed by [83]. Finally, the third approach, followed by [84], is more commonly used in constructing modern handwritten databases. It involves collecting words from writers via specified forms containing the exact desired words. The first approach is more suitable for developing databases for word-spotting purposes. Since it only saves coordinates of the bounding box containing the word, a given instance (word/subword) may contain parts from other neighboring instances due to overlapping effects. The third approach is more accurate but requires a lot of effort and time in building and distributing forms, collecting and verifying data. It is unsuitable for building Arabic historical databases because it is difficult to find cooperative calligraphs. The second approach is also challenging. Line and word extraction is quite difficult for several reasons.

However, regarding its accuracy, it remains the easiest to establish a database of historical words when raw data is available.

Many large databases have been developed for handwriting recognition in Latin script, providing researchers with sufficient resources to develop robust recognition and word spotting systems [17], [85]–[89]. The same cannot be said for Arabic-based scripts. Few databases have been developed for Arabic handwriting, and even fewer are publicly available. To address this gap, [83] introduced the IBN-SINA database of historical Arabic subwords based on an ancient manuscript from 1243. The authors used a clustering technique to label the subwords and converted the labels into bi-class problems for an SVM classifier to evaluate the database. However, the automatic labeling technique is still limited to specific applications, given that the total number of subwords in Arabic is unknown.

The IFN/ENIT database, as described by [84], is a public database of Handwritten Arabic words. The database includes 26459 words representing 937 Tunisian cities and villages, written by 411 volunteer writers. The database is widely used in research as it is available publicly for the research community and serves as a primary data source for many works in Arabic handwriting recognition. The database includes information about the character sequence, baseline, writer information, and ground truth. Its main purpose is for the segmentation and recognition of Arabic handwriting, but researchers have also used it for other tasks such as writer identification and pre-processing evaluation.

The KHATT database, which was introduced by [90], is an available resource for modern handwritten Arabic text. It contains 2000 paragraphs and 9000 lines written by 1000 different writers from different countries. The content comes from various sources and topics, and the forms are scanned and saved with 200 dpi, 300 dpi, and 600 dpi resolutions. The database also has 2000 other paragraphs with the same content. According to the authors, the ground truth is verified manually and provided in one text file per line with information about the writer. The database is used for several research purposes, including handwriting recognition, text pre-processing, text-to-line segmentation, and writer identification. It is split into three sets for training, testing, and validation, with a 70%, 15%, and 15% split, respectively. Experiments based on Hidden Markov Model (HMM) classifier were conducted using statistical features like pixel density and vertical edge derivatives.

The IESK-arDB database introduced in [18], is a database of Arabic words containing over 4,000 words and 6,000 isolated characters written by 22 different writers. The data covers various domains, such as security, banking, and city names, and includes both grayscale and binary versions. To annotate the data, each word in the database is associated with a feature XML file. Although the database is relatively small in terms of the number of words, the authors explain that a comparison of the frequency of letters with other databases shows the usefulness of their proposed database. In addition, the authors presented a methodology for subword extraction and overlapping resolution, which will be reviewed later in this chapter.

Authors in [91] developed an offline Arabic recognition database aimed at supporting automatic bank check processing systems. According to the authors, the database consists of

handwritten paragraphs and words contributed by 100 writers. However, the database is not publicly available, and there is limited information about it in the literature.

The VML-HD database, introduced by [82], is designed for recognition systems of historical Handwritten Arabic pages at the subword level. The database is based on 680 pages from 5 historical books. It is annotated at the subword level, with each page having an associated XML file containing the subwords' coordinates and transcription information. To validate the database, the authors used a radial descriptor graph and a radial descriptor [92] and calculated the hit rate for several subsets. Later in the work, a comparative study of VML-DB and MOJ-DB is conducted.

The authors of [93] presented a collection of Arabic characters and ligatures as a database. The database is created with contributions from 50 different writers and contains 15400 instances, including 9900 ligatures and isolated characters. Researchers can utilize this database to evaluate the effectiveness of Arabic character recognition systems.

2. The proposed method of subwords extraction from ancient Arabic documents

To accomplish the task of lines-to-subwords, we proposed a novel approach based on contour analysis to extract all subwords along with their points and diacritic signs. Detecting the baseline position is crucial in text image segmentation, but it is more challenging in old documents compared to modern handwriting. This is due to the random orientations of ancient books, the usage of outdated tools, and the fact that baseline position can change within the same line. Therefore, we adopt a method that detects multiple local baseline positions within the same line by dividing it into five areas. A local baseline is detected for each area using the algorithm presented in [35]. **Figure 58**.



Figure 58 : Local baseline detection

To initiate the segmentation process, all the contours in the line image are extracted and labeled. Next, a contour matrix is created to indicate the relationships between the contours. These relationships are classified as Parent, Child, or Brother. If contour B is entirely enclosed within contour A, then A is considered the parent of B, and B is considered the child of A. Conversely, if contours B and C share the same parent, they are considered brothers. **Figure 59** provides some examples of these relationships.



Figure 59: Relations of contours: C1 is a Brother of c2. C2 is a Parent of C3. C3 is a Child of C2

The primary contours representing the main body of the subword are identified by marking each contour with no parent and intersecting with the local baseline of its corresponding area. Then, using the relations matrix, contours' positions and sizes, additional contours representing holes, dots, and diacritic signs are associated with each corresponding primary contour. Given three contours, C^0, C^1, and C^2, they are associated with coordinates indicating their start and end positions, C^i_{Xstart}, C^i_{Xend}, C^i_{Ystart}, and C^i_{Yend}. A contour C^i is labeled as a hole of contour C^j if it is entirely inside contour C^j and has C^j as its parent and no child according to the relations matrix. **Figure 60** illustrates this concept. The association of diacritic signs and dots is complicated due to the variety of cases. The size and position of a dot or diacritic sign may change from one word to another and may appear above or below the main body. To associate an additional contour C^n with the primary body contour C^m, the overlap ratio between the two contours based on the position and size of the additional contour C^n is calculated. The midpoint MP of the primary contour is detected, and the addition is associated with the main body if it satisfies equation 1) when it is on the left side of the midpoint MP or equation 2) when it is on the right side of MP.

$$1) \quad \frac{|C^n_{Xstart} - C^m_{Xstart}|}{C^n_{Xend} - C^n_{Xstart}} < u$$

$$2) \quad \frac{|C^n_{xEnd} - C^m_{xEnd}|}{C^n_{xEnd} - C^n_{xStart}} < u$$

Where u is a fixed threshold found experimentally for each document. After analyzing the results of the tests conducted on 200 diacritical marks, it was established that in order to classify contour C^n as an extension of contour C^m, the overlap ratio should not exceed 0.6. **Figure 61** shows an example; C2 (Xstart=5.7, Xend=6.3) is a true addition of C0 (Xstart =1.2, Xend=6.2), where C1 (Xstart=6.1, Xend=6.6) is not.

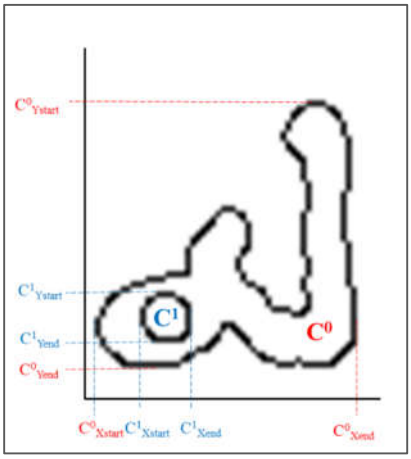

Figure 60 : Hole example: C2 is a hole of C1

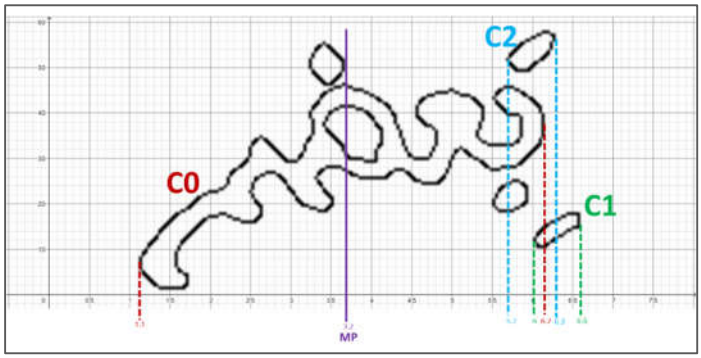

Figure 61 : Association of additions to the main body

Additionally, historical documents may contain instances where two or more subwords partially or entirely overlap. To the best of our knowledge, there is no prior research addressing subword complete overlap (super-positioning). In such cases when a subword B is entirely above another subword A, as shown in **Figure 62**, a size test is performed to determine whether B is an addition to A or not. The size of a subword is determined by its total number of pixels, and based on experiments conducted on 100 super-positioning cases, it was observed that an object B is considered an addition to another object A if size(B) is less than t=0.3 times the size of A, where t is set experimentally.

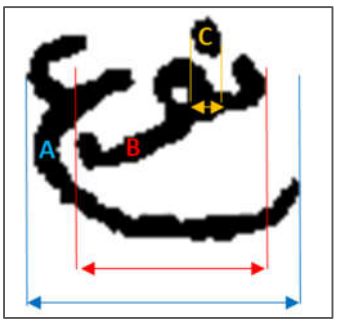

Figure 62 : Complete overlap of subwords

The extraction process can be summarized in the following algorithm :

```
done = [] //an empty list
// let Cnts be a list of all the extracted contours
for i=0 to count(Cnts) :
    if C^i not in done && C^i ∩ LocalBaseLine = True && Parent (C^i) = NULL :
        C^i is set to primary
        done . add(C^i)
        for j=0 to count(Cnts) :
            if C^j not in done : // not yet assinged
                if MP(C^j) < MP(C^i) : //C^j is on the left of C^i
                    if Eq1 = True &&  size(C^j) < t*size(C^i):
                            C^i.Secondaries.add(C^j)
                            done . add(C^j)
                    else : pass
                elif MP(C^j) > MP(C^i) : //C^j is on the right of C^i
                    if Eq2 = True && size(C^j) < t*size(C^i) :
                            C^i.Secondaries.add(C^j)
                            done.add(C^j)
                    else : pass
        else : pass
```

Separation of touching components

The issue of touching components has been a significant obstacle for contour-based methods and other approaches in the literature of handwritten text segmentation. This problem arises when two subwords touch each other, creating a false subword. This problem is most prevalent in cases where there are descending characters such as "و" or "ر," among others. An example of this problem can be seen in **Figure 63**.



Figure 63 : Examples of touching components

Our approach to addressing this issue involves localizing the descending characters within the word and detecting the segmentation points between the connected components. We have developed a novel technique for resolving touching characters, which involves identifying possible segmentation points between descending characters located within the angles of the lower contour. Initially, we remove any additional components using an 8-connected components analysis algorithm [94]. A size threshold of T = 0.2*P is set experimentally, where P represents the number of pixels in the main body of the word.

Extracting the word's lower contour involves using the Freeman chain code with 8-connectivity to connect the most right-up and most left-up points of the primary body contour. The lower contour is chosen because it represents the shape of descending characters, which are the primary cause of the touching components issue, and using it helps reduce the processing time. The resulting lower contour is then labeled from right to left, forming an 8-orientations vector that includes the lower contour variations and their corresponding (x,y) coordinates. A smoothing algorithm is applied to eliminate any noise present in the vector.

To extract the features of a descending character, we analyze the orientations vector and determine its start, bottom, and end coordinates, as well as its width and height. This information is depicted in **Figure 64**. Using this information, we can identify potential segmentation points between two descending characters, C1(start, bottom, end) and C2(start, bottom, end), where $C1_{end} \equiv C2_{start}$, as shown in **Figure 65**. Once the subwords have been identified, they are separated and saved in separate files.

