Automatic processing of Historical Arabic Documents: a comprehensive survey

Mohamed Ibn Khedher, Houda Jmila, Mounim El Yacoubi

Version of Record: https://www.sciencedirect.com/science/article/pii/S0031320319304455
Manuscript_2e132f158aee827c9ac4bf8cbe55c4d2

Automatic processing of Historical Arabic Documents: A comprehensive Survey

Mohamed Ibn Khedher[a], Houda Jmila[b], Mounim A. El-Yacoubi[b]

[a]*Samovar, CNRS, Télécom SudParis, Institut Polytechnique de Paris*
9 rue Charles Fourier, 91011 Evry Cedex, France
[b]*IRT SystemX, 8 avenue de la vauve, 91120 Palaiseau, France*

Abstract

Nowadays, there is a huge amount of Historical Arabic Documents (HAD) in the national libraries and archives around the world. Analyzing this type of data manually is a difficult and costly task. Thus, an automatic process is required to exploit these documents more rapidly.

Processing historical documents is a recent research subject that has seen a remarkable growth in the last years. Processing Historical Arabic Documents is a particularly challenging problem. First, due to complicated nature of Arabic script compared to other scripts and second because the documents are ancient.

This paper focuses on this difficult problem and provides a comprehensive survey of existing research work. First, we describe in detail the challenges making the automatic processing of Historical Arabic Documents a difficult task. Second, we classify this task into four applications of automatic processing of HAD: i) Analyze the document to extract the main text ii) Identify the writer of the document iii) Recognize some words or parts of the document in a reference dataset and iv) Retrieve and extract specific data from the document. For each application, existing approaches are surveyed and qualitatively described. Finally, we focus on available datasets and describe how they can be used in each application.

Keywords: Historical Arabic Documents, Writer Identification, Data Retrieval, Text Analysis, Text Recognition, Survey on Historical Arabic Documents.

Nomenclature

$APHAD$ Automatic Processing of Historical Arabic Documents

BoW Bag of Word

DA Document Analysis

DR Data Retrieval

DTW Dynamic Time Warping

GHT Generalized Hough Transform

HAD Historical Arabic Documents

HMM Hidden Markov Models

HOG Histograms of Oriented Gradients

IHP Islamic Heritage Project

IP Interest Point

OCR Optical Character Recognition

PAW Part of Arabic Word

SVM Support Vector Machine

TA Text Alignment

WC Writer Classification

WI Writer Identification

WR Word Recognition

WR Writer Retrieval

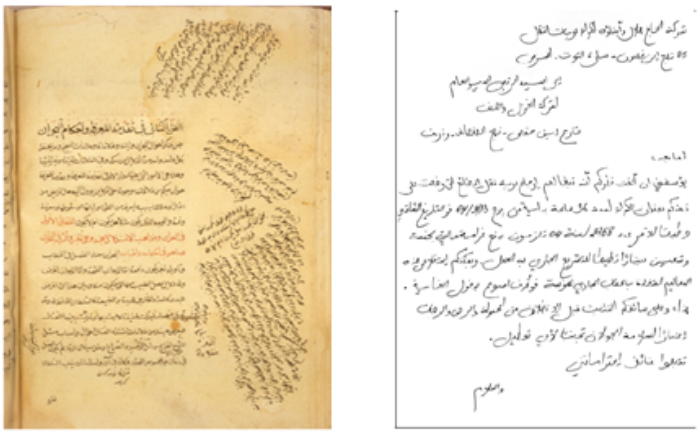
LCM Letter Connectivity Map

1. Introduction

Documents are frequently utilized in everyday life. The need to analyze them and understand their content is permanent. In the past, processing the documents was achieved *manually* due to the lack of *large and high quality digital datasets* where an *automatic* process can be learned. In fact, due to the scarcity of *i*) high quality digital scanning solutions and *ii*) high storage capacity devices, converting and storing document images from paper format to digital format was a challenging task. Recently, this task has become easier thanks to impressive advances in digital scanning and electronic storage solutions [87]. Consequently, the amount of numeric document datasets has expanded dramatically. This calls for automatic documents processing to understand their content and meet the users various queries such as *i*) searching some keywords in the document *ii*) identifying the writer of an unknown document, *iii*) indexing a document to facilitate its use, etc.

