

**THE APPLICATION OF NEW METHODS FOR OFFLINE
RECOGNITION IN PRINTED ARABIC DOCUMENTS**

A DISSERTATION
SUBMITTED TO THE PHD SCHOOL IN COMPUTER SCIENCE
OF THE UNIVERSITY OF SZEGED
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

BY HASSINA BOURESSACE



Supervior: Dr.János Csirik
Professor emeritus

Szeged, 2020

*To my loving parents for their endless love and
care and for their selfless devotion,
encouragement and support.*

Acknowledgements

First and foremost, praises and thanks must go to Allah Almighty, who blessed me with great love and fortune. I am especially thankful to Him for blessing me with the gift of knowledge, a loving family and a beautiful life.

I am very lucky and can never forget the close support and guidance of my supervisor, Professor Janos Csirik, for his good advice and help during the whole duration of my study here. I am also very grateful to my colleagues who helped me with my studies and stayed by my side during the tough and happy times during my PhD studies in the city of Szeged.

I would also like to thank the administrative staff for their help over these four years and David P. Curley for scrutinizing and correcting this thesis from a linguistic point of view. Last, but not least, I wish to thank my dear parents and all my siblings for their constant love and support. I would like to dedicate this thesis to them as a way of expressing my gratitude and appreciation.

This research was supported by the Stipendium Hungaricum scholarship Program. I am grateful for the opportunity they provided, which motivated me and has led to the submission of this thesis.

Contents

Acknowledgements	ii
Abbreviations	vi
List of Figures	xi
List of Tables	xii
1 Introduction	1
1.1 Motivation	2
1.2 Problem Statement and Aim of the Study	2
1.3 Methodological Approach and Innovative Aspects	5
1.4 Structure of the Thesis	6
1.5 Published Articles Related to the Dissertation	7
2 State of the Art	8
2.1 Binarization	8
2.1.1 Global Methods	9
2.1.2 Adaptive Methods	10
2.1.3 Methods Based on the Combination of Different Binarizations	11
2.2 Smoothing and Noise Reduction	11
2.2.1 Filtering Techniques	12
2.2.2 Smearing Techniques	15
2.3 Neural Networks and Deep Learning	17
2.3.1 Self-Organizing Map	17
2.3.2 Convolutional Neural Networks	18
2.4 Document Layout Analysis	20
2.4.1 Physical Structure Analysis	20
2.4.2 Logical Structure Analysis	24
2.5 Text Line/Word Segmentation	26
2.5.1 Text Line Segmentation	26
2.5.2 Word Segmentation	27

3	Arabic Document Database	29
3.1	Introduction	29
3.2	Features of the Arabic Language	31
3.3	Documents Image Selection	32
3.3.1	Characteristics of Newspaper/Magazine Pages	34
3.3.2	Documents Data Capture	35
3.3.3	Capture Parameters	36
3.4	Ground Truth File Description	39
3.5	Summary	40
4	Arabic Document Layout analysis	41
4.1	Printed Arabic Newspaper	42
4.1.1	Method Overview	43
4.1.2	Results and Discussion	58
4.1.3	Summary	61
4.2	Title Detection in Printed Arabic Newspaper Pages	62
4.2.1	Overview of the Method Used	63
4.2.2	Pre-processing	64
4.2.3	Title Segmentation	64
4.2.4	Subtitle Extraction	66
4.2.5	Results And Discussion	68
4.2.6	Summary	71
4.3	Smartphone-captured Arabic Newspaper Analysis	72
4.3.1	Method Overview	73
4.3.2	Distortion Correction	73
4.3.3	Morphological Operations	76
4.3.4	Connected Component Extraction	78
4.3.5	Conventional Neural Network Classification	79
4.3.6	Page Segmentation	81
4.3.7	Results and Discussion	82
4.3.8	Summary	84
5	Arabic Handwritten Word Detection	85
5.1	Introduction	85
5.2	Method Overview	86
5.3	Pre-processing	87
5.4	Segmentation	88
5.4.1	Smoothing Technique	88
5.4.2	Feature Extraction	89
5.4.3	Kohonen Map Learning	90
5.5	Experimental Results	92
5.5.1	Database	93
5.5.2	Training and Error Analysis	93

5.5.3 Results	94
5.6 Summary	96
Summary	97
Bibliography	102

Abbreviations

ANN	Artificial neural network
ARLSA	Adaptive Run-length Smoothing Algorithm
BLSTM	Bidirectional Long-Short Term Memory
BMU	Best Matching Unit
CC	Connected Component
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
CTM	Cut Text Minimization
DAL	Document Architecture Language
DAN	Document Analysis on Network
DAR	Document Analysis and Recognition
DLA	Document Layout Analysis
DTD	Document Type Definition
HPP	Horizontal Projection Profile
NLP	Natural Language Processing
OC	Overlapped Component
OCR	Optical Character Recognition
PP	Projection Profile
ReLu	Rectified Linear Unit
RLSA	Run Length Smearing/Smoothing Algorithm
RLSA	Run-length Smoothing Algorithm
SOM	Self-organizing Map
VPP	Vertical Projection Profile
XML	Extensible Markup Language

List of Figures

1.1	Examples of Arabic document pages	2
1.2	A hierarchy of document processing [90, page 2], [38].	3
1.3	Examples of documents analysis. Each type of layout is highlighted in a different color.	4
1.4	Examples of Arabic document used for related works.	5
2.1	The results of many possible thresholds applied globally on the image [109].	9
2.2	Example of a Otsu/manually binarization results: (a) The image in gray; (b) Histogram with the Otsu threshold (dashed) and manual threshold (dashed-dotted); (c) The Otsu threshold image; (d) A manually thresholded image [40].	10
2.3	Degraded document image example: (a) Original image; (b) Binarization results produced by the adaptive method [32].	11
2.4	Example of a combination of binarization results: (a)(b) Two degraded document image examples; (c)(d) Binarization results produced by Otsu's method; (e)(f) Binarization results produced by Sauvola's method; (g)(h) Combination-binarization method results. [21]	12
2.5	Example of the letter "e" with salt-and-pepper noise [1, page 19]. . .	13
2.6	Different possible ways of applying Gaussian filtering to a sample camera-captured warped document image [140].	13
2.7	Example of Median filter result: (a) An example of a degraded image; (b) A filtering result produced by the Median method. [21]	14
2.8	Example of some morphological operation results [133].	15
2.9	Example of a smearing operation result:(a) Document image example; (b) A smearing result using the run-length smoothing algorithm [134].	16
2.10	Example of a smearing operation result:(a) Document image example; (b) A smearing result using the binary transition count map [134]. . .	16
2.11	Example of smearing operation result:(a) Document image example; (b) A smearing result using the adaptive local connectivity map [134].	17
2.12	An simple CNN architecture, represented by five layers [81].	19
2.13	The physical structure of a newspaper page [53, 103].	20

2.14	An example of the RLSA algorithm: (a) The original Image; (b) Horizontal Smoothing; (c) Vertical Smoothing; (d) Final result of RLSA; (e) Results for blocks treated as text data [84].	21
2.15	An example of an X-Y-Cuts algorithm: (a) Document image; (b) Placement of cuts; (c) Zones subdivided; (d) The X-Y tree of the page layout structure [67].	22
2.16	An example of horizontal and vertical projection profiles of a document image [100].	23
2.17	An example of a physical structure representation [91].	24
2.18	The logical structure of a newspaper page [53,103].	25
2.19	An example of the logical structure representation [91].	26
3.1	An example of different shapes of an Arabic letter.	31
3.2	An example of Arabic vowels.	31
3.3	Examples of image-shapes: (a1) A rectangular picture on the right; (a2) Two combined rectangular pictures in the centre; (b1) A circular picture on the left; (b2) A combined circular and random picture; (c1) A random shape of picture in the centre; (c2) Random shape overlapping some text.	33
3.4	Examples of image texts-blocks and title-blocks: (a) AL-Quds; (b) AXTManal; (c) Hacen Promoter; (d) Beirut; (e) AL Hadith; (f) Kacstone; (g) AL-sharek Title; (h) Hacen Algeria; (i) Hacen Qatar; (j) Mariam; (k) MCS Topaz Brok out; (l) Hacen Extender X4 super fit; (m) Yakout; (n) Kufah.	33
3.5	Sample of a journal page component.	34
3.6	Sample images from the PATD database with different types of lighting conditions: (a) In sunlight; (b) Shaded; (c) In artificial light.	37
3.7	Sample images from the PATD database showing two types of motion blur: (a) Horizontal motion blur; (b) Vertical motion blur.	38
3.8	Sample images from the PATD database showing two levels of focus-blur with OCR accuracy: (a) Relatively out-of-focus blur (b) Totally out-of-focus blur.	39
4.1	The diagram of our DAR process.	44
4.2	An example of connected component labeling: (a) The original image; (b) Binarization result; (c) Labeling result.	45
4.3	Overall accuracy for the non-text threshold: W(image-width), H(image-Height).	47
4.4	Overall accuracy for the border/black thread threshold.	47
4.5	Overall accuracy for the text-black-band threshold.	48
4.6	An example of the detection process: (a) The binarized image of a non-text layout; (b) Converting the layouts to the negative version; (c) Applying Horizontal ARLSA; (d) Extracting the new bounding-boxes; (A) Example of an article-black-band; (B) Example of a figure layout.	49

4.7	Example of detecting and removing graphics: (a) The same newspaper page used in the previous step; (b) The graphic layout elimination result.	49
4.8	Example of histograms projection on newspaper page part.	50
4.9	Example of article/ block segmentation step: (a) Article segmentation result; (b) Block segmentation result.	50
4.10	Example of segmentation blocks into lines.	52
4.11	The word detection process: (a) The original line; (b) The negative version; (c) Bounding-box extraction after applying (RLSA algorithm + CCs labeling); (d) Space extraction; (e) Space filtration; (f) Line segmentation into words.	53
4.12	A sample of words detected in different articles.	53
4.13	Overall accuracy for M_l , DF_1 and DF_2 threshold values.	54
4.14	Example of legend detection process: (a) The original image; (b) Figure extracted; (c) Binarization result; (d) Bottom-part extracted; (e) Legend/figure detection result.	55
4.15	Example of legend detection process: (a) The original image; (b) Smearing result; (c) Bounding-box extraction result; (A1) The legend existent; (A2) The legend non-existent.	56
4.16	Example of a segmentation page and its logical structure: (a) Newspaper page along with its articles; (b) The detected logical elements.	57
4.17	Example of XML and DTD file: (a) XML file of the previous image; (b) DTD file of the previous image (Figure 4.10).	58
4.18	Example of some typical errors in the physical and logical phase.	61
4.19	Outline of our method proposed for Title/Subtitle detection.	63
4.20	The input image with a plot of the HPP on the right.	64
4.21	Examples of removing figures and black blocks.	65
4.22	The title segmentation results of our proposed method on Arabic text documents: (a) Newspaper page with textual data; (b) Magazine page with graphical/textual data.	66
4.23	The subtitle segmentation results for an Arabic document page.	67
4.24	The number of experiments of our data thresholds.	68
4.25	Samples of images used for the line segmentation method: (a) Ibrahim's data [55]: an article with the same text size in one column; (b) Soujanya et al.'s data [111]: an article with a different size font in one column; (c) Ayesha et al.'s data [93]: variability of font size and the possibility of multiple articles, which was restricted by the presence of vertical white spaces between them; (d) Our own data where several articles have different font sizes and figures.	70
4.26	Outline of our method proposed for Arabic document analysis.	74
4.27	Outline of graphical/textual labels.	74

4.28	Examples of skew types. The original skew is highlighted in blue and the resulting skew is highlighted in red.	75
4.29	Example of sharpness correction:(First) Original image,(Second) De-blurred image.	75
4.30	The binarized image of a shaded document image: (a) Original image; (b) Binarized image without Gaussian filter; (c) Binarized image without Gaussian adaptive thresholding; (d) Binarized image with (Gaussian filter +Mean adaptive thresholding+Gaussian adaptive thresholding).	76
4.31	Some results of opening and dilation operations: (a) The binarized image; (b) Morphological opening; (c) Morphological dilation.	77
4.32	The CC ordering according to its y-coordinate (each consecutive block of five CCs is highlighted in one color).	78
4.33	Selected example based on particular features:(a) The resultant image from the previous step is partially colored for the selected CC; (b) The resultant image after removing the red parts; (c) The resultant image after removing the black parts.	79
4.34	Results of ARLSA and RLSA: (First), using ARLSA, (second), using RLSA.	79
4.35	Examples of extracted patches in textual/graphical status:(A1) Patch of figure; (A2) Patch of text; (A3) Random selection of patches extracted from the previous figure; (a) Part of a resultant image from the previous step; (b) CC selection; (c) Bounding-box of CC; (d) The context image.	80
4.36	The proposed network architecture for patch classification using VGG-16 architecture [82].	80
4.37	Example of CNN results.	81
4.38	Example of label extraction.	82
5.1	Handwritten Arabic letters.	86
5.2	Example of a semi-word constituting an Arabic sub-word: (a) 4 semi-words; (b) 1 sub-word [97].	87
5.3	Outline of the method proposed for word extraction.	87
5.4	Example of a preprocessing step result: (a) The original image; (b) After preprocessing.	88
5.5	Example of projection histogram.	88
5.6	Example of text line segmentation result.	88
5.7	Text image after RLSA method.	89
5.8	Training data file sample	90

5.9	Example of the proposed method: (a) Original image; (b) Connected component labeling; (c) Feature extraction from every connected component; (d) We calculate the number of the gaps that exist between the CC; (e) The number of possible gap-words/gap-CCs between the CC are shown; (f) We select the best match result according to the feature data; (g) We segment the CC based on the SOM result.	91
5.10	A word segmenation example (1: between-word; 0: inter-word) . . .	92
5.11	The number of CCs as a function of training images.	93
5.12	Example of a randomly spacing in an Arabic text image: (a) The gaps are equal to each other though the a1 is a separator space between two words, while a2 is a space between two letters from the same word; (b) The gap-word is smaller the gap-letter, where b1 is a gap between two letters from the same word, while b2 is a separator space between two words.	94
5.13	Examples of overlapping problems and good segmentation in Arabic text: (a) The letter (و) and word (هو) are treated as one word; (b) The word above is connected to the word below so the line/word segmentation is ineffective; (c) The words were segmented correctly even when they were very close to each other.	95

List of Tables

3.1	Statistics of the image number used for database production.	32
3.2	Statistics of the database for different smartphone makes with the number of images for each type of phone.	35
3.3	Lighting condition statistics	37
3.4	Motion blur statistics	37
4.1	Example of some labels detected manually and automatically in one news- paper page.	59
4.2	Test results.	60
4.3	A comparaison with other approach.	60
4.4	Test results.	69
4.5	A comparaison with other approach.	70
4.6	The performance of the proposed method on different newspapers. . .	83
4.7	A performance comparaison with different approaches.	83
5.1	Test results.	95
5.2	Comparison with other approaches.	96

Chapter 1

Introduction

In today's world, a lot of the information is still recorded, stored and distributed in paper format, and because of the widespread use of smartphones for collecting and editing document information along with computer equipment for document processing software, the electronic document has become an indispensable element for the exchange of ideas and information during a communication process between people and machines.

Due to a the large number of documents that are continuously increasing every day, many questions and problems have appeared related to the storage of data, retrieval and information processing. This has led to the appearance of new areas of research such as document layout analysis and document element recognition. These elements are organized into a structure that conveys information about the document content to simplify the reading and interpreting step. Therefore, to design a system for recognition, indexing, search and automatic classification, or any other system designed for understanding printed and handwritten documents, it must first be able to recognize the document structure.

A technique called OCR (Optical Character Recognition), one of the earliest addressed computer vision tasks, has gained interest over the last two decades, where many approaches have been developed for diverse issues. Although OCR provides excellent results in many cases, it is limited to very specific use cases where text documents are still considered challenging tasks, especially in magazines and newspaper pages. Therefore, document layout analysis is an important step for OCR, which seeks to represent the document in a structured form by applying a set of computer techniques to facilitate reuse and recovery.

In this dissertation, we focus on this area of computer vision to tackle these tasks. This area involves many research tasks such as layout analysis, (handwritten/printed) text line segmentation, (handwritten/printed) text recognition, graphic element recognition, (text/non-text) element extraction, and document classification.

DLA is a specialist field that contains numerous areas, such as deep learning, image recognition, with big data technology as the most recently developed area.

In the following sections, we describe the motivation for this study, we describe the

main problems of the above tasks and our aims, we present the set of contributions and innovative aspects derived from these studies. Then, we will define the structure of this thesis.

1.1 Motivation

Over the last two decades, a large number of programs for the systematic recognition of writing have been developed, and now researchers have moved towards the analysis and logical labeling of documents, which is represented in a high-level structured form for understanding the hierarchical construction of its elements and the relationships among them automatically. Hence, the raw image can be replaced by a set of structured information exploitable by the machine, and millions of stored paper volumes can be replaced by computer files in XML format. These types of document analysis systems should cut the number of misfiled, misshelved and lost files and will increase automated sorting, automated text/non-text recognition and make it quicker and more accurate especially for Arabic documents, where great effort is still required to attain the performance for English documents. These objectives serve as a motivation for exploring prospective solutions for Arabic document image analysis and Arabic writing analysis.



Figure 1.1: Examples of Arabic document pages

1.2 Problem Statement and Aim of the Study

The analysis of document layouts is an important step for many areas especially for the OCR system where its input is a printed or handwritten text document without any graphic elements, hence if any document has a non-text element, the system will not give the perfect solution as expected. For this, DLA is a necessary step before OCR, which is used for extracting and recognizing all the existing elements on a page, either as text or non-text elements and specifying each element according to

its features. Therefore extracting information from a document means the extraction of all graphical-textual elements that exist on a document page (see Figure 1.2).

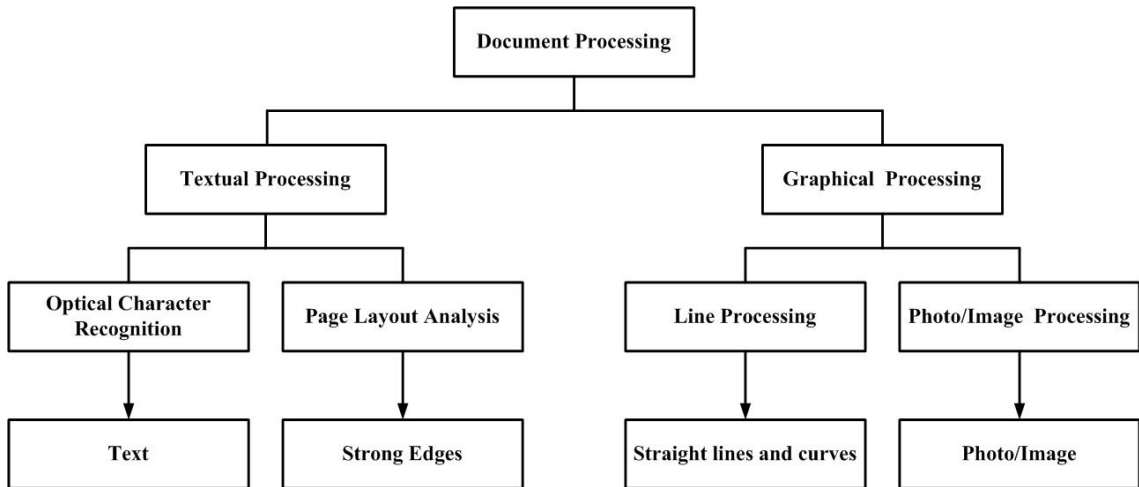


Figure 1.2: A hierarchy of document processing [90, page 2], [38].

These elements include many categories such as titles, comments, authors, and legends for the text extraction mode and pictures, tables and adverts for the non-text extraction mode. Each category has various types and different sizes as well because each document has a different form and structure (e.g. a newspaper page or a magazine page). Such a combination complicates the segmentation process for homogeneous regions, and it is made worse when the document has many imaging conditions (e.g., noise, uneven illumination, skew, perspective distortion and motion blur).

Examples of DLA by highlighting many distinct categories of page components are shown in Figure 1.3. Several algorithms and methods have recently been developed for DLA for English and other languages, but with documents in Arabic, many challenges still have to be overcome.

NLP is the basic phase in most language searches of the world, where the programs are generated in such a way that they can readily comprehend and manipulate human language text. Segmenting Arabic sentences is a crucial step for Arabic recognition as it is used in many natural language processing technologies such as parsing, machine translation, and research. Unlike English texts, in Arabic texts, there is no explicit space between all the words. For example, (العمل وأهميته) means: Work and its importance) is composed of three words (العمل، و ، أهميته)، in another sentence (العمل واجبنا) means: work is our duty) is composed of just two words, not three (العمل، واجبنا) as we can see the same structure, but the result is different because of the non-uniform spacing of words.

Many researchers have concentrated on the segmentation phase using various solutions and techniques such as line and page segmentation. The methods used produce good results in the case of English and Latin texts as bottom-up methods based on



Figure 1.3: Examples of documents analysis. Each type of layout is highlighted in a different color.

connected components [48], structural features [101], or both of them [95], but they fail in some text cases due to the type of structure, overlapping words, diverse writing styles, punctuation marks, dots, and diacritics. In handwritten texts, Arabic text writing has many features that make handwriting quite hard to process, where we have unique shapes at the letter level, and it may sometimes be different from time to time even it is made by the same person. For this, no hard-and-fast rule for segmentation can be applied.

All these challenges explain why many methods proposed in the last years focus on a particular form of document types or on a specific problems, which why they are difficult to generate to other structures and collections.

The main objective of this study is to create a new printed Arabic database, develop new methods and techniques for Arabic document layout analysis and Arabic handwritten text segmentation by improving the results of these tasks and making it more general than before. For this, we present several approaches based on RLSA, connected components, projection profiles and machine learning, which seek to overcome the main problems encountered here.

1.3 Methodological Approach and Innovative Aspects

The current state-of-the-art of Arabic document analysis methods [55, 93, 111] is mainly based on the extraction of text elements without graphical elements by choosing simple structure document pages as the dataset, where most of the given paper elements are in text format. Figure 1.5 shows examples of the dataset used for most of the Arabic document analysis studies. In addition, there is no current study that describes how to extract the logical structure from degraded Arabic images that were captured by a smartphone-camera (see Chapter 3).



Figure 1.4: Examples of Arabic document used for related works.

The proposed methods in this thesis can be divided into two sections where the first section focuses on Arabic document analysis and logical structure extraction under perfect and poor conditions. The second section is based on handwritten Arabic line segmentation where we present an improved extraction method specially designed for Arabic texts.

The main innovative aspects of this thesis are:

- The creation of a printed Arabic database, which may be regarded as the first Arabic database, containing (scanned/ smartphone-captured/ computer-vision) types that have been selected from the Arabic newspaper/magazine pages.
- The extraction of titles and their subtitles without the need to process the elements of the entire page based on geometric features, RLSA, and connected components.
- The extraction of physical and logical layouts from Arabic newspaper pages based on RLSA, projections profile analysis, and connected components labeling for physical structure extraction, certain rules of sizes and positions of the physical elements extracted earlier, and also based on a priori knowledge of specific properties of logical entities (titles, figures, authors, captions, etc.) for logical structure extraction, stored in an XML/DTD file.

- Applying a form analysis which is robust against degraded images and using deep learning (convolutional neural network) instead of an analysis with a fixed framework. The proposed method can be used in various structure formats with different types of text (font, size, and shape) and various figure shapes.
- Applying an efficient method for freestyle handwritten line segmentation where the main objective is to be able to process any Arabic handwritten text regardless of its writing style based on Kohonen's self-organizing neural network.

1.4 Structure of the Thesis

The remainder of the thesis is organized as follows:

In Chapter 2, we review state-of-the-art methods for binarization, smoothing, noise reduction and neural networks, followed by an outline and comparison of the latest related studies on segmentation and recognition document structures. Additionally, state-of-the-art of Arabic handwritten text line segmentation is presented along with related articles.

In Chapter 3, we present a review of the Arabic language, where in the first part we present the Arabic documents and their writing specifications and characteristics. Then our printed Arabic database is elaborated on, followed by a description of the ground truth file.

In Chapter 4, we present a set of methods specially developed for the task of Arabic document layout analysis. We outline several approaches for extracting textual and non-textual information from Arabic newspaper/magazine pages, and we propose two approaches based on a hybrid approach (bottom-up and top-down) and conventional neural network for this task. Next, we propose another approach that is based on RLSA, connected components and projection profiles for title and subtitle detection in a complex structure.

In Chapter 5, we present an advanced method based on a smoothing technique and Kohonen map learning for the task of Arabic handwritten line segmentation.

1.5 Published Articles Related to the Dissertation

Most of the ideas, tables and figures have appeared in five publications that covered these chapters:

- (1) Arabic Document Database (Chapter 3): A conference paper on printed Arabic text database for automatic recognition systems;
- (2) Printed Arabic newspaper (Section 4.1): A conference paper on recognition of the logical structure of Arabic newspaper pages;
- (3) Title Detection in Printed Arabic newspaper (Section 4.2): A Journal paper on title segmentation in Arabic document pages;
- (4) Smartphone-captured Arabic newspaper analysis (Section 4.3): A conference paper on a convolutional neural network for Arabic document analysis;
- (5) Arabic Handwritten Word Detection (Chapter 5): A conference paper on a self-organizing feature map for Arabic word extraction.

Chapter 2

State of the Art

Here, we focus on the well-known techniques for Arabic document layout analysis and Arabic handwritten text line segmentation. We describe binarization in the first section, where global methods are defined in Section 2.1.1 and adaptive methods in Section 2.1.2, followed by smearing and filtering techniques in Section 2.2. In addition to the preprocessing phase, a survey of neural networks is presented in Section 2.3. Unsupervised pre-trained networks, along with definitions of a self-organizing map and convolutional neural network model are given. Then we go through the main approaches for physical and logical structure recognition in Arabic documents in Section 2.4, presenting a comparison and summary of existing approaches. Then, in the last section, we review the existing suggestions for text line/word segmentation.

2.1 Binarization

For several decades, scanners were the most widely used tools for capturing a document image, hence numerous binarization methods were created for analyzing scanned documents like OCR [44, 115, 129]. Nowadays, cameras are widely available which can offer high-speed, flexible and non-contact document imaging, and they have become a nice alternative to the scanners. However, in contrast to scanners, the quality of camera-captured document images is lower because of perspective distortions, non-uniform shading, image blurring, character smearing (due to low resolution) and lighting variations. Attempts to tackle these problems has led to a binarization revolution [43, 69, 128, 137]. The principle concept of the binarization technique is to separate the pixels of the image using one or more thresholds into two classes, namely background pixels and foreground pixels.

In the following, we present state-of-the-art methods of global and adaptive binarization methods, and methods that use a combination of binarization methods.

2.1.1 Global Methods

In global binarization, the same single threshold is applied on every pixel using the gray value images as input, therefore the colored images have to be converted to gray tone conversion, and it can be transformed with the standard conversion:

$I(x; y) = 0.3R(x; y) + 0.59G(x; y) + 0.11B(x; y)$, where R, G and B are the Red, Green and Blue channels of the color image [68].

For an $m * n$ gray value image $I(x; y)$ with intensity values between 0 and 1 and a threshold $T(x; y)$, each image pixel is classified in the foreground (labeled as 1), and background (labeled as 0), resulting in the thresholded image $I_{th}(x; y)$, where $T(x; y) = T_g = \text{constant}$:

$$I_{th}(x, y) = \begin{cases} 1 & \text{if } I(x, y) > T(x, y) \\ 0 & \text{if } I(x, y) \leq T(x, y) \end{cases}$$

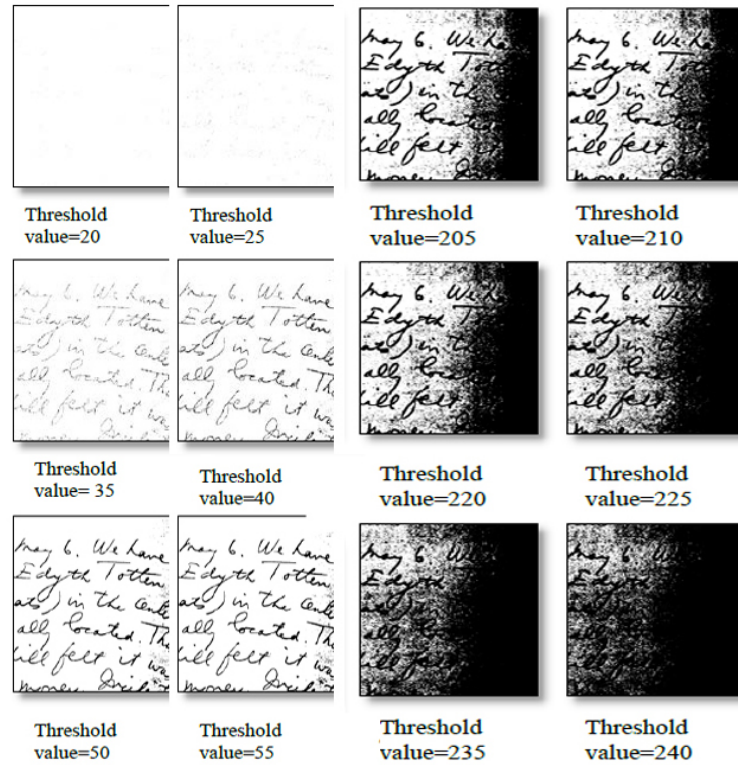


Figure 2.1: The results of many possible thresholds applied globally on the image [109].

Figure 2.1 shows the application of multiple manual thresholds on part of the document image. It can be seen that there is a range containing the values that may be regarded as the optimal solution while the other values produce poor results.

The most well-known method in the global threshold is the Otsu method, which involves an analysis of the distribution of the gray values. This method is based on

the calculation of the optimal threshold (see Figure 2.2) by maximizing the variance between pair groups of pixels of the local region defined by a structuring element. Many studies focused on global thresholding, where some of them were based on the probability of the classification error, entropic thresholding based on the gray-level spatial correlation histogram, and the most recent focused on Iterative Deep Learning for Otsu binarization enhancement [63, 125, 158].

Global methods can be used on documents that have uniform illumination with a stable background (scanned document).

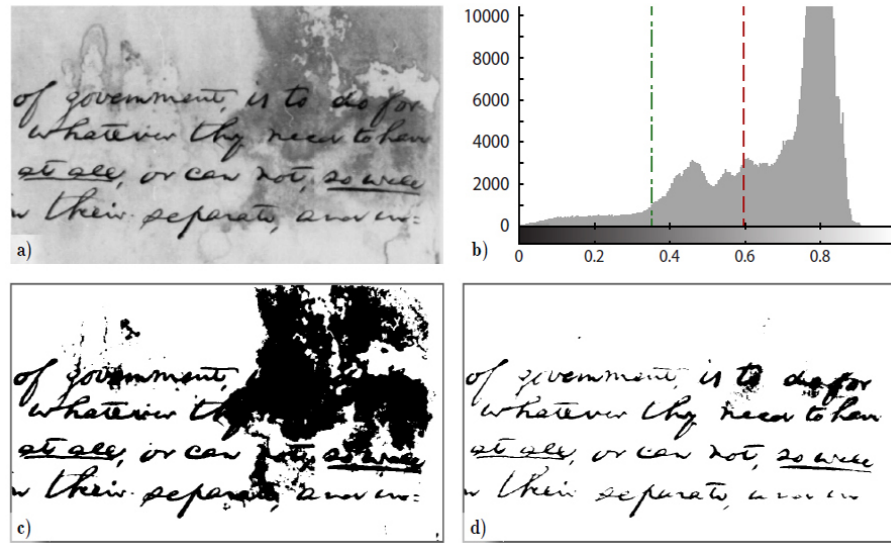


Figure 2.2: Example of a Otsu/manually binarization results: (a) The image in gray; (b) Histogram with the Otsu threshold (dashed) and manual threshold (dashed-dotted); (c) The Otsu threshold image; (d) A manually thresholded image [40].

In Figure 2.2, it can be seen that for uneven backgrounds the Otsu method will prove ineffective.