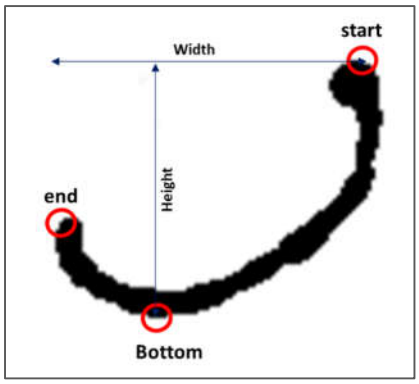

Figure 64 : A descending character "rae - ر"

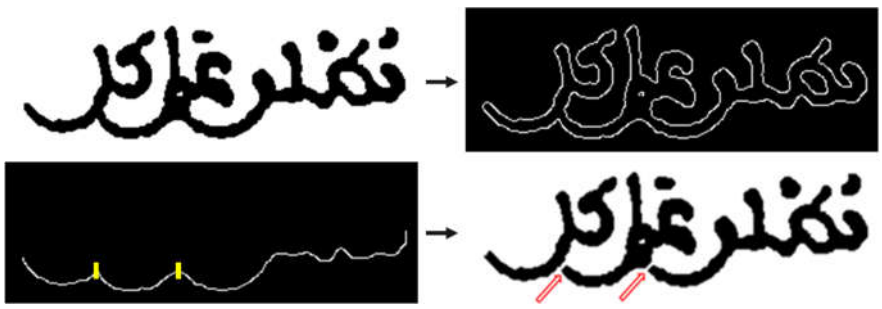

Figure 65 : Detection of Segmentation point

3. Database construction

3.1. Data collection

Our dataset was sourced from the Al-Qaraouiyine Library, which was established in Fez in 860 and affiliated with Al-Qaraouiyine University, recognized as the oldest educational institution in the world by UNESCO and the Guinness Records [95], [96]. The raw collected data consists of 129 pages, all written in ancient Arabic Al-Mojawhar style. They were captured as double-paged images as shown in **Figure 66**, and divided into four sets (a, b, c, and d). We randomly selected these pages from ten books authored by ten different writers during the 17th and 16th centuries, covering various subjects such as art, science, religion, and biographies. The chosen pages were scanned at 300dpi resolution and saved in PNG file format. However, due to the complexity of the construction process and lack of material resource, we found that processing all the available data would cost a lot of time and efforts. Therefore, we only used the first two sets (a and b), which consist of 64 pages, for the initial stage. Future versions of the database will incorporate the remaining sets that are currently not included. We follow a series of steps to extract the subwords from the raw text pages in order to construct our database. **Figure 67**.



Figure 66 : Example of the raw document

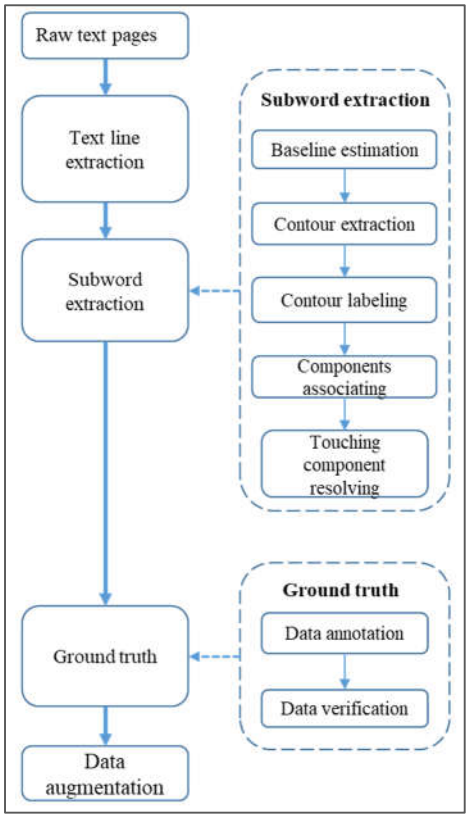

Figure 67 : The process of MOJ-DB database construction

3.2. Data pre-processing

In the initial phase, we conducted a quality check of the manuscripts to ensure the clarity of the text and the scan. Following this, we manually eliminated any extraneous objects present in the pages, such as borders, shapes, and paginations. The images of the pages were converted to binary format using the Niblack's algorithm of thresholding [72], and a median filter was applied to eliminate any noise. The approach of Connected Component Analysis, as explained by [97], coupled with morphological opening, was employed to eliminate large black spots that might have resulted from mishandling or the storage conditions of the books.

3.3. Subword extraction

Subsequently, following the initial pre-processing of the documents' images, we proceeded to apply method of text line extraction and skew correction, as thoroughly explained in the previous chapter. The primary objective of this stage was to accurately identify and isolate the text lines within the documents. To ensure optimal segmentation quality, manual refinement was applied to enhance the segmentation quality, thereby isolating each individual line and preparing them for subsequent subword extraction. Moving forward, the 1573 text lines obtained from the segmentation stage were segmented into subwords using our contour-based method extensively elaborated in the previous section. The objective of this segmentation is to successfully extract the containing subwords of the text lines, and resolve overlap.

3.4. Database Ground truth

During the construction of a database, exact ground truth associations are considered essential, and errors are not acceptable. To facilitate database annotation, we created a simple application with user interface to insert and link ground truth information with the corresponding image filename of the subword. In the second phase, we used the help of five individuals to validate the ground truth. We divided the database into five subsets; each person reviewed and validated four subsets. A ground truth instance was considered valid only if all four persons confirmed it. Otherwise, each person suggested their own ground truth for the word in question, and we chose the most commonly proposed one. We believe that this one-to-all verification strategy minimizes ground truth errors.

To ensure normalization and avoid encoding issues, we generated a Latin annotation of the ground truth for each subword. In the first stage, we assigned a Latin symbol to each letter of the Arabic alphabet, as shown in **Table 6**. In the second stage, we assigned a prefix to each character in the ground truth to indicate its position within the given subword. The prefixes we defined are 'B' for Beginning, 'M' for Middle, 'E' for End, and 'A' for Isolated. For example, the subword 'بيتهُم' is encoded as 'B_b | M_y | M_ta | M_h | M_s3 | E_m'.

Table 6 : Arabic to Latin Annotation

ر	r	ف	q	ء	a0	(	p1	ً	s8
ذ	thz	غ	gh	ة	ta	لأ	la2	ٍ	s7
د	d	ع	aa	ي	y	لإ	la1	ِ	s6
خ	kh	ظ	dz	و	w	لا	la0	ْ	s5
ح	ha	ط	tt	ه	h	آ	a6	ٌ	s4
ج	j	ض	da	ن	n	ؤ	a5	ُ	s3
ث	th	ص	sa	م	m	ئ	a4	ٌ	s2
ت	t	ش	ch	ل	l	ى	a3	َ	s1
ب	b	س	s	ك	k	إ	a2	*	st
ا	a	ز	z	ق	q	أ	a1	)	p2

In addition to the word transcription, we stored fundamental features of the subword image and the ground truth. In addition to the Arabic ground truth (arGT) and the filename, the header also includes the Latin annotation of the character sequence (ltGT) and the subword's coordinates (XStart, YStart, XEnd, YEnd) within the image (Coords). The IDs of the book, page, and line from which the subword is extracted are also included. The (Gen) field in the header denotes the origin of the subword and whether it is extracted (Gen=0) or generated (Gen=1) during data augmentation. Basic features of the subword image, such as baseline position (BL_y), the number of components (nbrComp), and the total number of foreground pixels (frgPixelCnt) are also saved. **Table 7** shows an example of the ground truth information.

Table 7 : The ground truth of the subword "يعْتَقِد"

fileName 0622-25-30-3		**arGT** يعْتَقِد			
Coords (Xs=91, Ys=23; Xe=268, Ye=111)		**ltGT** B_y | M_aa | M_s5 | M_t | M_q | M_s6 | E_d			**Gen** 1
yBL 71	**nbrCmp** 8	**frgPixelCnt** 3555	**idBook** 06	**idLine** 25	**idPage** 22

3.5. Data balancing

In Arabic text, certain letters are more common than others, such as the letter 'أ - alif', which is the most frequently used letter. Additionally, the subword 'alif' can be combined with the letter "ل - lam" to form the definite article of any Arabic noun, while the letter "و - waw" is often used as a conjunction between words. As a result, these two letters/subwords are more prevalent than any other subword in Arabic text, leading to unbalanced classes. This fact poses a challenge for machine learning engineers trying to build robust recognition models [98]. The initial database contained 38,497 subwords, and to address the imbalance, we reduced the number of occurrences for classes with more than 100 instances. On the other hand, we used data augmentation techniques to generate subword images for classes with less than 100 instances. Taking inspiration from the work of [99], we applied parameterized combinations of transformations such as horizontal and vertical distortion, morphological erosion, and dilation. These operations were randomly applied to a set of images for each class to generate new images, example in **Figure 68**. Our balanced database finally contains 560,000 subwords evenly distributed across 5,600 classes.

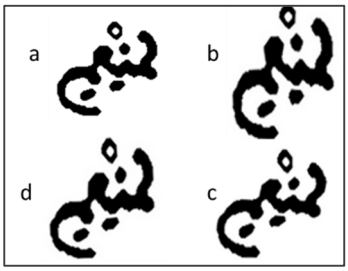

Figure 68 : Data augmentation with distortion, erosion and dilation

4. Experimental study

4.1. Subwords extraction

In order to assess the effectiveness of our subword extraction method, we assembled a set of 350 text lines that were randomly selected from our raw data **Figure 69**. The subwords

were subsequently extracted using our approach, as well as methods described in previous studies [18], [21], [78]. Following subword extraction, we manually reviewed the resulting subwords and eliminated cases of false extraction. It is noteworthy that all implementations and experiments were conducted using a Python-OpenCV environment (Python 3.7 with OpenCV 4.2 library) on a quad-core i7 PC with a clock speed of 3.4GHz and 16Gb of RAM. **Table 8** provides the number of valid subwords that were successfully extracted using each method, as well as the average required time for extracting all the subwords of one line.



Figure 69 : Examples of text lines used for the segmentation test

Table 8 :total number of subwords extracted with each method

Method	Number of extracted subwords	Number of different classes	One line extraction time (s)
[78]	6529	115	3.2
[18]	6988	127	2.9
[21]	7853	141	3.8
Our method	**8082**	**135**	**3.3**

To assess the subword extraction quality of each technique, we randomly sampled 6500 subwords from each of the acquired sets, which were evenly distributed across 100 classes. The feature vectors for each set were subsequently extracted using Histogram of Oriented Gradients (HOG) [100] and employed as test sets for an SVM classifier based on a polynomial kernel. The resultant classification outcomes are presented in **Table 9**.

Table 9 : Results of classification of subwords extracted with each method

Method	Accuracy
[78]	59.3%
[18]	74.8%
[21]	81.1%
Our method	**83.2%**

4.2. Database validation

To validate the consistency and robustness of our database, we conducted several experiments. Firstly, we performed handcrafted feature extraction using Histogram of Oriented Gradients (HOG), Edge Histogram Descriptor (EHD), and Scale-Invariant Feature Transform (SIFT) descriptors, taking inspiration from the works of [101], [102]. In a second experiment, we utilized a deep convolutional neural network (CNN) for feature extraction and classification. To extract more insights from these experiments, we divided the database into three sets based on the size of the subwords. Set A contains subwords with one or two letters, set B contains subwords with 3-5 letters, and set C contains subwords with more than six letters.

4.2.1. Handcrafted features

Table 10 presents the results of handcrafted feature extracted with HOG, EHD, and SIFT for subword classification with an SVM classifier based on a polynomial kernel. Each of the three sets (A, B, and C) was split into 70% for training and 30% for testing.