This paper focuses on the *Historical Arabic Documents*. As statistics mentioned in [87], over 90 million documents were written in Arabic script between the seventh and fourteenth centuries. About

Figure 1: An example of Historic Arabic Document (left) and Recent Arabic Document(right)



seven million documents, in various disciplines, survived the years. Processing these documents is required to make their knowledge accessible to the general public.

Processing Historical Arabic Documents is not well investigated in the literature. This task is challenging for two reasons: to the nature of Arabic script and second the poor quality of historical old documents. As shown in Fig.1, historical documents have very different patterns compared to recent ones. Thus, the complexity of processing a historical document is more challenging than processing a recent one. This is due essentially to the poor image quality of historical documents (in terms of resolution, etc.). Moreover, historical document generally miss some parts and their content is not available entirely. In this paper we focus on the complex task of *Automatic Processing of Historical Arabic Documents (APHAD)* and provides a comprehensive survey of existing research work.

The introduction is structured as follows. The first part discusses the APHAD applications. Second, our contributions are detailed and finally the paper organization is presented.

1.1. Applications

Processing Historical Arabic Documents offers a large range of interesting patterns classification and analysis problems. There are different ways of exploiting a document. In this survey, we have enumerated four applications based on our analysis of the state of the art. These applications are briefly described below.

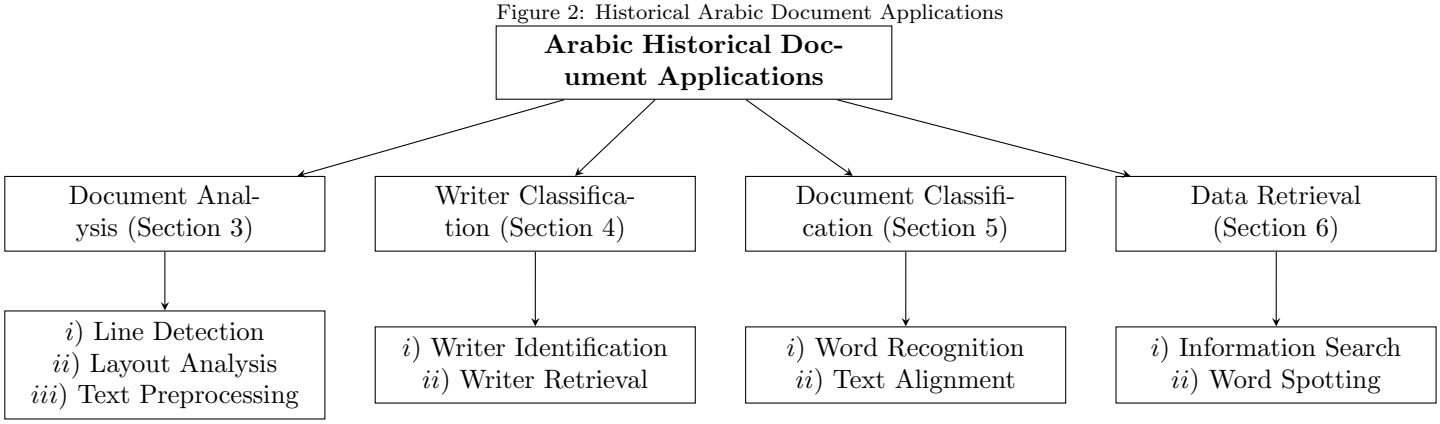
- **Document Analysis (DA):** As the quality of a historical document is often degraded, it cannot be used directly. As seen in the left of Fig.1, layout analysis and line segmentation are required to extract the main text in the document from the rest. We call this procedure *Document Analysis* and it can be seen as a pre-processing step before *writer classification* or *data retrieval*.

- **Writer Classification (WC):** often, a historical document misses some pages that include important information like the *metadata* of the document (year, writer, title, etc). When the writer

is unknown, we can try to find some information about him. This can be done first by comparing the document to other writer-known documents in the library (Writer Identification (WI)). Second, one can retrieve, out of the database, the documents images written by the same writer-even unknown (Writer Retrieval (WR)). We call these two procedures, *Writer Classification*.