2.1.2 Adaptive Methods

Adaptive methods define local regions in which separate threshold values $T(x, y)$ are calculated. The latter combines dynamic thresholding with local windows across the image to determine local thresholds, which means that in each window, the threshold value is calculated individually for each pixel using some statistics (such as the mean and median) obtained from the region (local window). As a result, different thresholds are produced for different image regions.

Figure 2.3 shows that in contrast with global thresholding, adaptive thresholding works quite well under a variety of conditions like non-uniform illumination, and an uneven background.

Many studies have focused on adaptive thresholding, which is used for separating

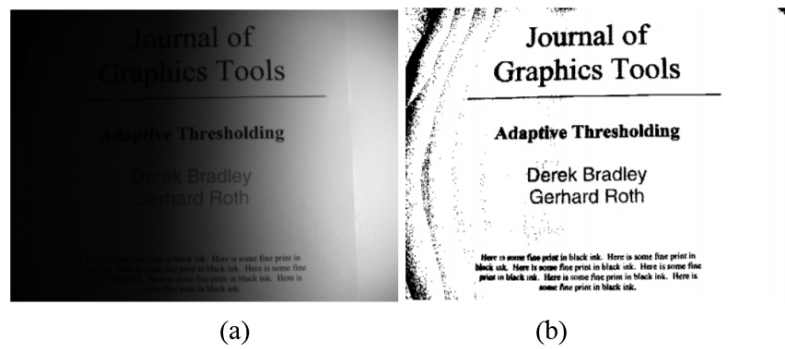


Figure 2.3: Degraded document image example: (a) Original image; (b) Binarization results produced by the adaptive method [32].

characters from the background, by estimating the background white level and subtracting it from the raw image [24, 98, 120]. Many techniques [19, 22] carry out a rectangular division of the gray value image depending on the character size. In another study [31], they adapted the Sauvola method, where the contrast and the mean gray levels of the image are normalized. In Dorini and Leite’s paper [86], a pixel is marked as background if the eroded value is closer to the actual pixel value and as foreground otherwise.

2.1.3 Methods Based on the Combination of Different Binarizations

The principle of these methods is to apply different thresholding methods to the same image then select the best result [58], or a feature vector is created and classified, e.g. a combination of global thresholding (Otsu’s method) and local thresholding (Sauvola method). As we see in Figure 2.4, the combination binarization method gave the best results for both example document images compared to those of Otsu and Sauvola’s methods. Below, thresholding methods, which are based on a combination of different binarization techniques, are described.

In [21] the algorithm divides the document image pixels into three sets, namely foreground pixels, background pixels and uncertain pixels. A classifier is then applied using the pre-selected foreground and background sets. In [58], they used different binarization methods with different parameters for each book. Therefore, a subset of each book is classified into one of 4 noise classes: bleed-through, the high similarity between background and foreground, variable background and all other images.

2.2 Smoothing and Noise Reduction

After the binarization step smoothing and noise reduction is performed, on the data to remove defects, reduce damage, and improve the quality of the noisy image. This is usually applied for visualization purposes and/or to prepare the ground for

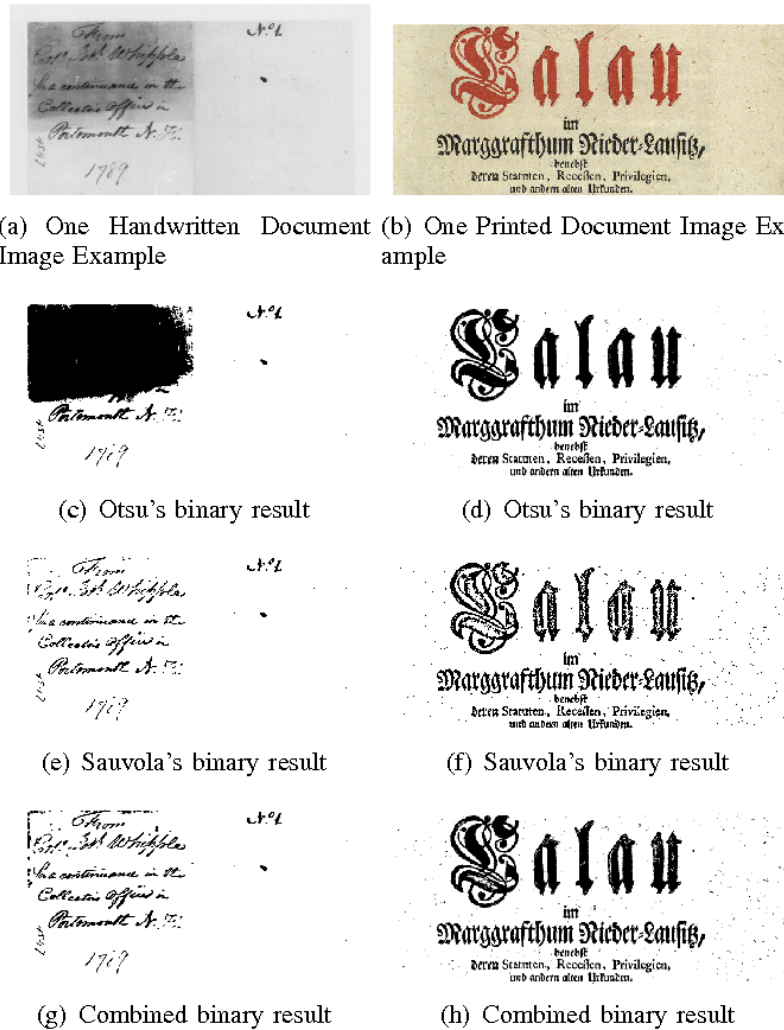


Figure 2.4: Example of a combination of binarization results: (a)(b) Two degraded document image examples; (c)(d) Binarization results produced by Otsu's method; (e)(f) Binarization results produced by Sauvola's method; (g)(h) Combination-binarization method results. [21]

further processing. The result is a clear image without any noise, which would be the best possible result for subsequent treatment of the chain like reducing the size of the image file, reducing the time required for subsequent processing and storage, and removing extraneous features that would otherwise cause subsequent errors in recognition. We note that here the process of reducing is called “filtering”.

In Figure 2.5, the isolated black pixels that exist along with the letter “e” represents the noise that should be removed to make the image more clearly.

2.2.1 Filtering Techniques

A Gaussian filter is the most common linear filter used to remove certain types of noise. It is based on the 2-D distribution as a point-spread function, and it is

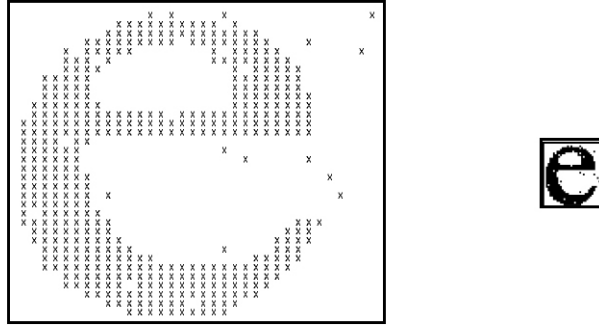


Figure 2.5: Example of the letter “e” with salt-and-pepper noise [1, page 19].

achieved by convolution. This convolution brings the value of each pixel into closer harmony with the values of its neighbors. which is given by the following rule;

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}},$$

where σ is the standard deviation of the distribution, and x,y are the image coordinates.

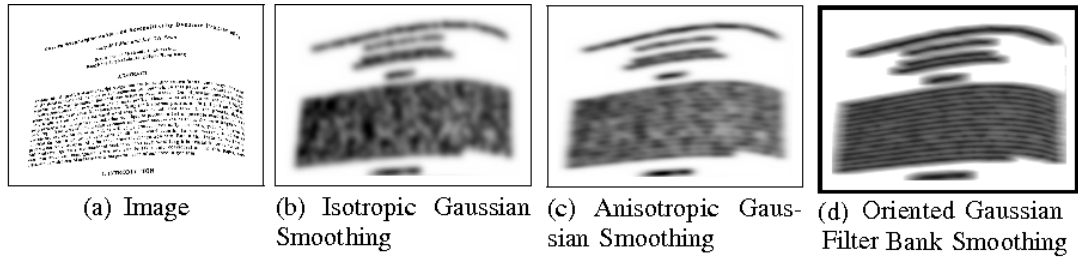


Figure 2.6: Different possible ways of applying Gaussian filtering to a sample camera-captured warped document image [140].

As shown in Figure 2.6, the result of this method is a blurred image in which the sharp edges are removed along with very thin lines.

As regards filters, the median filter is viewed as the most efficient non-linear filter for removing noise, which can be divided into many types such as weighted median, rank conditioned rank selection, and the relaxed median. More precisely, the median filter replaces a pixel by the median of the neighborhood pixels, instead of the average.

$$y[m,n] = \text{median}\{x[i,j], (i,j) \in \omega\}$$

where ω represents a neighborhood defined by the user, centered around the (m,n) location in the local window, and this window may be a round disc, a square, or a rectangle. The pixel at the center will be replaced by the median of all the pixel values inside the window.

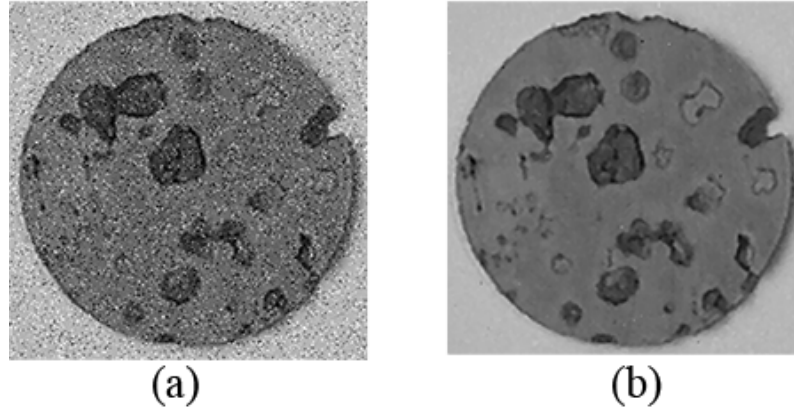


Figure 2.7: Example of Median filter result: (a) An example of a degraded image; (b) A filtering result produced by the Median method. [21]

As shown in Figure 2.7, all of the small black/white pixels have been transformed into colors similar to the pixel neighbors.

Morphological operations [71, 119] have many objectives, noise reduction being the most common objective that can be exploited from these techniques (see Figure 2.8). The basic morphological operations are erosion and dilation.

Erosion is the reduction of the region size, in which the level of this reduction is achieved by the interaction of a set called a structuring element with a set of pixels of interest in the image, where the structuring element has both a shape and an origin. Dilation is the opposite process, where the size of the region will be bigger, which means that new black pixels will be produced as boundaries of each black pixel region.

Let A be a set of pixels and let B be a structuring element [36]. Let $(B)_s$ be the reflection of B about its origin, followed by a shift s .

Erosion is represented by the $A \ominus B$ formula, where the dilation is represented by the formula $A \oplus B$:

$$A \oplus B = \{s \mid ((\hat{B})_s \cap A) \subseteq A\}$$

$$A \ominus B = \{s \mid (B)_s \subseteq A\}$$

These basic operations are usually combined and applied iteratively to erode and dilate many layers. The overall result may be an opening or closing operation.

Opening is represented by $A \circ B$ formula, and closing is represented by the formula $A \bullet B$:

$$A \circ B = (A \ominus B) \oplus B$$

$$A \bullet B = \cup \{(B)_z \mid (B)_z \subseteq A\}$$

Opening leads to boundaries being smoothed, narrow isthmuses broken, and

small noise regions are eliminated.

$$A \bullet B = (A \oplus B) \ominus B$$

Closing leads to boundaries being smoothed, narrow gaps joined, and small noise holes are filled.

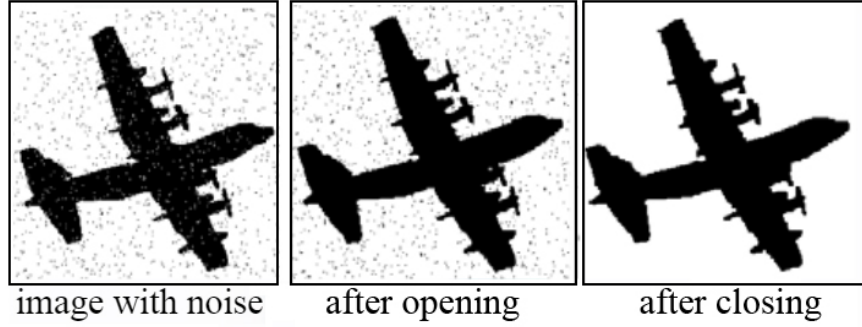


Figure 2.8: Example of some morphological operation results [133].

Many studies focused on this topic, and each of them had different goals. In [157], the researchers employed morphological filter recursivity, where the standard morphological operation is applied directly to the original input image. In [131], the author presented an efficient data reuse architecture to perform grayscale morphological operations, using a feedback loop path and a decoder/encoder pair comparator. In [28], they used a 2-D systolic architecture for gray-scale morphology operations, and this architecture can be used for non-rectangular flat structuring elements to get high-resolution images. In [141], they used a Partial-Result-Reuse architecture for mathematical morphological operations with flat structuring elements, where the partial results are kept and reused in subsequent operations to reduce hardware costs.

2.2.2 Smearing Techniques

The objective of smearing techniques is to create homogeneous regions (black and white blocks) which are applied on a binary sequence, where these regions may be text or non-text (only one type of data e.g text, graphic or image), these techniques being created and developed for a document analysis system. According to [134], there are three main types of smearing techniques. These are:

Run Length Smoothing Algorithm (RLSA)

This technique [84] transforms a binary sequence x into an output sequence y (homogenous region creation) according to the following rule:

The pixel value P (white pixels) in x is transformed into a black pixel if the number

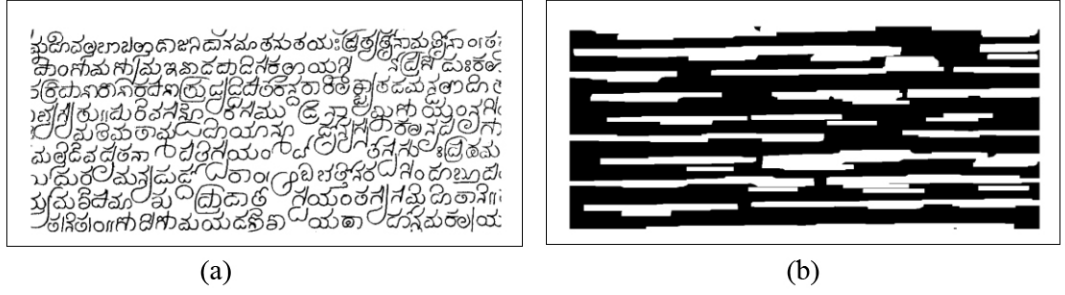


Figure 2.9: Example of a smearing operation result:(a) Document image example; (b) A smearing result using the run-length smoothing algorithm [134].

of adjacent white pixels is less than or equal to a predefined limit V , where the black pixels in x are stable in the output y (see Figure 2.9).

The RLSA is applied row-by-row as well as column-by-column to a document, producing two distinct bitmaps, where the result may be horizontal smearing/vertical smearing, or it may be combined by a logical AND operation between the two bitmaps. This technique is one of the most common algorithms used in page layout analysis and segmentation techniques.

Binary Transition Count Map

This technique [142] was developed to automatically read documents that have complex layout structures which include graphics. It is applied to a binary image using a sliding window for each pixel (0 or 1), and the output image is created by counting the shifts that exist in the window, where each pixel location is stored with the count value (see Figure 2.10).

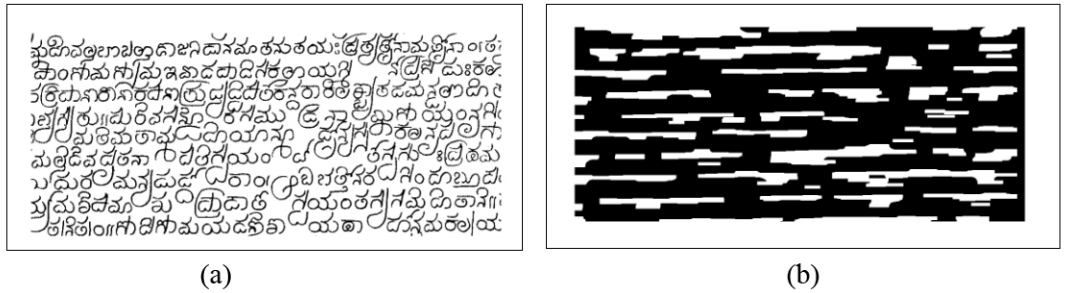


Figure 2.10: Example of a smearing operation result:(a) Document image example; (b) A smearing result using the binary transition count map [134].

Adaptive Local Connectivity Map

This technique [161] is based on connectivity features similar to local projection profiles, which can be directly extracted from grayscale images, using a sliding window, where each window cumulative sum is calculated around the pixel and the sum is

placed in the coordinates of the positioned pixel. It is used for retrieving text lines from handwritten historical documents (see Figure 2.11).

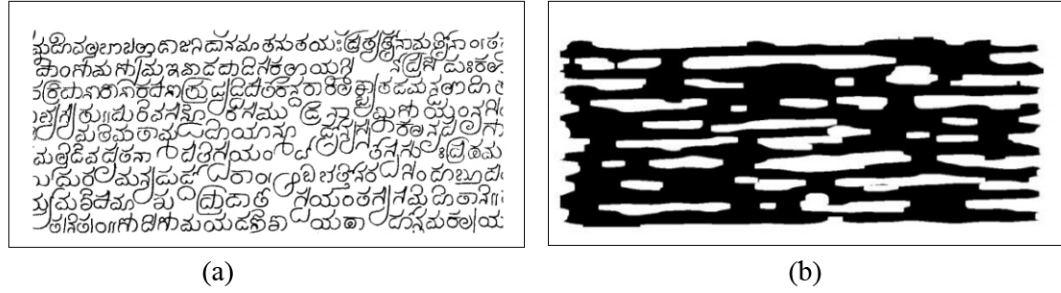


Figure 2.11: Example of smearing operation result:(a) Document image example; (b) A smearing result using the adaptive local connectivity map [134].

2.3 Neural Networks and Deep Learning

Machine learning research [148] seeks to adapt new data independently through iterations by learning from previous computations and transactions to provide positive and accurate results. It commences with inputting training data into the selected algorithm, in which the type of data affects the algorithm process. For algorithm testing, new data is fed into the Machine Learning algorithm, and both the prediction and results are compared. If the prediction is not as expected, the algorithm is re-trained multiple times until the optimal output is obtained. Machine Learning has two main areas, namely supervised learning and unsupervised learning, where each one has a particular purpose. It is widely used in clustering and classification tasks such as natural language processing, speech recognition, computer vision and reinforcement learning.

In the following, we selected some of the machine learning types that are commonly used for document layout analysis and natural language processing.

2.3.1 Self-Organizing Map

The self-organizing map (SOM) [145], also known as the Kohonen map, or self-organizing feature map, is an artificial neural network using unsupervised learning to cluster data samples. The map is usually two-dimensional, each map including a model vector in which learning is iteratively performed by processing one input sample at a time, where the best matching unit (BMU) is adapted to the input vector, and the model vectors are adapted to the input sample.

The training is often done in two phases:

- The learning effect is set high and the neighborhood size is kept large.
- The learning is conducted with fewer learning effects and fewer neighborhoods.

The steps of the SOM algorithm [144] are as follows:

1. Set $v = 0$ and initialise the map nodes $m_i(0)$ with small random values.
2. Find the closest map node $m_c(v)$ for input $x(v)$ based on the Euclidean distance.

$$\|x(v) - m_c(v)\| = \min_i \{\|x(v) - m_i(v)\|\}$$

3. Update the map nodes where, $h_{ci}(v)$ is the neighborhood kernel.

$$m_i(v + 1) = m_i(v) + h_{ci}(v)[x(v) - m_i(v)]$$

4. Set $v = v + 1$ and return to step 2 if $v \leq z$, where z_c and z_i are the position vectors of nodes c and i on the map.

$$h_{ci}(v) = h(\|z_c - z_i\|, v)$$

$$h_{ci}(v) = \begin{cases} \alpha(v), & \text{if } i \in N_c(t) \\ 0, & \text{if } i \notin N_c(t) \end{cases}$$

In the following an example of SOM algorithm is given for classification purposes:

- SOM creation using a training data set.
- BMU computation using the Euclidean distance.
- Mapping each training set sample to the map.
- Using training samples of mapped classes to find a class label for each node (the majority class determines the class of the node).
- Using BMU for mapping test set samples to the labeled map.

There are plenty of studies available that rely on SOM clustering, in which document text is included. In [143], they used SOM for Chinese word clusterization, while in [80] they used this type of clustering for word-sense discovery and demystification. Also, some experiments used SOMs for text retrieval [18, 64], as well as for text classification [23, 102]; and it is also used for handwritten text and character segmentation purposes [9, 108, 121].

2.3.2 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is an artificial neural network (ANN) with multi-layers (deep learning algorithm), where its input is given as an image pixels and its goal is to be able to discriminate the aspects/objects that form image. The CNN architecture includes many filters, which are set via training on the dataset, starting with random initialized values for the filter; then these values will be refreshed within

gradient backpropagation. CNNs are composed of three types of layers, namely convolutional layers, pooling layers, and fully-connected layers [81] (see Figure 2.12).

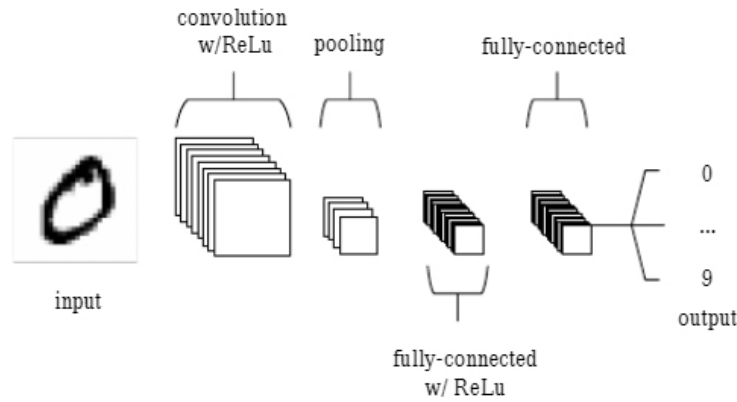


Figure 2.12: An simple CNN architecture, represented by five layers [81].

- The convolutional layers: They determine the output of neurons which are connected to local regions of the input via the calculation of the scalar product between their weights and the region connected to the input volume. The ReLu (Rectified linear unit) applies an activation function such as a sigmoid function to the output of the activation produced by the previous layer.
- The pooling layers: They perform downsampling along with the spatial dimensionality of the given input, to further reduce the number of parameters within that activation.
- The fully-connected layers: They perform the same tasks found in standard ANNs and attempt to produce class scores from the activations, which will be used for classification.

CNNs are used in a variety of research domains, in which document layout analysis is treated as one of the areas that has made great improvements in the performance through the use of CNN algorithms. In [75], they used CNN for handwritten historical document image segmentation using a single one convolution layer. Similar results are presented in [30] for old document segmentation using a fully connected network. In [79], they proposed a novel CNN-based method to accurately localize documents in real-time using the model localization problem as a key point detection problem. A fully convolutional network is presented in [152] for extracting semantic structures from document images by treating document semantic structure extraction as a pixel-wise segmentation task and they proposed a unified model for classifying pixels.

2.4 Document Layout Analysis

Because of the variety content of documents, new ways were developed for exploiting them. DLA is the first step that leads to the extraction of the physical and logical structure of the document, which is a key step in any application of OCR, document archiving systems, document storage, and retrieval systems.

The purpose of the DAR (Document Analysis and Recognition) procedure is to extract the important labels from the document image such as text, author, figure, table, title, text-line, for automatic document classification. To achieve this goal, we first outline all the related studies, and present the different techniques of document layout analysis. DLA guided by structural rules, to find two types of structures, namely the physical and logical labeling. Below we introduce the topic of Arabic document structure analysis, where the well-known methods for the analysis and recognition of document structures are described.

2.4.1 Physical Structure Analysis

Each document includes different labels such as text, lines, words and tables which represent the presence of the document (see Figure 2.13).

The physical structure of the document describes the layout of the document and



Figure 2.13: The physical structure of a newspaper page [53, 103].

the different text boxes and their order relative to each other, as well as all their text properties like size and font.

The classical extracting physical structure methods are often divided into three major classes of approaches, namely a bottom-up approach, a top-down approach and a hybrid approach and other techniques.

Bottom-up Approach

The objective of ascending methods is to extract the text/ non-text elements such as text and figure from the analyzed image; using for example, the connected components where the merge operation is applied until the document page is completely-assembled/ partially-assembled.

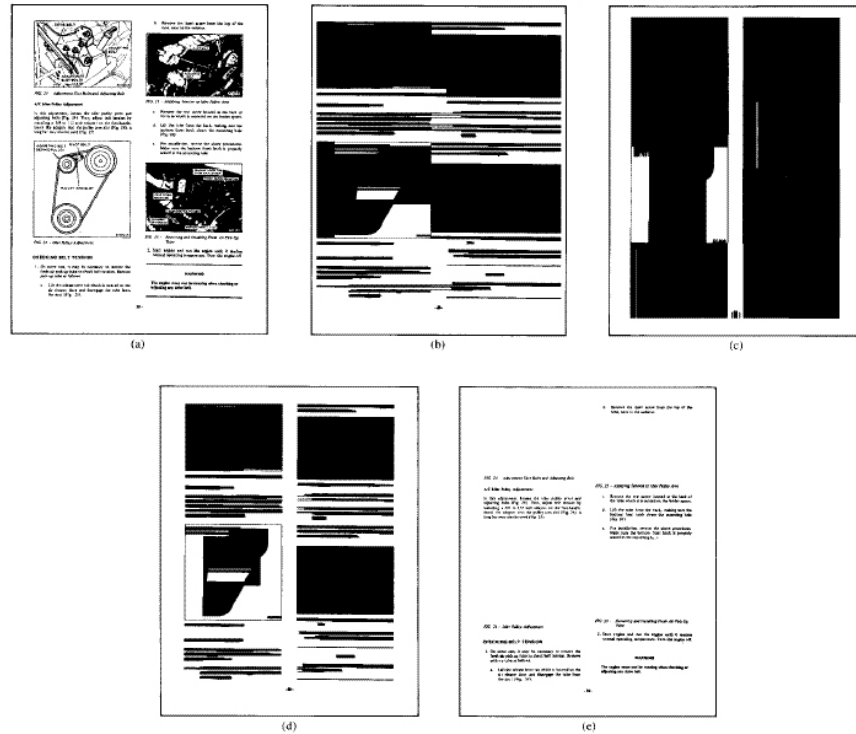


Figure 2.14: An example of the RLSA algorithm: (a) The original Image; (b) Horizontal Smoothing; (c) Vertical Smoothing; (d) Final result of RLSA; (e) Results for blocks treated as text data [84].

This approach naturally has problems, the independent primitives extraction can not always be performed, hence the system will make incorrect choices [42]. It requires a priori knowledge of the style used and appearance along with very high precision in the resolution of images.

An example of an algorithm that use the bottom-up strategy is the well-known RLSA smoothing technique. The RLSA algorithm [84, 136] allows the grouping of neighboring black pixels into regions by performing a horizontal, vertical smoothing of the image or by combining the two images obtained after horizontal and vertical smoothing. For this algorithm, a threshold values must be set beforehand, and this can lead to over-segmentation or sub-segmentation if a bad value is chosen (see Figure 2.14).

Top-down Approach

The methods used here are based on the cutting of the image into multiple areas by examining particular features of the nature of the processed document using a priori knowledge, and it attempts to divide the entities of the document iteratively to check the assumptions. However, it needs a prior knowledge of the document structure for good performance.

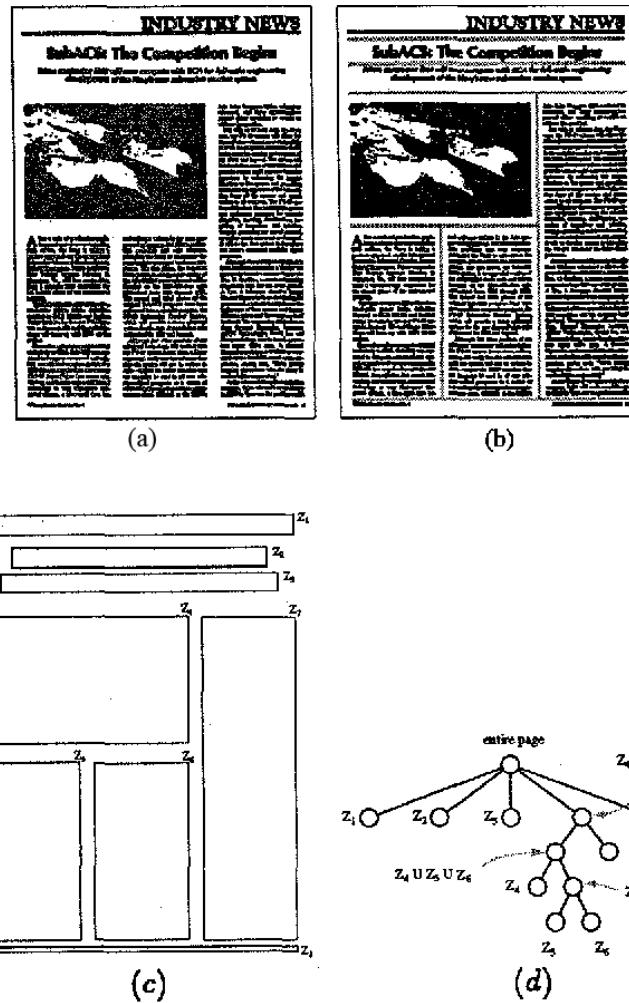


Figure 2.15: An example of an X-Y-Cuts algorithm: (a) Document image; (b) Placement of cuts; (c) Zones subdivided; (d) The X-Y tree of the page layout structure [67].

The most widely used method here is the X-Y-Cuts algorithm [50]. It can be adapted quite well to several document structures which are mostly composed of horizontal lines of text organized in paragraphs, and graphics for well-separated forms of the text. The algorithm recursively decomposes the image of the document into a sub-rectangle, where the division is done recursively on the densest areas of space. It can be represented by an X-Y tree, where the root corresponds to the whole

page and the leaves represent page blocks and each level represents the tree [78] (see Figure 2.15).

Another well-known algorithm is the projection profile analysis [66] which allows the separation of text blocks and the detection of lines. This is done to project the values of the black pixels or the thickness of the circumscribed rectangle of the characters, in horizontal and vertical directions to get two histograms, and these histograms are used in the division task. This method does not give a good performance on complex structures, and the figures must be correctly binarized to separate the lines correctly.



Figure 2.16: An example of horizontal and vertical projection profiles of a document image [100].

Hybrid Approach and Other Techniques

The hybrid approach combines bottom-up analysis to extract global primitives and top-down analysis to search for local primitives. Methods using both local and global primitives are now new avenues for research [26, 132].

Other types of methods combine different techniques for distinguishing the document elements such as text/non-text separation. Artificial neural networks have been extensively utilized for this purpose [37], where this technique relies on classification methods different from the previous methods (see Section 2.3). Binary classification trees are also used for extracting region classification data in various categories [118], while other techniques are based on morphological operations, such as contour analysis and the morphological distribution of text lines [94]. Many other studies

used text classification features, where the texture descriptors are one of the most commonly used features to discriminate between the set of document elements [15, 160].

Physical Structure Representation

Physical structures can be described by a tree or XML to transcribe the visible hierarchical links that exist among the labels (see Figure 2.17).