Table 10 : The results of Classification of handcrafted features

Descriptor/Set	HOG	HOG+EHD	SIFT
A	89.1%	95.3%	91.7%
B	85.3%	89.6%	84.8%
C	75.9%	78.4%	72.1
Complete data	78.65%	85.47%	80.6%

4.2.2. Learned features

We created a deep learning model utilizing a convolutional neural network, a type of feed-forward network commonly employed in supervised learning. Composed of convolution layers, pooling layers, and fully connected layers, CNN automatically extracts distinctive features that are resistant to shift and shape distortions in input characters, as noted by [103]. Our model comprises three convolution layers, each succeeded by a MaxPooling layer, as well as two additional fully connected layers and an output layer. **Figure 70**. The first convolution layer's input is normalized to M x N x C, with image dimensions of M = 162 and N = 162 and C denoting the number of color channels. (C=3 for RGB and C =1 for grayscale images. **Table 11** presents the results of experiments of the model, which includes a series of convolutional and pooling layers. Specifically, the first convolution involves 64 filters sized (5x5), followed by a MaxPooling layer that uses a (3x3) grid for sub-sampling. The second convolution applies 128 filters of size (5x5) and is followed by a MaxPooling layer of (3x3) and a dropout layer with a fraction of 0.5 to prevent over-fitting. The third convolution is executed with 256 filters of size (3x3) and a MaxPooling layer of (3x3), succeeded by two fully connected layers of

(512x1) and an output layer of (5600x1). To evaluate the model, the data is split into three sets, with 70% allocated for training, 20% for testing, and 10% for validation.

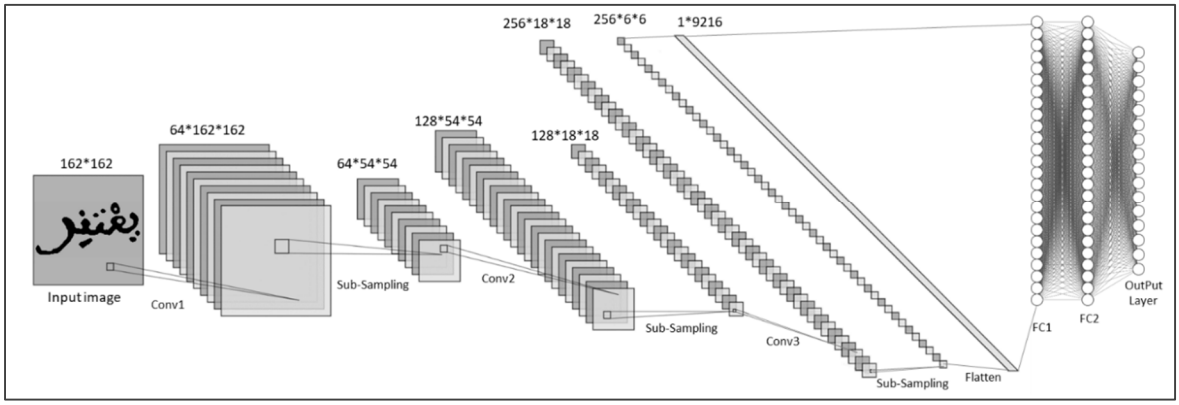

Figure 70 : The used CNN model

Table 11 : Results of subwords classification with CNN

Set	Accuracy
A	96.7%
B	91.7%
C	84.3%
Complete DB	89.6%

4.2.3. Database comparison

To compare the performance of the MOJ-DB database with other publicly available databases, we utilized the same CNN model and the HOG+EHD model with the SVM classifier. For this comparative analysis, we selected the IESK-arDB modern handwriting database, presented in [18], and two other historical handwriting databases: IBN-SINA [83] and VML-HD [82]. We pre-processed the subword images by binarizing and normalizing them to match the input layer size of our models. **Table 12** summarizes the classification results that we obtained.

Table 12 : Classification of learned and handcrafted features for different databases

Database/model	CNN	HOG+EHD with SVM
IESK-arADB	88.5%	83.1%
IBN-SINA	79.1%	77.6%
VML-HD	73.5%	69.4%
MOJ-DB	**89.6%**	**83.4%**

4.3. Discussion

4.3.1. Subword extraction

Based on the experiments conducted, it was observed that our contour-based approach was more effective than the other approaches, particularly those based on selecting bounding boxes. Table 8 In fact, the contour accurately describes the shape of the object and preserves all of its variations when extracting a subword with all its additions. On the other hand, when extracting a subword using a bounding box, the selection may include unwanted objects belonging to other subwords in cases of overlapping, resulting in low extraction quality, as demonstrated in the second subword extraction experiment. Our method surpassed the other methods in term of both quantity, as indicated in **Table 8,** and quality as proved by results in **Table 9**. We were able to extract more subwords than the other methods and to discover more classes, which proves its great precision despite the fact that it takes more time for one extraction .

4.3.2. Database construction

The classification results obtained demonstrate the consistency of MOJ-DB when compared to other databases. Initially, we used handcrafted features extracted with HOG, SIFT, and EHD and classified them using an SVM. the results presented in **Table 10** show that the chosen descriptors, especially the combination of HOG and EHD, demonstrated high ability to extract features of subwords.

Later, we classified the learned features of different databases using one CNN model, which gave a high extraction accuracy of 89.6% as shown in **Table 11**. The low classification rate for the IBN-SINA database , indicated in **Table 12**, can be attributed to insufficient and imbalanced data. In contrast, for the VML-HD database, the results are explained by the use of bounding boxes to extract images containing subwords, leading to inaccurate features due to the inclusion of additions from other subwords. To overcome this issue, the MOJ-DB database is built utilizing contour-based extraction and a process for touching component separation, ensuring that each image contains only the subword of interest. **Figure 71** illustrates the difference between the same subwords from MOJ-DB and VML-HD databases.



Figure 71 : Examples from MOJ-DB and VML-HD databases

Conclusion

This study introduces a new database, that we named MOJ-DB, which comprises Arabic historical text that is intended for segmentation-free recognition systems of Arabic historical text. The database comprises 560,000 subwords that have been extracted from 64 text pages, which were randomly selected from 10 historical books covering various subjects. The ground truth for this database has been manually provided at the subword level and has been iteratively verified to eliminate any potential errors. The consistency of the MOJ-DB database has been demonstrated through several experimental analyses and comparisons with other databases. Additionally, a novel method for extracting subwords from Arabic text lines has been introduced in this work. Furthermore, the proposed database is made openly accessible to the research community through the following link: https://github.com/Abdelhay-Z/MOJ-DB.

The proposed subword extraction system is based on the analysis of geometric features of the image's contour. These features include the start and end coordinates, size, and relative position to other contours and the baseline. Additionally, the system includes a process for resolving the connected components, which is often caused by descending characters touching. This challenge remains a significant obstacle for Arabic and Latin handwriting text segmentation systems. The present study tackles this challenge by examining the lower contour of the subword to identify possible segmentation points that could facilitate the separation of touching components. To evaluate the proposed subword extraction method, it was tested on the MOJ-DB database as well as other methods from previous literature. In an experimental comparative analysis, our proposed approach demonstrated superior performance in terms of extraction quality and quantity in comparison to the other techniques.

Chapter 4 . Comparison of feature extraction method for Al-Mojawhar characters recognition

Introduction

In the previous chapter, we introduced MOJ-DB database comprising words, subwords, and characters extracted from historical documents written in Al-Mojawhar style. Being a recently established resource, this database demands in-depth exploration and comprehensive analysis through experimentation. In this chapter, we conduct a study to investigate al-Mojawhar character recognition using the proposed MOJ-DB database. We also aim to provide a comparison of handcrafted features and learned features. In addition, we want to check for the best combination of feature extraction and classification methods for recognizing Al-Mojawhar characters. Therefore, multiple experiments were conducted, employing diverse combinations of feature description methods and classification techniques to identify the most suitable approach for the recognition.

It is commonly accepted that in the process of optical character recognition, feature extraction plays a significant role. This important stage involves identifying and selecting relevant characteristics from input data to effectively represent the underlying information and capture essential attributes for accurate recognition. This process helps overcome challenges posed by variations in font styles, sizes, orientations, and image quality. In this chapter, we investigate the use of the widely employed feature extraction methods in the field of optical character recognition. Specifically, we incorporated the Scale Invariant Feature Transform (SIFT) and Histogram of Oriented Gradient (HOG), the Zernike moment-based descriptor, the Gabor filter, and the pixel data values used as raw features.

Mostly in OCR systems, following feature extraction steps, comes the classification. It is an integral part of the overall recognition process. It heavily influences the performance and reliability of OCR systems as it helps categorize individual characters or character groups into predefined classes. It involves training machine learning models to identify patterns and features that distinguish different character classes. The character recognition experiments presented in this chapter use modern classifiers like the Multi-Layer Perceptron (MLP), Support Vector Machines (SVM), K-Nearest Neighbor (KNN), and Random Forest (RF). Additionally, we implemented a deep Convolutional Neural Network (CNN) in a final experiment to classify the character images.

This chapter is organized as follows: the following section presents the state of the art of Arabic character recognition. We discuss the previous works by highlighting the utilized feature extraction and classification methods as well as the databases used for training and evaluating. In Section 3, we summarize the various conducted experiments on Arabic Al-Mojawhar characters. We present the obtained results of recognition, which are discussed in Section 4, before closing this chapter with conclusion as a final section.

1. State of the art of feature extraction and classification methods of Handwritten Arabic Character

Until recently, there had been a notable absence of research focusing on the recognition of ancient writing styles. While a limited number of publications have emerged in recent years aimed at advancing printed and handwritten Arabic OCR systems, the outcomes achieved thus far can be regarded as modest. The intricate shapes of Arabic letters present a significant challenge, necessitating the utilization of more sophisticated shape descriptors to generate precise and reliable features. After reviewing the literature, we cite some of the reference works that have addressed the problem of Handwritten Arabic character recognition, mentioning the feature extraction and classification techniques used, along with the database against which the evaluations were conducted.

In [104], the authors conducted a comparative study on various feature extraction methods used for Arabic handwriting recognition. The selected methods included Gabor Filter, Wavelet Transform, Fourier Transform, and Hough Transform, which were evaluated based on their ability to extract invariant features. To assess the precision of the different methods, the authors employed the Euclidean Minimum Distance Classifier (EMDC) as classifier. Each feature extraction method was tested on the IFN/ENIT dataset. The reported results indicated that Hough Transform and Gabor Filter were more effective in extracting representative and invariant features.

In the work of [105], they presented a model for Arabic word recognition based on the Scale Invariant Feature Transform (SIFT) as the feature extraction method and SVM as the classifier. The features extracted using SIFT were grouped into clusters using K-means clustering. The proposed system was tested on the AHDB database and reported a success recognition rate of 99.08%.

In [106], the authors employed Gabor filter features as input for an SVM classifier utilizing both RBF and polynomial kernels for Handwritten Arabic character recognition. The proposed model was tested using the HACDB database, consisting of 66 distinct classes. The reported classification rates were 88.77% for the RBF-based SVM and 70.82% for the polynomial-based SVM.

In [107], the authors proposed a system for Handwritten Arabic character recognition. The system employed a multi-class SVM with an RBF kernel combined with the Histogram of Oriented Gradient (HOG) feature extractor. The authors also explored the use of Gabor filter as a handcrafted feature descriptor. The HOG-SVM system was evaluated using the IFN/ENIT database and achieved a recognition accuracy of 98.5%, while the Gabor-SVM system achieved 92.8% accuracy.

In [[108], the authors utilized the Freeman chain code for Arabic character identification. The process involved defining a bounding box around the character, including auxiliary parts such as dots and symbols. The chain code was extracted from the character's shape and reduced to a minimal 7-digit form. The encoded chain code was combined with additional statistical features to create an 11-digit feature sequence. A confusion matrix was calculated to evaluate the effectiveness of the proposed method, with the authors claiming accuracy ranging from 92% to 97%.