- **Document Classification:** In the context of historical documents, it is often difficult to recognize each character separately. It is however possible to conclude some information about the document by considering a higher level of granularity: a sub-word, a word, or even the whole document, as described below.
  - *Word Recognition (WR):* The idea is to consider some words or sub-words of the document and recognize them through a matching with known words in the reference dataset to understand the content of the document.
  - *Text Alignment (TA):* The whole document is considered and compared with another known document in the reference dataset to check whether they have the same content. Mostly, the two documents are one text image and its potential transcript.
- **Data Retrieval (DR):** the objective is to retrieve/search a specific information/data from a query document, like searching for a keyword, the occurrence of a keyword in the text, etc.

Figure 2: Historical Arabic Document Applications



1.2. Contributions

This paper focuses on the Automatic Processing of Arabic Historical Documents. In the past, researchers were interested on processing *i*) *Arabic Non-Historical documents* [76, 67, 48, 75, 7] or *ii*) *Non-Arabic Historical Documents* [44, 49, 64]. As discussed in the introduction, processing Historical Arabic Documents is a more challenging task. Thanks to the increasing number of HAD digital datasets, this research subject is gaining more attention. Analyzing existing solutions can help understanding

the problem and identifying other issues. However, to the best of our knowledge, there is no survey summarizing the Automatic Processing of Arabic Historical Documents literature. This paper fills this gap and provides two essential contributions:

- Although some surveys of problems similar to the APHAD exist, they can not provide a comprehensive view of the APHAD problem. In fact, the solutions proposed for other document types (non degraded, non Arabic) are not necessarily adapted to Arabic Historic Documents. For example, Optical Character Recognition (OCR) [90] used for non degraded documents is not suitable to Historical Documents. *This paper summarizes existing solutions only proposed and validated on Historical Arabic Datasets.*
- For each HAD application, existing approaches are discussed and qualitatively compared, then some recommendations are provided. We think that this survey can be a first guide to researchers interested on APHAD.

1.3. Paper organization

The paper is structured as follows. Section 2 describes the challenges involved on the APHAD. Sections 3, 4, 5 and 6, present the state of the art of the four APHAD applications, respectively, Document Analysis, Writer Classification, Document Classification and Data Retrieval. For each task, the existing approaches are described in detail. Section 7, focuses on datasets and softwares used on the APHAD problem. Finally, conclusions and perspectives are drawn in section 8.

2. Challenges related to Historical Arabic Documents

The processing of a Historical Arabic Documents is a difficult task. The challenges can be grouped into two categories:

1. Challenges due to the nature of Arabic script.
2. Challenges due to the fact that the documents are ancient.

Next sections describe in detail each challenge category.

2.1. Challenges related to Arabic script

Analyzing Arabic script (cf. Fig 1) is difficult for the following reasons:

- Arabic text is cursive, the letters are joined together along a writing line.
- Arabic text contains small marks (i.e dots, punctuations below or above letters, etc.) that can change the signification of a word, and should be taken into account by any document analysis system.

- Letters have different shapes depending on whereabouts they are found in the word. The same letter at the beginning and end of a word can have a completely different appearance. In addition to dots and other marks, this broaden the size of the alphabet to about 160 characters [11].
- Arabic word can be composed of one or several Parts of Arabic Word (PAW). A PAW is a connected component which can refer to a diacritic sign, a single letter, a sequence of letters or a whole word.
- An Arabic letters can be horizontally and/or vertically ligatured. This makes the segmentation a difficult task.

2.2. Challenges due to Historical Documents

The analysis of a Historical Document is a difficult task due to its degradation through time. It can undergo several types of degradations:

- **Chemical degradations** due to the variation of temperature, humidity, brightness, and air pollution. These chemical reactions can lead to *i*) yellowish coloring of the paper or *ii*) discoloration caused by inks and pigments.
- **Biological degradation** caused by living beings (for example, the degradation of a part of the document by rats).
- **Human degradations** caused by the humans like: annotations added to document, scratches, etc.
- **Degradation due to the digitizing step**: the resolution of the digitization, the support of the digitization, the standard of compression, etc.

In the next sections, we describe the four APHAD applications and analyze existing solutions. Table 1 gives a general overview of the surveyed approaches. For each application, we summarize the details of the proposed approaches, the evaluation database and the obtained results (the accuracy). However, It is worth to note that these approaches should not be compared based on the obtained results as the authors used different evaluation protocols, different metrics, and different databases.

3. Document Analysis

Historical Documents have an important cultural value and it is important to exploit their content which is a challenging task. In some cases, a Document Analysis step can be applied on a document image before its exploitation (eg. writer identification, word spotting, text recognition, etc.). Document analysis can include: *i*) *Line Detection*, *ii*) *Layout Analysis* to separate the main text from the side text of a document image, and *iii*) *Image preprocessing* to improve the quality of a document image for a better use.