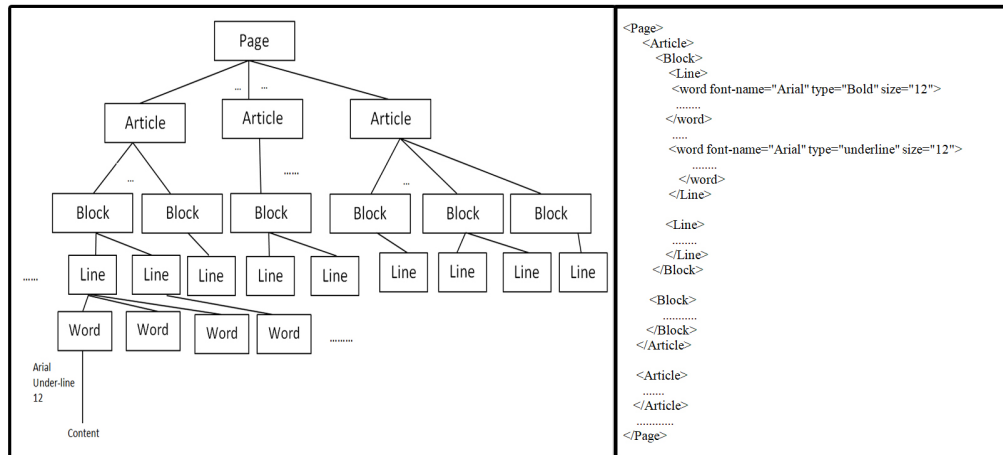


Figure 2.17: An example of a physical structure representation [91].

2.4.2 Logical Structure Analysis

The logical structure of a document is a designation of the semantic content of the document, and thus the correspondence among the physical regions and their function. The logical labeling consists of attributing to the different physical regions, that exist in the document such as a paragraph, a header, an abstract, a footer or proverb. This labeling is the goal of the DLA system (see Figure 2.18).

This recognition process consists of two steps, namely block labeling and physical structure transformation, where the first step may be before, after or at the same time as the second step [91].

- Block labeling: assigning logical tags to the physical blocks extracted previously, where it is done by the extraction of the features and the classification.
- The physical structure transformation: merging physical blocks belonging to the same logical entity and determining a reading order among the logical entities.

According to [62], several methods have been proposed for the extraction of logical structures, which are grouped into four main types:



Figure 2.18: The logical structure of a newspaper page [53, 103].

- The structural method: This method is applied directly to data representation structures using transforming tools or grammar inference [62] for going from a physical structure to a logical structure using various techniques, which is represented in the form of a tree or graph.
- Artificial intelligence methods: These rely on the construction of rules from different pieces of information extracted at the physical level to find the logical structure, where heuristics and a Document Architecture Language (DAL) description language were used for rule building. The possible relationships among the logical components cannot be represented in simple terms.
- Probabilistic and learning-based methods: These represent the elements that have been generated by a set of probability distributions. The goal is to adapt the data, using probabilities, due to the lack of regularity. Several probabilistic techniques have been applied like Bayesian networks, generalized n-grammars, probabilistic grammatical analysis, and Hidden Markov Models. This type of method is suitable for documents with complex structures.
- The multi-criteria classification method: This approach is more or less based on complex classifiers that adapt the proposed system to conventional forms of recognition tools. [62].

Logical Structure Representation

The logical structure of a document describes its semantic content, and it can be represented by a tree (like the physical structure) and encoded in XML (see Figure 2.19).

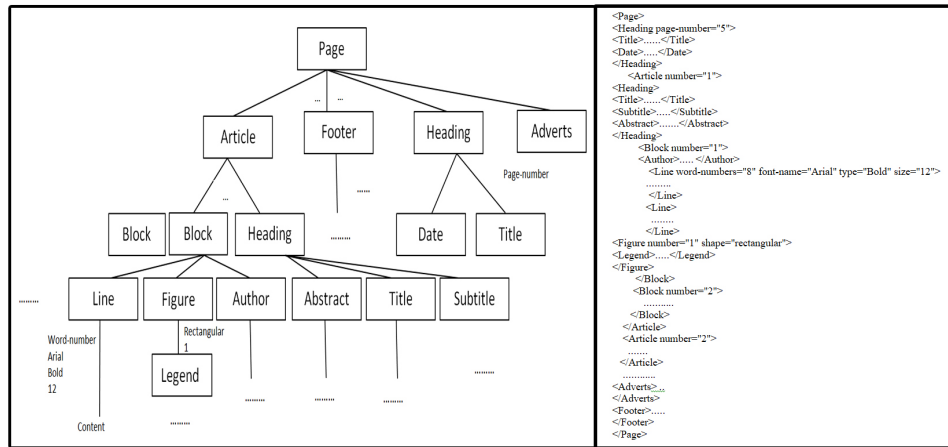


Figure 2.19: An example of the logical structure representation [91].

As can be seen in the figure, the tree structure contains more detailed information than the physical structure, which is represented at the text level. Here, the structure provides semantic information about all the existing elements in the document such as the title, date, adverts, footer, heading, author, table and legend.

2.5 Text Line/Word Segmentation

To obtain the optimal solution for the text line/word segmentation phase from different perspectives, several methods have been devised and developed, and each of them has its own special characteristics.

2.5.1 Text Line Segmentation

The text Line Segmentation methods are divided into five categories, which cover the printed and handwritten text, the projection-based approach, the grouping approach, Hough-based methods, morphology-based methods, and other approaches. These are presented below.

- The projection-based approach: This type of method is based on projection profile analysis using both the horizontal and vertical projections. From the horizontal profile, the black pixel values of each line can be extracted, and in the vertical profile, line-gaps can be determined. Many studies used this method for printed and handwritten text segmentation in a variety of languages [12, 106].
- The grouping approach: This method is based on the grouping technique, where the grouping can be done for connected components and pixels. Some studies based on this approach use geometric connections between the components, like distances and angles. [89, 96, 130].

- Hough-based: With this method [88] the curved line problem can be handled, where the skew orientation is obtained using the Hough transform to the gravity center of each CC that exists on the document. The essential points will be extracted then the lines set will be calculated according to the matching score with points sets [49, 122, 156].
- Morphology-based: The most popular approaches in morphology-based methods are the smearing-based approaches, and this model produces excellent results even in the case of skewed and curved lines. With this technique, the morphological operation, the RLSA, and the ARLSA are used, where the CC bounding-box in the smeared image is treated as text lines [110, 117, 153].
- Other approaches: Other types of methods include: The graph-based approach [60], this method being based on graph property estimation. It asserts that the word-gaps in a text line are less than the distance between two adjacent text lines [83]. The CTM (Cut Text Minimization) Approach [29] is based on finding a path in text line-gaps to be separated, which decreases the text line pixels cut by the segmentation line. Active contours (snakes) [99] is another method that has provided good results compared to other techniques, while statistical approaches are less commonly used for this task [154].

2.5.2 Word Segmentation

Most of the word segmentation studies are based on an analysis of geometric relationship of neighboring components, where these components may be CCs or OCs (connected components or overlapped components).

The related works can be divided into two parts:

- The calculation of the adjacent components distance;
 - The classification of CC-gaps into word gaps or within word-gaps.
- Distance metrics: In [51], they presented eight different distance metrics, namely the bounding-box distance, the minimum and average run-length distance, the Euclidean distance and different combinations of them which depend on several heuristics. An improvement method was presented in [149] which added a new distance metric (convex hull-based metric). In [65], they used a component-based method for gap classification using the baseline.
 - Trees: In [147], they extended classical word extraction techniques by incorporating a tree structure exploiting the gap context data instead of the gap classification threshold.
 - Neural networks: A neural network was presented in [47] to define the segmentation points, after a calculation of a feature vector for every possible segmentation point. In [25] the classification was performed by an improvement

architecture of the neural network, where the feature vector contained eleven features. In [151], they calculated an SVM-based gap metric for adjacent CCs within each text line, along with the threshold value for CC-gap classification. In [3], they based it on the SOM algorithm, where the CC-lengths are clustered to separate the groups of letters/ subwords/ words, while the CC-gaps are clustered to determine the boundaries of each word.

Chapter 3

Arabic Document Database

In this chapter we present an Arabic database which was created for the task of Arabic DLA and other things. First, we survey the printed Arabic text databases available in the literature. Second, we describe some properties of Arabic script and a variety of fonts and types of Arabic printed script. Third, we give some details of our database and provide a statistical description of aspects relating to the database. Fourth, we describe the ground truth file. Finally, we give a brief summary of our work.

3.1 Introduction

Documents have again come to the fore over the past few years and transforming them into digital images has raised the interest of computer science researchers. Various algorithms and systems have been created and developed in this area and much progress has been made in document recognition systems. For this reason, a systematic testing database in a recognition system needs to be created, by scanning then capturing a document image, which is easier now due to the evolution of cameras, mobile devices and the improving quality of cameras and smartphones.

In spite of this abundance, there are few Arabic databases with complex structure due to the lack of optical systems that can read the graphical Arabic papers. Hence the motivation for creating a new database with a new dataset for the quality assessment of camera-captured document images. The details of a printed Arabic text database for recognition research provide information needed for successful recognition. Here, an off-line Arabic recognition system needs to be designed by Arabic language researchers, and it can be used to validate samples by removing illumination problems, orientation problems, low-resolution, noise or checking the structure of a given page.

In the Arabic databases, there is no available database that includes printed Arabic text documents in newspaper/magazine format that can be downloaded or processed. Only a PDF newspaper exist that are freely available for commercial purposes. Here, we present an overview of the current Arabic printed databases con-

taining documents that were scanned or captured.

- The APTID / MF (Arabic Printed Text Image Database / Multi-Font) database [46] consists of 387 pages of Arabic printed documents scanned in grayscale format and 300 dpi resolution. These documents cover 1,845 text-blocks and provide a ground truth file for each text-block and they also contain an Arabic printed character image dataset with 27,402 samples. This database covers only one newspaper on different pages, and this coverage limitation is a disadvantage when developing a recognition application.
- The DARPA (Defense Advanced Research Projects Agency) Arabic corpus [116] contains 345 Arabic printed pages scanned with a 600 dpi resolution created from books, magazines, the newspapers and computer-generated documents in 4 fonts. However, the DARPA corpus is currently not freely available.
- The PATDB (Printed Arabic Text Database) database [13] was obtained from 6954 scanned pages of different forms of Arabic printed text (viz. books, chapters, advertisements, magazines, newspapers, and reports) scanned with 200, 300, and 600 dpi resolutions. Despite this, the PATDB database covers only a small percentage of a newspaper with 3.0% representing 210 pages. And, there is no smartphone-captured format.
- The APTI (Arabic Printed Text Image) database [45] contains 45,313,600 Arabic printed word images that cover approximately 250,000,000 characters taken from Arabic proper names, general names, country/town/village names, Arabic prepositions, using 113,284 Arabic printed word images with different fonts; 10 Arabic fonts, 10 font sizes and 4 font styles with a 72 dpi resolution using a computer program. However, this database does not include newspaper pages or any other document format pages.
- The ATID (Arabic Text Images Dataset) database [39] is organized into two groups. The first group represents the database of the printed documents with 16472 document pages captured from 116 different paper documents. The second includes handwritten documents with 9088 document pages captured from 64 different paper documents. In the two groups, 142 different images were captured per document (71 captures per phone). This paper describes the first and the only public offline images database up till now for both printed and handwritten Arabic mobile captured documents with a plain and simple background under varying capture conditions (blurry, different perspective angles and lighting conditions) using modern smartphones called Samsung Galaxy S6 edge and iPhone 6S plus. Unlike the existing database, there is a smartphone-captured format. However, the SmartATID database contains only one newspaper selected from the APTID/MF database [159]. Also, they did not cover the entire page nor part of page of the newspaper, they selected some articles where each picture covered one simple article.

- The ALTID (Arabic/Latin Text Images Database) database [57] contains 1845 Arabic text and 2328 Latin text images taken from 731 pages of Latin and Arabic printed documents in grayscale format and 300 dpi resolution. The handwritten dataset was created by 17 individuals of different ages and educational levels and the dataset includes 460 Arabic and 582 Latin text-blocks. This database does not contain any images captured using a smartphone camera.

3.2 Features of the Arabic Language

The Arabic language has unique features that distinguish it from most other languages, and this has had an impact on the development of document image analysis and recognition applications [1,159] (see Figure 3.1). These features make it difficult for automatic recognition, and they arise from its cursive nature and the variation of its character forms according to their position in the word [107] (see Figure 3.2).

Ending form	Middle form	Beginning form	Isolated form	Ending form	Middle form	Beginning form	Isolated form	Ending form	Middle form	Beginning form	Isolated form	Ending form	Middle form	Beginning form	Isolated form
ا	-	-	ا	ض	ض	ض	ض	د	-	-	د	ك	ك	ك	ك
ب	ب	ب	ب	ط	ط	ط	ط	ذ	-	-	ذ	ل	ل	ل	ل
ت	ت	ت	ت	ظ	ظ	ظ	ظ	ر	-	-	ر	م	م	م	م
ث	ث	ث	ث	ع	ع	ع	ع	ز	-	-	ز	ن	ن	ن	ن
ج	ج	ج	ج	غ	غ	غ	غ	س	س	س	س	ه	ه	ه	ه
ح	ح	ح	ح	ف	ف	ف	ف	ش	ش	ش	ش	و	-	-	و
خ	خ	خ	خ	ق	ق	ق	ق	ص	ص	ص	ص	ي	ي	ي	ي

Figure 3.1: An example of different shapes of an Arabic letter.

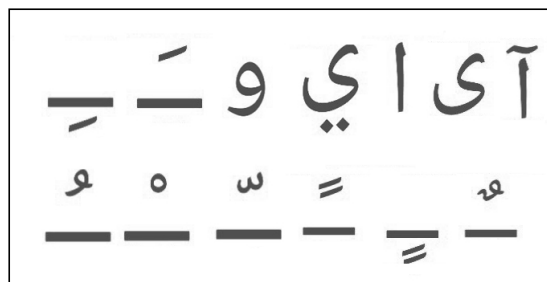


Figure 3.2: An example of Arabic vowels.

The Arabic characters change form not only according to their position in the word, but also according to the calligraphic style used. Calligraphy is highly developed in the Arab-Muslim world. In fact, Arabic writing may appear in different calligraphic styles such as Neskhi, Thoulthi, Diwani [127]. An analysis of Arabic documents reveals the existence of a diversity of writing fonts for the printed paper because of the variety and the abundance of Arabic calligraphy styles [46].

3.3 Documents Image Selection

We sought to create a database of Arabic newspapers that would include a wide range of structures, pictures, sizes, fonts and certain images of varying smartphone-captured quality.

In order to generate the Arabic newspaper printed text images dataset, we selected a set of document images taken from ten newspapers (Alriad, Alsharek, Alhadaf, Alnabar, Alshourouk, Alshorouk almisri, Alsharek al-Awsat, Aswaq Qatar, Alayam, Akhersaa), composed of 200 newspaper pages; 705 articles, where each article can appear on more than one image (see Table 3.1), and fourteen fonts written in three writing styles (normal, italic and bold).

Table 3.1: Statistics of the image number used for database production.

Newspaper	One article	Two articles	Three articles	More than Three
Alriad	41 images	15 images	6 images	13 images
Alsharek	36 images	8 images	5 images	14 images
Alhadaf	28 images	6 images	4 images	9 images
Alnabar	46 images	18 images	7 images	17 images
Alshourouk	74 images	21 images	10 images	15 images
Alshorouk almisri	46 images	12 images	8 images	13 images
Alsharek al-Awsat	35 images	11 images	7 images	15 images
Aswaq Qatar	36 images	13 images	6 images	14 images
Alayam	37 images	15 images	3 images	13 images
Akhersaa	79 images	26 images	10 images	19 images
Total	458 images	145 images	66 images	141 images

This set of document pages was printed with a HP laser printer, and the pages were scanned at 300 dpi resolution in grayscale format with a HP scanner. Using both the printed and screen versions, we created a database for a smartphone-captured Arabic text database. Afterwards, we used 810 documents to capture 2954 images with mobile phones for our text image database. The images were then stored in JPG format. Many pictures may appear on one page with different shapes, which may have a rectangular, circular or random shape in different positions to the left, right or in the middle of the article. Also, many of them include article-text or title-text. Figure 3.3 shows pictures containing six shapes.

The page images are divided into article-blocks with manual segmentation that may contain one article, several articles or all the articles on one page. Each page



Figure 3.3: Examples of image-shapes: (a1) A rectangular picture on the right; (a2) Two combined rectangular pictures in the centre; (b1) A circular picture on the left; (b2) A combined circular and random picture; (c1) A random shape of picture in the centre; (c2) Random shape overlapping some text.

has a unique structure due to the diversity of its contents, which appear in many article-shapes, image-shapes, title-sizes, title-fonts, text-sizes, and text-fonts.



Figure 3.4: Examples of image texts-blocks and title-blocks: (a) AL-Quds; (b) AXTManal; (c) Hacen Promoter; (d) Beirut; (e) AL Hadith; (f) Kacstone; (g) ALsharek Title; (h) Hacen Algeria; (i) Hacen Qatar; (j) Mariam; (k) MCS Topaz Brok out; (l) Hacen Extender X4 super fit; (m) Yakout; (n) Kufah.

The images of PATD generated using 14 different fonts are shown in Figure 3.4:

AL-Quds, AXTManal, Hacen Promoter, Beirut, AL Hadith, Kacstone, ALsharek Title, Hacen Algeria, Hacen Qatar, Mariam, MCS Topaz Brok out, Hacen Extender X4 super fit, Yakout and Kufah. These fonts were selected to cover various shapes of Arabic printed words ranging from simple fonts with few superpositions and ligatures (AL-Quds) to more complex fonts rich in superpositions, ligatures, and flourishes (MCS Topaz Brok out). Figure 3.4 shows a set of fonts with many sizes (6, 8, 10, 12, 14, 16, 18 and 24 points.).

3.3.1 Characteristics of Newspaper/Magazine Pages

The properties of the newspaper/magazine pages exploited in this database are the following (see Figure 3.5):



Figure 3.5: Sample of a journal page component.

- Advertising areas (may be absent).
- Several articles with different formats (long, short, main articles, ... etc.) where each article contains:

One or more titles with different formats.

A summary of the article below the titles (may be absent).

One or more figures, where each figure is accompanied or not by a legend, is overlapped or not by a text (it may be title/subtitle/simple text). The article may contain no figures.

An author's name. This name can be found either at the beginning or the end of the article.

One or more columns of text.

One or more bands (full rectangles) encompassing text (white), and they correspond to titles or secondary articles. These bands may be absent.

- Some items are surrounded by a black rectangle. Some others include straight horizontal lines serving as dividers between parts of the article.
- Horizontal or vertical straight lines used as dividers between the different articles can be found on the page.

3.3.2 Documents Data Capture

With different state-of-the-art smartphones, we created a new smartphone-captured database by capturing the document images for the PATD scanned database in a reproducible and controlled environment. As a result, most of the procedures were performed manually in realistic environments with different lighting conditions and various geometric distortions of the images.

To guarantee realistic results we used different lighting conditions affected by motion and perspective-angles distortions in various positions with rolling adjustable height and a rolling camera. We tested scene-related distortions under different lighting conditions by capturing images in a closed room with artificial light and outside with sunlight, blurred motion depending on the variation in the focal distance during the capture process.

The image capturing was carried out by hand using the smartphone via a Bluetooth to trigger the capture; this performance produces digitized data due to the manual variations of a paper document or human hand, lighting conditions and the location of the focal distance from the smartphone camera to the document page. For building and supporting the PATD, we used three modern smartphones, namely Samsung Galaxy S7 edge, Samsung Galaxy s3, and iPhone s6 (see Table 3.2), whose cameras are fitted with different sensor technologies, have various focal distances and they are able to capture images with different resolutions (13MP and 8MP).

Table 3.2: Statistics of the database for different smartphone makes with the number of images for each type of phone.

Smartphone	Screen version	Printed version	Total
iPhone 6	199 images	1427 images	1626 images
Samsung s7	/	928 images	928 images
Samsung s3	/	400 images	400 images
Total	199 images	2755 images	2954 images

3.3.3 Capture Parameters

As we mentioned previously, our Arabic database contains two datasets; namely, the first one generated with scanned Arabic printed text images, and the second by smartphone-captured Arabic printed and digital text images with different settings. The results were quite varied, ranging from low quality in dark conditions with some blur effects to high quality in good lighting conditions. We used three smartphones with a focus-select feature of the camera hardware to generate a series of images with focal blur depending on the variation in focal distance. The focus distance was decided randomly for each image. The location of capture was varied and the lighting conditions changed according to the time of day (morning, noon, evening). With the captured newspaper images, we used different parameter settings, which ensured a uniform acquisition for all the images of the database.

- Background: Without a background just the color of the newspaper page is present in the image.
- Smartphone setting: All the smartphones used for capturing had the flash deactivated in each case.

The smartphone's camera had different parameter settings, and this ensured different acquisition conditions when generating a variety of images for the database using three smartphones. The conditions are:

- Smartphone camera: 3 smartphones.
- Position of the smartphone camera: two positions.
 - Parallel with Y-axis of the document, longitudinal incidence angle.
 - Parallel with X-axis of the document, lateral incidence angle.
- Distance between the camera and the document: 10 cm, 20cm, 24 cm, and 30cm.
- Light level for the camera: three lighting conditions
- Focus blur: one value
- Motion blur: two types.

The images that were taken by the smartphone cameras were sorted into two types of distortions, namely single and multiple distortions. For a single distortion, we had different lighting conditions, out-of-focus blur and motion blur. The distorted images were captured at different positions, distances and times of the day.

Lighting Distortion

With lighting distortions, the pictures were taken for each document under three different lighting conditions given below.

- Light condition 1: Daylight.
- Light condition 2: Daylight + shadow of the object on part of the document.
- Light condition 3: Night + lamp (artificial light) indoors.

Table 3.3: Lighting condition statistics

Total	Daylight	Shaded	Artificial Light
2755 images	895 images	901 images	959 images



Figure 3.6: Sample images from the PATD database with different types of lighting conditions: (a) In sunlight; (b) Shaded; (c) In artificial light.

Motion Blur

For motion blur, the images were captured in the above-mentioned lighting conditions, and in various positions and distances using the same focus blur. The presence of motion blur is due to certain types of movement of the hand or object at a certain speed and in a certain direction.

Table 3.4: Motion blur statistics

Total	Horizontal motion blur	Vertical motion blur	Out-of-focus blur
711 images	199 images	205 images	307 images

There are basically two types of motion blur, namely horizontal and vertical motion blur. The act of image capture may occur at any moment of the motion; hence we made images with different degrees of blur.



Figure 3.7: Sample images from the PATD database showing two types of motion blur: (a) Horizontal motion blur; (b) Vertical motion blur.

The new versions of the smartphones handled the problem of handshake, in contrast to the old versions, but motion blur still exists if it exceeds the acceptable ratio of motion.

Out-of-focus Blur

For the out-of-focus blur, the images were captured in many positions and at different distances with the same motion blur in three different lighting conditions. Out-of-focus blur occurs if the distance between the camera and the document is very small or the focusing position is not in the center but at an edge. It may be on different sides: left of, right of, above, below the document edge or over the whole document (see Table 3.4).

Multiple distortions may contain one or many distortions that were shown above. By taking a reference capture, we captured them using two camera positions, three lighting conditions and two motions blur values, which is a blurry motion and blurry focus. This led to many captures per document for each dataset.



Figure 3.8: Sample images from the PATD database showing two levels of focus-blur with OCR accuracy: (a) Relatively out-of-focus blur (b) Totally out-of-focus blur.

3.4 Ground Truth File Description

An essential component of any database is the presence of ground truth data [123]. Each image of our two datasets (the smartphone-captured an Arabic printed document dataset and a scanned Arabic printed document dataset) is provided with the following ground-truth information:

- A reproduction of the text in a document using the Free Online OCR program in a PDF format of the newspaper.
- The types of distortion in each document.
- The ID of a captured document.

The XML file is grouped into five main parts:

- Specs: Here, we present the encoding of image and article numbers.
- Content: This part provides the fonts types, subtitles, titles, text, legends, author and image numbers in each article.
- Font: Here, we provide the font name which exists in the document pages.
- Smartphone: This states the type of camera used to capture the image document.
- Distortion: This provides all the information about the distortions for each image and for each capture parameters such as illumination conditions, out-of-focus blur, motion blur and perspective.

The next figure presents an example of the XML file of a newspaper page, which contains all the necessary ground-truth information that specifies the page.

```

<page_type="jpg" nb_Articles="2">
<Resources language="Arabic" code="ar"/>
<smartphone_type> iPhone 6 </smartphone_type>
<distortion>
<light_condition> Night + lamp (artificial light) indoors. </light_condition>
<motion_blur> not available </motion_blur>
<focus_blur> not available </focus_blur>
<distance_value> 20 cm </distance_value>
<Perspective_status> Parallel with Y-axis of the document </Perspective_status>
</distortion>
<articles> <article id="1" font= "Kacstone">
<title t="أمانة مسابقة القرآن الكريم" تعرف بنشاطاتها "font= "ALsharek Title"/>
<subtitle> not available </subtitle>
<author> الرياض - "الرياض" </author>
<image nb="1" legend ="من زيارة نائب وزير الشؤون الإسلامية للسم الأمانة" />
<article_content >
سجلت الأمانة العامة لمسابقة القرآن الكريم المحلية والدولية حضورا مميزا ومشاركة متميزة في المهرجان الوطني للتراث والثقافة "الجنادرية 32" ضمن جناح وزارة الشؤون
الإسلامية والدعوة والإرشاد، التي يشرف عليه الإدارة العامة للعلاقات والإعلام بالتعاون مع الجهات ذات العلاقة.
وتهدف مشاركة الأمانة العامة لمسابقة القرآن الكريم إلى التعريف ببعض الجهود التي تبذلها المملكة في خدمة القرآن الكريم وتكريم أهله وتشجيع حفظه من خلال دعم المسابقات
الفرائية والمحلية بها.
</article_content>
</article>
<article id="2" font= "Kacstone">
<title t="نزلاء السجون يعرضون إبداعاتهم للزوار" font= "ALsharek Title" />
<subtitle> not available </subtitle>
<author> الرياض - "الرياض" </author>
<image nb="1" legend ="التقيب محمد الزهراني" />
<article_content >
أقرت الإدارة العامة للسجون مساحة كبيرة في جناحها بالمهرجان لعرض إنتاج النزلاء والنزيلات من المواد الاستهلاكية التي تدرّبوا عليها بالتعاون مع المؤسسة العامة للتدريب
التقني والمهني وأوضح التقيب محمد بن علي الزهراني من إدارة التأهيل والإصلاح بإصلاحية جدة أن هذا الإنتاج لن يتم بيعه خلال المهرجان، وقال إنه سباع في مناسبات أخرى
مثل أسبوع التنزيل الخليجي والمعارض المنظمة تحت إشراف الغرف التجارية بمختلف مناطق المملكة.
</article_content>
</article>
</articles>
</page>

```

3.5 Summary

In this chapter, we presented our new type of printed database, namely an off-line large database collected from two different sources, 810 images scanned in color format with 600 dpi resolution and 2954 smartphone-captured images with 450 dpi resolution, giving a total of 3764 images created from pages, which were split into scanned and captured images to provide a test set for Arabic text recognition research, it is freely available to interested researchers at the following webpage <http://www.inf.u-szeged.hu/patd>. The PATD database contains a wide range of sizes, fonts, styles and structures selected from ten Arabic newspapers and each newspaper has unique features including image-shape, image-size, text-size, text-font and, article-structure. The PATD database provides information about single and multiple capture distortions of the images of newspapers, and it allows one to handle problems under real conditions; moreover, it contains a complete ground truth with all the details of the specs, fonts and, distortions in the images of the PATD database. The database is freely available for research purposes and we hope it will assist the Arabic printed text recognition research community.

Chapter 4

Arabic Document Layout analysis

In this chapter, we describe the proposed approaches for logical structure extraction and document analysis, more specifically for Arabic newspaper pages. We will explain and demonstrate how we can move from a raw page image to a set of exploitable structured information representing the logical organization of the document. Each section focuses on elucidating the processes and explaining the various steps in detail. We will describe the working environment, the necessary tools, and the general architecture of our application. We will also describe the different processing steps in our application with the help of various figures. We will then present some test results and we discuss these results, and explain the reasons for the good and bad performance of our three approaches.

We will begin by outlining the logical structure extraction method for limited Arabic newspaper pages (PDF version), which is presented in Section 4.1. In the next section, we will present a title and subtitle detection method for Arabic newspaper pages. In the last section, we introduce an improved method for the efficient analysis of the smartphone-captured newspaper/magazine pages.

4.1 Printed Arabic Newspaper

Here, we present a system for recognizing the logical structure (hierarchical organization) of Arabic newspapers pages. These are characterized by a rich and variable structure. They may contain several articles composed of titles, figures, author's names and figure captions. However, the logical structure recognition of a newspaper page is preceded by the extraction of its physical structure. This extraction is performed in our system using a combined method which is essentially based on RLSA [84], PP (Projection Profile) analysis, and CC (connected component) labeling [11, 70, 155]. Logical structure extraction is then performed based on certain rules of sizes and positions of the physical elements extracted earlier, and also on an a priori knowledge of certain properties of logical entities (titles, figures, authors, captions, etc.). Lastly, the hierarchical organization of the document is represented as an XML file generated automatically. To evaluate the performance of our system, we tested it on a large set of newspaper images.

There are many studies that focus on this topic, beginning with physical structure extraction.

- The method of Liu et al. [41] used a bottom-up [14] method based on the grouping of CCs. For merging text lines into blocks, neighboring CCs are taken into account and the best CC pair is chosen for the merge. Filtering is then applied to remove CCs of small size, following by title labeling and the graphic image separation from a graph.
- Another bottom-up method was proposed by Mitchell and Yan [113], which groups the rectangular regions that contain the most pixels in the foreground to build patterns. The component segmentation is carried out by more than three pixels. The size, shape, and range of pixel values are the characteristics used in the classification of the entity. Next, patterns are grouped to form lines and blocks.
- Hadjar and Ingold [42] proposed an algorithm using a bottom-up approach based on CCs. The only difference when extracting the blocks is that it is performed by merging them into large areas.
- Cinque et al. [85] proposed a method called DAN that consists of three steps. First, preprocessing is applied. On the resulting image, they apply a quad-tree technique to cut the document into small blocks. The chopping result is the entry of the third step: the merge. This step applies pre-classification criteria to merge similar blocks into larger regions.
- Antonacopoulos et al. [5] proposed a method that starts with a binarization technique, followed by black /white separator detection, where the result of these steps is a set of empty rectangles. The next step is to segment the page

using a hybrid technique. The text regions are separated from the non-text using the statistical properties of the text. Then, they extract the text lines and the regions. The lines of text are detected from previously extracted text regions.

As for the logical structure extraction phase, it has many new improvements. However, it is still quite limited compared to the physical extraction phase.

- In [77], in conjunction with the extraction of the logical structure of the journal pages, the authors propose labeling the extracted blocks in figures, titles and texts. The figures are separated from the text in the first step of the extraction of the physical structure, and rules relating to the dominant height of characters and the average distance between the lines of text are used to perform the logical labeling of the text blocks into titles and texts.
- In [146], the authors proposed a method for the logical segmentation of articles in old newspapers. The purpose of the segmentation was to extract metadata from the digitized images by using a method of pixel sequence classification based on conditional random fields, associated with a set of rules that defines the notion of an article within a newspaper copy.

In this section, we are interested in recognizing the logical structure of the hierarchical organization of a category of documents with a complex structure, namely newspaper pages. This section is organized as follows. In Subsection 4.1.1, we present our recognition approach. Then we present our experimental results on Arabic newspaper page segmentation in Subsection 4.1.2, and lastly in Subsection 4.1.3 we give a brief summary.

4.1.1 Method Overview

Our system was designed to handle Arabic newspaper pages, and we chose the daily newspaper called Echorouk [54] for our test corpus. The pages of this newspaper vary considerably in their structure and this makes their treatment and analysis quite difficult. Our approach has two parts, namely the extraction of the physical structure and recognition of the logical structure, where the first part seeks to analyze the document image in order to recognize its physical structure. Our system combines two steps: pre-processing to improve the quality of the input image, and segmentation to separate the physical entities contained in the document. The second part also has several steps: labeling by logical labels, physical entities previously extracted and generating a structured XML/DTD files that represent the logical organization of the document, and the generation of a dynamic tree, representing the hierarchical organization of the document.