Alternatively, in [109], SVM was also employed in combination with the Histogram of Oriented Gradient (HOG) descriptor as a feature extractor. Prior to feature extraction, the input images underwent several preprocessing operations, including cleaning, binarization, color normalization, and word segmentation into smaller windows. The resulting feature vectors were then fed into the SVM classifier. The proposed model was applied to a private dataset comprising Jordanian city names, and a recognition success rate of 99% was reported.

In [110], the authors proposed a Convolutional Neural Network (CNN) model for Handwritten Arabic character recognition. The model consisted of three convolutional layers, three MaxPooling layers, two fully connected layers, and an output layer with 29 units. The ReLU activation function was employed for the convolutional layers, while SoftMax was used for the output layer. The model was tested on a private database created by the authors, containing 47,000 images, as well as the Handwritten Arabic Characters Database (AHCD) database [14]. The CNN model achieved an accuracy of 97% on the AHCD database, while on the private database, the accuracy did not exceed 88%.

In [111], a CNN model for offline Arabic character recognition was presented. The model consisted of three convolutional layers, one fully connected layer, and one output layer. The dropout technique with a probability of 0.5 was employed to mitigate overfitting. The model was experimented on both the AHCD and AIA9K databases, achieving recognition accuracies of 97.6% and 94.8%, respectively.

In [112], the authors introduced the Shared-Hidden-Layer Convolutional Neural Network (SHL-CNN) for character recognition across multiple languages. The model architecture consists of two parts. The first part, shared among all languages, includes two convolutional layers, two MaxPooling layers, two contrast normalization layers, and two local convolutional layers. The second part is a non-shared layer consisting of a SoftMax fully connected layer. While Arabic letters were not specifically included in the study, promising results were obtained.

In [113], a Convolutional Neural Network (CNN) model is proposed for Arabic character recognition. The model comprises four convolutional layers, two MaxPooling layers, and three fully connected layers. The activation function used for all layers except the output layer is ReLU, while the output layer utilizes SoftMax. The authors investigated various optimizations and data augmentation algorithms. The proposed model was evaluated on the AHCD and Hijja datasets, achieving recognition success rates of 98% and 91%, respectively.

In [114], the authors conducted a novel study utilizing a hybrid model for Handwritten Arabic character recognition. They introduced a recognition model based on a deep

convolutional neural network (CNN) as a feature extractor. The extracted features were subsequently fed into three machine learning models for classification: Support Vector Machines (SVM), XGboost, and a neural network with SoftMax fully connected layers. The authors tested various architecture combinations with different parameters using the Hijja dataset. Among the tested combinations, the CNN-SVM hybrid model achieved the highest recognition rate, with a success rate of 96.3%.

In [115], a hybrid model for handwritten character recognition was proposed, combining CNN and SVM. The CNN architecture comprised three convolutional neural layers, each followed by a MaxPooling layer. A fully connected layer and an output layer were incorporated, with the output layer subjected to a dropout function and then fed to the SVM classifier. The DCNN-SVM model was tested on a private database containing 16,800 images, achieving a classification error rate of 4.9%, which is highly promising.

2. Study of feature extraction and classification methods for Al-Mojawhar characters

The primary goal of this study is first to investigate the MOJ-DB database, to confirm the differences with modern handwriting. Moreover, this study aims to check for the best combinations of feature extraction and classification methods that may give good recognition results of Al-Mojawhar characters. To conduct our study, we used characters selected from MOJ-DB database. We also used AHCD database of modern handwritten characters.

Standard Arabic characters generally have four different shapes for the four positions within a word, namely the beginning, middle, end, and the isolated form of the character. In contrast, some of the Al-Mojawhar characters can have one or more shapes for the same position. **Figure 72**. However, some characters like (أ, و, د, ر , ز, ذ) don't have an in-middle or in-beginning shape. They are called ending characters as their presence in a word splits it into subwords. In conclusion, the total number of shapes of Al-Mojawhar go up to 76 different shapes.

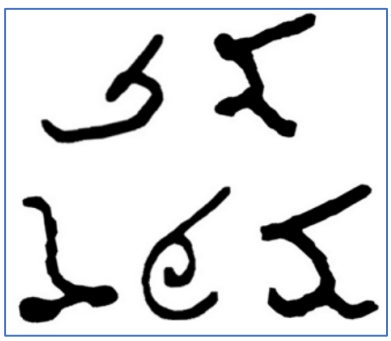

Figure 72 : Different shapes for the same letter (kaf - ك)

In this study, we employed various techniques of feature extraction and classification for the recognition of Al-Mojawhar characters. These two phases play a crucial and risky role in developing an optical character recognition system, as they involve converting the graphical representation of characters into a numerical format that can be digitally manipulated and edited. As seen in the aforementioned works, the existing literature offers several effective methods for extracting representative features and classifying handwritten characters. However, the complexity introduced by the variations in shape and additional signs poses significant challenges when dealing with ancient Arabic characters.

To extract features, we examined the application of Scale-invariant feature transform (SIFT) [100], Histogram of Oriented Gradient (HOG) [116], Zernike moments [117], Gabor filter [118]. Additionally, we investigated the use of raw pixel data to construct feature vectors. On the other hand, for classification, we employed various techniques, including the multi-layer perceptron, Support Vector Machines (SVM) with RBF and Polynomial kernels, K-Nearest Neighbor (KNN), and Random Forest. Additionally, we utilized a Deep Convolutional Neural Network (CNN) for its ability to perform both feature extraction and classification tasks.

2.1. Feature extraction methods

As described in [100], the SIFT offers a unique representation of local image characteristics that remains unaffected by diverse transformations, thereby facilitating object matching and recognition. SIFT descriptor encompasses several steps including the identification of keypoints, the improvement of their localization accuracy, the determination of their orientations, and the computation of a local image descriptor based on gradient orientations. **Figure 73.**

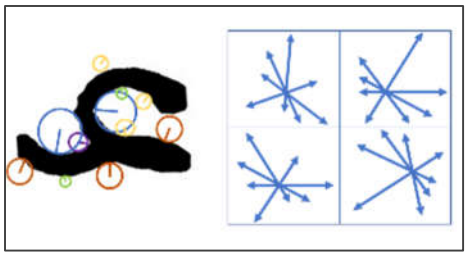

Figure 73 : SIFT Keypoint detection

Within the same category, the HOG descriptor presented first in [116], is recognized as a powerful method for feature extraction in pattern recognition. It operates by dividing the image into multiple connected and equal cells. For each one, the edge gradients and orientations are computed, and a one-dimensional histogram is created by combining these values. **Figure 74**. The global feature vector is then obtained by concatenating these histograms, representing the overall characteristics of the global image.

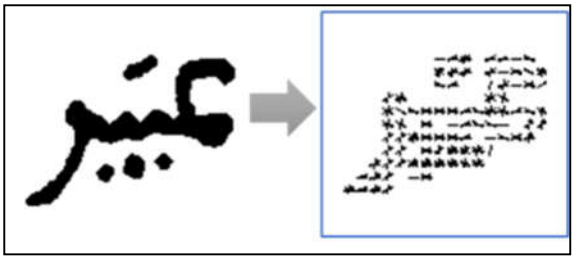

Figure 74 : HOG descriptor simulation

Despite the apparent similarities between HOG and SIFT, these two descriptors exhibit notable differences. HOG is employed for extracting global features, which capture the overall distribution of gradients or edge orientations within an image. In contrast, SIFT is utilized for extracting local features, focusing on distinctive keypoints that are robust to variations in scale, rotation, and illumination.

Zernike moments were first introduced by physicist and Nobel Prize winner Fritz Zernike in the 1930s, and first used in pattern recognition by [117]. These moments are based on orthogonal radial polynomials and offer a distinct representation of an entity without any redundant information. This unique description provided by Zernike moments is valuable in feature extraction.

Feature extraction with Gabor filter [118] involves convolving an image with a combined Gaussian and sinusoidal kernel. The resulting response captures spatial frequency and orientation information. Analyzing the magnitude or other characteristics of the response allows the extraction of unique features related to texture and edges. These features can be further studied or employed for tasks like classification and object recognition.

2.2. Classification methods

A multi-layer perceptron (MLP) is a neural network which can utilized for character classification. It consists of interconnected layers of neurons, where the input is a vector representation of the character image. **Figure 75**. The MLP adjusts weights during training to extract relevant features and learn from the input data. The hidden layers allow the network to capture intricate patterns, while the output layer produces the final classification decision. MLPs are widely used OCRs due to their ability to learn and distinguish between different characters.

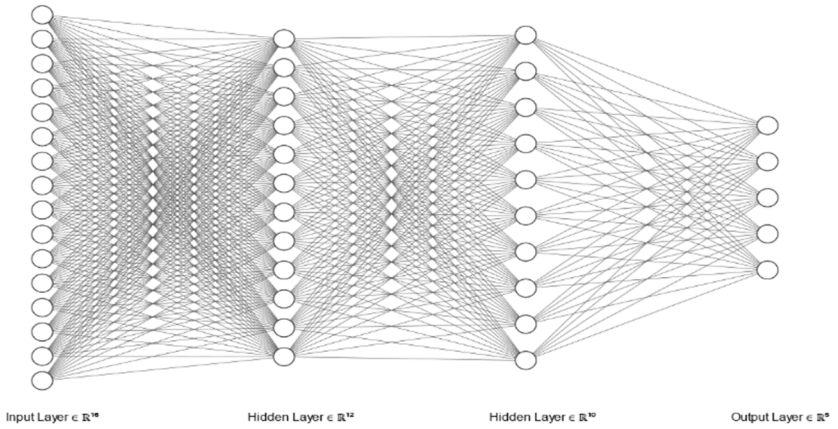

Figure 75 : Multi-Layer perceptron

Support Vector Machines (SVMs) are potent supervised learning methods widely used for classification and regression problems. Its fundamental objective is to identify an optimal hyperplane that maximizes the margin between data points belonging to different classes. By carefully selecting support vectors, a subset of data points lying closest to the decision boundary, SVM constructs this hyperplane. **Figure 76.** The algorithm can handle datasets exhibiting both linear and non-linear separability by employing kernel functions that map the data into higher-dimensional feature spaces.

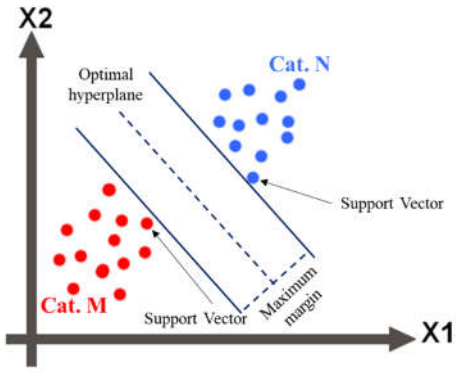

Figure 76 : Support Vector Machine simulation

Random Forest is a powerful ensemble learning algorithm that also finds extensive application in both classification and regression tasks. It addresses the limitations of individual decision trees by constructing a multitude of trees through random selection of features and data samples. Each tree operates independently to make predictions, and the final output is determined by a voting mechanism. **Figure 77**. It is highly regarded for its accuracy, simplicity, and flexibility.

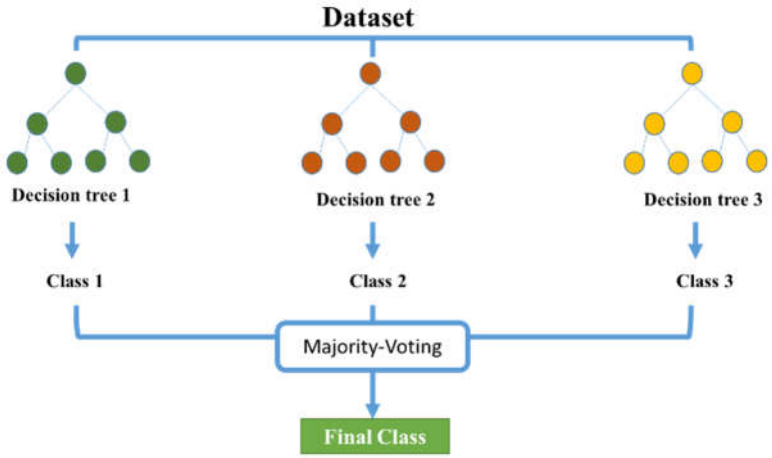

Figure 77 : Random Forest simulation

The K-Nearest Neighbor algorithm, referred to as k-NN, is a supervised learning method that employs proximity to perform classifications or predictions. It is commonly employed as a classification algorithm, operating under the assumption that neighboring points tend to share similar characteristics. **Figure 78**. However, the performance of K-NN is subject to various factors, such as the choice of K and the distance metric employed. Despite these considerations, KNN retains its popularity and remains widely employed in pattern recognition.