Table 1: Summary of research work related to APHAD.

Application	Category	Publication	Dataset	Year
Writer Classification				
Writer Classification	Model-free	[41]	IHP	2014
		[19]	WAHD	2015
		[12]	IHP/KHATT	2017
	Model-based	[40]	IHP	2014
Data Retrieval				
Word spotting	Model-free	[11]	Private (23 pages)	2011
		[87]	Private (20 pages)	2013
		[105]	Private (40 pages)	2013
		[10]	Private (23 pages)	2014
		[53]	Private (20 pages)	2014
		[81]	Private (12 pages)	2014
		[55]	VML	2016
		[39]	HADARA80P	2017
	Model-based	[84]	Private (20 pages)	2008
		[36]	Ibn-Sina	2015
		[58]	Ibn-Sina	2016
		[22]	VML	2018
Text Analysis				
Text Line Detection	Top-down	[17]	Private (217 pages)	2011
		[83]	Private (315 pages)	2014
	Hybrid	[31]	Private (836 pages)	2014
Layout Analysis	Model-free	[14]	Public (38 pages)	2014
		[54]	Public (38 pages)	2016
	Model-based	[29]	Public (38 pages)	2012
		[23]	Public (38 pages)	2018
Text Preprocessing	Text skew correction	[35]	Private	2010
	Image binarization	[8]	Private	2014
		[37]	Private	2015
Text Classification				
Text Recognition	Model-free	[15]	40,000 sub-words	2012
	Model-based	[5]	22,218 sub-words	2017
		[4]	4,124 sub-words	2018
Text Alignment	Model-free	[32]	(6 pages)	2015
		[80]	Private (6 pages)	2013
		[16]	Not mentioned	2011
	Model-based	[56]	Public (72 pages)	2017
Datasets				
Writer classification	Not degraded	[3]	WAHD Dataset: IHP subset	2017
		[3]	WAHD Dataset: NLJ subset	2017
Word spotting	Not degraded	[71]	IBN-SINA	2010
		[74]	HADARA80P	2014
		[52]	VML-HD	2017
Quality Assessment	Degraded	[91]	MHDID	2018
		[92]	VDIQA	2010

Softwares				
Data Retrieval	Word spotting	[70]	—	2011
	Annotation	[57]	Private (> 10 pages)	2014
		[24]	—	2013
Text Analysis	Text Preprocessing	[25]	National library of Tunisia	2007
	Simplifying the reading	[13]	Public (25 pages)	2015
Text Recognition	Text Classification	[95]	HADARA80P *and others*	2010

3.1. Line Detection

Table 2: Summary of related works to Line Detection task. Accuracy corresponds to the line detection precision.

Work	Description	Input	Dataset	Accuracy
Top-down Approaches				
[17]	Apply a Energy Map algorithm [63], then compute seems using dynamic programming. Seams propagate according to energy maps	Gray-scale document image	Dataset of 217 page (2520 lines). It includes Arabic, English and Spanish Historical Documents	96.05%
[83]	Apply an Energy Map algorithm [18] then compute seems	• Gray-scale document image • Binary document image	Dataset of 315 page (5431 lines). It includes Arabic, English and Chinese Historical Documents	• 98.85% • 97.30%
Hybrid Approaches				
[31]	Binarization based on component-tree + Assign components to lines based on Energy minimization	Gray-scale document image converted to Binary format	Datasets written in different language (Arabic, English, Hebrew, etc.) at varying range of image quality. It includes 521 pages	97.22%

The segmentation of a document image into a set of text lines can be an important step for several HAD applications such as text alignment, word spotting, etc. Although text line detection has received a lot of attention, it still faces many challenges related to the: *i*) document image quality and *ii*) the writing style. In fact, a historical document is usually of low quality due to its age, its storage conditions and the frequent manipulations. Often, historical documents include various noise, like spot and holes. The latter makes the line extraction step difficult. Moreover, each writer has his/her own style and gives rise to various text line segmentations. In addition to these difficulties, we mention: *iii*) the presence of touching components from successive lines and *iv*) the presence of punctuation and diacritic symbols interlines.