Image segmentation involves partitioning the image of newspaper page into several

related regions. The three approaches for document segmentation are the bottom-up approach, the top-down approach, and the mixed approach (see Section 2.4 for more detail).

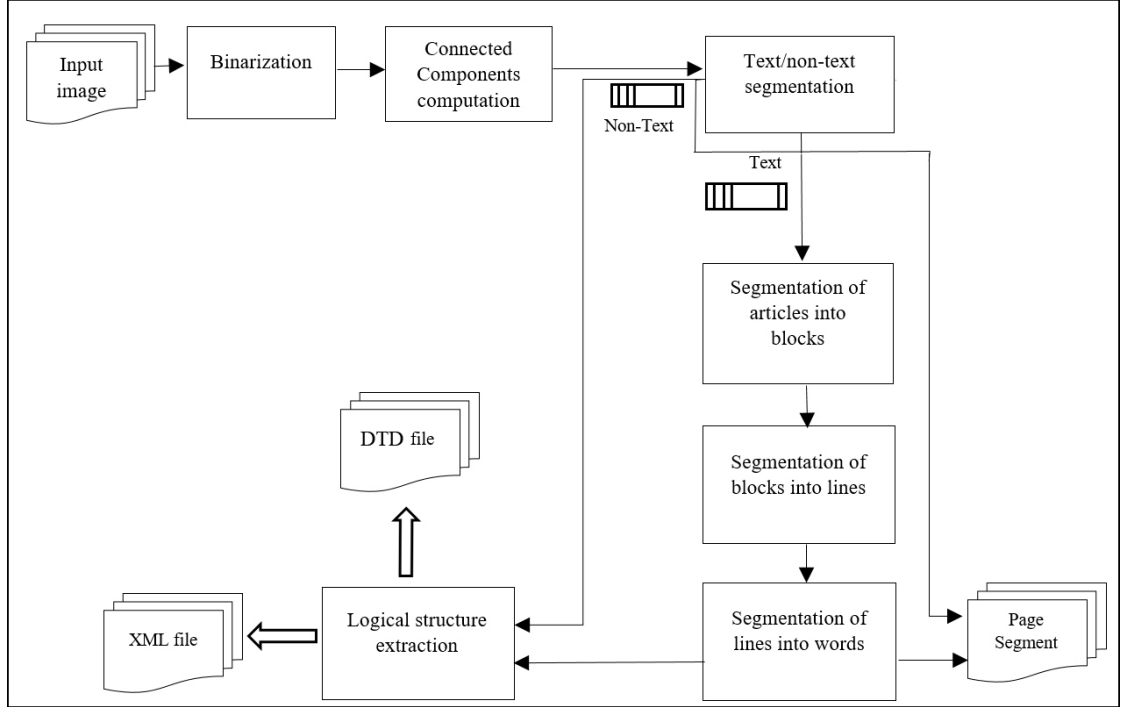


Figure 4.1: The diagram of our DAR process.

In our study, we perform a mixed segmentation. We commence with an upward segmentation that starts from the pixels of the image and merges them into CCs. Then the CC information is used to separate the graphic components of the page (figures, bands, rectangles, and straight lines). Next, to divide the text of the newspaper page into articles, we use mixed segmentation based on the analysis of PP, the RLSA smoothing algorithm, and the labeling of CCs. Lastly, we apply a descending segmentation to divide the articles of the page into blocks, the blocks into lines and lines into words. The diagram above represents our approach steps.

Dataset Features

As mentioned earlier, Echourouk newspaper was chosen as a test corpus for our approach evaluation. The features of the Echourouk daily newspaper pages used in the test are as follows.

Most of the pages that we used have 6 to 12 articles;

All the pages have a header, while the footer rarely exists;

Most of the figures have a rectangle shape;

The majority of pages have black article(s) and bands which contain a title/ subtitle/ author;

The legend may exist/be absent; The legend may be found at the side/below/above/

or inside the figure;

Most of the pages that we used have 1 to 10 figures;

Some of the articles are bordered by rectangle line;

Each article contains 2 blocks to 6 blocks;

The author may be mentioned;

Many pages have vertical/horizontal separated lines;

Each article has a different size/position/structure.

Pre-processing

Before segmenation we must do some pre-processing, which has two steps, namely a transformation into grayscale and Otsu thresholding. The aim of these transformations is to construct an image for the labeling of CCs (see Figure 4.2).

Connected Component Labeling

The labeling of the CCs involves merging the neighboring black pixels into a separate unit. The result of the labeling of the CCs is a colored image where each CC is displayed in a different color (see Figure 4.2).



Figure 4.2: An example of connected component labeling: (a) The original image; (b) Binarization result; (c) Labeling result.

Let L be the label matrix, CCs be all connected components and CCi be the i_{th} connected component of binarized image. Every CCi is characterized by the following set of features:

- $B(CC_i)$ is the bounding box of CC_i with (X_{li}, Y_{li}) , (X_{ri}, Y_{ri}) is the top-left and bottom-right coordinate, H_i and W_i is the height and width of $B(CC_i)$.
- $Csize(CC_i)$ is the number of pixels of CC_i .

- $Bsize(CCi)$ is the size of $B(CCi)$, $Bsize(CCi) = Wi * Hi$;
- $Cdens(CCi)$ is the ratio of $Csize(CCi)$ and $Bsize(CCi)$;

$$Cdens(CCi) = \frac{Csize(CCi)}{Bsize(CCi)}, Cdens \in (0,1]$$

- A_{HW} is the aspect ratio denotes the of width and height of CCi , $A_{HW} \in (0, 1]$;

$$A_{HW} = \frac{\min(H_i, W_i)}{\max(H_i, W_i)}$$

- $Holap(CCi)$ is the collection of connected components exist on the same row with CCi (the same horizontal line).

Text/Non-text Segmentation

Taking into account the fact that the header and the footer are always bordered at the top or bottom by a horizontal straight line, the detection of the header/footer relies on the detection of these dividing lines.

In order to detect the separating line of the header/footer, we applied the following conditions;

- The widest $B(CCi)$ is extracted from the top/bottom part of the page (1/6 page height).
- If the Wi is greater than (image-width/2), then this component is treated as the dividing line of the header/footer.
- Lastly, all CCs above the line separating the header are treated as header-components and all the components below the foot dividing line are treated as footer-components (see Figure 4.7).

Separating text/non-text components is a key step before decomposing the text of the page, and it has several steps. It begins with the detection of the header/footer (already extracted), the figure detection, then the detection of the band/ rectangle/ figure/ Black thread, and after the removal of all the detected components.

For this purpose, we used formulas and conditions based on the CC features; the CCi is considered as a non-text component if it satisfies one of the following conditions:

- $Hi > image - Height/10$, this is correct for all the vertical non-text layouts because the text elements cannot exceed this height threshold ;
- $Wi > image - width/11$, this is correct for all the horizontal non-text layouts because the simple text or title cannot pass this width threshold;
- $Cdens(CCi) > 0.9$, this is correct for all the tiny straight-lines that can be found under the authors/legend, because the $Csize(CCi)$ of this element in the most

cases covers the whole $Bsize(CCi)$, after taking into account a binary image with noise (gradient line). These thresholds were chosen with preliminary experiments to provide accurate results (see Figure 4.3).

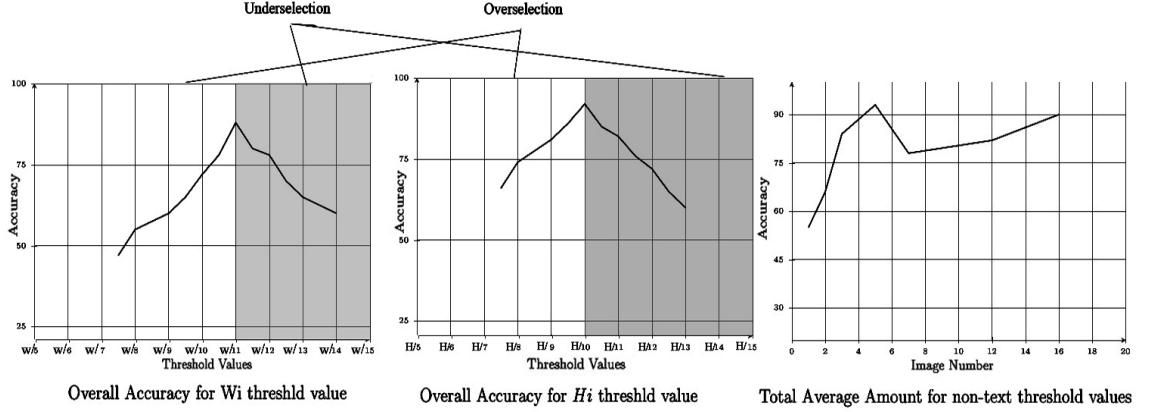


Figure 4.3: Overall accuracy for the non-text threshold: W(image-width), H(image-Height).

Next, for dividing the non-text elements into four layouts (black bands/ rectangles/ black thread/ figure), we used the following conditions:

- Borders: $Cdens(CCi) < 0.1$, the $Csize(CCi)$ of this layout in the most cases just covers the $Bsize(CCi)$ boundaries.
- Black threads: $Cdens(CCi) > 0.9$, the $Csize(CCi)$ of this layout in the most cases covers the whole $Bsize(CCi)$, after taking into account a binary image with noise (see Figure 4.4).

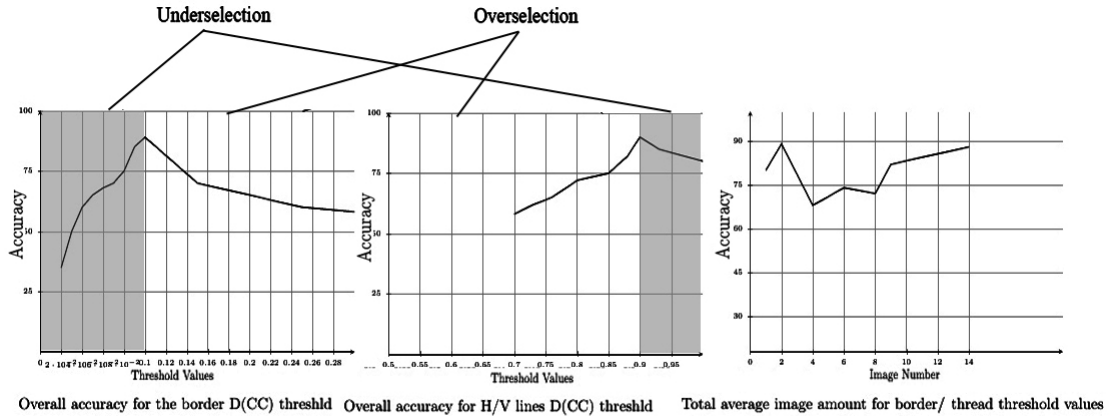


Figure 4.4: Overall accuracy for the border/black thread threshold.

We reserved the data of these layouts then removed, so we just have the figures and the black bands on the page. We notice that there are two types of black band; namely a title-black-band and article-black-band;

- For title-black-band detection: $A_{HW} < 0.15$ recall that all layouts of this type are very wide compared to their length (see Figure 4.5).

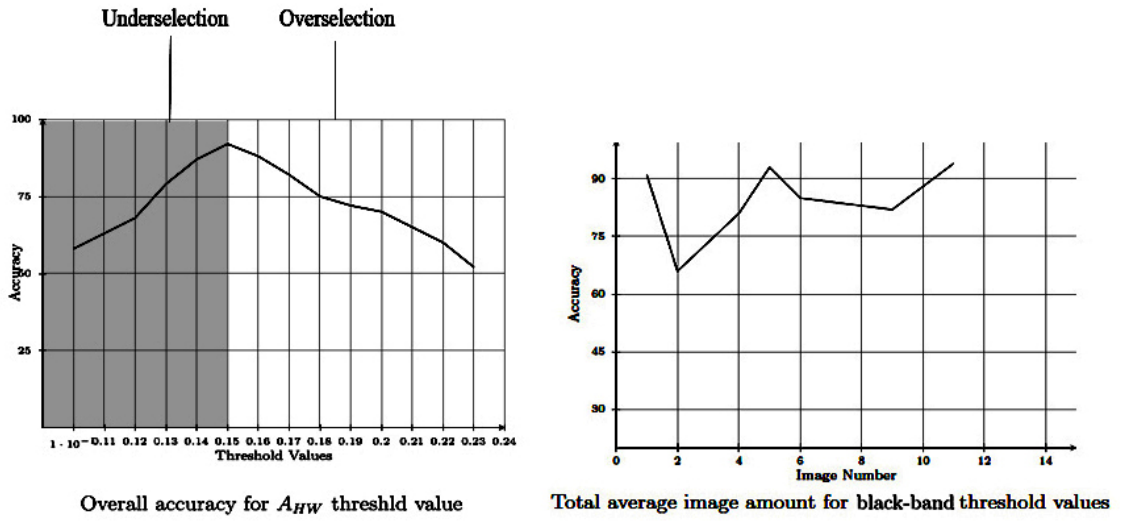


Figure 4.5: Overall accuracy for the text-black-band threshold.

Many threshold values were tested for each graphic layout, after a lot of tests (Figure 4.3, 4.4, 4.5), we selected the ranges that had the best test results and contained the most best thresholds compared to other values, while the other line graph for each layout shows the number of images that were used for threshold values extraction along with their results.

- For figure and article-black-band layouts the CC features cannot be so helpful in this case because these layouts are very similar to each other in some circumstances, so we will employ other ideas which are expressed in the following steps:

- Convert the layouts to the negative version;
- Apply horizontal adaptive RLSA, its threshold being proportional to the W_i where $(W_i/2)$;
- Label CCs for the new regions, where the $CC_{ii} = Holap(CC_{ii})$;
- Calculate the row numbers $CC_{ii} = Rn$ for each original $B(CC_i)$;
- Apply adaptive threshold proportional to the H_i ;
if $Rn > H_i / average(B(Rn)_{height})$, then this CC_{ii} is an article-black-band; else it is a figure layout;
- Convert the figure layouts to their original pixels (negative version) (see Figure 4.6);

As shown in Figure 4.7, each non-text layout is highlighted in one of four colors where each color represents one type of layout. The yellow color represents the header, the magenta color represents the black stripes, the red color represents the rectangle layout which surrounds the article, and the blue color represents the figure layout. Afterwards, we eliminate all the detected labels except the black bands because it contains text information, so we have to convert black bands labels to the negative version to extract this information.

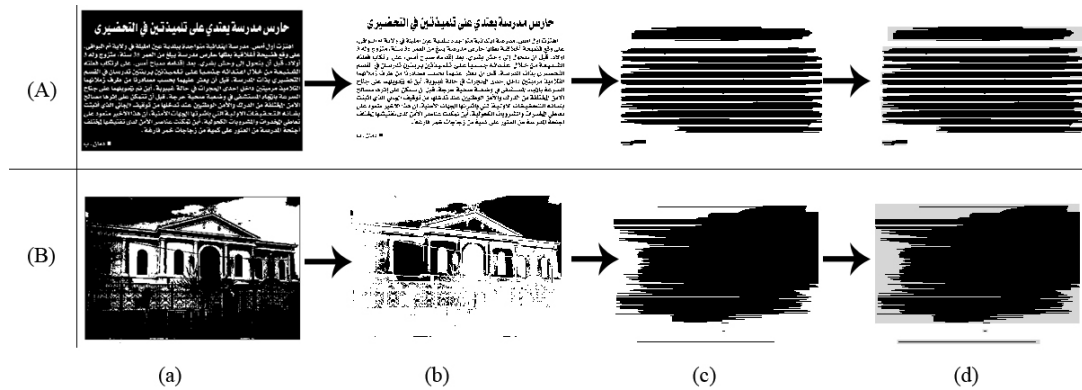


Figure 4.6: An example of the detection process: (a) The binarized image of a non-text layout; (b) Converting the layouts to the negative version; (c) Applying Horizontal ARLSA; (d) Extracting the new bounding-boxes; (A) Example of an article-black-band; (B) Example of a figure layout.



Figure 4.7: Example of detecting and removing graphics: (a) The same newspaper page used in the previous step; (b) The graphic layout elimination result.

Article/Block Segmentation

After separating the text/non-text, the next step is to divide a text into articles. The decomposition of the text into an article is carried out in our system using the RLSA algorithm and PP analysis. Here VRLSA and HRLSA were used for the small-bounding-box/big-bounding-box, then we applied the AND operation between the vertical and horizontal PP analysis using the histogram projection (see Figure 4.9). This method consists of calculating the number of black pixels accumulated in the horizontal or vertical directions in order to identify the separation positions.

The histogram of horizontal projections is first obtained by calculating the number of black pixels for each line of the image. The corresponding horizontal pro-



Figure 4.8: Example of histograms projection on newspaper page part.

jection histogram will consist of peaks and valleys. The valleys represent spaces of separation between articles.

Next, we calculate the histogram of vertical projections on the page delimited by a valley of the histogram of horizontal projections (see Figure 4.8).

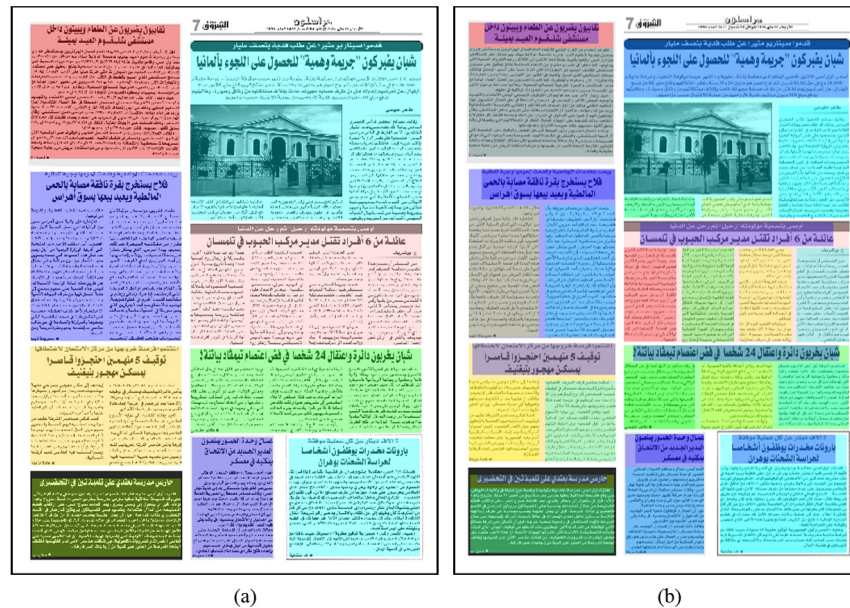


Figure 4.9: Example of article/ block segmentation step: (a) Article segmentation result; (b) Block segmentation result.

After separating the articles, the next step is the decomposition of each article into blocks. Thus, two types of text blocks are distinguished, namely the header block representing the headings, and the text columns. The decomposition of the article into blocks is carried out in our system using VPP and HPP analysis.

In header block extraction; the valleys in the vertical projection histogram correspond to the separation spaces between the text columns. The calculation of the

histogram of vertical projections is repeated for each valley of the histogram of horizontal projections. The valley is treated as a separator between the header block and the rest of the article which contain the rest of columns found from the maximum vertical projections. The header block will therefore be the part of the article between the start of the article and this separator.

Column extraction: The columns are obtained from the histogram of vertical projections of the part of the article below the header block. The valleys of this histogram constitute the spaces of separation between the articles.

As shown in Figure 4.9, each article is highlighted in a different color and we did the same thing for the block segmentation to help us differentiate each other.

Segmentation of Blocks in Lines

The next step in extracting the physical structure is the decomposition of each block into lines. To do this, we used the technique of line segmentation implemented in [52]. This technique relies on the application of a horizontal projection on each block separately in order to extract the lines that compose it. Here, we do the following:

- Suppose \hat{f} is the considering binary block with $a * b$ as the size of it.
- Calculate the histogram of the horizontal projections P of each block.

$$P = \left\{ p_x \left| p_x = \sum_{y=1}^b \hat{f}(x, y), 1 \leq x \leq a \right. \right\}$$

- Extract the local minima (LM): treating the histogram as a discrete function $f(x)$, for k ranging from 1 to the $histogramsizes^{-1}$, k is treated a local minimum if $f(k-1) > f(k)$ and $f(k+1) > f(k)$.
- Filtrate LM :
 - if $(threshold(TL) < LM_{width})$ then eliminate LM , where TL is the width of the longest $LM/2$;
 - if $SM < 2 * (MD)/3$ then eliminate the longer of the two very similar LMs ; where MD =the average distance; and SM =the distance between two successive minima that matches the text height. The remaining minima correspond to the gaps between the lines of the text.
- Assign the existing black pixels in the separator zones to the nearest line of text.

As shown in the Figure 4.10, the segmentation process was represented by a colored rectangle for each line. We notice that even if the lines are very close to each other, the segmentation results were very accurate and efficient.

Segmentation of Lines in Words

CC labeling and RLSA smoothing are applied on each text-line separately to extract the words from it. Hence the segmentation of a line into words is carried out

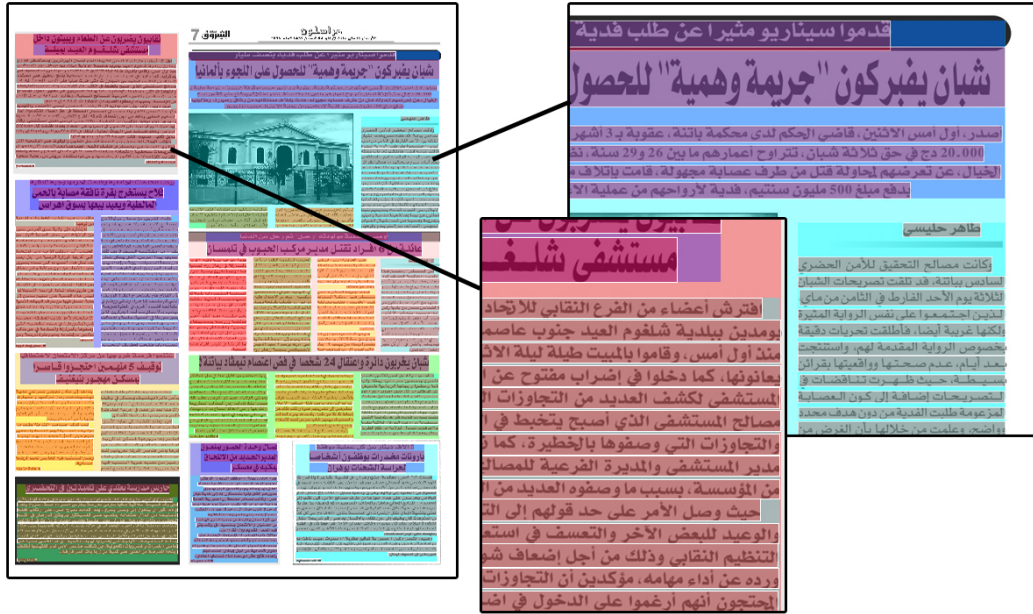


Figure 4.10: Example of segmentation blocks into lines.

as follows (see Figure 4.11):

- Apply VRLSA to interconnect diacritical points to words. The smoothing threshold is set to the value of boundig-box height of each line.
- Use HRLSA with a threshold equal to 2 to connect the tiny sub-words of the same word.
- Label the CCs for each line.
- Apply filtering to improve the separation between successive words, using the following steps:
 - Order the CC s (left to right) then extract all the $B(CC_i)$;
 - Extract all the spaces S_i between every consecutive CC using

$$(Xr_i, Yr_i) \text{ for } B(CC_i), \text{ and } (Xl_{i+1}, Yr_{i+1}) \text{ for } B(CC_{i+1});$$

- Calculate the ratio R_{hw} for each space S_i , where

$$R_{hw} = Si_{width}/line_{height}, R_{hw} \in (0, 1];$$

- Filtrate the spaces resultant using a threshold T , where

$$T = 0.116 \text{ and } Si \text{ is a gap-word if } R_{hw} < T \text{ else it is treated as gap-letter;}$$

- Eliminate all the gap-letters, then divide the line based on to the gap-word values.

These values having been chosen after preliminary experiments to provide accurate results. We tested them on just a few pages because each newspaper page can have a huge number of words (each page can have over 1000 words), Figure 4.12 presents a page-part of segmented lines to words.

Author extraction: After analyzing many Echrouk newspaper pages, we noticed that the author's name is found either in the first line of the first column or in the last line of the last column. Exploiting this information, for each article we applied the following procedure:

- Extract $B(b_1)$ and $B(b_{last})$, the first and last bounding-box (the columns);
- Extract $B(l_1)$ from $B(b_1)$ and $B(l_{last})$ from $B(b_{last})$, the first and last bounding-box (the lines);
- Calculate the mean line-gap M_l using all the bounding-box-lines except the first line-gap, where the line-gap

$$l = Yl_{i+1} - Yr_i;$$

- Calculate the horizontal line-gaps between the $B(b_{last})$ and $B(l_{last})$, where

$$DF_1 = Xr_{B(b_{last})} - Xr_{B(l_{last})}, DF_2 = Xl_{B(b_{last})} - Xl_{B(l_{last})};$$

- Applying three thresholds;

if $M_l/(Yl_2 - Yr_1) < 0.2$, then the 1st line will be labeled as the author's name;
 else if $DF_1/(Xr_{B(b_{last})} - Xl_{B(b_{last})}) > 0.1 \wedge DF_2/(Xr_{B(b_{last})} - Xl_{B(b_{last})}) < 0.1$ then
 the last line will be labeled as the author's name.

Figure 4.13 represents the distribution of threshold values based on the M_l , DF_1 and DF_2 values used for author extraction, where the gray region treated as the precise range of the correct detection that can give the best results compared to other values in the white region.

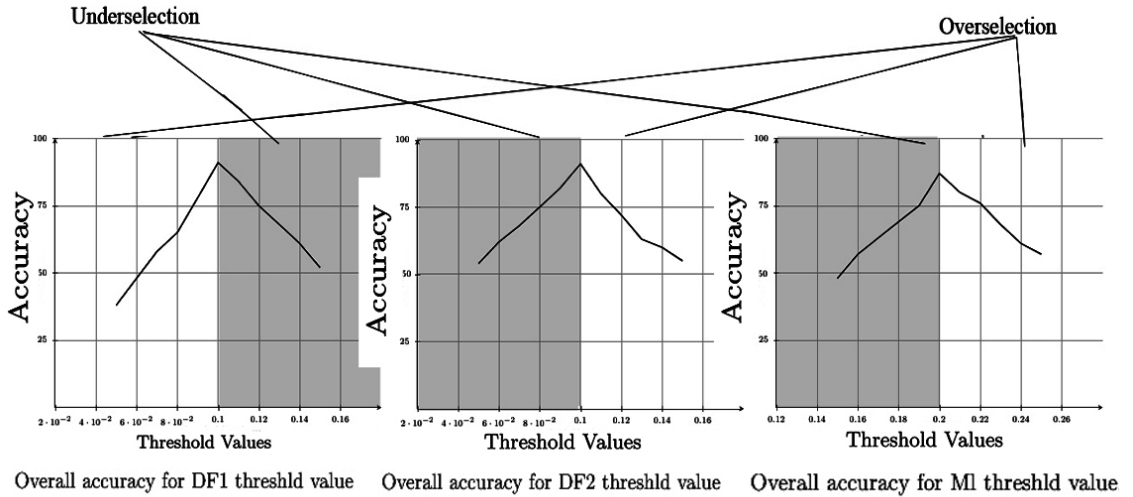


Figure 4.13: Overall accuracy for M_l , DF_1 and DF_2 threshold values.

Legend extraction: The legend is found either below/next to the figures or inside the figures in these newspaper structures. Hence we apply two heuristic filters, the first filter is used to extract the overlapped legend (see Figure 4.14) while the second

is used for the legend that is outside the figure (see Figure 4.15), noting that the two filters are connected to each other.

- Overlapped legend

- Calculate the mean of all the line-heights M_l that exist in the *article_figure* except the title-height;
- Extract the bottom-part B from figure, where $B_{height} = M_l$;
- Convert B to the negative version N_B ;
- Apply HRLSA with threshold $t = B_{width}/15$, then VRLSA with threshold $t = B_{height}/10$;
- Labeling CCs;
- Calculate the histogram H of the horizontal projection of N_B ;
- Extract all the values Vs , where $Vs = 0$ from H ;
- Test the Vs , if Vs were placed on the top/bottom in N_B and $CC_{number} = 1$ then $CC_{bounding-box}$ will be labeled as a legend;
- Change the figure bottom-coordinates (remove the legend part).

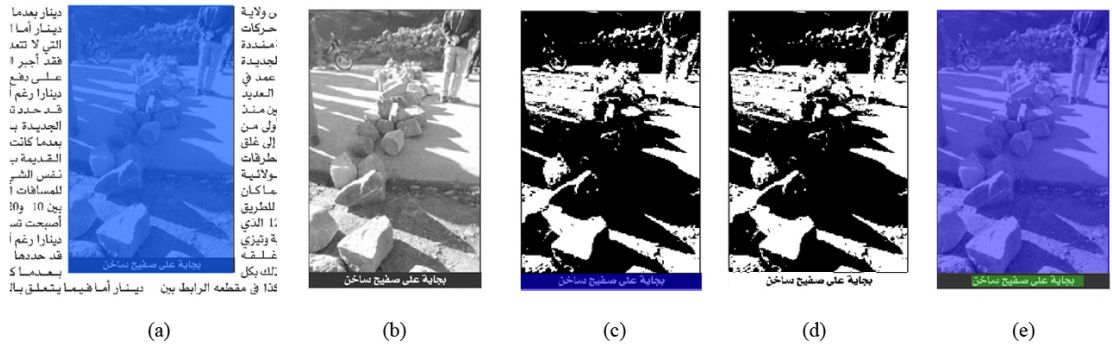


Figure 4.14: Example of legend detection process: (a) The original image; (b) Figure extracted; (c) Binarization result; (d) Bottom-part extracted; (e) Legend/figure detection result.

In case that the first filter did not give legend part, then we apply the second filter:

- Normal legend

- Extract the bottom image, where

$$\begin{aligned} image_{width} &= figure_{width} \\ image_{height} &= figure_{height}/3; \end{aligned}$$

- Apply the smearing technique on the bottom image, starting by using HRLSA with $threshold1 = image_{width}/15$, then VRLSA with $threshold2 = image_{height}/10$ to merge all the neighboring words to one CC;
- Extract the bounding-boxes $B(CCi)$ of all the resultant CCs;
- Apply geometric rules:
 - for $B(CCi)_{number} = 1$, if;

$$((Xr_{B(\text{figure})} - Xl_{B(\text{figure})}) - (Xr_{B(CC1)} - Xl_{B(CC1)}) > 10)) \wedge (Yr_{B(CC1)} - Yl_{B(CC1)} < \text{image_height}/4)$$

then this line will be labeled as a legend;

- for $B(CCi)_{\text{number}} > 1$;

Extract the first and second B(CCi) based on the y-coordinates, then if;

$$((Xr_{B(\text{figure})} - Xl_{B(\text{figure})}) - (Xr_{B(CC1)} - Xl_{B(CC1)}) > 10)) \wedge (Yr_{B(CC1)} - Yl_{B(CC1)} < \text{image_height}/4) \wedge (Yl_{B(CC2)} - Yr_{B(CC1)} > Yr_{B(CC1)} - Yl_{B(CC1)})$$

then the first B(CC1) will be labeled as a legend;

- Apply the same filter to the other surrounding images;



Figure 4.15: Example of legend detection process: (a) The original image; (b) Smearing result; (c) Bounding-box extraction result; (A1) The legend existent; (A2) The legend non-existent.

The rest of the elements are labeled using the following rules, Figure 4.16 shows the different detected logical entities:

- Header/Footer: The header and the footer are the first elements identified during the extraction of the physical structure. The labeling of these elements is the attribution of the labels "Page header" and "Footer" to the header and foot areas located in the image.
- Article: In this step, we just numbered the resultant articles from the first phase taking into account their positions on the page, and because the Arabic newspaper pages are read from right to left, we started from the top-right item which is labeled "Article 1", and so on.
- Block: There are two types of block labeling, the first one being the header block (title), and the other blocks represent the text columns.