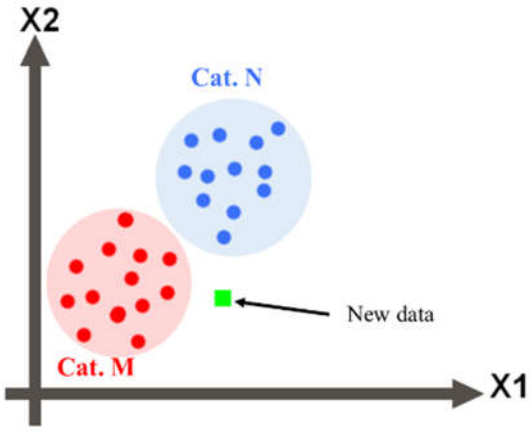

Figure 78 : K-nearest Neighbor simulation

Convolutional neural networks (CNNs) belong to a specific subset of feed-forward neural networks primarily utilized for image processing, serving as either shape descriptors or complete classification models. As stated in [103], CNNs offer the advantage of automatically extracting salient features that possess a degree of invariance to shifts and distortions in input characters. Typically, a CNN model comprises two main components. The first component utilizes convolution and pooling layers to function as a feature extractor, while the second component consists of multiple fully connected layers within a neural network that acts as a classifier.

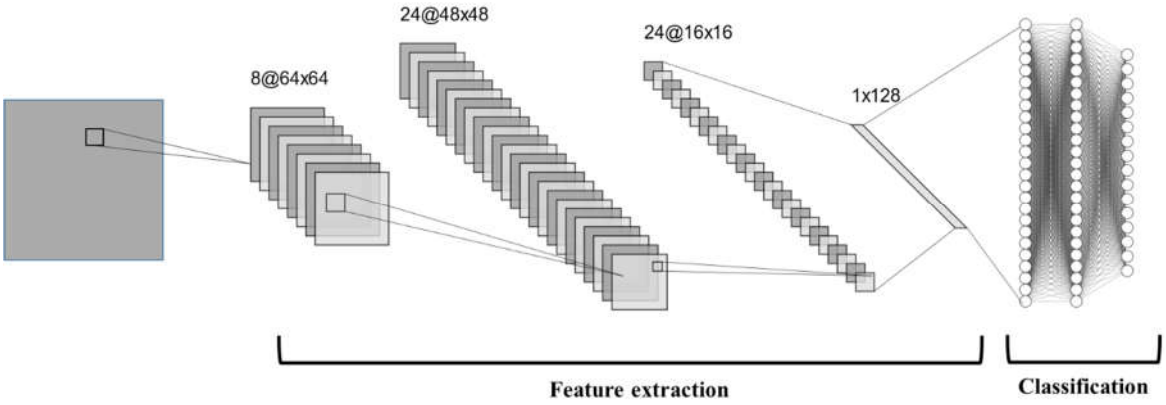

Figure 79 : Convolutional Neural network architecture

In order to assess the processing challenges associated with ancient Arabic handwriting compared to modern handwriting, our experiments also incorporated the AHCD (Arabic Handwritten Characters Database). This database contains a total of 16,800 images representing modern Handwritten Arabic characters, evenly distributed across 28 different classes. **Figure 80**. The AHCD database is recognized for its reliability and is commonly utilized for evaluating the performance of Arabic handwriting recognition systems.

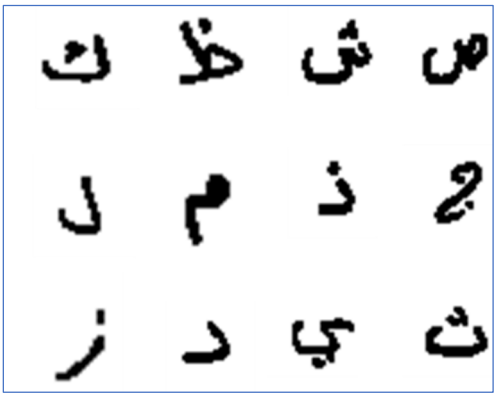

Figure 80 : Samples from AHCD Database

3. Experimental study

For the purpose of exploring MOJ-DB and evaluating the various methods described before, we used character images selected from our database described in the previous chapter. The initial dataset underwent pre-processing steps, including binarization and denoising techniques. Subsequently, the images were normalized to a size of 50x50 pixels. The initial MOJ-DB database consisted of 60 images for each of the 76 character classes. To ensure a comprehensive evaluation, we utilized an augmented database containing 600 instances per class. We note that the experiments were conducted in a Python-OpenCV environment, utilizing Python version 3.9 and OpenCV version 4.2 library. The hardware setup comprised a quad-core Ryzen 7 PC with a clock speed of 2.8GHz and 16GB of RAM.

3.1. Handcrafted features classification

We generated feature vectors from the MOJ-DB database using various methods, including HOG, SIFT, Zernike Moment, Gabor filter, and raw pixel data. Different parameters were tested for feature extraction. In the initial round of experiments, we employed a multi-layer perceptron (MLP) as classifier. The MLP consisted of four layers, each followed by a dropout function with a ratio of 0.5. The model was trained for 10 epochs with a batch size of 50.

In a subsequent experiment, we evaluated the performance of two types of SVMs: RBF-based and polynomial-based. These SVMs were trained using the different sets of features. The best results were obtained by identifying the optimal parameter combinations, specifically C=5, Gamma=0.05 for the RBF-based SVM, and degree=3 for the polynomial-based SVM. Furthermore, we conducted experiments using a KNN classifier with various values for the K parameter. The most favorable outcomes were achieved when setting K=3.

To explore the classification potential of Random Forest, we trained the model on the selected features. We tested different numbers of estimators, and the best results were obtained when using 64 estimators. The resulting test accuracies (%) obtained from the aforementioned experiments are presented in **Table 13**.

Table 13 : The test accuracy (%) of classification for the different models' combinations

	MLP	**SVM-RBF**	**SVM-Poly**	**KNN**	**RF**
Raw data	73.2	81.8	83.1	73.5	69.3
HOG	82.2	84.6	88.8	83.1	84.2
SIFT	70.7	68.7	75.8	66.8	77.6
Zernike Moment	76.1	79.7	74.6	82.1	81.6
Gabor filter	77.2	76.0	77.8	78.5	86.8

In another experiment within the same test environment, we conducted a time cost analysis of the four descriptors used for handcrafted features. For this purpose, we randomly chose 700 character images from the MOJ-DB database, which were resized to a dimension of 50x50 pixels. We calculated the processing time required for feature extraction using each descriptor for the entire set of images. The results of this analysis are summarized in **Table 14**.

Table 14 : Time cost for each features extractor

Feature extractor	**Time cost (s)**
HOG	1.88
SIFT	1.36
Zernike Moment	2.3
Gabor filter	0.46

In scenarios where the available data is insufficient for training machine learning models effectively, data augmentation can be employed as a technique to address this issue. It involves transforming existing data instances to generate additional data samples. In this study, we applied data augmentation by introducing distortions and slight rotations to the initial character images. As a result, we created two additional databases, one with 240 instances per character class and another with up to 600 instances. To assess the impact of data augmentation on the MOJ-DB characters, we trained an SVM with a polynomial kernel using HOG features on the three databases. This choice was based on the superior performance of the SVM model in previous experiments. The results of this evaluation are presented in **Table 15**.

Table 15 : The results of classification using SVM-HOG on the augmented databases

Database	Test Accuracy %
MOJ-DB_60	81.4
MOJ-DB_240	84.6
MOJ-DB_600	88.8

3.2. Learned features classification

As seen in the previous section, the use of deep learning in OCR research has increased significantly in last years, demonstrating remarkable accuracy. It has been widely used in OCR applications, particularly for Latin character recognition. One of this research's goals is investigating the effectiveness of deep learning techniques for recognizing and extracting features of Al-Mojawhar Arabic characters.

Since deep learning models require lot of data in order to extract features, we evaluated the performance of various convolutional neural network architectures using the augmented 600-version MOJ-DB and AHCD databases. In conclusion, the most promising results are achieved by the model summarized as follows: Input(50x50x1) → Conv (32) → MaxPool (2x2) → Conv (64) → MaxPool (2x2) → Conv (128) → MaxPool(2x2) → Flat → FC (128) → FC (128) → Output (76/28). It is composed of an input layer accepting as input 50x50 binary images, three convolution layers with subsequent MaxPooling layers, two fully connected layers for classification, and an output layer. This latter comprises 76 neurons for the tests on MOJ-DB and 28 neurons for AHCD representing the target classes. **Figure 81** provides a general illustration of the model used. The implementation was conducted on Keras platform with two different optimization functions; namely Adam and RMSprop. **Table 16** presents the obtained results after 30 epochs of training with a learning rate of 0.0001.

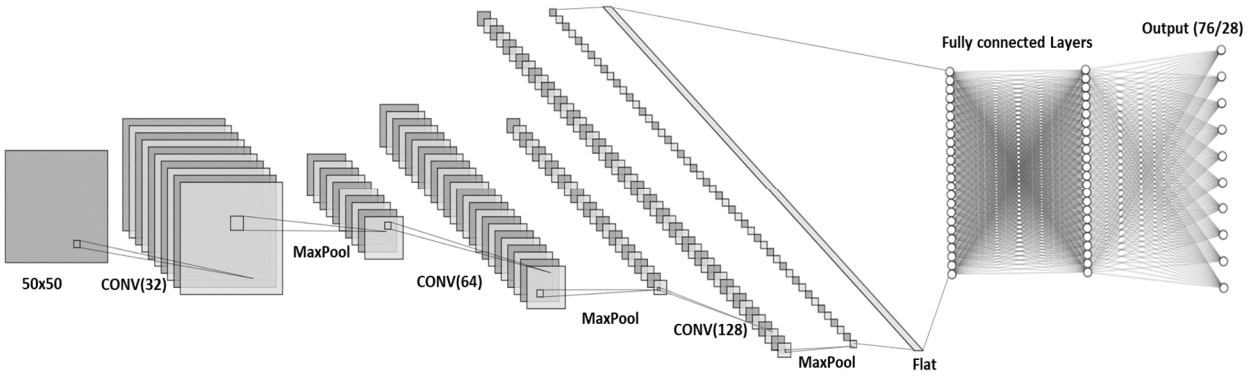

Figure 81 : CNN architecture

Table 16 : The results of classification using the CNN model

Optimizer	Test Accuracy on MOJ-DB_600	Test Accuracy on AHCD_600
Adam	95.6 %	98.5 %
RMSProp	95.2 %	97.9 %

4. Discussion of results

The majority of previous studies conducted on MOJ-DB have consistently indicated that the Al-Mojawhar style, characterized by its intricate details and patterns, exhibits significantly greater complexity when compared to contemporary handwritten characters. These findings highlight the need for researchers to devote more dedication, attention, and rigorous analysis when of the ancient Arabic script.

Additionally, the performance evaluation conducted on different feature extraction methods reveals that the Histogram of Oriented Gradients (HOG) and Gabor filter techniques consistently outperform other methods in terms of accuracy, robustness, and discriminative strength as indicated in **Table 13.** Furthermore, the features extracted using these two descriptors not only exhibit exceptional representation quality but also maintain a remarkably high level of consistency and reliability across all the classifiers employed in the experiments.