Remember that this paper focus only on research works related to Historic Arabic Documents. Therefore, approaches proposed for non-Historic-Arabic Documents are not referenced. Overall, Text line detection approaches can be divided into three classes: *i*) top-down, *ii*) bottom-up, and *iii*) hybrid.

• Top-down approaches divide the document image into regions, often recursively, in order to extract lines. One line is considered as a region of the input document image.

• Bottom-up approaches link basic elements, such as pixels or connected components, to form text lines.

• Hybrid approaches fuse techniques from top-down and bottom-up classes.

In the context of Historic Arabic documents, only few research works investigated the text line detection problem. These works lie either on the Top-down or Hybrid categories, as described below.

3.1.1. Top-down approaches

Most of these approaches define an energy function and aim to minimize it. Given an input document image, the basic idea is to apply an Energy Map algorithm to minimize the defined energy function, then seams are computed (medial and separating seams). Geometrically, the medial seams determine a text line and the separating seams define the upper and lower boundaries of the text line. In other words, the medial seam is defined as the chain of pixels that crosses the text area of a text line. The separating seam is defined as the path that passes between two consecutive rows.

The approach of [17] takes as input a gray-scale document image, and outputs a set of text lines. It consists of two stages: *i*) Apply a distance transformation to the document image and *ii*) Extract the two seams (medial seams and separating seams). The authors selected the Energy Map algorithm proposed in [63], then seams are computed using dynamic programming. The medial and separating seams propagate according to the energy maps, which are defined based on the constructed distance transform. The approach is evaluated on a dataset of 217 pages (2520 lines). It includes Arabic, English and Spanish Historical Documents [1]. The Arabic part includes 97 pages (900 lines) from "Juma Al-Majid Center for Culture and Heritage" [Abe].
In [83], the authors of [17] evaluated their approach on more different datasets. They proposed an equivalent approach that takes as input a binary document image, instead of gray-scale document image. The approach shares the same principle of [17]. Its idea is to seek paths that cross the middle of text lines (the "black" regions). Similar to [17], it starts by computing an energy map of the input document based on the Sign Distance Transform [18], then it uses dynamic programming to compute the minimum energy path. The approach is validated on a dataset of Arabic, English and Chinese documents. The Arabic part includes 156 pages of 3319 lines.

3.1.2. Hybrid approaches

In [31], the authors proposed a Hybrid approach to segment the text into lines. It consists of segmenting a document image into components, then components are fused to form lines. This approach includes three major steps: *i*) Line enhancement for slight skew, *ii*) Image binarization, and *iii*) Energy-based minimization. Given a gray-scale document image as input, first, a Gaussian based multi-scale

[1] The datasets and the corresponding ground truth are available at http://www.cs.bgu.ac.il/∼abedas/#publication.

Figure 3: Examples of Layout of Historical Arabic Documents

filter is applied to enhance text lines in the image. Then, a binarization algorithm is applied, based on component-tree [72]. Finally, an energy minimization step is applied to assign connected components to text lines. In evaluation, the authors used public datasets written in different languages having variable image qualities, such as ICDAR2013 [96] and ICDAR2009 [43]. Moreover, the authors constructed their own dataset [2] composed of 25 historical manuscripts with 261 text lines written in Arabic from the Islamic Heritage Project (IHP [3]).

3.1.3. Discussion

As shown in the literature, a line detection system takes as input a gray-scale or a binary image. As mentioned in [87], the use of gray-scale format is preferred since applying image binarization on low quality images can introduce noise and various artifacts.

Note that the performance of a line detection system depends on the nature of the input images. Most datasets used for evaluation contain non-degraded documents. We think that conceiving line detection systems for degraded document requires more investigation. In some cases, to better detect lines, some preprocessing steps should be applied to detect and correct the skew angle of the text as proposed in [35].

Finally, it is worthy to mention the work of [79] even if it is out of the scope of this review since it is not validated on a HAD datasets. The authors try to detect text lines in a large collection of Hebrew medieval manuscripts written in Judeo-Arabic, Hebrew, Aramaic and Yiddish languages using the Hebrew Alphabet. The proposal is interesting since it investigates line detection in corrupted and damaged historic manuscripts.

3.2. Layout Analysis

The layout analysis is the task of separating the side-notes from the main-text. Figure 3 shows examples of historical documents including side-notes and main-text. Over the time, people added their notes and remarks to the main document page (main-text) which led to an irregular layout format. The main text and side notes have different characteristics. For example, the main-text is generally horizontally oriented while the side-node can have different orientations and location on the page margins.