Figure 4.16: Example of a segmentation page and its logical structure: (a) Newspaper page along with its articles; (b) The detected logical elements.

- Columns: These blocks represent the text columns "Column" and the order number where the columns are arranged, and naturally because it is an Arabic page, "Column 1" starts from the rightmost column of each article.
- Text-lines: the labeling here corresponds to their order in the block (from top to bottom). So the first row in each column is labeled "line 1", and so on.

XML/DTD files

This step is very important in our system because it summarizes all the labels extracted in a well organized and structured fashion. We chose both the XML and DTD formats because they are widely used in the area of electronic document management and they also allow the exchange of results. We match each log page with a corresponding XML annotation file (Figure 4.17 shows the XML/DTD files for the tested image).

Each generated XML file contains the following image details:

- Name, format, height, and width, and article numbers that exist on the page.
- Header/Footer position.

For each article:

- Article position along with its author, the number of columns, the number of figures.
- A header block position, and the existing title levels.

- Title position, its level, and the number of words in the title.
- Figure position, and the legend of it if it exists.
- Column position, its number, and the number of rows in the column.
- The position of each line, its number, and the number of words in the line.
- The position of each word, and its number in the line or the title.

<pre> <?xml version="1.0" encoding="UTF-8"?> <PAGE name="0023" type="jpg" width="2000" height="3015"/> <heading box="803;0;1872;233" /> <ARTICLE num= "1" BOX= "72;177;754;857" nb_columns="1"> <heading BOX="148;177;685;284" > <title num="1" nb_words="7" box = "148;177;685;228"> <word num="1" box = "148;177;330;228" /> <word num="2" box = "428;177;684;228" /> <word num="3" box = "331;177;418;228" /> </title> </heading> <column NUM = "1" BOX="72;302;753;856" > <line num="1" nb_words="14" box = "76;302;732;325"> <word num="1" box = "132;302;217;325" /> <word num="2" box = "91;302;127;325" /> </line> <line num="2" nb_words="13" box = "76;327;753;351"> </line> </column> <author NUM_Line= "23" NUM_columns= "2" BOX="74;1756;199;856919"/> </ARTICLE > <ARTICLE num= "2" BOX= "804;265;1863;1168" nb_columns="3"> <FIGURE num= "1" BOX= "804;582;1499;1047" > <Dimension width= "695" height= "465"/></FIGURE> </ARTICLE > </PAGE> </pre>	<pre> <ELEMENT PAGE (ARTICLE+,heading?,footer?)> <!ATTLIST PAGE nom CDATA #REQUIRED type CDATA #REQUIRED width CDATA #REQUIRED height CDATA #REQUIRED> <ELEMENT ARTICLE (FIGURE*,heading,column+,author)> <!ATTLIST ARTICLE num ID #REQUIRED BOX CDATA #REQUIRED nb_columns CDATA #REQUIRED> <ELEMENT FIGURE (Dimension,LEGEND?)> <!ATTLIST LEGEND width CDATA #REQUIRED height CDATA #REQUIRED> <!ATTLIST LEGEND BOX CDATA #REQUIRED> <ELEMENT heading (Title+)> <ELEMENT Title (word+)> <!ATTLIST Title num ID #REQUIRED nb_word CDATA #REQUIRED box CDATA #REQUIRED> <ELEMENT word EMPTY> <!ATTLIST word num ID #REQUIRED box CDATA #REQUIRED> <ELEMENT column (Title+)> <ELEMENT Line (word+)> <!ATTLIST Line num ID #REQUIRED nb_word CDATA #REQUIRED box CDATA #REQUIRED> <ELEMENT word EMPTY> <!ATTLIST word num ID #REQUIRED box CDATA #REQUIRED> <!ATTLIST author num ID #REQUIRED nb_columns CDATA #REQUIRED box CDATA #REQUIRED> <ELEMENT heading (#PCDATA)> <ELEMENT footer (#PCDATA)> </pre>
(a)	(b)

Figure 4.17: Example of XML and DTD file: (a) XML file of the previous image; (b) DTD file of the previous image (Figure 4.10).

In addition, it is possible to build a dynamic tree that is enriched and well organized at each stage of the processing in our system. This tree provides all the information of the page in a dynamic and well-organized, structured and hierarchical form, and it can be treated as a navigation tool inside the page, where the component tree of the page allows one to easily locate in a single click any physical or logical element of the page (titles, articles, lines, authors, captions, figures, etc.).

4.1.2 Results and Discussion

In order to validate our system, we used all the images in the corpus, more than 100 pages being taken from the Echourouk website [54](see dataset features section). The evaluation was performed on JPG images generated from PDF files, in order to evaluate the methods on noise-free images, and because the proposed method may be used to perform a layout analysis of encrypted documents.

The viewing of the results of the experiments was done using the Java Runtime

Environment. In fact, the user can detect any part of the page: articles, authors, pictures, header, footer, columns, lines, words, titles, and get all the information about the whole page, number of articles, number of (columns, lines, words) present in each article, and the user can also toggle the view of the different layers: image text separation, threads, text line extraction and line merging into blocks, and word extraction.

The logical labels of the various elements of the test images were found manually on our own in all the steps of newspaper recognition. Then we applied our system to all the test images so as to label them automatically (see Table 4.1).

Table 4.1: Example of some labels detected manually and automatically in one newspaper page.

Label	Automatically detected	Manually detected
Page header	1	1
Footer	0	0
Figures	1	1
Blocks	18	18
Lines	315	315
Words	2042	1980
Legends	0	0
Authors	9	9

The automatic labeling results of each image were compared with the actual labels (manually set) to determine the recognition rate. In order to verify the general applicability of our system, we varied the test images, so that they contained a different number of articles, with different positions, and also contain straight lines, strips, figures; and so on.

Table 4.2 summarizes the average recognition rate for each logical entity. In this table, we can see that the system has managed to recognize most of the existing logical entities, and it had a recognition rate of 91.90% using 55 images.

With our solution, many results were improved, such as the text/non-text segmentation, logical labeling and label extraction, differentiate between the similar blocks or that merged for a certain type of frame (a non-closed rectangle).

When comparing the recognition performance of structures (physical and logical) with other studies, we found that the identification and verification results are quite different due to the diversity of the structured pages and the nature of detected elements. However for evaluation purposes, we selected the most similar one among the other studies to ours in our study (see Table 4.3).

According to the Hadjar and Ingold [76], they achieved excellent scores for two different Arabic newspaper pages, where they obtained (50.53%, 99.81%) for article detection, 97.59% for figure segmentation, 96.51% for line segmentation and 95.21% for block segmentation. However, their study had five labels (article(with/without border), block, line, image) compared with our study which can detect more than

Table 4.2: Test results.

Label	Recognition score
Page header	99.23%
Footer	89.45%
Figures	90.20%
Black bands	95.70%
Borders	88.55%
Straight horizontal lines	98.87%
Articles	90.03%
Blocks	90.32%
Lines	99.85%
Words	75.08%
Columns	90.28%
Legends	93.10%
Authors	94.16%
Average	91.90%

Table 4.3: A comparaison with other approach.

Algorithms	Tested on	Extracted label	Recognition score
Proposed method	55 images extracted from (Alshorouk) newspaper	13 labels extracted	91.90%
Hadjar and In-gold	50 images extracted from (Alhayat+Annahar) newspapers	5 labels extracted	87.93%(Annahar) 93.08% (Alhayat)

ten labels. In the article detection step, their method works well when there are bordering lines (straight-vertical-line/straight-horizontal-lines/rectangle-form) and they achieved a score of 99.81% for these types of articles, but it performed less well in other cases where they achieved a score of 50.53% for an article that has no boundary data. Furthermore, there was no logical structure extraction.

Although our program performed well, many problems were encountered in the case of page structure (see Figure 4.18). Some of these were:

- The spaces between the words were irregular;
- The legend was part of the figure;
- The large article included a small article;
- There was very luminous figure, or one that contained writing (overlap problem);
- The shape of the black bands was non-rectangular;

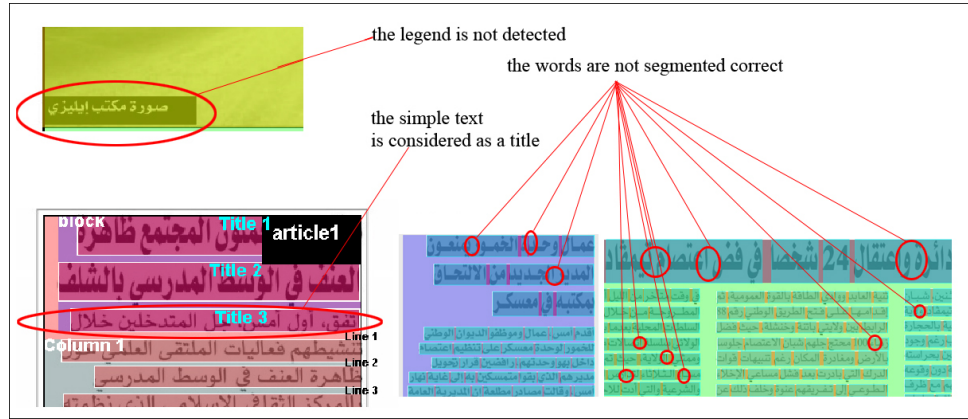


Figure 4.18: Example of some typical errors in the physical and logical phase.

- The footer/header was not separated by a dividing line.

In general, our application provides good results. These results in our view are encouraging considering that the dataset contained a great variety of newspaper pages.

4.1.3 Summary

In this section, we presented a system that converts a raw image of a newspaper page into a set of structured data that can be used to represent the logical organization of a document. The extraction of the logical structure is performed by labeling the different physical elements extracted. Several tests were conducted to assess the performance of our system and the results we got are encouraging. In the current state of our project, we have an application that satisfactorily meets the goals set at the beginning.

4.2 Title Detection in Printed Arabic Newspaper Pages

Recent studies on text line segmentation have not focused on title segmentation in complex structure documents, which may represent the upper rows in each article of a document page. Many methods available cannot correctly distinguish between the titles and the text, especially when it contains more than one title.

The application of text-line segmentation is not always easy to do, due to the existence of skew, script variations, noise, text-lines with different sizes and different fonts. One especially problematic issue, which is key aim of our study, is the line segmentation of a large scale heading, which must be done in such a way that we can represent it as titles and their subtitle detection in Arabic document pages. The existing approaches for text-line extraction cannot correctly distinguish the titles from the text, especially when it contains more than one title. Real text in documents often contains titles and subtitles, and such text lines cannot be precisely identified with state-of-the-art methods.

A wide variety of title detection methods for documents can be classified and incorporated in many techniques: Active Contour Model (Snake), HPP, VPP, CCs, the Bounding box-based method, smearing method, the Hough Transform (HT), and using HMMs. Here, the main studies of text-line detection methods are outlined.

- Bukhari et al. [139] presented a robust text-line segmentation approach against skew, curl and noise, which is based on an active contour model (Snake) with the novel idea of several baby snakes and their convergence in a vertical direction using the ridges which are found by applying multi-oriented anisotropic Gaussian filter banks, it is computationally expensive.
- In [8, 17], they applied HPP and VPP techniques for the text-line segmentation approach by finding the inter-line gap and taking into consideration the separation between two consecutive lines.
- In [33, 56, 126], they applied a smearing method; namely smearing the consecutive black pixels in the horizontal projection, then the pixels between them were marked in black if the distance between any two was less than a threshold value. However, it fails when there is no space between two consecutive lines or overlapping lines.
- In [93], they could not extract Arabic text documents with large-scale headings and titles; moreover, it is not efficient in the case of a document with a complex structure.

This inspired us to develop a new method that can extract not just one title, but also every title and subtitle on a document page. In this section, we present a new text-line detection method for complex-structured documents where the detected text is

treated as a title or subtitle and each page contains many titles corresponding to the number of articles.

The section is organized as follows: In part 2, the related work is described. In part 3, we describe each step of our algorithm in detail. Experiments and results are presented in part 4 and finally, in part 5 we provide a brief summary.

4.2.1 Overview of the Method Used

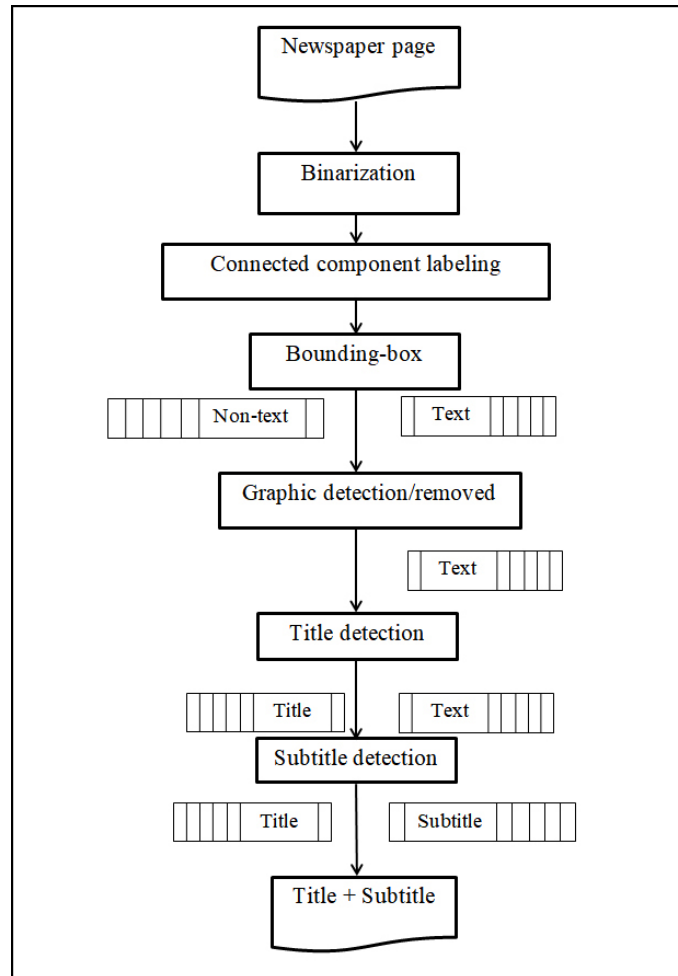


Figure 4.19: Outline of our method proposed for Title/Subtitle detection.

Titles are the key elements of documents because there are no page documents without titles and subtitles (see Figure 4.19). The size of these titles is not always larger than other text on the page, especially when the title belongs to a small article. Nevertheless, subtitles are usually found above or below the title where the space and the size between the subtitle and the article text are identical. These generic characteristics present challenges in the Arabic language in terms of text-line extraction from a document page. Figure 4.20 illustrates the problem where the spaces between the peaks did not provide useful information for title extraction.



Figure 4.20: The input image with a plot of the HPP on the right.

The conditions applied for subtitle and title detection will be determined by using the following geometrical features:

- Height: CC bounding box height,
- Width: CC bounding box width,
- Aspect Ratio: Width divided by height,
- Solidity: Area of the CC (in pixels) divided by the area of its convex hull,
- Area: Number of pixels in the CC,
- Position: CC coordinates.

4.2.2 Pre-processing

We used global thresholding to produce a clear image that simplifies the processing of the later steps. For this step, the Otsu binarization method [105] is used to transform the image into two possible pixel values (0 and 1) to reduce the noise and overcome the illumination issue that arises during the scanning process.

We know of course that document pages may contain more than one figure. These figures consist of the largest proportion of pixels that in some cases give us imprecise information and this could lead to poor results in the subsequent steps. Moreover, the existence of black blocks could corrupt the essential parts that are needed later on. To overcome these problems and facilitate title and subtitle segmentation, we used the same formulas that we applied in Section 4.1, with constraints on the size of the CCs, the ratio of height and width, and the density of black pixels in the CC (see Figure 4.21).

4.2.3 Title Segmentation

The detection of the titles is done by taking into account the fact that not just the height of the titles is important, but also the number of pixels in each component and its position coordinates. Here, our proposed method is based on RLSA and the CC labeling technique. Horizontal RLSA [90] is then applied to the resulting image

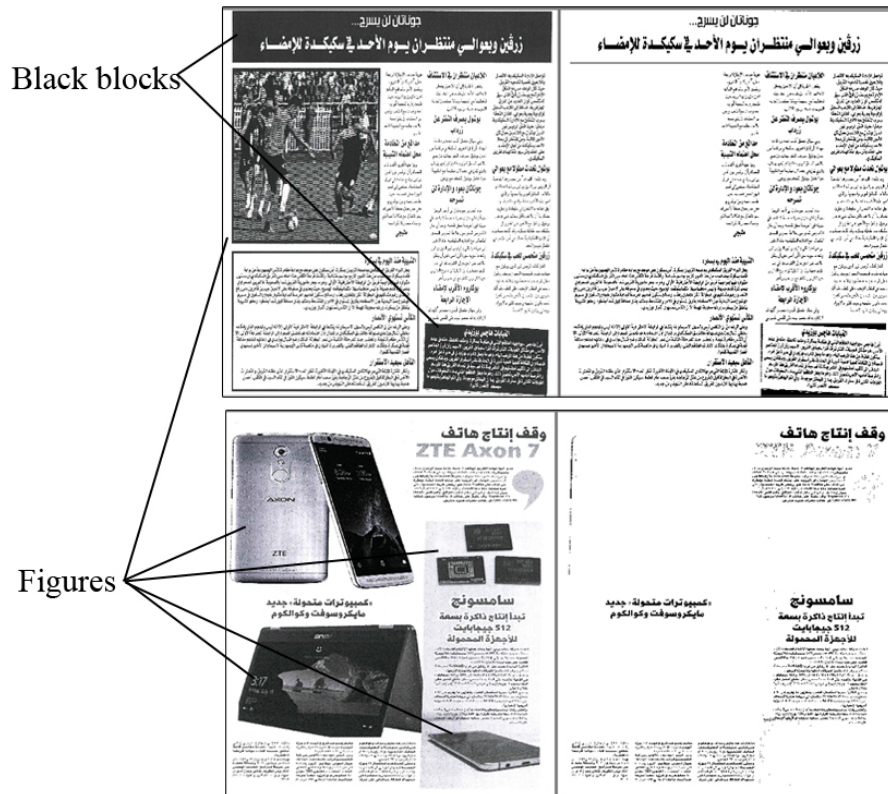


Figure 4.21: Examples of removing figures and black blocks.

of the preceding step to clear spaces between words of the same line of text and Vertical RLSA is used to connect the diacritic marks to the corresponding words. Let L_0 be a horizontal segment of unit length. The Run-Length Smoothing closing algorithm fuses nearby pixels of the binary image X by γL_0 , where γ is a size parameter.

$$RLSA(X) = X \oplus \gamma L_0 \ominus \gamma L_0 \quad (4.2.1)$$

The horizontal and vertical smoothing thresholds were determined empirically, namely (with threshold 1=1%) and (with threshold 2=0.85%) proportional to the size of the page, respectively.

Actually, the characters of the titles are usually larger than those of the lines of simple text. In this case, the threshold of the horizontal RLSA was previously too small to connect the words of a big title. To remedy this problem, we applied a second horizontal RLSA with a larger threshold (with threshold 3= 1.55 % proportional to the size of the page) only on the parts of the image containing probable major titles. These are composed of CCs whose height is greater than (1.5 x the most common text height in the document). We then applied another labeling of the CCs on the RLSA smoothed image. As the words of a single line of text (simple or title) become connected, each line of text is treated as a separate component. A CC is treated as a title if its height is greater than (1.2 x the most common text

height in the document); otherwise it is treated as a simple line of text (see Figure 4.22).

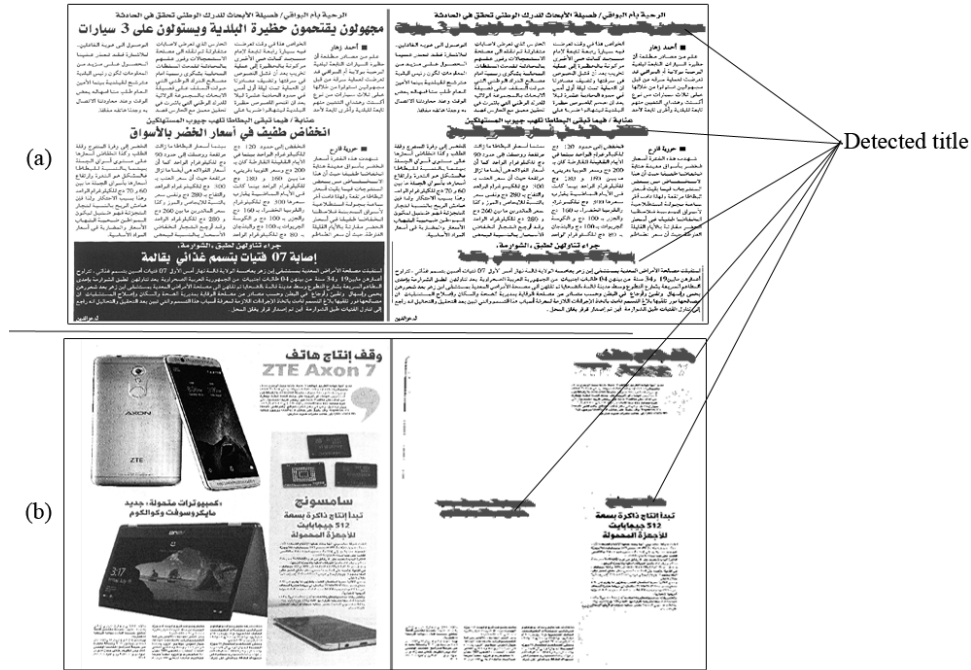


Figure 4.22: The title segmentation results of our proposed method on Arabic text documents: (a) Newspaper page with textual data; (b) Magazine page with graphical/textual data.

4.2.4 Subtitle Extraction

However, these techniques only provided us with the main titles, not subtitles. Other criteria must be used to add the other titles. For this, we combined two criteria, namely the size of the CC of the previous step and its position relative to the main titles. Here, the other titles are extracted using the PP method (see Figure 4.23). Let L_1 and L_2 denote the lines of text that are above and below a main title T respectively, V_1 denote the image width and V_2, V_3 denoting the heights. Now let:

- $V_1 = 1.25\%$, $V_2 = 3.35\%$, $V_3 = 0.07\%$;
- (x_1, y_1) : the coordinates of the bottom left-hand corner of L_1 ;
- (x_2, y_2) : the coordinates of the bottom right-hand corner of L_1 ;
- (z_1, k_1) : the coordinates of the top left-hand corner of T ;
- (z_2, k_2) : the coordinates of the bottom right-hand corner of T ;
- (z_3, k_3) : the coordinates of the bottom left-hand corner of T ;
- (z_4, k_4) : the coordinates of the top right-hand corner of T ;
- (x_3, y_3) : the coordinates of the top left-hand corner of L_2 ;
- (x_4, y_4) : the coordinates of the top right-hand corner of L_2 .

The lines of text L_1 and L_2 are treated as subtitles if they satisfy the following conditions:

-The height of L_1 and $L_2 > (\text{Threshold } 4) \cdot 1.15\%$ proportional to the size of the page document;

- $(|z_1 - x_1| < V_1) \wedge ((|y_1 - k_1| < V_2) \vee (|y_2 - k_4| < V_2))$

- $((k_1 - y_1 > V_3 \vee |y_2 - k_4| > V_3) \wedge (|y_1 - k_1| > V_3 \vee k_4 - y_2 > V_3))$

- $(|x_3 - z_3| < V_1) \wedge ((|k_3 - y_3| < V_2) \vee (|k_2 - y_4| < V_2))$

- $((y_3 - k_3 > V_3 \vee |k_2 - y_4| > V_3) \wedge (|k_3 - y_3| > V_3 \vee y_4 - k_2 > V_3))$



Figure 4.23: The subtitle segmentation results for an Arabic document page.

We proceed in the same way with other subtitles when they exist by letting (x_1, y_1) be the coordinates of the lower left-hand corner of T, (x_2, y_2) be the coordinates of the lower right-hand corner of T, (z_1, k_1) be the coordinates of the upper left-hand corner of subtitle L, (z_2, k_2) be the coordinates of the bottom right-hand corner of subtitle L in a recursive way until no line satisfies these conditions. Noting that these lines are checked by aspect ratio value Ar and solidity S , where these CCs treated as lines if $Ar < 0.15$ and $0.5 < S < 0.87$, the Ar value was tested and already checked in the previous study (see Section 4.1) while the S value chosen by taking into account that all the bold text lines after smearing technique have high solidity value.

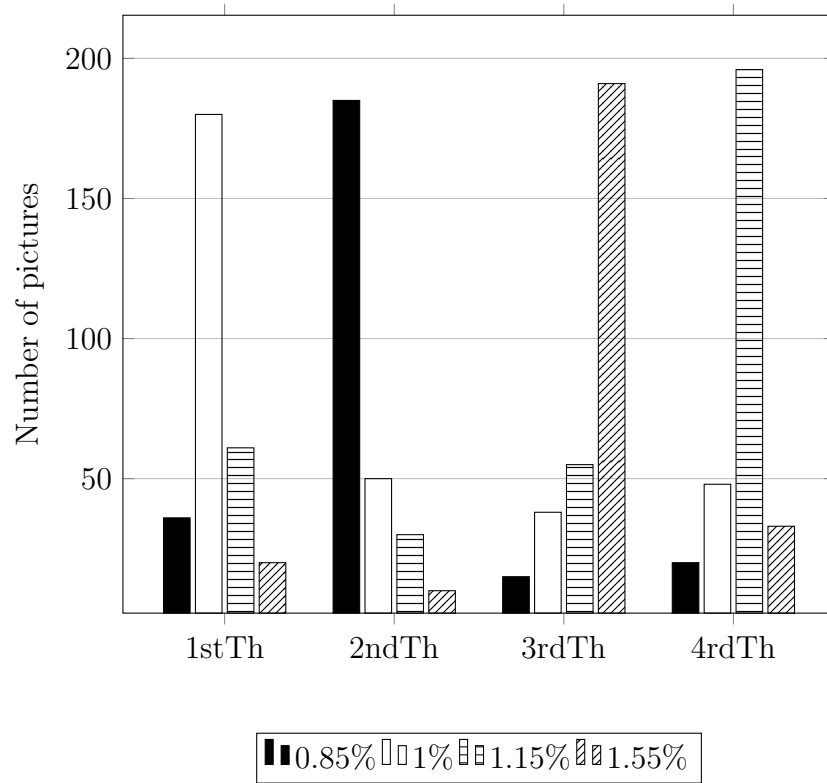


Figure 4.24: The number of experiments of our data thresholds.

Figure 4.24 shows the number of document pages with each threshold, where the threshold is computed by getting image information under valid conditions.

Every threshold in this chart is used for title or subtitle detection, which has four tests (with proportional values of 0.85, 1, 1.15 and 1.55, where these values are found by calculating the median of proportional values of the images (the ratio is extracted using information about the dimension and size of the document image, all the titles and subtitles being on each individual page)).

Here, the y parameter denotes the number of images that matched this proportional value. We took the best and the highest number column for each threshold. Our results were tested on over three hundred pages to check the accuracy and performance of our system.

4.2.5 Results And Discussion

Our method can be used in two modes; namely, the application returns just the cropped titles and subtitles, or it returns the whole page with colored titles and subtitles. Both demonstrate the segmentation phase in a clear way.

To assess our proposed system, we used the same criteria as in that described in [92], which is the title-segmentation accuracy in percentage terms. The constraints are given below.

- 1) If a single CC of a line is segmented to another line, this error is counted as two

line errors.

2) If n subtitles and simple text are merged together, then it is counted as n line errors.

3) If n titles and subtitles are merged together, then it is counted as n line errors.

We used the following formula for computing all the errors:

$$Accuracy\% = 100 - (E/Totalnumberwords) * 100$$

where E = the segmentation error.

The algorithm was tested on three hundred scanned pages at 300 dpi got from the PATD (see Chapter 3). The algorithm gives excellent scores, which may be as high as 98.02% for titles, and 98.15% for subtitles. Table 4.4 below lists the results obtained during the testing process with various font types, styles, and sizes.

Table 4.4: Test results.

Font Type	Title extraction	Subtitle extraction
AL-Quds	98.18 %	97.96 %
AxTManal	98.15 %	98.23 %
Beirut	/	98.87 %
AL-Quds Bold	98.45 %	98.17 %
Kacstone	97.56 %	97.55 %
Alshrek Titles	97.78 %	/
Total	98.02 %	98.15 %

Due to the lack of articles with the same goal in Arabic documents, we evaluated the performance of our approach by comparing it with related articles that have similar goals such as line segmentation (see Table 4.5, see Figure 4.25).

In [93], they took a binarized image as input and the algorithm returned a data file that contains a segmented image. Though it went well (99%) for line segmentation with different fonts, it cannot be applied to a complex structure, due to a dependence on the VPP in the first phase of page segmentation, where there must always be a vertical white space on the whole page between the articles. Therefore there are incorrectly segmented lines with poor detection in the case of the absence of vertical white space in the page image.

In [55], the authors proposed a robust method for line segmentation and they achieved a score of 97.8% for both simplified and traditional text fonts (97.3% for simplified font and 98.4% for a traditional font), based on splitting one region into many smaller regions in a repetitive way until no more regions require splitting using a HPP and a set of constraints. However, as the program does not work with



Figure 4.25: Samples of images used for the line segmentation method: (a) Ibrahim's data [55]: an article with the same text size in one column; (b) Soujanya et al.'s data [111]: an article with a different size font in one column; (c) Ayesh et al.'s data [93]: variability of font size and the possibility of multiple articles, which was restricted by the presence of vertical white spaces between them; (d) Our own data where several articles have different font sizes and figures.

a complex structure that has more than one article and different sizes of text on the same page, it is not possible to extract the lines for both normal text or large size text from each article if it exceeds an article on the image page or if it contains variable font size texts in the same article.

Table 4.5: A comparison with other approach.

Algorithms	Tested on	Segmentation level	Recognition score
Proposed method	complexe structure	line segmentation (title/subtitle)	98.08%
Ibrahim	simple structure	line segmentation	97.8%
Soujanya et al.	simple structure	line segmentation	98%
Ayesh et al.	simple structure	line segmentation	99%

Another study [79] focused on the line segmentation of low-quality documents, by investigating different text-line segmentation algorithms like PP, the RLSA and

ARLSA; and by applying HPP they achieved a score of 100% on English documents that had varying spaces. RLSA achieved an accuracy of 96% on overlapping documents, and ARLSA achieved an accuracy of 99% on English documents with overlapping components. However, PP cannot handle images where the text lines are overlapping or touching. RLSA and ARLSA fail if there is any overlap between two text lines. Although the program can handle many size fonts, in the previous study they based it on documents which had just one article hence one title had no more than this, and their approach cannot be applied to pages with a complex structure e.g. when there are many articles, figures, and titles.

In our previous study (see Section 4.1), we got a score of 99.85% for text based on pages taken from the same newspaper. Although the lines are well segmented, a good segmentation line does not mean necessarily good title and subtitle detection. And because every article has a title, this leads us to assume that good article detection means a good title and subtitle extraction, therefore because the performance of article detection was 90.03% it means the title detection cannot exceed this even in the best cases.

4.2.6 Summary

Title segmentation plays a significant role in the segmentation step for the identification of any article in any random document. We handled the problem of distinguishing text and overlapping-lines with small font size, and for large fonts, using RLSA, CCs and PP in scanned pages. We evaluated the proposed method on three hundred text images using the PATD database. The results presented here are superior to those of existing algorithms that perform the same task and our method is more general.