Regarding computational speed, **Table 14** indicates that the Gabor descriptor is the fastest, while SIFT and HOG require similar processing times, with a slight advantage for SIFT. On other side, the experiments confirmed the theory that invariant moments are the slowest descriptors for feature extraction.

For the classifiers, the experimental results summarized in **Table 13** and obtained through a comprehensive evaluation and rigorous examination, demonstrate that the multi-layer perceptron, despite its potential, exhibits overall limited efficacy and suboptimal performance in accurately classifying images of Al-Mojawhar characters compared to the other classifiers. In addition, the results also reveal that the polynomial-based Support Vector Machine (SVM) presented superior performance compared to K-Nearest Neighbors (KNN),

Random Forest (RF) and Multilayer Perceptron (MLP), with regard to classification rates across the majority of feature sets.

Specifically, the fusion of the polynomial-based SVM classifier and the informative Histogram of Oriented Gradients (HOG) feature set, resulted in a remarkable classification accuracy rate of 88.8%, highlighting a discriminative power and robustness in effectively categorizing and distinguishing patterns within Al-Mojawhar characters.

Specifically in this research, the proposed CNN model yielded superior results, surpassing the performance of SVM, KNN, MLP, and Random Forest classifiers for the AL-Mojawhar character recognition. **Table 16.** The achieved results, with an accuracy of 95.6% using the Adam optimizer and 95.2% using RMSProp, can be attributed to the exceptional capability of the CNN in extracting relevant features through convolution and pooling. These outcomes prove that learned features derived from deep learning models are more precise and representative compared to handcrafted features for ancient Arabic handwriting recognition.

Finally, we must mention that data augmentation serves as a convenient solution for machine learning problems to provide the models with sufficient amount of data by generating new instances for each of the initial data classes. By augmenting the data ten times (x10), the recognition rate knew a notable increase of approximately 5%, which holds great significance in the context of character recognition. **Table 15** demonstrates the effectiveness of data augmentation in improving the performance of the models.

Conclusion

In this chapter, our primary focus was the examination of the MOJ-DB database. Through the implementation of various designed experiments, our aim was to identify the most appropriate features and classifiers that would facilitate the accurate recognition of Al-Mojawhar calligraphy writing. Our investigation encompassed the utilization of several widely recognized classifiers, namely a neural network, support vector machines, Random Forest, K-Nearest Neighbors (KNN), and a convolutional neural network (CNN). In addition, this study focused also on the use of some of the most known feature descriptors employed in OCR. Prominent techniques such as Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), Zernike Moment, and Gabor filter were applied and evaluated.

To construct robust and efficient systems, we implemented various combinations of classifiers and descriptors. By exploring all the mentioned descriptors and classifiers, we aimed to identify the best combination between classifiers and descriptors, thereby establishing highly effective recognition systems. Subsequent to a series of rigorous and carefully executed experiments, it became evident that the Support Vector Machine (SVM) exhibited the highest accuracy when combined with a polynomial kernel and supplied with Histogram of Oriented Gradients (HOG) feature vectors, achieving an accuracy rate of 88.8%.

By employing the technique of data augmentation, we successfully expanded the size of our database, thereby facilitating the enhancement of classification results of the SVM-HOG configuration. This augmentation significantly improved the classification accuracy from an

initial 81.4% to 88.8%. Subsequently, we proceeded to construct a Convolutional Neural Network (CNN) model with the inherent capability to accurately recognize the Al-Mojawhar writing characters. Remarkably, the developed CNN model presented an exceptional success rate of 95.6% on the augmented database.

Conclusions and Perspectives

1. Conclusions

Through this thesis, we investigate the segmentation and recognition of optical Arabic text. Recognizing the need for further research in this area, we aim to contribute to the field of Arabic Optical Character Recognition by proposing various ideas and methods related to pre-processing, segmentation, feature extraction, and classification of Arabic printed and ancient text.

Our first contribution is a novel hybrid method for segmenting multi-font printed Arabic text. This method utilizes primarily the contour of words as a feature to identify segmentation points and extract individual characters. Furthermore, in a second contribution, we have developed a reference corpus specifically designed for evaluating and comparing segmentation systems. This corpus takes into account challenging scenarios that segmentation systems may encounter. By providing researchers with this corpus, we aim to facilitate accurate insights and advancements in the field of Arabic OCR. The results of our method of printed multi-font word-to-character segmentation demonstrated a high accuracy rate of 94.74% outperforming the other methods included in this study.

Shifting our focus to a more intricate research domain, we explored the digitization of Moroccan historical documents. Motivated by a strong drive and enthusiasm, our aim was to contribute to the preservation of these documents, safeguarding their cultural value and invaluable content. We firmly believe that implementing an Optical Character Recognition (OCR) system specifically designed for this type of text is essential to ensure the long-term storage and exploration of this historical material. Moreover, our objective extended to revitalizing the Moroccan calligraphic Al-Mojawhar style, as well as promoting its artistic aspects for enhanced learning and understanding. Within this context, our contributions include a preprocessing method to correct the skew angle of ancient documents using the projection method. We have also developed a technique for extracting text lines from Moroccan ancient documents written in the Al-Mojawhar style, resolving the problem of overlapping and touching lines. The proposed method yielded promising results with a segmentation accuracy of 90.94%.

Furthermore, we delve deeper in the segmentation of Arabic historical documents, by proposing a contour-based method for the extraction of subwords from text lines. Our proposed method outperformed the previous works in terms of both quantity and quality. The extraction method enables us creating a large database, which we named MOJ-DB consisting of 560000 of Al-Mojawhar subwords. This database, being the largest of its kind, aims to address the main challenges encountered in historical text research, by serving as a standard resource for researchers to build and evaluate their models. The proposed method of subword extraction presented a high accuracy with a successful segmentation rate of 83.2%

Lastly, we conducted extensive experiments using state-of-the-art methods of feature extraction and classification to study the recognition of Al-Mojawhar characters of MOJ-DB. Notably, we found that deep learning models exhibit superior capabilities in extracting more precise features compared to modern statistical descriptors.

Through these contributions, we strive to advance the field of Arabic OCR and facilitate the understanding, preservation, and appreciation of Arabic text, particularly in the context of historical and artistic documents.

2. Perspectives and future works

In addition to the valuable contributions made by this thesis, there are several promising directions for future research that can build upon our results and further advance the understanding of Arabic optical text recognition (OCR). Through our study, we have addressed significant challenges in OCR and presented notable contributions. However, in the process, we have also identified novel research aspects that require exploration and additional contributions, among them we cite the following noteworthy points :

- Improving the segmentation technique of printed Arabic text by using recent state-of-the-art methods of text segmentation like deep learning. Also, studying of the global approach of recognition with new segmentation-free methods. Developing our corpus is also one of our future interests as we can add new standard fonts and texts features to extend it in order to cover all types of Arabic text.
- Contributing to the optical text pre-processing prior to segmentation and recognition of historical text. In fact, despite being decisive, the phase of pre-processing does not catch the attention of researchers. We believe that improving the pre-processing operations on optical text can help enhance the accuracy of subsequent segmentation or recognition. Indeed, the segmentation and recognition of ancient scripts need an automatic image pre-processing to perform noise and side notes' removal, border and stamp elimination, and foreground text separation.
- As previously stated in "chapter 3, Data collection section", we used half of the available data to build MOJ-DB database. We intend to add the other half of pages, thus reaching one million subword images.
- Improving our subword extraction method by addressing its current limitations by firstly introducing dynamic parameters to adapt the extraction process to other writing styles. Secondly, developing a robust system for the separation of touching components.
- Developing applications of Al-Mojawhar recognition, on the line, words and character levels.
- Exploring other Moroccan calligraphic styles like Al-Mabsout, Thuluth, Zmami … etc.
- Finally, developing a complete OCR system of historical text recognition, incorporating the complete system of segmentation and classification.

References

[1] B. Al-Badr and R. M. Haralick, “Segmentation-free word recognition with application to Arabic,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 1, pp. 355–359, 1995, doi: 10.1109/ICDAR.1995.599012.

[2] M. Shatnawi, “Off-line handwritten Arabic character recognition: A survey,” *Proc. 2015 Int. Conf. Image Process. Comput. Vision, Pattern Recognition, IPCV 2015*, pp. 52–58, 2015.

[3] A. M. Nazif, “A System for the Recognition of the Printed Arabic Characters,” Faculty of Engineering, Cairo University, 1975.

[4] H. El Abed, V. Margner, M. Kherallah, and A. M. Alimi, “ICDAR 2009 Online Arabic handwriting recognition competition,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 1388–1392, 2009, doi: 10.1109/ICDAR.2009.284.

[5] M. T. Parvez and S. A. Mahmoud, “Offline Arabic handwritten text recognition: A survey,” *ACM Comput. Surv.*, vol. 45, no. 2, 2013, doi: 10.1145/2431211.2431222.

[6] S. Suganya, “Analysis of Feature Extraction of Optical Character detection in Image Processing Systems,” pp. 1–8, 2015.

[7] A. Zoizou, A. Zarghili, and I. Chaker, “A new hybrid method for Arabic multi-font text segmentation, and a reference corpus construction,” *J. King Saud Univ. - Comput. Inf. Sci.*, 2018, doi: 10.1016/j.jksuci.2018.07.003.

[8] A. Zoizou, A. Zarghili, and I. Chaker, “Skew Correction and Text Line Extraction of Arabic Historical Documents,” *Commun. Comput. Inf. Sci.*, vol. 1108, pp. 181–193, 2019, doi: 10.1007/978-3-030-32959-4_13.

[9] A. Zoizou, A. Zarghili, and I. Chaker, “MOJ-DB: A new database of Arabic historical handwriting and a novel approach for subwords extraction,” *Pattern Recognit. Lett.*, vol. 159, pp. 54–60, 2022, doi: 10.1016/j.patrec.2022.04.040.

[10] L. Likforman-Sulem, A. Zahour, and B. Taconet, “Text line segmentation of historical documents: A survey,” *Int. J. Doc. Anal. Recognit.*, vol. 9, no. 2–4, pp. 123–138, 2007, doi: 10.1007/s10032-006-0023-z.

[11] M. Arivazhagan, H. Srinivasan, and S. Srihari, “A statistical approach to line segmentation in handwritten documents,” *Doc. Recognit. Retr. XIV*, vol. 6500, p. 65000T, 2007, doi: 10.1117/12.704538.

[12] I. Bar-Yosef, N. Hagbi, K. Kedem, and I. Dinstein, “Line segmentation for degraded handwritten historical documents,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 1161–1165, 2009, doi: 10.1109/ICDAR.2009.191.

[13] L. Likforman-Sulem, A. Hanimyan, and C. Faure, “A Hough based algorithm for extracting text lines in handwritten documents,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2, pp. 774–777, 1995, doi: 10.1109/ICDAR.1995.602017.

[14] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, “Text line detection in handwritten

documents," *Pattern Recognit.*, vol. 41, no. 12, pp. 3758–3772, 2008, doi: 10.1016/j.patcog.2008.05.011.

[15] S. Malakar, S. Halder, R. Sarkar, N. Das, S. Basu, and M. Nasipuri, "Text line extraction from handwritten document pages using spiral run length smearing algorithm," in *2012 international conference on communications, devices and intelligent systems*, 2012, pp. 616–619.

[16] Z. Shi and V. Govindaraju, "Line Separation for Complex Document Images Using Fuzzy Runlength," *Proc. - First Int. Work. Doc. Image Anal. Libr. - DIAL 2004*, pp. 306–312, 2004, doi: 10.1109/dial.2004.1263259.

[17] U. V. Marti and H. Bunke, "The IAM-database: An English sentence database for offline handwriting recognition," *Int. J. Doc. Anal. Recognit.*, vol. 5, no. 1, pp. 39–46, 2003, doi: 10.1007/s100320200071.