Layout analysis approaches can roughly be grouped into two classes: *model-based approaches* and *model-free approaches*. A model-based approach consists of modeling the side-notes and the main-text characteristics using Machine Learning algorithms whereas model-free approach directly compare these characteristics without any *prior* learned model.

3.2.1. Model-free approaches

In [14], the authors proposed a segmentation-based approach using filters. It is a top-down approach composed of two major steps: *i*) first, the document image is coarsely segmented using a texture-based filter and *ii*) second, the segmentation result is refined iteratively. In fact, in the first step, Gabor filters [42] are applied to segment the input document image into text regions according to their texture characteristics. Then, depending on the filter responses and using empirical thresholds, the pixels are classified as main-text or side-notes. In the second step, a global refinement scheme is applied to the coarse segmentation from the previous step. It consists of minimizing an energy/cost function previously defined.

An interactive approach, proposed in [54], is composed of two major steps: *i*) Scribbles drawing and *ii*) Layout segmentation. The process (*i* and *ii*) is repeated iteratively and at each iteration the user refines the layout segmentation by drawing a new scribble. In fact, the user draws, using the mouse, two types of scribbles: 1) background scribbles (the region should be removed) and 2) foreground scribbles (the region should never be removed and should stay in the foreground after each iteration). Then, according to the position of these scribbles and their characteristics, the layout segmentation method is performed. For layout segmentation, similar to [14], a bank of Gabor filters are applied to classify the selected various regions. In fact, to separate the side-note from the main-text, the authors define a metric that takes into account the characteristics of each text type including the size of the font, the writing style, the text orientation, etc.

[2]The dataset is available at http://www.cs.bgu.ac.il/∼rafico/GT.zip.
[3]https://library.harvard.edu/collections/islamic-heritage-project

Table 3: Summary of related works to Layout Analysis task. Accuracy corresponds to the layout extraction precision.

Work	Features	Classifier	Input	Dataset	Accuracy
Model-free approaches					
[14]	Gabor filters	Empirical thresholds	Binary format	38 Handwritten pages from 7 manuscripts	98.36%
[54]	Gabor filters	Predefined metric	Binary format	Same dataset as [14]	N/A
Model-based approaches					
[29]	• **Shape features:** height, foreground area, orientation, etc. • **Context features:** neighbors components	AutoMLP	Binary format	Same dataset as [14]	94.85%
[23]	Fully Convolutional Network		Original format	Same dataset as [14]	87.50%

3.2.2. Model-based approaches

In this category of approaches, the layout segmentation task is formulated as a classification problem. A *prior* model is learned from a training dataset to model the characteristics of note-side and main-text. In [29], the characteristics are learned via the AutoMLP [27] classifier. While in [23], a Fully Convolutional Network classifier is used.

In [29], each component from the document image is classified as `main-text` or `side-notes`. The approach consists of three major steps: *i*) Features extraction, *ii*) Training and *iii*) Labeling refinement. First, each component is described by two types of features related to its shape and context: 1) Component shape features are related to its height, foreground area, orientation, etc. and 2) Component context features are related to its neighbors components. Second, an AutoMLP [27] classifier is trained using a manually labeled dataset. The classifier takes as input 13 thousand main-text components and 12 thousand side-notes components. Finally, after classifying all the components of the input document image, each component label is refined (updated) using its neighbor labels, keeping a spatial coherence.

In [23], the two steps, features extraction and training are fused in a Fully Convolutional Network (FCN) classifier. It takes as input patches of the two classes (`main-text` and `side-notes`) and outputs the predicted class. The FCN classifier is learned from 100.000 patches and validated on 20.000 patches.

3.2.3. Discussion

As shown in Tab.3, all the approaches are evaluated on the same dataset with a small size of 38 Handwritten pages. The use of model-based approaches, specially Deep Learning like in [23], will be more beneficial if large datasets are used. In this case, the training dataset could be augmented via

augmentation techniques as detailed in [4].

3.3. Text Preprocessing

The text preprocessing task consists of transforming the original input document image into another format for better use. There are several text preprocessing tasks in the document analysis context. In the HAD context, they are mostly related to: *i*) Image binarization and *ii*) Text skew angle correction. Usually, the authors integrate the preprocessing steps to their approaches and do not evaluate them separately. This explains the few number of works cited below.