4.3 Smartphone-captured Arabic Newspaper Analysis

Here, we apply a CNN to segment a document image into its page components. Our approach consists of two main steps. Firstly, we apply a new method of extracting layouts based on sharpness/smoothing filters, adaptive thresholding techniques, morphological operations, along with a connected-component labeling and adaptive RLSA for the patch extraction phase. Secondly, the extracted patches will be put into six classes (text, table, figure, title, legend, author) using a CNN, and four other classes (straight-line, text-line, block, article) using projection profile analysis and geometric features. The method was tested on smartphone-captured newspaper images selected from the Printed Arabic Text Database (see Chapter 3). There are a number of significant challenges that segmentation algorithms must overcome. One of these is the quality deterioration of the scanned newspaper due to time and the complex layout of the newspaper pages [20].

There are relatively few papers that focus on DLA for complex structure documents written in Arabic. In this section, we shall discuss some of the recent ones.

- Bukhari et al. [138] presented a method for text/non-text segmentation along with reading flow determination. The dataset used contained scanned documents. In their procedure, binarization was applied by using Otsu [105] and Sauvola [73] methods, then they used the Bloomberg method [36] for text and non-text partition along with a ridge-based line detection technique for text line detection. Despite the good results, this method can only extract a single figure which is treated as the only non-text element that exists on a given page.
- Amer et al. [59] presented a method based on CNN [10] to classify the documents regions into text/ non-text. Using a fast Hough transform with a block adjacency graph [104] and Bradley's adaptive thresholding method as a pre-processing step, ARLSA [27] was utilized for black zone building. These zones were classified into text and non-text by using two techniques, namely Zone-Based and Patch-Based classification. The method was tested on three types of newspapers. In spite of the excellent results, this procedure only handles 2 classes (text/non-text), where the images tested were extracted from PDF files.
- Amany et al. in [16] proposed a method for text/non-text classification in Arabic documents. Using the Sauvola method [73] for the binarization step, along with median filters [72] for noise removal, a Gaussian smoothing filter was used to clean and remove both Gaussian and marginal noise. For skew correction they applied the Radon transform [112], and the resulting image was segmented into multiple CCs and each CC was classified into text or non-text

using an SVM. Lastly, the text zones were segmented into lines by clustering CCs and then into words by clustering the spaces between words and between standalone characters of the same word using k-means [32]. The dataset employed was an Arabic historical document with different font types/sizes. They got very good results. However, there was no logical structure of the extraction (e.g. articles, titles, subtitles, authors, tables, legends).

- Alshameri et al. in [4] presented a method for text/non-text segmentation and text line extraction from document images, where they used RLSA, CCs for text segmentation and an SVM for figure detection, by applying the AND-ing and ORing operations to set the correct bounding-box for each category (text/figure). This technique gave interesting results, but the application of their RLSA is efficient only in certain special cases, where specific thresholds have to be applied, and specific vertical/horizontal projections are used to distinguish between CCs with a special spatial structure. Also, their extraction just extracted three classes (text, text-line, figure).

4.3.1 Method Overview

In this section, a method for DLA using CNN is proposed. We commence with an Arabic document image as input for text/non-text classification and logical structure recognition. The input is corrected and improved by sharpness/ smoothing filters, adaptive thresholding, morphological operations. The resulting image is transformed into CCs via a pixel connectivity technique, using Adaptive RLSA to reduce the numbers of CC, then we transform each CC into a patch (a small image covers the CC and its surrounding pixels) by a cropping technique, where each patch will be classified into six regions. These identified and localized regions are merged using a CNN score along with geometric features for label extraction. Finally, each article is segmented into blocks and lines by using a projection profile analysis. The method is outlined in the pipeline in Figure 4.26.

Regarding the PATD database content, there is statistical information associated with textual and graphical labels on each page of Arabic newspaper images. The pie chart (Figure 4.27) shows the percentage of elements that were used in this study.

4.3.2 Distortion Correction

The images of the database used were captured by three smartphone cameras under different lighting conditions (daylight/ shaded/ night), where some of them were affected by focus blur. Therefore numerous types of distortions had to be overcome to determine the outcome of the following steps.

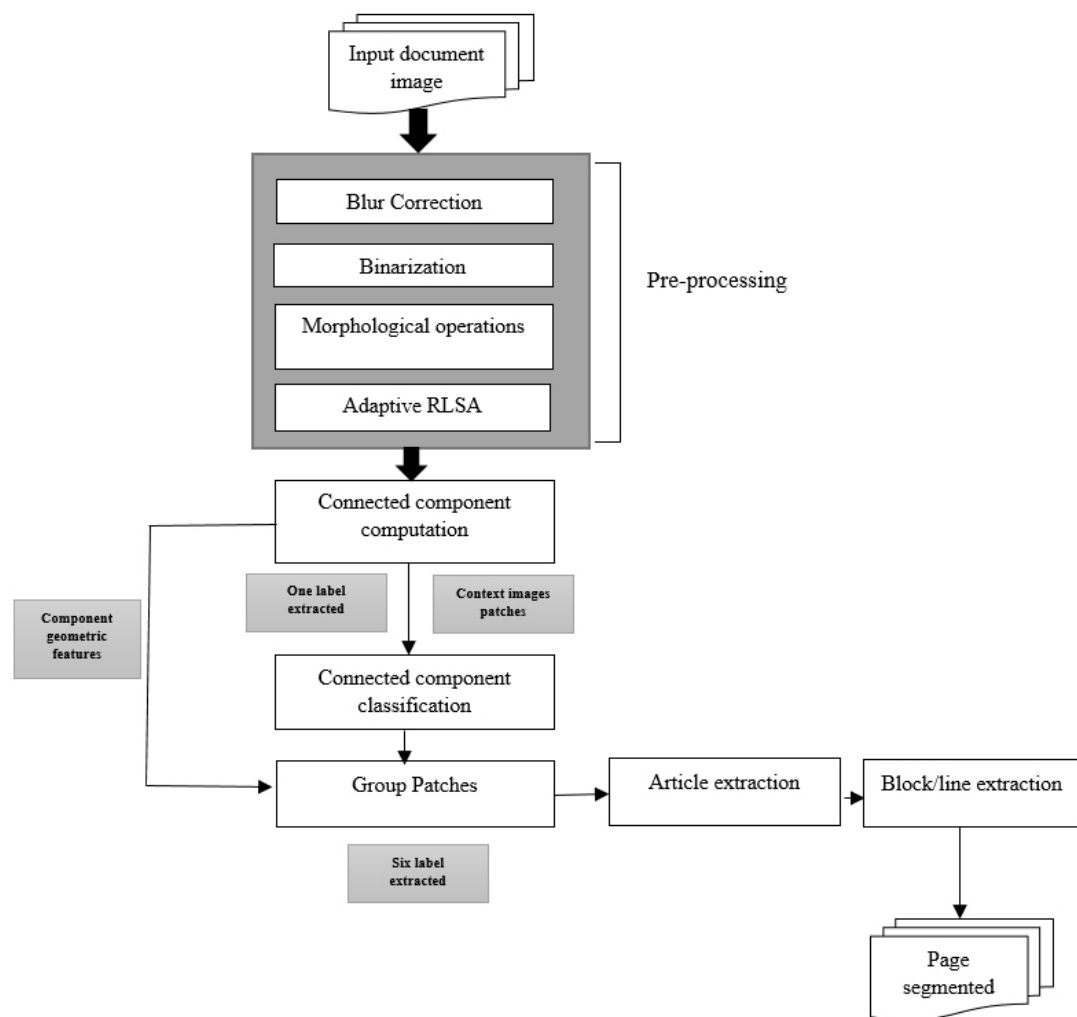


Figure 4.26: Outline of our method proposed for Arabic document analysis.

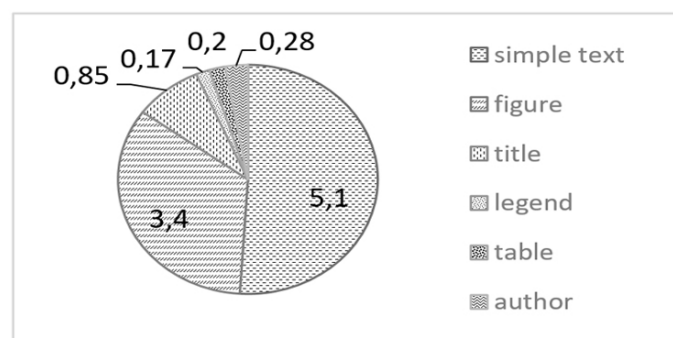


Figure 4.27: Outline of graphical/textual labels.

Skew

Many simple documents have a zero skew, but other types such as newspaper or magazine originally have a partial skew which can cover different parts of the paper.

Because of the nonexistence of original newspaper/magazine pages, the pages were printed and manually scanned and then captured using a smartphone camera, hence another skew could have been introduced. This skew may cause problems in text baseline extraction and DLA techniques.



Figure 4.28: Examples of skew types. The original skew is highlighted in blue and the resulting skew is highlighted in red.

However, the original and new skew were not corrected at this level because if all the parts were corrected as zero skews even for the original skew, the original document structure would vanish and we would lose substantial spatial information. Figure 4.28 shows examples of the original and a skewed version.

Focused Blur

Some of the documents in the database are blurred, and this distortion will cause a problem when recognizing the elements. For this we use a sharpness filter for image restoration, with the following kernel $K = [-1, -1, -1, -1, 9, -1, -1, -1, -1]$ (see Figure 4.29), but we did not deal with the problem of motion blur that exists for some



Figure 4.29: Example of sharpness correction:(First) Original image,(Second) De-blurred image.

documents in the database, which in our view can be handled by a Wiener filter [87] as one of the options for motion-blurred document images.

Binarization

To distinguish between foreground and background on degraded images, a smoothing process was used as the first step to remove shadow sharpness by applying a Gaussian filter as a blurring operation. Because the image was captured at different times of the day, the illumination level was not constant. Moreover, the shadow present can complicate the problem. In such cases, adaptive thresholding can handle these issues (Mean adaptive + Gaussian adaptive), where these algorithms determine the threshold for a pixel based on a small region around it, so we have different thresholds for different regions of the same image (see Figure 4.30).



Figure 4.30: The binarized image of a shaded document image: (a) Original image; (b) Binarized image without Gaussian filter; (c) Binarized image without Gaussian adaptive thresholding; (d) Binarized image with (Gaussian filter + Mean adaptive thresholding + Gaussian adaptive thresholding).

4.3.3 Morphological Operations

In the background and foreground detection stage we seek to generate the patches for the CNN, where these patches are created by CCs. In this step, before connected

component computation, images must be transformed by Adaptive RLSA to reduce the patch numbers because the newspaper page is very big and it contains a huge number of elements that will increase the processing time. The ARLSA method combined adjacent pixels to transform the document image into black blocks which will be used for patch construction later on. We used ARLSA rather than RLSA because it can combine different components with an automatic threshold based on the size and shape of the components which have variable font sizes.

To improve the results of adaptive RLSA, we applied some of morphological operations on the binarized document image, to delete the diacritics, points and make the words more flexible (see Figure 4.31). The document image is then morphologically opened which is represented by $B \circ A$:

- An erosion ($B \ominus A$) with a structuring element followed by a dilation (\oplus) with the same element, where the used element is a vertical short dash equal to the mean gap between the word and its points [35, 36]. Namely,

$$I1 = B \circ A = (B \ominus A) \oplus A \quad \text{where}$$

$$A = \frac{1}{n} * \sum_{i=1}^n x_i$$

B= binary image
 A = arithmetic mean (structuring element)
 n = the number of spaces between the word and its points
 x_i = the numerical distances

- A dilation operation then is applied with a horizontal element equal to the mean gap between the words. These operations were applied to merge the small regions and remove the tiny regions. The following figure shows the results of morphological operations.

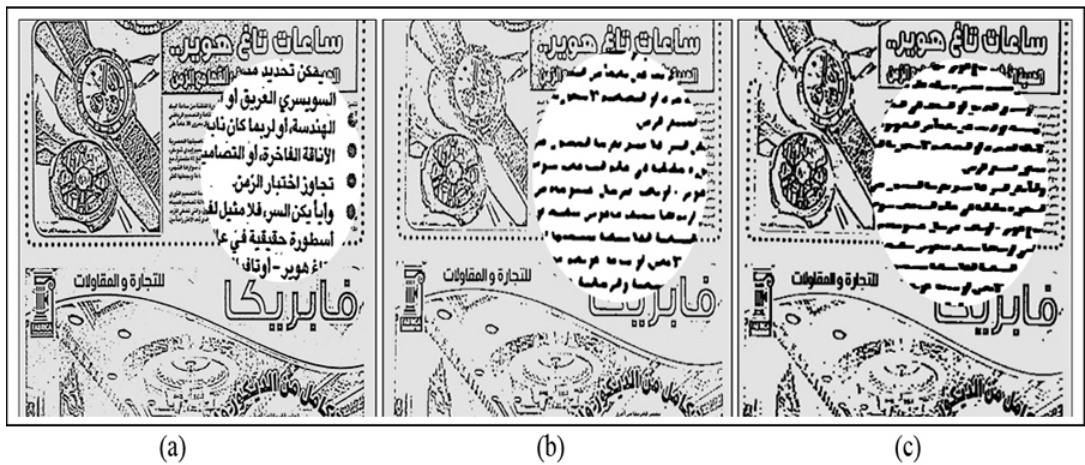


Figure 4.31: Some results of opening and dilation operations: (a) The binarized image; (b) Morphological opening; (c) Morphological dilation.

4.3.4 Connected Component Extraction

Connected components were extracted by grouping image pixels into components based on pixel connectivity, where each CC contained similar pixel intensity values. After grouping pixels, each CC constructed was labeled with a different color. The extracted CCs were grouped from top to bottom by its position of the y-coordinates with right-to-left reading flow, where each CC was labeled with a unique identification number to permit easy retrieval later on (see the figure below).

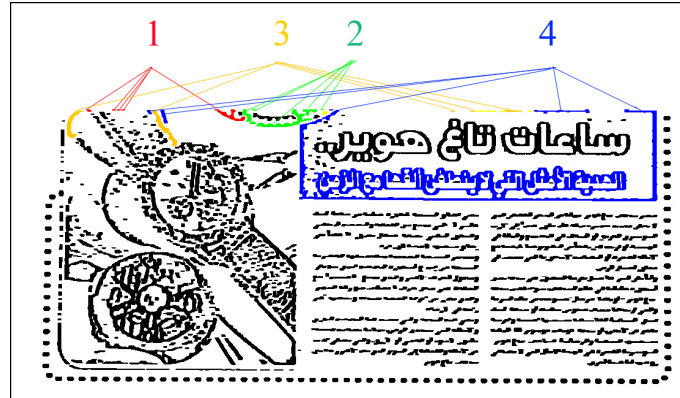


Figure 4.32: The CC ordering according to its y-coordinate (each consecutive block of five CCs is highlighted in one color).

As can be seen in Figure 4.32 there is no uniform arrangement of CCs, hence neighboring position features cannot be applied (which is very different from the simple structure used in previous studies). To overcome this obstacle, we selected other features using the bounding-box.

We started by ordering CCs from top to bottom according to their height and the recurrence of the same height if it exists. Because the text covers approximately 64% of the page (see Figure 4.27), text-CC is then the most frequent one, while the lowest ones are non-text elements; so for text-CC detection, we divided the page-CC into many levels according to recurring and length values, then colored the highest and the lowest levels in white. Thus the remaining CCs were text components (see Figure 4.33).

We applied horizontal ARLSA by adaptive threshold related to the mean bounding-box height, and an adaptive window related to the CC size. In the same way we applied adaptive thresholding for other elements by performing the reverse operation (white to black/ black to white). Then we computed CCs again (see Figure 4.34).

At this level we extracted a straight horizontal/vertical line label by applying geometric rules on the CCs that have a lower occurrence percentage, then we removed them to avoid merge problems in the later steps. These rules had already been applied and tested in a previous study (see Section 4.1).

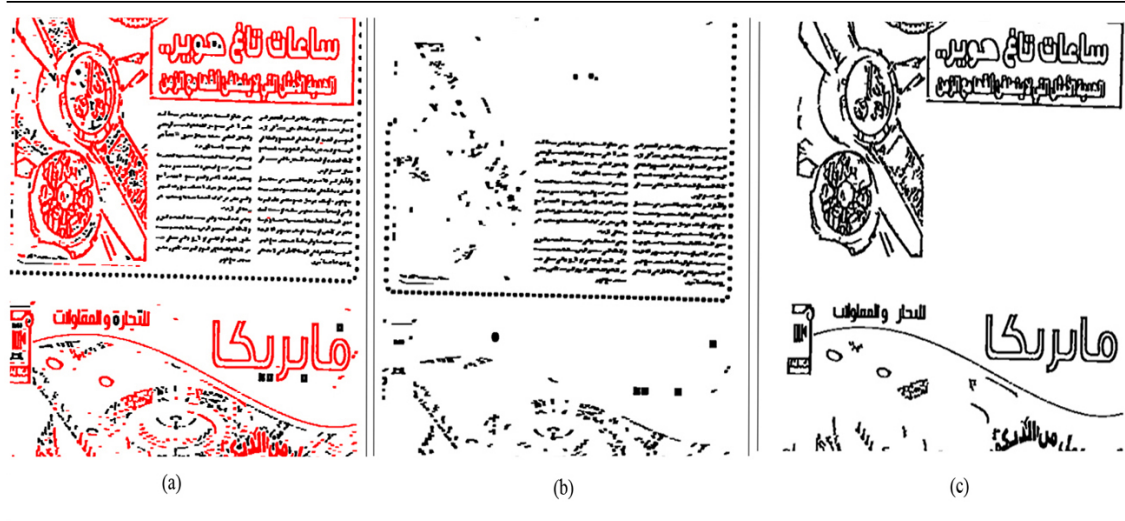


Figure 4.33: Selected example based on particular features:(a) The resultant image from the previous step is partially colored for the selected CC; (b) The resultant image after removing the red parts; (c) The resultant image after removing the black parts.



Figure 4.34: Results of ARLSA and RLSA: (First), using ARLSA, (second), using RLSA.

4.3.5 Conventional Neural Network Classification

Most of the previous studies used CC features for classification. In our study we used patches, where each patch is got from a cropped square image containing the CC at the center along with its surroundings.

As shown in Figure 4.35, we represented the bounding-box of each CC in blue to define the inputs of CNN, while the inclusion of the surroundings aids the classification process.

We chose VGG-16 [82] trained with ImageNet [61] for our analysis problem because it is one of the most efficient Deep CNNs for image recognition.

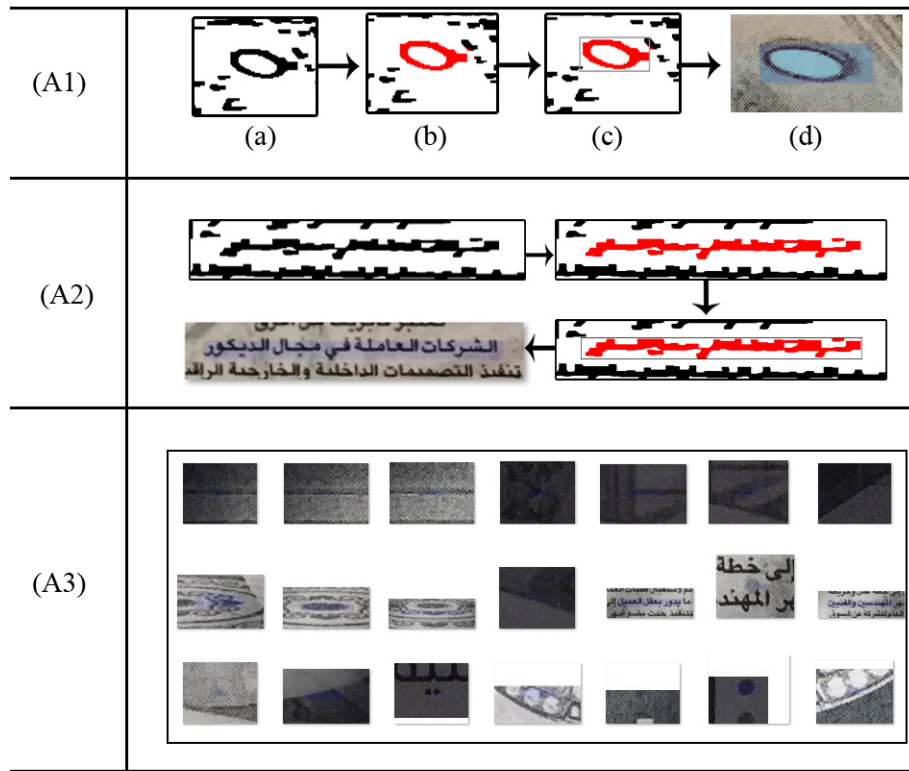


Figure 4.35: Examples of extracted patches in textual/graphical status: (A1) Patch of figure; (A2) Patch of text; (A3) Random selection of patches extracted from the previous figure; (a) Part of a resultant image from the previous step; (b) CC selection; (c) Bounding-box of CC; (d) The context image.

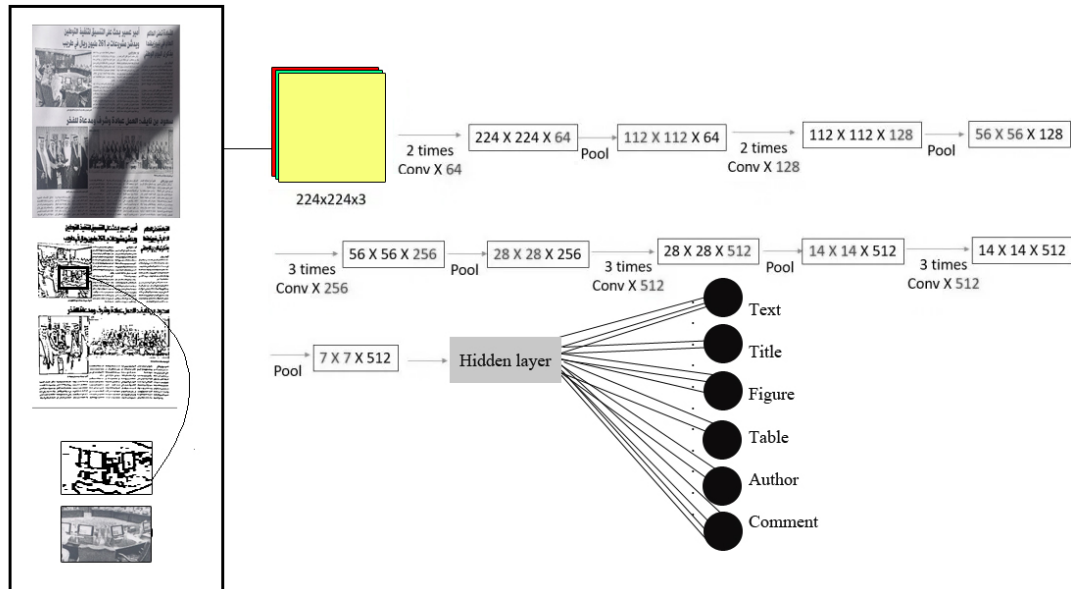


Figure 4.36: The proposed network architecture for patch classification using VGG-16 architecture [82].

As shown in Figure 4.36, the input to the convolutional layer has a fixed size of 224 x 224 RGB image (patch), using 3x3 with a stride of $s=1$ in convolutional layers stacked on top of each other in increasing depth. It is followed by max-pooling 2x2 with a stride of $s=2$ decreasing height and width. Overall, VGG 16 consisted of 13 convolutional layers, five max pooling and three fully connected layers, leading to a 7x7x512 matrix output.

Although VGG-16 predicts 1,000 classes of images, we needed only six (text, title, figure, table, author, comment). These six classes were not included in ImageNet, so we needed to modify the output model (freezing all the other layers), and train 24,576 + 6 bias parameters (6 classes) out of 138 million (1,000 classes) [82] from the last layer for new training by adding a new prediction layer.



Figure 4.37: Example of CNN results.

Figure 4.37 shows an example of extracting the elements from the patches using CNN, where each element was defined by its type such as text, figure and title in red color.

4.3.6 Page Segmentation

After the CCs were individually classified (see Figure 4.37), we combined all the CCs that were treated as a part of the same class by merging the closest CCs that had the same properties. Using geometric features such as height, width, aspect ratio, convex hull, solidity, and area, each detected CC was compared with the nearest CCs. If they had a high similarity score of CNN and geometric features, then they were merged. We repeated this step until no CC had a neighbor with the same properties.

Now, because every zone had been classified properly, we could extract the articles.

We know that every article has a title and this title is always at the top of this article (top-right/top-left/top-center), and each title has its own text. Using this information, we applied a vertical projection, taking into account the direction of title zone coordinates, so the skewed articles were detected correctly for each article-zone rectangle, which started from the title bounding-box covering the article-text to the next title bounding-box. If there was a title without text, it was connected to the title below.

Next, taking into consideration the article direction, we segmented the article-text into blocks using the VPP and HPP analysis. Then we segmented each block into lines using the same rules that had already been used in a previous study (see Section 4.1).

Figure 4.38 shows an example of label extraction results, where each label is highlighted in a different color.

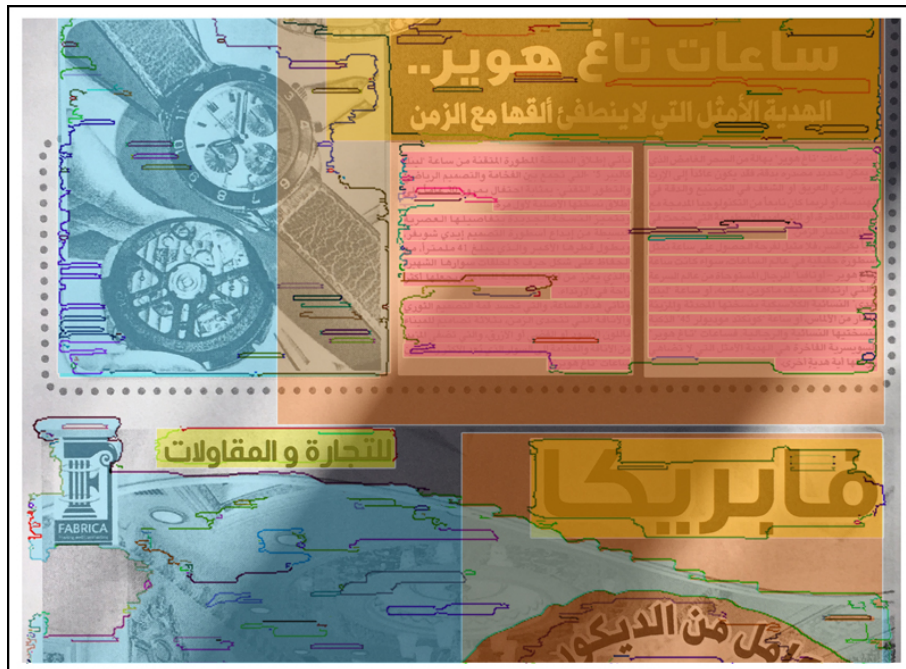


Figure 4.38: Example of label extraction.

4.3.7 Results and Discussion

To validate our method, all the algorithms used were implemented and the experiments were performed using JAVA and a computer system that had 14 GB RAM, INTEL (R) Xeon(R) CPU E550 @ 2.53 GHz with a Windows 7 operating system, using the Deeplearning4J (DL4J) framework and VGG-16 model.

The training set used 203,000 samples extracted from 1000 newspaper pages and 58,000 samples extracted from 400 newspaper pages were used for testing and validation.

The training time of the CNN took nearly six hours. The system was evaluated on

120 smartphone-captured document images taken from ten Arabic newspapers (see Chapter 3). The performance was measured based on a patch-based classification. The total number of patch classifications in 120 smartphone-captured document images was 8,523 patches; where 5,540 patches were text, 68 patches were tables and 2,915 patches were figures. The labels extracted by the CNN method are presented in Table 4.6.

Table 4.6: The performance of the proposed method on different newspapers.

LABEL/ Newspaper	Text	Title	Legend	Author	Figure	Table
Akhersaa	0.901	0.997	0.981	0.972	0.985	0.993
Alsharek	0.941	0.963	0.963	0.963	0.898	/
Alhadaf	0.759	0.780	0.905	0.916	0.732	/
Alnahar	0.915	0.921	0.969	0.945	0.903	/
Alshourouk	0.891	0.923	0.972	0.961	0.900	/
Alshorouk almisri	0.864	0.841	0.952	0.948	0.839	0.941
Alriad	0.853	0.889	0.971	0.973	0.845	0.959
Alsharek al-Awsat	0.851	0.894	0.958	0.952	0.832	/
Aswaq Qatar	0.798	0.761	0.915	0.909	0.747	0.967
Alayam	0.864	0.882	0.952	0.948	0.849	0.965
Total	0.864	0.882	0.952	0.948	0.849	0.965

The average classification success rate of our method on the 120 test images was 92.06% including the percentage of (text, title, figure, legend, comment, author, article, block, text-line, straight horizontal/vertical line) label extraction. This is a good result, considering that we have ten labels under many distorted conditions compared to previous studies that had perfect conditions.

Table 4.7: A performance comparison with different approaches.

Algorithms	Tested on	Used method	Recognition rate
Previous method (see Section 3.1)	55 images from one PDF newspaper	Mixed method	91.90%
Ibrahim et al	40 images collected from three PDF newspapers	CNN based on zone classification	74.50% (2 classes)
		CNN based on patch classification	91.00% (2 classes)
Proposed method	120 images collected from ten smartphone-captured newspapers	CNN based on patch classification/geometric features	92.06%

Here, our results (see Table 4.7) cannot be directly compared to the previous studies because, to the best of our knowledge, none of the methods classified these ten labels and because the test set used was not the same as that in the related papers (we used ten different newspapers). However, we should mention that some

of our test images were extracted from the same newspapers (Alshourouk, Alsharek, and Alriad) that were used in the previous studies [59], hence the comparison does give an indication of how well our method actually works.

Like any other program, ours has some issues in some steps, which are listed below:

- In the ARLSA step, the figure-CC and text-CC have been merged;
- In a patch classifying step, the author label is treated as a legend;
- In the patch collecting step, two horizontal figures are treated as one.

4.3.8 Summary

Here, we presented a simple and flexible machine learning-based method for an Arabic DLA in smartphone-captured conditions. We offered new ideas for ARLSA implementation and showed that a pre-trained model can be very effective for the extraction of new classes (document labels). Several tests were conducted to evaluate the performance of our system and the results we got are competitive.

Chapter 5

Arabic Handwritten Word Detection

This chapter describes the system architecture of the proposed Arabic handwritten text line segmentation (word spotting) procedure in detail with the proposed algorithms and the various steps. After the Introduction and related work in Section 1, in Section 2, we outline the methodology used to segment words. In Section 3, we present the experimental results, then in Section 4, we provide our summary.

5.1 Introduction

Arabic word spotting is a key step for Arabic NLP and the text recognition task. Many recent studies have addressed segmentation problems in the Arabic language. However, many issues still have to be overcome due to the virtually unlimited variety of handwritten styles.

In this chapter, we present a new approach for segmenting the image of an Arabic text into its individual words. Our approach consists of two main steps.

1. In the first step, a set of features is extracted from connected components using the Run-length smoothing algorithm (RLSA).
2. In the second step, spatially close connected components that are likely to belong to the same word component are grouped together. This is done via a learning technique called the self-organizing feature map (Kohonen map).

We evaluated our approach on 300 line images with different sizes and fonts for handwritten text using AHDB. Our results suggest that our approach efficiently segments lines.

Several methods of the text segmentation phase have been resolved and reported in the literature. The following papers describe approaches that were developed for the process;

- Many previous studies on word spotting focused on CCs (connected components) by extracting the distances between adjacent CCs using a metric

distance such as the Euclidean distance, the bounding box distance or the convex hull metric [51, 149, 150], and then classifying this distance to determine whether they are inter-word or inter-character gaps.

- In [7], Belabiod et al. proposed a method using a CNN (Convolutional Neural Network) [114] to extract the input features, then they applied a BLSTM (bidirectional Long Short Term Memory), which was followed by a CTC function (Connectionist Temporal Classification) [135] where the CTC decoder output was a sequence of "word-spaces". This method that was tested on the KHATT Arabic database achieved a word segmentation rate of 80.1%.
- Al-Dmour et al. [2] calculated the CC length and the distance between them. The lengths are used to classify the CCs into a words or sub-words where the purpose of a metric distance is to decide whether to classify them as separation gaps or not. Lengths and gaps are then clustered to identify an optimal threshold for word distance and to distinguish between "between-words" or "within-words". This method was tested on the AHDB dataset [124] and it attained a spotting rate of 86.3%.