[18] M. Elzobi, A. Al-Hamadi, Z. Al Aghbari, and L. Dings, "IESK-ArDB: A database for handwritten Arabic and an optimized topological segmentation approach," *Int. J. Doc. Anal. Recognit.*, vol. 16, no. 3, pp. 295–308, 2013, doi: 10.1007/s10032-012-0190-z.

[19] A. Lamsaf, M. Aitkerroum, S. Boulaknadel, and Y. Fakhri, *Text Line and Word Extraction of Arabic Handwritten Documents*, vol. 1. Springer International Publishing, 2019.

[20] A. Al-Dmour and R. A. Zitar, "Word extraction from Arabic handwritten documents based on statistical measures," *Int. Rev. Comput. Softw.*, vol. 11, no. 5, pp. 436–444, 2016, doi: 10.15866/irecos.v11i5.9384.

[21] M. Tanvir Parvez and S. A. Mahmoud, "Arabic handwriting recognition using structural and syntactic pattern attributes," *Pattern Recognit.*, vol. 46, no. 1, pp. 141–154, 2013, doi: 10.1016/j.patcog.2012.07.012.

[22] B. Parhami and M. Taraghi, "Automatic recognition of printed Farsi texts," *Pattern Recognit.*, vol. 14, no. 1–6, pp. 395–403, 1981, doi: 10.1016/0031-3203(81)90084-4.

[23] A. Amin and G. Masini, "Machine recognition of cursive Arabic words," in *SPIE 26th International Symposioum on Instrument Display*, 1982, p. . 286-292.

[24] F. El-Khaly and M. A. Sid-Ahmed, "Machine recognition of optically captured machine printed Arabic text," *Pattern Recognit.*, vol. 23, no. 11, pp. 1207–1214, 1990, doi: 10.1016/0031-3203(90)90116-3.

[25] H. Goraine, M. Usher, and S. AL-Emami, "Offline printed Arabic character recognition," no. May, pp. 71–74, 1992, [Online]. Available: http://etheses.nottingham.ac.uk/2601/.

[26] H. Al-sadoun, "A New Structural Technique for Recognizing Printed Arabic Text," *Joumal Pattern Recognit. Artif. Intell.*, vol. 9, no. 1, pp. 101–125, 1995.

[27] B. M. F. Bushofa and M. Spann, "Segmentation of Arabic characters using their contour information," *Digit. Signal Process. Proceedings, 1997. DSP 97., 1997 13th Int. Conf.*, vol. 2, pp. 683–686, 1997, doi: 10.1109/ICDSP.1997.628443.

[28] V. Margner, "Sarat - A System for the Recognition of Arabic Printed Text," *IEEE*, pp. 561–564, 1992.

[29] L. Peng, C. Liu, X. Ding, and H. Wang, “Multilingual document recognition research and its application in China,” *Proc. - Second Int. Conf. Doc. Image Anal. Libr. DIAL 2006*, vol. 2006, no. March, pp. 126–132, 2006, doi: 10.1109/DIAL.2006.27.

[30] C. Oiivier, H. Miled, K. Romeo, and Y. Lecourtier, “Segmentation and Coding,” pp. 264–268, 1996.

[31] T. Sari, L. Souici, and M. Sellami, “Off-line handwritten Arabic character segmentation algorithm: ACSA,” *Proc. - Int. Work. Front. Handwrit. Recognition, IWFHR*, pp. 452–457, 2002, doi: 10.1109/IWFHR.2002.1030952.

[32] B. P. Gaikwad, “Automatic Video Scene Segmentation to Separate Script for OCR,” pp. 9–15, 2013.

[33] B. M. F. Bushofa and M. Spann, “Segmentation and recognition of Arabic characters by structural classification,” *Image Vis. Comput.*, vol. 15, no. 3, pp. 167–179, 1997, doi: 10.1016/S0262-8856(96)01119-5.

[34] S. Naz, M. I. Razzak, K. Hayat, M. W. Anwar, and S. Z. Khan, “Challenges in baseline detection of Arabic script based languages,” *Stud. Comput. Intell.*, vol. 542, pp. 181–196, 2014, doi: 10.1007/978-3-319-04702-7_11.

[35] H. Boubaker, M. Kherallah, and A. M. Alimi, “New algorithm of straight or curved baseline detection for short Arabic handwritten writing,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 778–782, 2009, doi: 10.1109/ICDAR.2009.265.

[36] A. Al-shatnawi and K. Omar, “A Comparative Study between Methods of Arabic Baseline Detection,” no. August, pp. 73–77, 2009.

[37] A. Al-shatnawi and K. Omar, “Detecting Arabic Handwritten Word Baseline using Voronoi Diagram,” no. August, pp. 18–22, 2009.

[38] D. Chakraborty and U. Pal, “Baseline detection of multi-lingual unconstrained handwritten text,” *Pattern Recognit. Lett.*, vol. 74, pp. 74–81, 2016, doi: 10.1016/j.patrec.2016.02.003.

[39] L. Zheng, A. H. Hassin, and X. Tang, “A new algorithm for machine printed Arabic character segmentation,” *Pattern Recognit. Lett.*, vol. 25, no. 15, pp. 1723–1729, 2004, doi: 10.1016/j.patrec.2004.06.015.

[40] S. N. Nawaz, M. Sarfraz, and A. Zidouri, “RECOGNITION USING NEURAL NETWORKS,” pp. 1328–1331, 2003.

[41] T. Y. Zhang and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” *Commun. ACM*, vol. 27, no. 3, pp. 236–239, 1984, doi: 10.1145/357994.358023.

[42] A.-E. Samir, “Machine Recognition of Handwritten and Typewritten Arabic Characters,” University of Reading, 1988.

[43] HILDITCH CJ, “Linear Skeletons From Square Cupboards,” pp. 403–420, 1969.

[44] H. Freeman, “On the Encoding of Arbitrary Geometric Configurations,” *IRE Trans. Electron. Comput.*, vol. EC-10, no. 2, pp. 260–268, 1960, doi: 10.1109/TEC.1961.5219197.

[45] J. Cowell and F. Hussain, “Thinning Arabic Characters for Feature Extraction,” pp. 181–185, 2001.

[46] L. Ding and A. Goshtasby, “On the canny edge detector,” *Pattern Recognit.*, vol. 34, no. 3, pp. 721–725, 2001, doi: 10.1016/S0031-3203(00)00023-6.

[47] R. Mehran, H. Pirsiavash, and F. Razzazi, “A Front-End OCR for Omni-Font Persian/Arabic Cursive Printed Documents,” no. February, pp. 56–56, 2006, doi: 10.1109/dicta.2005.3.

[48] L. Peng, C. Liu, X. Ding, and H. Wang, “Multilingual document recognition research and its application in China,” *Proc. - Second Int. Conf. Doc. Image Anal. Libr. DIAL 2006*, vol. 2006, pp. 126–132, 2006, doi: 10.1109/DIAL.2006.27.

[49] M. Sarfraz, S. N. Nawaz, and A. Al-Khuraidly, “Offline Arabic text recognition system,” *Seventh Int. Symp. Signal Process. Its Appl. 2003. Proceedings.*, vol. 2, 2003, doi: 10.1109/ISSPA.2003.1224906.

[50] A. M. Elgammal and M. A. Ismail, “A graph-based segmentation and feature extraction framework for Arabic text recognition,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2001-Janua, pp. 622–626, 2001, doi: 10.1109/ICDAR.2001.953864.

[51] M. M. Altuwaijri and M. A. Bayoumi, “Arabic text recognition using neural networks,” *Proc. IEEE Int. Symp. Circuits Syst. - ISCAS ’94*, vol. 6, pp. 415–418, 1994, doi: 10.1109/ISCAS.1994.409614.

[52] M. A. A. Mousa, M. S. Sayed, and M. I. Abdalla, “Arabic Character Segmenta ation Using Projection-Based Approach h with Profile ’ s Amplitude Filter,” pp. 122–126, 2013.

[53] M. Omidyeganeh, K. Nayeb, R. Azmi, and A. Javadtalab, “A new segmentation technique for multi font Farsi/Arabic texts,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. II, pp. 757–760, 2005, doi: 10.1109/ICASSP.2005.1415515.

[54] A. M. Gouda and M. A. Rashwan, “Segmentation of Connected Arabic Characters Using Hidden Markov Models 1,” *Comput. Eng.*, no. July, pp. 14–16, 2004.

[55] N. ahmed Shaikh, Z. A. Shaikh, and G. Ali, “Segmentation of Arabic text into characters for recognition,” *Commun. Comput. Inf. Sci.*, vol. 20 CCIS, pp. 11–18, 2008, doi: 10.1007/978-3-540-89853-5-4.

[56] H. Ghaleb, P. Nagabhushan, and U. Pal, “Segmentation of Offline Handwritten Arabic Text,” in *2017 IEEE International Workshop on Arabic Script Analysis and Recognition (ASAR) Segmentation*, 2017, pp. 41–45, doi: 10.1109/ASAR.2017.8067757.

[57] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane, “Arabic hand-written text-line extraction,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2001-Janua, pp. 281–285, 2001, doi: 10.1109/ICDAR.2001.953799.

[58] V. Katsouros and V. Papavassiliou, “Segmentation of Handwritten Document Images into Text Lines,” *Image Segmentation*, 2012, doi: 10.5772/15923.

[59] R. Saabni, A. Asi, and J. El-Sana, “Text line extraction for historical document images,” *Pattern Recognit. Lett.*, vol. 35, no. 1, pp. 23–33, 2014, doi: 10.1016/j.patrec.2013.07.007.

[60] A. Garz, A. Fischer, H. Bunke, and R. Ingold, “A binarization-free clustering approach to segment curved text lines in historical manuscripts,” *Proc. Int. Conf. Doc. Anal.*

Recognition, ICDAR, pp. 1290–1294, 2013, doi: 10.1109/ICDAR.2013.261.

[61] A. Garz, A. Fischer, R. Sablatnig, and H. Bunke, "Binarization-free text line segmentation for historical documents based on interest point clustering," *Proc. - 10th IAPR Int. Work. Doc. Anal. Syst. DAS 2012*, pp. 95–99, 2012, doi: 10.1109/DAS.2012.23.

[62] Z. Shi, S. Setlur, and V. Govindaraju, "Text extraction from gray scale historical document images using adaptive local connectivity map," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2005, pp. 794–798, 2005, doi: 10.1109/ICDAR.2005.229.

[63] F. Yin and C. L. Liu, "Handwritten text line extraction based on minimum spanning tree clustering," *Proc. 2007 Int. Conf. Wavelet Anal. Pattern Recognition, ICWAPR '07*, vol. 3, no. December 2007, pp. 1123–1128, 2008, doi: 10.1109/ICWAPR.2007.4421601.

[64] il Koo Hyung and I. Cho Nam, "Text-Line Extraction in Handwritten Chinese Documents Based on an Energy Minimization Framework," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1169–1175, 2012, doi: 10.1109/TIP.2011.2166972.

[65] U. Pal and S. Datta, "Segmentation of Bangla unconstrained handwritten text," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2003-Janua, pp. 1128–1132, 2003, doi: 10.1109/ICDAR.2003.1227832.

[66] W. Boussellaa, A. Zahour, H. Elabed, A. Benabdelhafid, and A. Alimi, "Unsupervised block covering analysis for text-line segmentation of Arabic ancient handwritten document images," *Proc. - Int. Conf. Pattern Recognit.*, no. 1, pp. 1929–1932, 2010, doi: 10.1109/ICPR.2010.475.

[67] N. Ouwayed, A. Belaïd, and F. Auger, "General text line extraction approach based on locally orientation estimation," *Doc. Recognit. Retr. XVII*, vol. 7534, no. January, p. 75340B, 2009, doi: 10.1117/12.839518.