Regarding the *Image binarization* task, the authors of [8] proposed a thresholding-based approach. First, the input document image is transformed into the Neutrosophic sets domain [45] where each pixel is described by three values reflecting its truth/false/indeterminacy of belonging to a set of bright pixels. Second, an adaptive thresholding method [89] is applied to obtain the binary image. In [37], the team of [8] proposed an Automatic selection approach of thresholds. It is based on the Artificial Bee Colony optimization algorithm [51] to find optimum threshold values. In [46], the same team proposed a whale optimization algorithm, incorporating a fuzzy c-means objective function to obtain optimal thresholds.

Regarding the *Text skew angle correction* task, a simple approach was proposed in [35]. After a binarization step, two stages are performed: *i*) Angle skew detection and *ii*) Skew correction. First, the center of each connected component is searched and its contour is formed by the nearest ellipse. The baseline is detected as the principal axis of the ellipse. In the second step, the orientation of connected components is defined as a skew between its principal axis and the reference axis. To correct the orientation, the image is rotated according to the estimated direction.

Discussion

Often, the binarization step is avoided since it can produce noise and affect the performance of document processing tasks. Most researches prefer to use the original format that contains the maximum of available information. Regarding the text skew information, it can be helpful for some HAD tasks like the writer classification where scribes can be identified by their writing style. However, in others HAD tasks like line detection, text skew can be seen as a disruption and should be corrected for better segmentation. Text skew correction task is not well investigated. The only work tackling the problem is [35], but no information is mentioned about the size of the used dataset, thus we can not confirm the performance of the approach.

4. Writer Classification

4.1. Problem description

Given a query document image, it is important to know his/her writer (Writer Identification) or retrieve, out of a set of documents, the document images written by the same writer (Writer Retrieval). In both cases, we are talking about Writer Classification. In more detail, these two applications can be defined as follows :

- The **writer identification** task assigns a *query document* written by an unknown writer to a known writer in a reference dataset.

- The aim of the **writer retrieval** task is to select, or rank, the reference documents that are probability written by the same writer as the query document's based on their similarities. Writer retrieval can be used in the following scenario: having a part of a manuscript, the writer retrieval helps finding the rest of the manuscript in a huge amount of documents.

In this paper, we group these two procedures under the name of Writer Classification (WC). During the last years, WC has gained more interest, essentially due to *i*) the huge amount of historic manuscripts with unknown authors (eg. in the archive of national libraries), and *ii*) the large number of documents found with scattered pages.

In the literature, a WC approach is generally performed in two steps: *i*) writer representation and *ii*) writer features matching. Often, a preprocessing step is applied to reduce noise.

1. In the **writer representation** step, the writer is described by a set of features extracted from the document image to reflect his/her writing style (for example, the alignment of the text, the density of words per line, etc.).

2. During the **writer features matching** the query manuscript features are compared to those of a reference dataset composed of manuscripts with known writers.

Depending on the input document, we can enumerate two features matching levels, *i*) manuscript level and *ii*) page level:

- At the **manuscript level**, each query manuscript is considered as one request which writer needs to be determined.

- At the **page level**, each page from the query manuscript is classified independently of the others pages, then all query page labels are combined to determine the writer of the query manuscript.

Depending on the solution used for features matching, WC approaches can be divided into two categories:

• **Model-free approaches** where the features are compared using a distance metric [12] or the Nearest Neighbor classifier [19, 41]

• **Model-based approaches** where the features are compared to a pre-trained model, learned from a training dataset. In [40], the pre-trained model is learned using the Support Vector Machine classifier [98].

Table 4: Summary of existing work focusing on the Writer classification task. Accuracy corresponds to the writer classification rate.

Work	Features	Classifier	Dataset	Accuracy
Model-based approaches				
[40]	**Local Features based on IP** • Scale-Invariant Feature Transform • Oriented Basic Image Features **Global Features** • Histogram of Oriented Gradients • Histogram of Oriented Gradients • Contour-Based Features	SVM	**IHP**: Handwritten manuscripts	84.10%
Model-free approaches				
[41]	Some features as [41]	Nearest-Neighbor	**IHP**: Handwritten manuscripts	92.50%
[19]	• Contour-Based Features • Gradient-Based Features	Nearest-Neighbor	Private dataset of 10,000 Handwritten documents	93.95%
[12]	• Contour-Based Features • Globalizing local key point descriptors	Distance metric	• **WAHD dataset**: Handwritten manuscripts • **KHATT dataset**: Non-historical Handwritten pages	• 76.00% • 85.50%

4.2. Survey of existing approaches

In the next sections, we give a survey of existing research work that have been proposed for historical Arabic documents writer classification. A handful of papers addressed this problem as summarized in Tab.4. Below, we describe each method and give analyses and discussion.