5.2 Method Overview

As we mentioned in the Introduction (see Chapter 1), many letters are not joined to the adjacent letter, even in the middle of the word. Each letter has up to four distinct forms, based on its position (beginning, middle, end, or isolated) within or between the word. The figure below shows a sample of numerous letter positions in Arabic text along with their shapes.









Isolated	Beginning	Middle	End
			
			

Figure 5.1: Handwritten Arabic letters.

Figure 5.2 shows that a single word can contain one or more spaces and have many semi-words or sub-words.

The spotting word method is outlined in Figure 5.3. The input of the schema is a handwritten Arabic text image and the output is its segmentation result represented by extracted words. The sections later on will explain how the proposed method works.

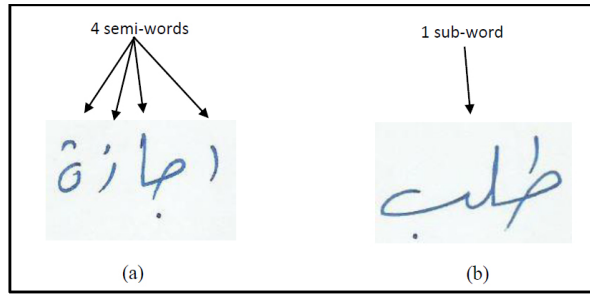


Figure 5.2: Example of a semi-word constituting an Arabic sub-word: (a) 4 semi-words; (b) 1 sub-word [97].

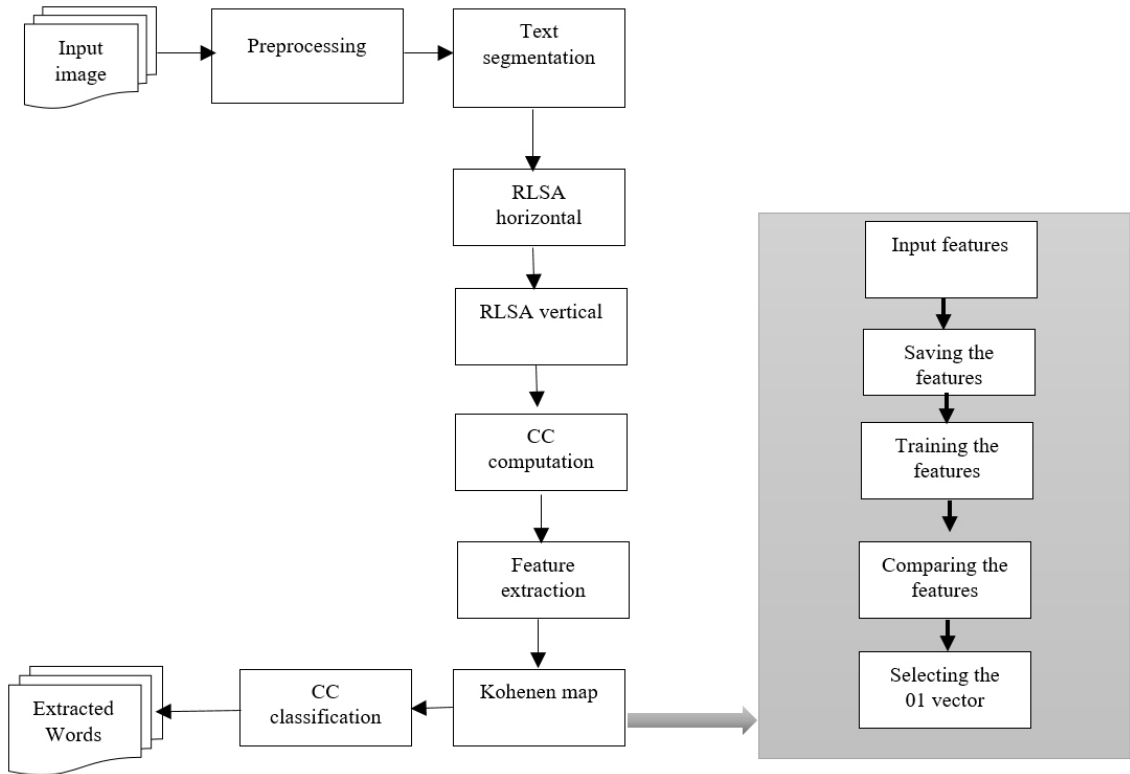


Figure 5.3: Outline of the method proposed for word extraction.

5.3 Pre-processing

The pre-processing step consists of two parts (see Figure 5.4):

1. In the first part, images must be binarized to enhance the image for better performance. For this, we used the well-known Otsu method.
2. In the second part, we remove noise using a median filter to delete the very small items (noise) caused by the acquisition process with a scanner.

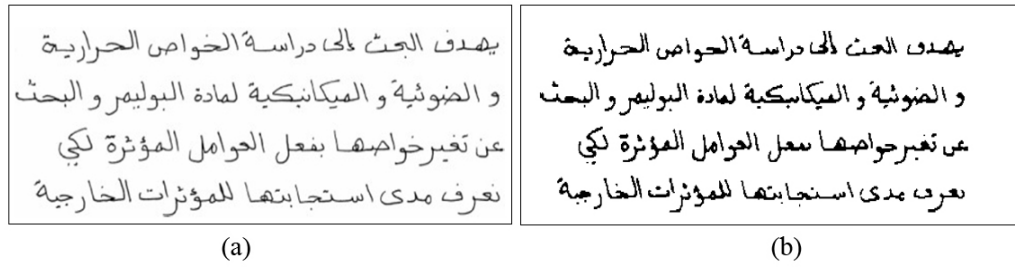


Figure 5.4: Example of a preprocessing step result: (a) The original image; (b) After preprocessing.

5.4 Segmentation

The dataset images that have been used contain several lines, and for this we used the earlier approach (see Section 4.1) for line segmentation which is based on an application of a horizontal projection, local minima and conflict resolution.

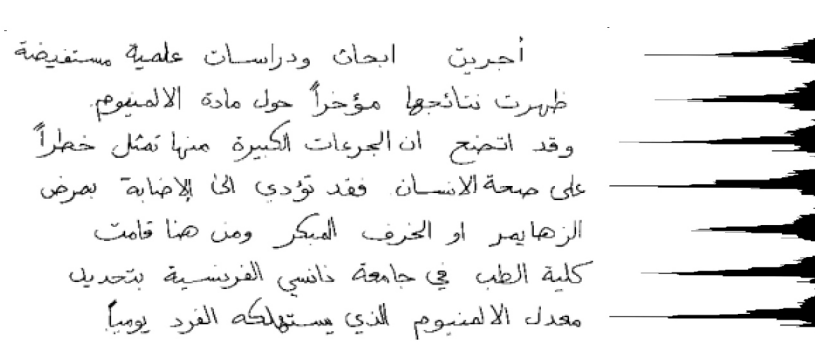


Figure 5.5: Example of projection histogram.

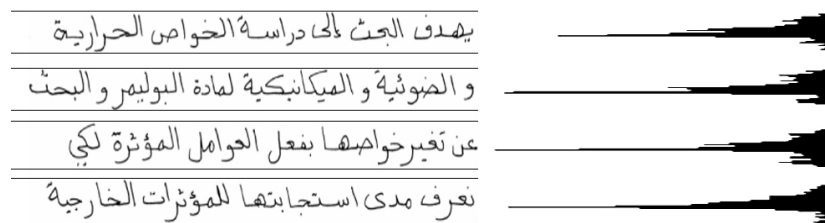


Figure 5.6: Example of text line segmentation result.

As shown in Figure 5.6, the segmentation result was efficient for handwritten text, because the lines were well spaced; and the skew does not affect the process because the lines are sufficiently straight horizontally.

5.4.1 Smoothing Technique

As a result, all the images become just one line, and we will apply smearing technique along with CC labeling on each line image:

- Every resultant image must be normalized to a height equal to 45 x the proportional width of to the height value for applying a fixed threshold in the later steps.
- For CC (connect component) extraction, we used the RLSA [90] algorithm beforehand in order to minimize and reduce the number of CCs such as letter dots and diacritics where they exist because it will enhance the results of the later steps. So rather than having two CCs cover one letter with dots, these will be just one CC (see Figure 5.7).
- Starting from the HRLSA with threshold=1 to vertical RLSA with threshold=30, these thresholds were chosen by testing many values on some 50 images. As a result, we found these values suitable for most normalized Arabic text images.

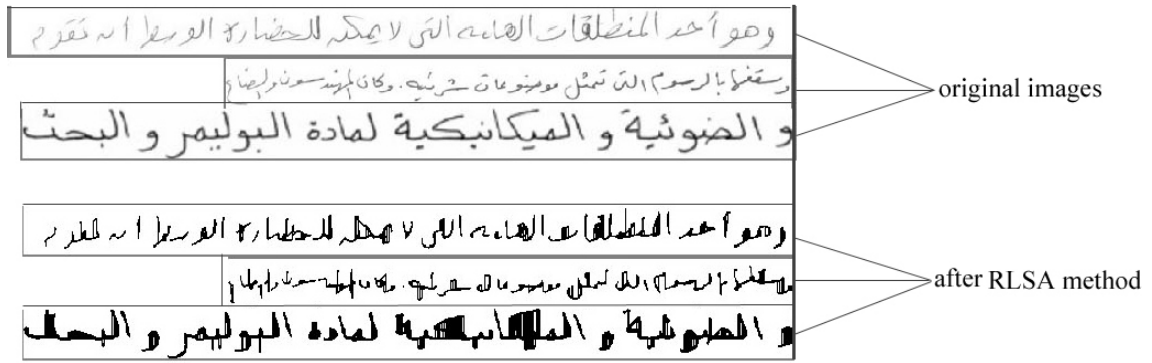


Figure 5.7: Text image after RLSA method.

5.4.2 Feature Extraction

Now, the connected components must be computed and the features extracted. Each CC represents every region that exists in the image, which may be a dot, a letter, word or half word.

To classify these connected components of the Arabic text, we extracted six features from each CC. Then, we assigned geometric information of the CCs to its corresponding position in the text. Here, we will use the following geometric features related to the size and shape of the CCs:

- Height: The height of the connected component bounding box;
- Width: The width of the connected component bounding box;
- Aspect Ratio: The width divided by the height;
- Distance: the gap measure between an adjacent connected component;

- Area: The number of pixels in the connected components;
- Position: Horizontal CC coordinates using the bounding box, taking the center value between the left and right box.

5.4.3 Kohonen Map Learning

Connected component features are then used as the input for SOM clustering [145]. The features of the connected components are fed into the input neurons, where;

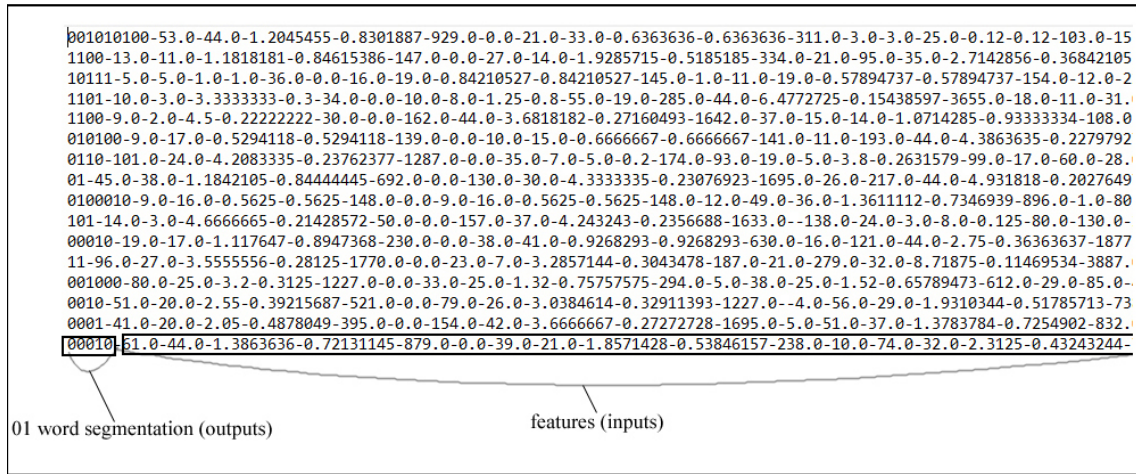


Figure 5.8: Training data file sample

- The number of inputs represent the multiplication product between the features and CCs. Since the total number of features is six for each CC, the number of inputs is $6 \times \text{CC numbers}$ (one input for each feature for every CC).
- The number of output neurons corresponds to the number of possible ways that 01 can be represented in the text with segments of 0 for inter-word and 1 for between-words (see Figure 5.8). Hence if the text contains six words, the output total will be 2^5 neurons (which equals the number of possibilities of the given segmentation vector (01)).
- The total number of connections (weights) is related to the number of input and output neurons. For instance, if we wish to segment an image containing five words (and divide into seven CCs), the features will be 42 (inputs), while in the competition layer there will be $2^4 = 16$ neurons; so the total number of connections will be $42 \times 16 = 672$ connect weights.

The training and learning processes contain four steps :

- Calculate the number of input neurons/output neurons;

- Construct a training set that has the same number of samples as the input number.
- Copy the input features to the training set; this is repeated for all the samples of the input patterns.
- Calculate the total number of winning neurons; (the winner neuron will be selected); If there is no satisfactory number of winners, one neuron is declared the winner.

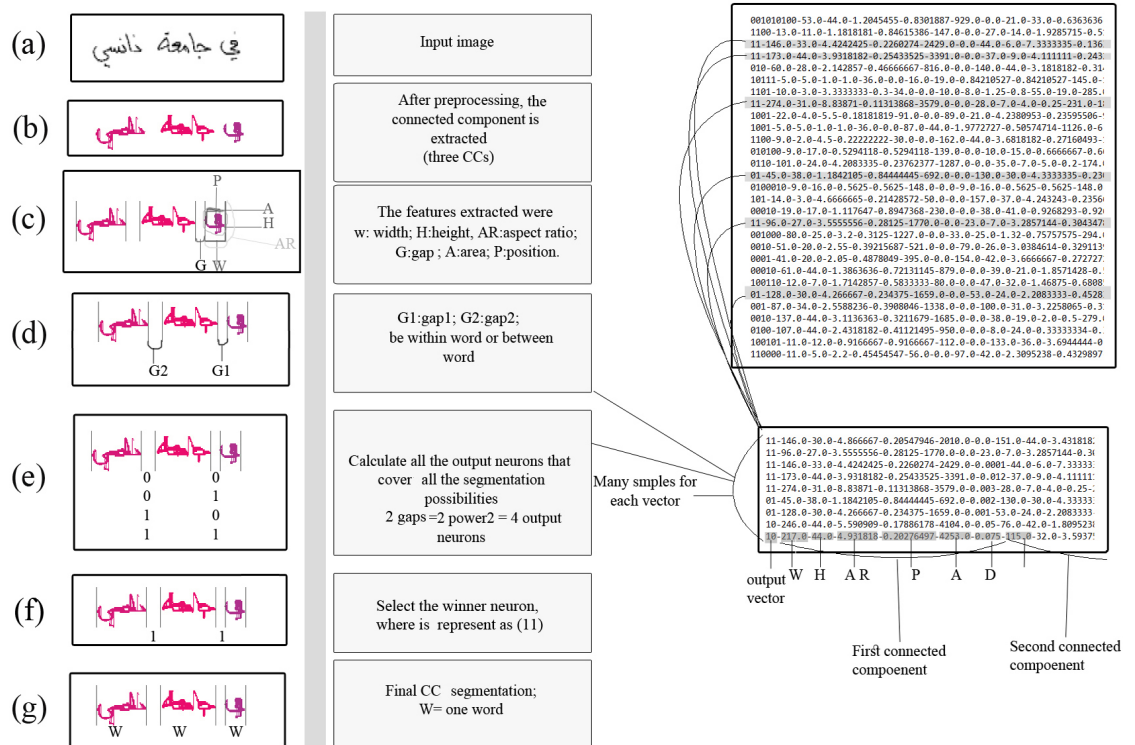


Figure 5.9: Example of the proposed method: (a) Original image; (b) Connected component labeling; (c) Feature extraction from every connected component; (d) We calculate the number of the gaps that exist between the CC; (e) The number of possible gap-words/gap-CCs between the CC are shown; (f) We select the best match result according to the feature data; (g) We segment the CC based on the SOM result.

As shown in the figure above, the selection from the datafile depends on the number of gaps that exist between the CCs, where the winning vector is the closest one to the test sample. What is more, the vertical RLISA can enhance the segmentation process because it linked the CCs that are separated from each other. However, in some cases, it is not suitable because sometimes writers or authors overlap their letters when they intend to write two isolated words.

The successive neuron can be found by using a Mexican hat function [34] that describes synaptic weights between neurons in the Kohonen layers (input and output layers). The competitive learning rule defines the change Δw_{ij} applied to the synaptic weight w_{ij} , where x_i is the input signal and α is the learning rate parameter.

$$\Delta w_{ij} = \begin{cases} \alpha (x_i - w_{ij}), & \text{if neuron } j \text{ wins the competition} \\ 0, & \text{if neuron } j \text{ loses the competition} \end{cases}$$

However, it does not give us the vector, but the winning neuron index number. For this, the Kohonen Map neuron approach is used to find the vector that matches the winning neuron, which will return a matrix of vectors. The index of each matrix-vector corresponds to the neuron-index number that recognizes the correct word extraction vector by computing the minimum Euclidean distance between any two vectors.

The Euclidean distance is a pair of input vectors X and weight vector W_j where x_i and w_{ij} are the i^{th} elements of the vectors X and W_j , respectively. It is defined as follows:

$$d = \|\mathbf{X} - \mathbf{W}_j\| = \left[\sum_{i=1}^n (x_i - w_{ij})^2 \right]^{1/2}$$

A Kohonen neural network learns by evaluating and optimizing a weight matrix in a regular way. Beginning with an initial random weight matrix, the training will start and evaluate the weight matrix to find the error estimate. If it is under 10% the training procedure terminates; otherwise, it will go on optimizing until the weight falls within the desired bound (see Figure 5.10).

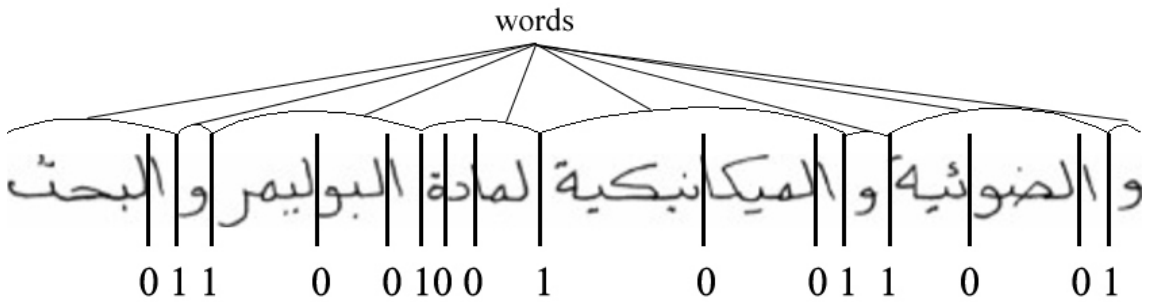


Figure 5.10: A word segmenation example (1: between-word; 0: inter-word)

5.5 Experimental Results

The proposed method was implemented in Eclipse Java oxygen.2 using the Java programming language. Several experiments were conducted to verify the performance

of the proposed method.

5.5.1 Database

We tested our approach on 300 line images extracted from the AHDB database using 35 different handwritten Arabic documents.

5.5.2 Training and Error Analysis

For mapping the CC to a word or sub-word, we used 7356 training line images, and these images were converted to CCs, where each line image contained multiple CCs (the CC number depends on the line length and the writing style).

A long text means numerous of connected components must be covered by thousands of training images to cover all the possibilities with many versions (01) to give good results. For this, we reduce the number of training image combinations for the long Arabic texts, where they exist, by checking the number of connected components. If there are more than nine CCs, the text will be divided automatically based on the maximum extracted space between the connected components examined.

Therefore each long line image that exceeds the threshold (nine CCs) will be divided into two line images, and so on. We recalculated the number of line images after division, so the number will be 10,245 line images (containing fewer than nine CCs) rather than 7356 training images (containing more than nine CCs).

In the figure below, we have plotted the distribution for the Arabic text image proportion along with the number of connected components.

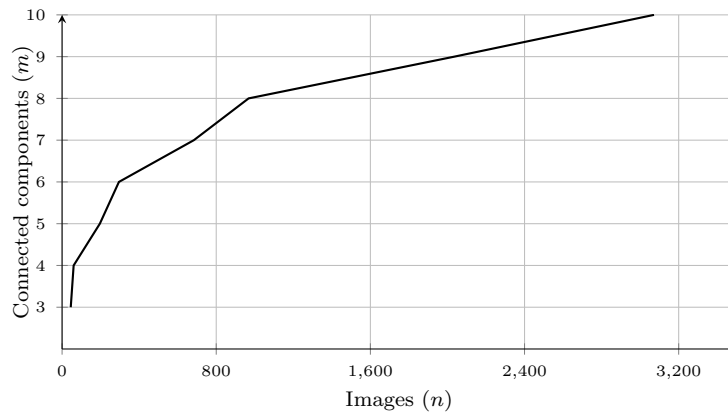


Figure 5.11: The number of CCs as a function of training images.

The evaluation process was performed using two types of error, namely the spacing/ overlapping errors and the validity method.

Spacing and Overlapping

Most Arabic texts in handwritten form typically have a spacing or overlapping problem. The spaces between adjacent words are present at random so there are no fixed thresholds that can be determined, and an overlapping problem occurs when two letters or characters have zero or less space between them, and they are usually regarded as belonging to different words, not the same word (see Figure 5.12).

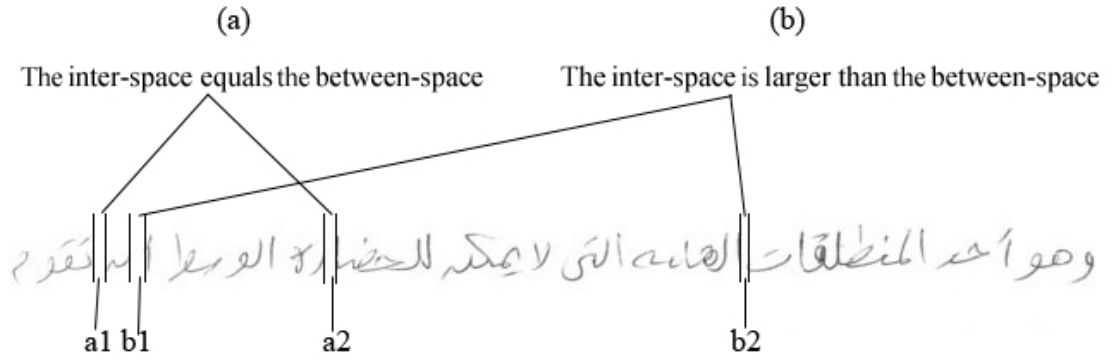


Figure 5.12: Example of a randomly spacing in an Arabic text image: (a) The gaps are equal to each other though the a1 is a separator space between two words, while a2 is a space between two letters from the same word; (b) The gap-word is smaller the gap-letter, where b1 is a gap between two letters from the same word, while b2 is a separator space between two words.

Method Validity

The proposed algorithm was applied with a Kohonen map. With this method, we partially solved the spacing and overlapping problems with a training data file that contains connected component features extracted from a wide variety of text word positions and writing shapes.

However, it will not be effective when the two words have no space between each other either in horizontal or vertical segmentation and it will be treated as one word (see Figure 5.13).

5.5.3 Results

To evaluate our proposed method, we used several criteria for measuring segmentation accuracy in percentage terms. The constraints for error detection are:

- 1) If two words are treated as one word;
- 2) If one word is treated as two words;
- 3) If a part-word and word are treated as two words;
- 4) If a part-word is treated as one word.

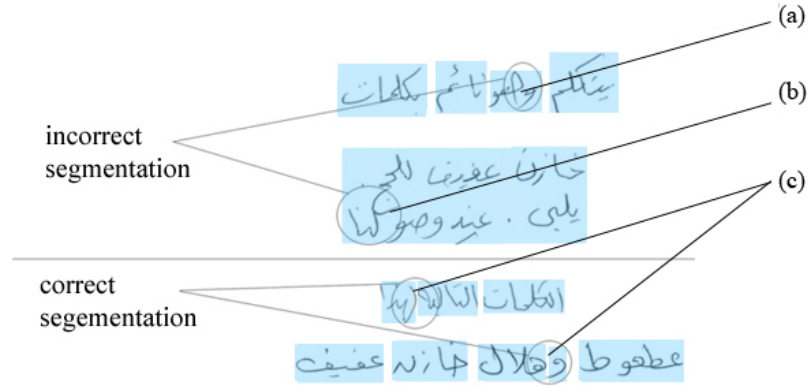


Figure 5.13: Examples of overlapping problems and good segmentation in Arabic text: (a) The letter (و) and word (هوا) are treated as one word; (b) The word above is connected to the word below so the line/word segmentation is ineffective; (c) The words were segmented correctly even when they were very close to each other.

We used the following formula for computing all the errors:

$$Accuracy\% = 100 - (E/Totalnumberwords) * 100$$

where E = the word segmentation error.

Some results obtained using this method are listed in Table 5.1. We processed 300 line images for test purposes, which attained a score of 87.54%. The best results listed had a correct word extraction rate of 100%.

Table 5.1: Test results.

Image No.	No.words in the line	No.Wrong Seg. words	Correct Seg. rate
1	14	4	71.42 %
2	15	1	93.34 %
3	14	0	100.00 %
4	19	5	73.48 %
5	13	0	100.00 %
6	20	4	80.00 %
7	18	2	88.89 %
.	.	.	.
.	.	.	.
300	17	1	94.11 %
Total	5400	672	87.54%

In Table 5.2, our results were presented along with results from other researchers working on handwritten Arabic texts, which were obtained from previously published studies using the same dataset (AHDB), and other databases. We notice that our results are quite good compared to those of the previous methods described in

the literature, but due to the diversity of data from one study to another (e.g. some of the databases are skewed), we cannot guarantee that this method will be suitable for every dataset.

Table 5.2: Comparison with other approaches.

Method	Measures	Tested on	Based on	Correct rate
Jawad et al. [65]	Within-word and between-word gaps	200 images extracted from the IFN/ENIT database	Bayesian criteria	85 %
Ayman et al. [2]	Within-word and between-word gaps	25 images extracted from the AHDB database	Kmeans	84.8 %
Al-Dmour et al. [3]	Within-word and between-word gaps + CC lengths	35 images extracted from the AHDB database	Kohonen Neural Network	86.3 %
Belabiod et al. [7]	batch features	200 line images extracted from the KHATT database	CNN+ BLSTM+ CTC	80.1 %
Proposed method	CC geometric features	300 line images extracted from the AHDB database	Kohonen Neural Network	87.54 %

5.6 Summary

In this chapter, a new robust and flexible approach based on a machine learning technique was proposed for word spotting. The segmentation system began by preprocessing a text using Otsu binarization and the Median filter, then we applied vertical and horizontal RLSA to eliminate dots and diacritics, so as to compute connected components that are then used as building blocks. This latter is used for extracting the most important features that will provide the input for the Kohonen map. To evaluate the potential of the method, we used 300 line images extracted from the AHDB database. According Table 5.2, our method is competitive.

Summary

In this dissertation, we handled many issues of image analysis which is including: PDF newspaper page recognition based on hybrid approach (bottom-up and top-down), title extraction using the analysis of projection profiles, smart-phone captured newspaper analysis based on deep learning and handwriting word segmentation by applying a neural network. The smartphone-captured images used were extracted from the Printed Arabic Text Database (PATD), and were created and collected by us using the PDF version.

The dissertation consists of four main parts, namely state-of-the-art techniques presented in Chapter 2, creating a printed Arabic database in Chapter 3, an Arabic newspaper pages analysis in Chapter 4, and Arabic word segmentation in Chapter 5.

State-of-the-art

In Chapter 2, we discussed recently published articles and gave general definitions concerning different methods and approaches.

We described all the methods that were used in this dissertation, including the preprocessing step which contains definitions of binarization techniques, followed by smoothing and noise reduction approaches. Then the neural network and deep learning was given by presenting an interpretation of two types of neural network, called the self-organizing map and convolutional neural networks. After, we furnished a general view of the document layout analysis, outlined the popular approaches used in related Arabic studies, and the results of our analysis. Then we concluded by presenting a review of the different types of approaches used for line/word segmentation.

Creating a printed Arabic database

In Chapter 3, an Arabic database was presented for analysis and recognition purposes that covers the Arabic magazines and newspapers.

The advantage of having such a text database is that it allows one to make a reasonable comparison between old and new results because the evaluation of studies published previously were based on individual selections and they were chosen

in a random way from newspaper/magazine websites, which makes these studies hard to compare with each other, and this leads to inconclusive comparisons. What makes this database useful is that it covers complex structures not simple ones, which means that many figures with different shapes, overlapping situations between different elements (e.g figure with title/text/legend, different types of font/figure/structure/author/legend, different sizes of text/title/figure) can be processed.

The images were obtained via scanner/smartphone-cameras, where each capturing was performed twice in one day (daylight/night), under two conditions (shaded/non-shaded), and where there were two different light sources (artificial-light/sun-light). The database contains two types of images (blurred/non-blurred) that are transformed into XML files, which contained all the required information for the Arabic page elements. This database contains 3764 images including captured/scanned versions.

An Arabic newspaper page analysis

Here, we seek to improve the recognition of Arabic newspaper pages. The purpose is to understand the nature of the document layouts that are organized hierarchically, as well as the relationships among these layouts. Furthermore, the analysis-of-document-structures is considered as a key step in many document recognition-and-understanding applications such as indexing, searching and automatic classification of documents.

This chapter presents many methods that are used to transform a raw image of a journal page into a set of exploitable structured data.

To achieve this goal, we made three programs that are represented in three sections. In the first section, we extracted the physical and logical structures from the documents. The extraction of the physical structure allows the separation between articles, blocks, text columns, lines, etc. So, for this purpose we used a mixed-method; we performed upward segmentation to label the various connected components of the page. These were then grouped and used to separate text and graphics. Next, we followed a descending segmentation (via horizontal and vertical projections) to extract the articles, the blocks and the lines of text. However, the extraction of the logical structure from a log page is carried out in order to understand the hierarchical organization of its elements and to create a navigation interface inside the log page by listing all the elements (titles, columns, figures, authors, legends, footer, etc.). The extraction of the logical structure is done by labeling the different extracted physical elements. This labeling is mainly based on certain rules of the size and position of the page elements and also on a priori knowledge of certain properties of logical entities (titles, figure, author, legend ... etc.). Several tests were conducted to assess the performance of the developed system and the results we obtained are encouraging.

In the future, we intend to adapt the processing steps based on the characteristics and organization of the document; Develop the proposed system so that it is generic and applicable to other newspapers, not just Echorouk; Include other pre-processing modules, in particular, to allow the analysis and recognition of images from the scanning of newspaper pages by a scanner or camera; Improve the figure extraction step so that it can be applied to all forms of figures in the newspaper; Introduce other approaches that attempt to extract text from figures/tables and improve the article extraction step so that it is efficient even there are overlap situations between articles (a small article inside a larger one).

These points encouraged us to develop new methods that can overcome these challenges, and this is why new applicants were developed. In the next stage we will focus on improving the title recognition, then we will try to solve most of the problems that currently exist using deep learning techniques.

In Section 4.2, the proposed title/subtitle extraction method was presented. The main research contribution of this extraction was the detection of title and subtitle without requiring an analysis of all the elements that exist on the page. In addition, the proposed method can improve the article detection step, hence improve the document analysis step. The method is based on a combination of connected component labeling and projection profiles analysis using RLSA, a minimum bounding-box, and geometrical features. It was tested on scanned Arabic newspaper/magazine pages (complex structure) extracted from the PATD database (see Section 3). Our results indicate that the proposed method is efficient for many fonts and sizes where there are no skew/blur/overlapping obstacles. In our future plans, we intend to improve the title/subtitle detection for document images that have overlapping/ distortion problems.