[68] S. Capobianco and S. Marinai, "Text Line Extraction in Handwritten Historical Documents," in *Italian Research Conference on Digital Libraries*, 2017, pp. 68–79, doi: 10.1007/978-3-319-68130-6-6.

[69] M. H. Mohamed Dyla and F. Morain-Nicolier, "Text line segmentation and binarization of handwritten historical documents using the fast and adaptive bidimensional empirical mode decomposition," *Optik (Stuttg).*, vol. 188, no. March, pp. 52–63, 2019, doi: 10.1016/j.ijleo.2019.04.128.

[70] S. M. A. Bhuiyan, R. R. Adhami, and J. F. Khan, "A novel approach of fast and adaptive bidimensional empirical mode decomposition," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2, pp. 1313–1316, 2008, doi: 10.1109/ICASSP.2008.4517859.

[71] N. E. Huang *et al.*, "The empirical mode decomposition and the Hubert spectrum for nonlinear and non-stationary time series analysis," *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, 1998, doi: 10.1098/rspa.1998.0193.

[72] W. Niblack, *An introduction to digital image processing*. Prentice-Hall International, 1986.

[73] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognit.*, vol. 33, no. 2, pp. 225–236, 2000, doi: 10.1016/S0031-3203(99)00055-2.

[74] R. G. Casey and E. Lecolinet, “survey of Methods and Str in Character Segmentation,” *IEEE Anal.*, vol. 18, no. 7, pp. 690–706, 1996, doi: 10.1109/34.506792.

[75] K. Y. Wong, R. G. Casey, and F. M. Wahl, “Document Analysis System,” *IBM J. Res. Dev.*, vol. 26, no. 6, pp. 647–656, 2010, doi: 10.1147/rd.266.0647.

[76] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Trans. Syst. Man, Cybern.*, vol. 9, no. 1, pp. 62–66, 1979, [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4310076.

[77] A. SAVITZKY and M. J. E. GOLAY, “Smoothing and Differentiation of Data by Simplified Least Squares Procedures,” *Anal. Chem*, vol. 2, no. 8, pp. 1627–1639, 1964, doi: 10.1021/ac60214a047.

[78] N. Aouadi and A. K. Echi, “Word Extraction and Recognition in Arabic Handwritten Text,” *Int. J. Comput. Inf. Sci.*, vol. 12, no. 1, pp. 17–23, 2016, doi: 10.21700/ijcis.2016.103.

[79] N. Aouadi and A. Kacem, “A proposal for touching component segmentation in Arabic manuscripts,” *Pattern Anal. Appl.*, vol. 20, no. 4, pp. 1005–1027, 2017, doi: 10.1007/s10044-016-0543-1.

[80] L. Kang and D. Doermann, “Template based segmentation of touching components in handwritten text lines,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 569–573, 2011, doi: 10.1109/ICDAR.2011.120.

[81] R. O. Duda and P. E. Hart, “Use of the Hough Transformation to Detect Lines and Curves in Pictures,” *Commun. ACM*, vol. 15, no. 1, pp. 11–15, 1972, doi: 10.1145/361237.361242.

[82] M. Kassis, A. Abdalhaleem, A. Droby, R. Alaasam, and J. El-Sana, “VML-HD: The historical Arabic documents dataset for recognition systems,” pp. 11–14, 2017, doi: 10.1109/asar.2017.8067751.

[83] R. F. Moghaddam, M. Cheriet, M. M. Adankon, K. Filonenko, and R. Wisnovsky, “IBN SINA: A database for research on processing and understanding of Arabic manuscripts images,” *ACM Int. Conf. Proceeding Ser.*, pp. 11–17, 2010, doi: 10.1145/1815330.1815332.

[84] M. Pechwitz *et al.*, “IFN/ENIT-database of handwritten Arabic words,” *Proc. of CIFED*, vol. 2, pp. 1–8, 2002, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.1245&rep=rep1&type=pdf.

[85] A. Fischer, V. Frinken, A. Fornés, and H. Bunke, “Transcription alignment of Latin manuscripts using hidden Markov models,” *ACM Int. Conf. Proceeding Ser.*, pp. 29–36, 2011, doi: 10.1145/2037342.2037348.

[86] R. Christian, W. Christoph, N. Maximilian, B. Andreas, W. Maximilian, and S. Uwe, *Mixed Model OCR Training on Historical Latin Script for Out-of-the-Box Recognition and Finetuning*, vol. 1, no. 1. Association for Computing Machinery, 2021.

[87] K. Khurshid, C. Faure, and N. Vincent, “Word spotting in historical printed documents using shape and sequence comparisons,” *Pattern Recognit.*, vol. 45, no. 7, pp. 2598–2609, 2012, doi: 10.1016/j.patcog.2011.10.013.

[88] U. Reffle and C. Ringlstetter, “Unsupervised profiling of OCRed historical documents,” *Pattern Recognit.*, vol. 46, no. 5, pp. 1346–1357, 2013, doi: 10.1016/j.patcog.2012.10.002.

[89] U. Springmann, D. Najock, H. Morgenroth, H. Schmid, A. Gotscharek, and F. Fink, “OCR of historical printings of latin Texts: Problems, prospects, progress,” *ACM Int. Conf. Proceeding Ser.*, pp. 71–75, 2014, doi: 10.1145/2595188.2595205.

[90] S. A. Mahmoud *et al.*, “KHATT: An open Arabic offline handwritten text database,” *Pattern Recognit.*, vol. 47, no. 3, pp. 1096–1112, 2014, doi: 10.1016/j.patcog.2013.08.009.

[91] S. Al-Ma’adeed, D. Elliman, and C. A. Higgins, “A data base for Arabic handwritten text recognition research,” *Proc. - Int. Work. Front. Handwrit. Recognition, IWFHR*, vol. 1, no. 1, pp. 485–489, 2002, doi: 10.1109/IWFHR.2002.1030957.

[92] M. Kassis and J. El-Sana, “Word spotting using radial descriptor graph,” *Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR*, vol. 0, pp. 31–35, 2016, doi: 10.1109/ICFHR.2016.0019.

[93] N. Lamghari and S. Raghay, “DBAHCL: database for Arabic handwritten characters and ligatures,” *Int. J. Multimed. Inf. Retr.*, vol. 6, no. 3, pp. 263–269, 2017, doi: 10.1007/s13735-017-0127-x.

[94] L. G. Shapiro, “Connected Component Labeling and Adjacency Graph Construction,” *Mach. Intell. Pattern Recognit.*, vol. 19, no. C, pp. 1–30, 1996, doi: 10.1016/S0923-0459(96)80011-5.

[95] UNESCO World Heritage Centre, “Medina of Fez,” 1981. http://whc.unesco.org/en/list/170.

[96] M. Young, *The Guinness Book of World Records 1998*, 1st ed. 1998.

[97] D. G. Bailey and C. T. Johnston, “Optimised single pass connected components analysis,” *Proc. 2008 Int. Conf. Field-Programmable Technol. ICFPT 2008*, no. May, pp. 185–192, 2008, doi: 10.1109/FPT.2008.4762382.

[98] H. Kaur, H. S. Pannu, and A. K. Malhi, “A systematic review on imbalanced data challenges in machine learning: Applications and solutions,” *ACM Comput. Surv.*, vol. 52, no. 4, 2019, doi: 10.1145/3343440.

[99] M. D Bloice, C. Stocker, and A. Holzinger, “Augmentor: An Image Augmentation Library for Machine Learning,” *J. Open Source Softw.*, vol. 2, no. 19, p. 432, 2017, doi: 10.21105/joss.00432.

[100] D. G. Low, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, pp. 91–110, 2004.

[101] M. J. Mohammed, S. M. Tariq, and H. Ayad, “Isolated Arabic handwritten words recognition using EHD and HOG methods,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 2, p. 801, 2021, doi: 10.11591/ijeecs.v22.i2.pp801-808.

[102] L. Chergui and M. Kef, “SIFT descriptors for Arabic handwriting recognition,” *Int. J. Comput. Vis. Robot.*, vol. 5, no. 4, pp. 441–461, 2015, doi: 10.1504/IJCVR.2015.072193.

[103] H. C. Shin *et al.*, "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016, doi: 10.1109/TMI.2016.2528162.

[104] H. Hassen and M. Khemakhem, "A Comparative study of Arabic handwritten characters invariant feature," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 12, pp. 62–68, 2011, doi: 10.14569/ijacsa.2011.021209.

[105] A. K. A. Hassan, B. S. Mahdi, and A. A. Mohammed, "Arabic handwriting word recognition based on scale invariant feature transform and support vector machine," *Iraqi J. Sci.*, vol. 60, no. 2, pp. 381–387, 2019, doi: 10.24996/ijs.2019.60.2.18.

[106] M. Elleuch, H. Lahiani, and M. Kherallah, "Recognizing Arabic Handwritten Script using Support Vector Machine classifier," *Int. Conf. Intell. Syst. Des. Appl. ISDA*, vol. 2016-June, pp. 551–556, 2016, doi: 10.1109/ISDA.2015.7489176.

[107] M. Elleuch, A. Hani, and M. Kherallah, "Arabic handwritten script recognition system based on HOG and gabor features," *Int. Arab J. Inf. Technol.*, vol. 14, no. 4A Special Issue, pp. 639–646, 2017.

[108] H. Althobaiti and C. Lu, "A survey on Arabic Optical Character Recognition and an isolated handwritten Arabic Character Recognition algorithm using encoded freeman chain code," *2017 51st Annu. Conf. Inf. Sci. Syst. CISS 2017*, pp. 2–7, 2017, doi: 10.1109/CISS.2017.7926062.

[109] N. A. Jebril, H. R. Al-Zoubi, and Q. Abu Al-Haija, "Recognition of Handwritten Arabic Characters using Histograms of Oriented Gradient (HOG)," *Pattern Recognit. Image Anal.*, vol. 28, no. 2, pp. 321–345, 2018, doi: 10.1134/S1054661818020141.

[110] N. Altwaijry and I. Al-Turaiki, "Arabic handwriting recognition system using convolutional neural network," *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2249–2261, 2021, doi: 10.1007/s00521-020-05070-8.

[111] K. Younis and A. Khateeb, "Arabic Hand-Written Character Recognition Based on Deep Convolutional Neural Networks," *Jordanian J. Comput. Inf. Technol.*, vol. 3, no. 3, p. 186, 2017, doi: 10.5455/jjcit.71-1498142206.

[112] J. Bai, Z. Chen, B. Feng, and B. Xu, "Image character recognition using deep convolutional neural network learned from different languages," *2014 IEEE Int. Conf. Image Process. ICIP 2014*, pp. 2560–2564, 2014, doi: 10.1109/ICIP.2014.7025518.

[113] Nesrine Wagaa, Hichem Kallel, and Nedra Mellouli, "Improved Arabic Alphabet Characters Classification Using Convolutional Neural Networks (CNN)," *Comput. Intell. Neurosci.*, vol. 2022, p. 16, 2021.

[114] N. Alrobah and S. Albahli, "A Hybrid Deep Model for Recognizing Arabic Handwritten Characters," *IEEE Access*, vol. 9, pp. 87058–87069, 2021, doi: 10.1109/ACCESS.2021.3087647.

[115] M. Shams, A. A. Elsonbaty, and W. Z. El Sawy, "Arabic handwritten character recognition based on convolution neural networks and support vector machine," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 8, pp. 144–149, 2020, doi: 10.14569/IJACSA.2020.0110819.

[116] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005,

pp. 886–893, doi: 10.1007/978-3-319-24947-6.

[117] A. Khotanzad and Y. H. Hong, “Invariant Image Recognition by Zernike Moments,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 5, pp. 489–497, 1990, doi: 10.1109/34.55109.

[118] X. Wang, X. Ding, and C. Liu, “Gabor filters-based feature extraction for character recognition,” *Pattern Recognit.*, vol. 38, no. 3, pp. 369–379, 2005, doi: 10.1016/j.patcog.2004.08.004.