In Section 4.3, we attempted to solve the problems that were tackled in earliest studies by utilizing the deep learning technique, where many of the Arabic DLA obstacles were solved or partially solved. With this method all structure types, and figure shapes can be covered, and many of the distortion problems were handled. This method is based on a convolutional neural network for each extracted patch, using sharpness/smoothing filters, adaptive thresholding techniques, morphological operations, along with a connected component labeling, ARLSA, projection profile analysis, and geometric features. Using our system, ten labels were extracted (text, table, figure, title, legend, author, straight-line, text-line, block, article) from the smart-phone captured Arabic document images. The images we tested were obtained from the PATD database (see Chapter 3), and the results look quite promising.

In the future, we would like to create an automatic system that can extract not just the structures and labels, but also the content itself for indexing and searching purposes. Hence the exploration could be performed by article-title, author or legend text for extracting the required information without needing to use any manual

archiving approach.

Arabic word segmentation

In Chapter 5, the Arabic word detection method was presented. With this proposed method, we improved some of the results that have been published. Our method based on a self-organizing feature map (Kohonen map), using connected component labeling and the Run-length smoothing algorithm. The evaluation was made using the AHDB database, where the images of this database contain a paragraph, and because our method was based on lines not paragraphs, we first segment each image into lines using the method described in Section 4.1.

In our experiments, we sought to demonstrate the efficiency of our method. We selected a dataset that includes several types of paragraph, with different authors and scripts. We showed that our method is able to handle most of the situations regarding the irregular spaces between words. The results obtained support our initial hypothesis and they are definitely better than those of the related studies that were evaluated on the same dataset.

In the future, we would like to extend this method so that it covers touching words, and extract the words from images that have many strange shapes and lines along with actual words that were hastily written or scribbled.

Publications of the Author

Publications Accepted by the PhD School in Computer Science

- (1) H. Bouressace and J.Csirik, Printed Arabic Text Database for Automatic Recognition Systems, 5th International Conference on Computer and Technology Applications, pp.107-111, 2019.
- (2) H. Bouressace and J.Csirik, Recognition of the logical structure of Arabic newspaper, 21st International Conference on Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence (Springer), Vol.11107, No.3, pp.251-258, 2018.
- (3) Bouressace Hassina, Title Segmentation in Arabic Document Pages, Journal of WSCG, pp.45-50, 2019.
- (4) H. Bouressace and J.Csirik, A Convolutional Neural Network for Arabic Document Analysis. IEEE 18th International Symposium on Signal Processing and Information Technology (ISSPIT), pp.1-6, 2019.
- (5) H. Bouressace and J.Csirik, A Self-Organizing Feature Map for Arabic Word Extraction. 22nd International Conference on Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence (Springer), Vol.11697, pp.127-136, 2019.

Bibliography

- [1] A. AbdelRaouf, C.A. Higgins, and M. Khalil. A Database for Arabic Printed Character Recognition. In International Conference Image Analysis and Recognition, Springer, pp.567-578, 2008.
- [2] A. Al-Dmour and F. Fraij. Segmenting arabic handwritten documents into text lines and words. International Journal of Advancements in Computing Technology (IJACT), 6(3):109–119, 2014.
- [3] A. Al-Dmour and R.A. Zitar. Word extraction from Arabic handwritten documents based on statistical measures. International Review On Computer and Software (IRECOS) 11, 2016.
- [4] A. Alshameri, S. Abdou, and K. Mostafa. A Combined Algorithm for Layout Analysis of Arabic Document Images and Text Lines Extraction. International Journal of Computer Application, Vol.49, No.23, pp.30-37, 2012.
- [5] A. Antonacopoulos, C. Clausner, S. Papadopoulos, and S. Pletschacher. Historical Document Layout Analysis Competition. Proceedings of the 11th International Conference on Document Analysis and Recognition, pp.1516-1520, 2011.
- [6] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos. IC-DAR 2009 Page Segmentation Competition, 10th International Conference on Document Analysis and Recognition, University of Salford, pp.1370-1374, 2009.
- [7] A. Belabiod and A. Belaïd. Line and Word Segmentation of Arabic handwritten documents using Neural Networks, University of Lorraine, 2018.
- [8] A. Cheung, M. Bennamoun, and N.W. Bergmann. An Arabic optical character recognition system using recognition based segmentation. Pattern recognition, Vol.34, No.2, pp.215-233, 2001.
- [9] A. Husain, Z. Safdar, and A. Muhammad. A Self Organizing Map based Urdu Nasakh character recognition. pp.267-273, 2009.
- [10] A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet classification with deep convolutional neural networks, 2012.

- [11] A. Simon, J.C. Pret, and A.P. Johnson. A Fast Algorithm for Bottom-Up Document. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.19, No.3, pp. 273-277, 1997.
- [12] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane. Arabic Hand-written Text-line Extraction, in *Proceedings of the Sixth International. Conference on Document Analysis and Recognition, ICDAR*, pp.281–285, 2001.
- [13] A.G. AL-Hashim and S. Mahmoud. Printed Arabic Text Database (PATDB) for Research and Benchmarking. In: *proc. of 9th Wseas International Conference on Applications of Computer Engineering*, pp.62-68, 2010.
- [14] A.K. Jain, and B. Yu. Document Representation and Its Application to Page Decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence Journal*, Vol.20, pp.294–308, 1998.
- [15] A.K. Jain, N.K. Ratha, and S. Lakshmanan. Object detection using gabor filters, *Pattern Recognition*, Vol.30, No.2, pp.295-309, 1997.
- [16] A.M. Hesham, M.A. Rashwan, H. Al-Barhamtoshy, S.M. Abdou, A.A. Badr and I. Farag. Arabic document layout analysis. *Pattern Analysis and Applications*, pp.1-13, 2017.
- [17] A.M. Zeki, M.S. Zakaria, and C.-Y. Liong. Segmentation of Arabic characters: A comprehensive survey. *International Journal of Technology Diffusion*, Vol.2, No.4, pp.48-82, 2011.
- [18] A.S. Drigas and J. Vrettaros. Using the self-organizing map (SOM) algorithm, as a prototype e-content retrieval tool. In *Proceedings of the Eighth International Conference on Computer Science and Application, Lecture Notes in Computer Science*, Vol.5073, pp.14-230, 2008.
- [19] B. Gatos, I. Pratikakis, and I.J. Perantonis. Adaptive degraded document image binarization. *Pattern Recognition*, pp.17–327, 2006.
- [20] B. Gatos, S. Mantzarisl, and A. Antonacopoulos, First International Newspaper Segmentation Contest. *Proceedings of the 6th International Conference on Document Analysis and Recognition*, pp.1190-1194, 2001.
- [21] B. Su, S. Lu, and C.L. Tan. Binarization of historical document images using the local maximum and minimum. In *Proceedings of the International Workshop on Document Analysis Systems (DAS)*, pp.159–166, 2010.
- [22] B. Su, S. Lu, and C.L. Tan. Combination of Document Image Binarization Techniques. *International Conference on Document Analysis and Recognition*, pp.22-26, 2011.

- [23] B.H. ChandraShekar and G. Shoba. Classification of documents using Kohonen's self-organizing map. *International Journal of Computer Theory and Engineering*, Vol.1, No.5, pp.610-613, 2009.
- [24] B.T. Mitchell and A.M. Gillies. A model-based computer vision system for recognizing handwritten ZIP codes, *Machine Vision and Applications*, Vol.2, pp.231-243, 1989.
- [25] C. Huang and S. Srihari. Word segmentation of off-line handwritten documents. In: *Proceedings of the Document Recognition and Retrieval (DRR) XV, IST/SPIE Annual Symposium*, 2008.
- [26] C. Kai, Y. Fei, and L. Cheng-Lin. Hybrid Page Segmentation with Efficient Whitespace Rectangles Extraction and Grouping. *Proceedings of the International Conference on Document Analysis and Recognition*, pp.958-962, 2013.
- [27] C. Singh, N. Bhatia, and A. Kaur. Hough transform based fast skew detection and accurate skew correction methods. *Pattern Recognition* 41, No.12, pp.3528-3546, 2008.
- [28] C. Torres-Huitzil. Fast hardware architecture for grey level image morphology with flat structuring elements. *IET Image Processing*, Vol.8, No.2, pp.112-121, 2013.
- [29] C. Weliwitage, A. L. Harvey, and A. B. Jennings. Handwritten Document Offline Text Line Segmentation. In *Proceedings of Digital Imaging Computing: Techniques and Applications*, pp.184-187, 2005.
- [30] C. Wick and F. Puppe. Fully Convolutional Neural Networks for Page Segmentation of Historical Document Images, 2017.
- [31] C. Wolf, J.M. Jolion, and F. Chassaing. Text Localization, Enhancement and Binarization in Multimedia Documents. *International Conference on Pattern Recognition, (ICPR)*, pp.1037-1040, 2002.
- [32] D. Bradley and G. Roth. Adaptive thresholding using the integral image. *Journal of Graphics Tools* 12, No.2, pp.13-21, 2007.
- [33] D. Brodic and Z.N. Milivojevi. Text line segmentation with the algorithm based on the oriented anisotropic Gaussian kernel. *Journal of Electrical Engineering*, Vol.64, No.4, pp.238-243, 2013.
- [34] D. Miljković. Brief review of self-organizing maps, 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1061-1066, 2017.

- [35] D.S. Bloomberg and P. Maragos. Generalized Hit-Miss Operations, in SPIE Conf. on ImageAlgebra and Morphological Image Processing, Vol. 1350, pp. 116-128, 1990.
- [36] D.S. Bloomberg. Multiresolution morphological approach to document image analysis. In: Proceedings 1st International Conference on Document Analysis and Recognition, pp.963-971, 1991.
- [37] D.X. Le, G.R. Thoma, and H. Wechsler. Classification of binary document images into textual or nontextual data blocks using network models, Mach. Vision Appl, Vol.8, pp.289-304, 1995.
- [38] E. Mustafa. Page layout analysis and classification in complex scanned documents. Thesis, Rochester Institute of Technology, 2011.
- [39] F. Chabchoub, Y. Kessentini, S. Kanoun, and V. Eglin. SmartATID: A mobile captured Arabic Text Images Dataset for multi-purpose recognition tasks. International Conference in Frontiers on Handwriting Recognition, 2016.
- [40] F. Kleber. Document Image Analysis Preprocessing of Low-Quality and Sparsely Inscribed Documents. Technical Report, Vienna University of Technology, 2014.
- [41] F. Liu, Y. Luo, M. Yoshikawa, and D. Hu. A New Component based Algorithm for Newspaper Layout Analysis, Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR), IEEE Computer Society, pp.1176-1179, 2001.
- [42] F. Montreuil. Extraction de structures de documents par champs aléatoires conditionnels: application aux traitements des courriers manuscrits. PHD thesis, Rouen university, 2011.
- [43] F. Shafait and T. M. Breuel. Document image dewarping contest. In 2nd International Workshop on Camera-Based Document Analysis and Recognition, 2007.
- [44] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.941-954, 2008.
- [45] F. Slimane, R. Ingold, S. Kanoun, A.M. Alimi, and J. Hennebert. A New Arabic Printed Text Image Database and Evaluation Protocols. In: proc. of 10th International Conference on Document Analysis and Recognition, ICDAR, pp.946-950, 2009.

- [46] F.K. Jaiem, S. Kanoun, M. Khemakhem, I. El Abed, and J. Kardoun. Database for Arabic Printed Text Recognition Research. ICIAP, Part I, LNCS 8156, pp.251–259, 2013.
- [47] G. Kim and V. Govindaraju. Handwritten phrase recognition as applied to street name images, *Pattern Recognition* 31, pp.41–51, 1998.
- [48] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis. Text line and word segmentation of handwritten documents. *Pattern Recognition* 42, pp.3169–3183, 2009.
- [49] G. Louloudis, B. Gatos, I. Pratikakis, and K. Halatsis. A Block-Based Hough Transform Mapping for Text Line Detection in Handwritten Documents. *Proceedings of the Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.
- [50] G. Nagy, S. Seth, Hierarchical representation of optically scanned documents. *17th Conference on Pattern Recognition*, pp.347–349, 1984.
- [51] G. Seni and E. Cohen. External Word Segmentation of Offline Handwritten Text Lines. *Pattern Recognition*, 27(1), pp.41–52, 1994.
- [52] H. Boufersaoui and I. Frihi. Extraction of the logical structure of documents, Master’s Thesis of Media Engineering, University May 08, 1945-Guelma, 2015.
- [53] <https://alriadia.online/>
- [54] <https://www.echoroukonline.com/newspaper/echorouk-yawmi/>
- [55] I. Abuhaiba. Segmentation of discrete Arabic script document images, *Al Azhar University*, Vol.8, No.1810-6366, pp.85–108, 2006.
- [56] I. Aljarrah, O. Al-Khaleel, K. Mhaidat, M. Alrefai, A. Alzu bi, and M. Rababah. Automated system for Arabic optical character recognition with lookup dictionary. *Journal of Emerging Technologies in Web Intelligence*, Vol.4, No.4, pp.362–370, 2012.
- [57] I. Chtourou, A. Cheikh Rouhou, F. Jaiem, and S. Kanoun. ALTID: Arabic/Latin Text Images Database for recognition research, In *ICDAR*, pp.836–840, 2015.
- [58] I.B. Messaoud, H. Amiri, H.E. Abed, and V. Margner. Document preprocessing system-automatic selection of binarization. In *10th IAPR International Workshop on Document Analysis Systems (DAS)*, pp.85–89, 2012.
- [59] I.M. Amer, S. Hamdy, and M.G.M. Mostafa. Deep Arabic document layout analysis. In *Proceedings of the IEEE Eighth International Conference on Intelligent Computing and Information Systems*, pp.224–231, 2017.

- [60] I.S.I. Abuhaiba, S. Datta, and M.J.J. Holt. Line Extraction and Stroke Ordering of Text Pages. Proceedings of the Third International Conference on Document Analysis and Recognition, pp.390-393, 1995.
- [61] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. FeiFei. ImageNet Large Scale Visual Recognition Competition, 2012.
- [62] J. Duong. Étude des documents imprimés: Approche statistique et contribution méthodologique. PHD thesis, Claude Bernard University, 2005.
- [63] J. Fan, R. Wang, L. Zhang, D. Xing, and F. Gan. Image sequence segmentation based on 2d temporal entropic thresholding. Pattern Recognition Letters, 17(10), pp.1101–1107, 1996.
- [64] J. Fernandez, R.Mones, I. Diaz, J. Ranilla, and E.F. Combarro. Experiments with self organizing maps. In CLEF 2003, Lecture Notes in Computer Science, Vol.3237, pp.358-366, 2004.
- [65] J. H AlKhateeb, J. Jiang, J. Ren, and S. Ipson. Interactive knowledge discovery for baseline estimation and word segmentation in handwritten Arabic text. Recent Advances in Technologies, Maurizio A Strangio (Ed.), 2009.
- [66] J. Ha, L.T. Phillips, and R.M. Haralick. document page decomposition using bound boxes of connected components of black pixels. In Book, vol 2422, pp.140-151,1995.
- [67] J. Ha, R.M. Haralick, and I.T. Phillips. Recursive X-Y cut using bounding boxes of connected components. Proceedings of 3rd International Conference on Document Analysis and Recognition 2, Vol.2, pp. 952-955, 1995.
- [68] J. He, Q.D.M. Do, A.C. Downton, and J.H. Kim. A comparison of binarization methods for historical archive documents. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), pp.538-542, Vol.1, 2005.
- [69] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: a survey. International Journal of Document Analysis and Recognition, pp.84-104, 2005.
- [70] J. Liang, J. Ha, R. M. Haralick, and I. T. Phillips. Document Layout Structure Extraction Using Bounding Boxes of Different Entities. in Proc. of 3rd IEEE Workshop on Applications of Computer Vision, pp. 278-283, 1996.
- [71] J. Serra. Image Analysis and Mathematical Morphology, Academic Press, London, 1982.

- [72] J.H. Wang and L.D. Lin. Improved median filter using min-max algorithm for image processing. *Electron, Lett.* 33(16), pp.1362–1363, 1997.
- [73] J.J. Sauvola and M. Pietikinen. Adaptive document image binarization. *Pattern recognition*, No.2, pp.225-236, 2000.
- [74] J.L. Fisher, S.C. Hinds, and D.P. D’Amato. A rule-based system for document image segmentation. in *Pattern recognition, 10th Internatinal Conference*, Vol.1, pp.567-572, 1990.
- [75] K. Chen, M. Seuret, J. Hennebert and R. Ingold. Convolutional Neural Networks for Page Segmentation of Historical Document Images. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp.965-970, 2017.
- [76] K. Hadjar, and R. Ingold. Arabic Newspaper Page Segmentation, 7th International Conference on Document Analysis and Recognition, pp.895-899, 2003.
- [77] K. Hadjar, O. Hitz and R. Ingold. Newspaper Page Decomposition using a Split and Merge Approach, 6th International Conference on Document Analysis and Recognition (ICDAR), pp.1186-1189, 2001.
- [78] K. Hadjar. Une étude de l’évolutivité des modèles pour la reconnaissance de documents arabes dans un contexte interactif. PHD thesis, Fribourg University, 2006.
- [79] K. Javed and F. Shafait. Real-Time Document Localization in Natural Images by Recursive Application of a CNN. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp.105-110, 2017.
- [80] K. Linden. Word sense discovery and disambiguation. Academic dissertation, University of Helsinki, 2005.
- [81] K. O’Shea and R. Nash. An Introduction to Convolutional Neural Networks. ArXiv e-prints, 2015.
- [82] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
- [83] K.S. Sesh Kumar, A. M. Namboodiri, and C.V. Jawahar. Learning Segmentation of Documents with Complex Scripts. In *Fifth Indian Conference on Computer Vision, Graphics and Image Processing*, LNCS 4338, pp.749-760, 2006.
- [84] K.Y. Wong, R.G. Casey, and F.M. Wahl. Document analysis system. *IBM Journal of Reasrch and Devlopment*. Vol.25, pp.647-656, 1982.

- [85] L. Cinque, S. Levialdi, and A. Malizia. DAN: an automatic segmentation and classification engine for paper documents. Fifth IAPR International Workshop on Document Analysis Systems, LNCS 2423, pp.491-502, Princeton, 2002.
- [86] L. Dorini and N. Jeronimo Leite. A Multiscale Morphological Binarization Algorithm. Computer Vision, Imaging and Computer Graphics. Theory and Applications, pp.283-295, 2010.
- [87] L. Guan and R.K. Ward. Restoration of Randomly Blurred Images by the Wiener Filter, IEEE Transactions on Acoustics, Speech and Signal Processing ,Vol. 37, No. 4, 1989.
- [88] L. Likforman-Sulem, A. Hanimyan, and C. Faure. A Hough based algorithm for extracting text lines in handwritten documents. Third International Conference on Document Analysis and Recognition, Vol.2, pp. 774-777, 1995.
- [89] L. Likforman-Sulem, and C. Faure. Extracting text lines in handwritten documents by perceptual grouping, Advances in handwriting and drawing: a multidisciplinary approach, pp.117-135, 1994.
- [90] L. Ogorman and R. Kasturi. Executive briefing: document image analysis, IEEE Computer Society Press, ISBN 0-8186-7802-X, 1997.
- [91] L. Robadey. 2 (CREM): Une méthode de reconnaissance structurale de documents complexes basée sur des patterns bidimensionnels, Doctoral thesis, University of Friborg-Suisse, 2001.
- [92] M. Arivazhagan, H. Srinivasan, and S. Srihari. A statistical approach to line segmentation in handwritten documents. International Society for Optics and Photonics, 2007.
- [93] M. Ayesh, K. Mohammad, and A. Qaroush. A Robust Line Segmentation Algorithm for Arabic Printed Text with Diacritics. IS and T International Symposium on Electronic Imaging, pp.42-47, 2017.
- [94] M. Bulacu, R. Koert, L. Schomaker, and T. Zant, Layout analysis of handwritten historical documents for searching the archive of the cabinet of the Dutch queen, In International Conference on Document Analysis and Recognition, pp.23-26, 2007.
- [95] M. Elzobi, A. Al-Hamadi, and Z.A. Aghbari. O-line handwritten Arabic words segmentation based on structural features and connected components analysis. Communication Papers Proceedings: The 19th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp.135-142, 2011.

- [96] M. Feldbach and K.D. Tönnies. Line Detection and Segmentation in Historical Church Registers. Sixth International Conference on Document Analysis and Recognition, pp.743-747, 2001.
- [97] M. Kadhm. Arabic Handwritten Text Recognition and Writer Identification. University of Technology, Phd thesis, 2017.
- [98] M. Kamel and A. Zhao. Extraction of binary character/graphics images from grayscale document images, CVGIP: Graphical Models and Image Processing, Vol.55, No.3, pp.203-217, 1993.
- [99] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. International Journal of Computer Vision, Vol.1, No.4, pp.321-331, 1988.
- [100] N. Anoop and J. Anil. Document Structure and Layout Analysis. 2007.
- [101] N. Aouadi and A. Kacem-Echi. Word extraction and recognition in Arabic handwritten text. International Journal of Computing & Information Sciences 12, pp.17-23, 2016.
- [102] N. Chowdhury and D. Saha. Unsupervised text classification using Kohonen's self-organizing network. In Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics, Lecture Notes in Computer Science, Vol. 3406, pp.715-718, 2005.
- [103] N. Journet. Analyse d'images de documents anciens: une approche texture. Phd thesis, University of La Rochelle, 2007.
- [104] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, and N. Papamarkos. Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation path. Image and Vision Computing 28, No.4, pp.590-604, 2010.
- [105] N. Otsu. A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. 9(1), pp.62-66, 1979.
- [106] N. Tripathy and U. Pal. Handwriting Segmentation of Unconstrained Oriya Text, in International Workshop on Frontiers in Handwriting Recognition, pp.306-311, 2004.
- [107] N.B. Amara. On the problematic and Orintations in recognition of the Arabic Writing. In: CiFED, pp.1-10, 2002.
- [108] P. Agarwal, Hand-Written Character Recognition Using Kohonen Network. Vol.2, Issue.3, pp.112-115, 2011.

- [109] P. Farshid, A. Siti, and S. Shahnorbanun. Peak signal-to-noise ratio based on threshold method for image segmentation. *Journal of Theoretical and Applied Information Technology*, Vol.57, No.2 pp.158-168, 2013.
- [110] P. P. Roy, U. Pal, and J. Lladós. Morphology based handwritten line segmentation using foreground and background information. In *International Conference on Frontiers in Handwriting Recognition*, pp.241–246, 2008.
- [111] P. Soujanya, V. K. Koppula, K. Gaddam, and P. Sruthi. Comparative study of text line segmentation algorithms on low quality documents. *CMR College of Engineering and Technology Cognizant Technologies*, 2010.
- [112] P.A. Toft. *The Radon Transform-Theory and Implementation*. Kgs. Lyngby, Denmark: Technical University of Denmark, 1996.
- [113] P.E. Mitchell, and H. Yan. Newspaper Document Analysis featuring Connected Line Segmentation, 6th International Conference on Document Analysis and Recognition, pp.1181-1185, 2001.
- [114] P.Y. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, Vol.2, pp.958-962, 2003.
- [115] R. Cattoni, T. Coianiz, S. Messelodi, and C. M. Modena. Geometric layout analysis techniques for document image understanding: a review. Technical report, IRST, 1998.
- [116] R. Davidson and R. Hopely. Arabic and Persian OCR Training and Test Data Sets, *Proc. Of Symp. on Document Image Understanding Technology*, 1997.
- [117] R. P. d. Santos, G. S. Clemente, T. I. Ren, and G. D. C. Cavalcanti. Text line segmentation based on morphology and histogram projection. In *Proceedings of the 10th International Conference on Document Analysis and Recognition*, pp.651-655, 2009.
- [118] R. Sivaramakrishnan, I.T. Phillips, J. Ha, S. Subramaniam, and R.M. Haralick. Zone classification in a document using the method of feature vector generation, In *International Conference on Document Analysis and Recognition*, Vol.2, pp.541-544, IEEE, 1995.
- [119] R. Srisha and A. Khan. *Morphological Operations for Image Processing: Understanding and its Applications*. 2013.
- [120] R.G. Casey and K.Y. Wong. Document analysis systems and techniques, in *Image Analysis Applications*, R. Kasturi and M.M. Trivedi (eds), pp.1-36, 1990.

- [121] R.I. Gandhi and K. Iyakutti. An Attempt to Recognize Handwritten Tamil Character Using Kohonen SOM. pp.188-192, 2009.
- [122] R.O. Duda and P.E. Hart. Use of the hough transformation to detect lines and curves in pictures, *Commun. ACM*, Vol.15, pp.11-15, 1972.
- [123] S. Ahmed, M. Imran Malik, M. Zeshan Afzal, K. Kise, M. Iwamura, A. Dengel, and M. Liwicki. A Generic Method for Automatic Ground Truth Generation of Camera-captured Documents. in *arxiv.org*, 2016.
- [124] S. Al-Ma'adeed, D. Elliman, and C.A. Higgins. A database for Arabic handwritten text recognition research. In: *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*. pp.485-489, 2002.
- [125] S. He and L. Schomaker, DeepOtsu: Document Enhancement and Binarization using Iterative Deep Learning, *Computer Vision and Pattern Recognition*, Vol.91, pp.379-390, 2019.
- [126] S. Hussain, S. Ali, and Q. Akram. Nastalique segmentation-based approach for Urdu OCR. *International Journal on Document Analysis and Recognition (IJ DAR)*, Vol.18, No.4, pp.357-374, 2015.
- [127] S. Kanoun, A.M. Alimi, and Y. Lecourtier. Affixal Approach for Arabic Decom-posable Vocabulary Recognition: A Validation on Printed Word in Only One Font. In: *ICDAR*, pp.1025–1029, 2005.
- [128] S. Lu and C. L. Tan. The restoration of camera documents through image segmentation. In *Proceedings 7th IAPR workshop on Document Analysis Systems*, pp.484-495, 2006.
- [129] S. Mori, C.Y. Suen, and K. Yamamoto. Historical review of OCR research and development. *Proceedings of the IEEE*, pp.1029-1058, 1992.
- [130] S. Nicolas, T. Paquet, and L. Heutte. Text Line Segmentation in Handwritten Document Using a Production System. *Proceedings of the 9th IWFHR*, pp.245-250, 2004.
- [131] S. Ong and M.H. Sunwoo. A morphological filter chip using a modified decoding function. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol.47, No.9, pp. 876–885, 2000.
- [132] S. Raymond. Hybrid Page Layout Analysis via Tab-Stop Detection. *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 241-245, 2009.
- [133] S. Sukhvinder and G. Surender. Role of Mathematical Morphology in Digital Image Processing: A Review. *International Journal of Scientific Engineering and Research*. pp.1-3, 2014.

- [134] S. VishwasH and B.A. Thomas. Impact of Smearing Techniques on Text Line Localization of Kannada Historical Scripts. 2015.
- [135] S.F. A Graves, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: International Conference on Machine learning. pp.369-376, 2006.
- [136] S.N. Srihari and D. Wang. Classification of newspaper image blocks using texture analysis. Computer Vision Graphics and Image Processing, Vol.47, No.3, pp.327-352, 1989.
- [137] S.S. Bukhari, F. Shafait, and T. M. Breuel. Dewarping of document images using coupled-snakes. In Proceedings of Third International Workshop on Camera-Based Document Analysis and Recognition, pp.34-41, 2009.
- [138] S.S. Bukhari, F. Shafait, and T.M. Breuel. Layout Analysis of Arabic Script Documents. in Guide to OCR for Arabic Scripts, Springer, Ch 2, pp.35-53, 2012.
- [139] S.S. Bukhari, F. Shafait, and T.M. Breuel. Segmentation of curled text lines using active contours. In Document Analysis Systems, The Eighth IAPR International Workshop on IEEE, pp.270-277, 2008.
- [140] S.S. Bukhari, F. Shafait, and T.M. Breuel. Text-Line Extraction Using a Convolution of Isotropic Gaussian Filter with a Set of Line Filters. International Conference on Document Analysis and Recognition, pp.579-583, 2011.
- [141] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen. Partial-result reuse architecture and its design technique for morphological operations with flat structuring elements. IEEE Transactions on Circuits and Systems for Video Technology, Vol.15, No.9, pp.1156-1169, 2005.
- [142] T. Akiyama and N. Hagita. Automated entry system for printed documents. Pattern Recognition, pp.1141-1151, 1990.
- [143] T. Kohonen and H. Xing. Contextually self-organized maps of Chinese words. Advances in Self-Organizing Maps, Lecture Notes in Computer Science, Vol.6731, pp.16-29, 2011.
- [144] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. SOM-PAK: The self-organizing map program package. Technical Report A31, Helsinki University of Technology, 1996.
- [145] T. Kohonen. The self-organizing map. Proceedings of the IEEE, 78(9), pp.1464-1480, 1990.

- [146] T. Palfray, D. Hébert, P. Tranouez, S. Nicolas, and T. Paquet. Segmentation logique d'images de journaux anciens, Francophone International Conference on Writing and Document, pp.317, 2012.
- [147] T. Varga and H. Bunke. Tree structure for word extraction from handwritten text lines, in: The Eight International Conference on Document Analysis and Recognition, pp.352–356, 2005.
- [148] T.M. Mitchell. Machine Learning. McGraw-Hill, 1997.
- [149] U. Mahadevan and R.C. Nagabushnam. Gap metrics for word separation in handwritten lines. Third International Conference on Document Analysis and Recognition, pp.124-127, 1995.
- [150] U.V. Marti and H. Bunke. Text Line Segmentation and Word Recognition in a System for General Writer Independent Handwriting Recognition. Sixth International Conference on Document Analysis and Recognition, pp.159-163, 2001.
- [151] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis. Handwritten document image segmentation into text lines and words. Pattern Recognition 43, pp.369–377, 2010.
- [152] X. Yang, E. Yumer, P. Asente, M. Kralej, D. Kifer, and C. L. Giles. Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4342-4351, 2017.
- [153] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger. A new algorithm for detecting text line in handwritten documents. In International Workshop on Frontiers in Handwriting Recognition, pp.35–40, 2006.
- [154] Y. Li, Y. Zheng, D. Doermann, S. Jaeger, and Y. Li. Script-independent text line segmentation in freestyle handwritten documents. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.30, No.8, pp.1313-1329, 2008.
- [155] Y. Pan, Q. Zhao, and S. Kamata. Document Layout Analysis and Reading Order Determination for a Reading Robot. 10th IEEE Region Conference, pp. 1607-1612, 2010.
- [156] Y. Pu, and Z. Shi. A natural learning algorithm based on hough transform for text lines extraction in handwritten documents. In Proceedings of the Sixth International Workshop on Frontiers in Handwriting Recognition, pp.637–646, 1998.

-
- [157] Y. Shih, C.T. King, and C.C. Pu. Pipeline architectures for recursive morphological operations. *IEEE Transactions on Image Processing*, Vol.4, No.1, pp.11-18, 1995.
 - [158] Y. Xiao, Z. Cao, and T. Zhang. Entropic thresholding based on gray-level spatial correlation histogram. In *19th International Conference on Pattern Recognition (ICPR)*, pp.1-4, 2008.
 - [159] Y.M. Alginahi. A Survey on Arabic Character Segmentation, *International Journal on Document Analysis and Recognition (IJDAR)*, 16(2), pp. 105-126, 2013.
 - [160] Z. Kato, and T.-C. Pong. A markov random field image segmentation model for color textured images, *Image and Vision Computing*, Vol.24, No.10, pp.1103-1114, 2006.
 - [161] Z. Shi, S. Setlur, and V. Govindaraju. Text Extraction from Gray Scale Historical Document Images Using Adaptive Local Connectivity Map. *Proceedings of the International Conference on Document Analysis and Recognition, IC-DAR*, 2005.