4.2.1. ***Model-free approaches***

The authors of [41] evaluated different features and used the *Nearest-Neighbor classifier* [33] for matching. The evaluated features are of two types: *i*) *Local Features* based on Interest Point (IP) and

ii) *Global Features* based on image descriptor. The IP-based descriptor exploits some locally detected IPs or patches extracted from the document image. The detected IPs or patches have significant variations of image intensity values in their local neighborhood. Contrary to an IP-based descriptor, an image-based descriptor exploits the entire available page image to describe writers. The evaluated local features are the *Scale-Invariant Feature Transform* [68] and the *Oriented Basic Image Features* [65]. As Image-Based Features, the authors assessed the *Histogram of Oriented Gradients features* [34] and *Contour-Based Features* [30]. Note that to match two page features having a variable number of IPs, the authors converted, for each page, all IPs features to one feature. Moreover, a preprocessing step was applied to minimize the inherent noise and obtain a clean binarized version of the manuscript pages.

A similar work was proposed in [19]. It used the *Contour-Based Features* and the *Gradient-Based Features* to describe a document image, and the *Nearest Neighbor classifier* for matching. The major difference compared to [41] is the dataset used for the evaluation (cf. Tab.4).

Historical documents may be disorderly archived and pages from different manuscripts, written by different writers, could be ranged in the same document. The belief that there is one writer of a historical document might decline the performance of any writer analysis system. In this context, in [12], the authors propose to apply an *intra-document analysis(IDA)* process, on each manuscript before the writer classification step. This process aims to discard the pages which are suspected of not being written by the main (ground truth) writer. The IDA process consists of two steps: *i*) a clustering step and *ii*) a filtering step. The clustering step is based on a hierarchical clustering which groups data over a variety of scales by creating a cluster tree. The cluster with the maximum number of pages is kept since it contains pages written by the main writer. After uncertain pages were filtered, the writer classification approach is applied. For image description, the authors used the *Contour-Based Features* and a *key-points-based* descriptor. The matching steps is based on usual distances like *the cosine distance* and the χ^2 *distance*.

4.2.2. ***Model-based approaches***

In the writer identification task, it is hard to assume that a query manuscript associated with an unknown writer is necessarily written by an author existing in the reference database. Finding one reference manuscript which reveals the highest similarity with the query document does not systematically imply that the two manuscripts were written by the same scribe. In this context, in [40], the same authors of [41] proposed a two-stages approach to reject non correct identification. In the first stage, for each query page, the best ranked pages and their corresponding writers in the reference dataset are retrieved. The authors used the same features and classification method, proposed in [41], to select the most similar pages from the reference dataset. The result is a ranked list of retrieved pages, sorted by the *internal confidence* of the retrieval system in the correctness of the respective page. In a second

stage, from the retrieval results, a set of features are passed to a two-classes Support Vector Machine (SVM) to accept or reject the decision.

4.3. Discussion

As shown in the literature, the writer classification task was often tackled while assuming two hypotheses:

1. All the pages of the query manuscript were written by the same scribe.

2. The reference dataset include systematically the actual writer of the query manuscript.

Some studies have ignored one or the other of these hypotheses ([12] ignored 1) and [40] ignored 2)). However, given that these two hypotheses have the same importance, an approach resolving the writer classification issue while respecting *both of them* is required.

During the writer classification step, more parameters that can temporarily influence and change the scribe writing style like the writer anatomic conditions, age, etc., should be investigated and taken into account. Such parameters were considered in [2] where the authors focused on classifying writing styles in the same document. The authors proposed to use *Siamese Nets* [28] based on *Siamese convolutional Neural Network* to measure the small changes in the writing style of the same person.

As shown in Tab.4, a document image is mostly described by texture features that reflect the scribe handwriting style. The use of such features is well coherent with the use of handwriting datasets for their evaluation. However, in *machine printed document datasets*, the scribe writing style is absent, thus the distinction between the scribes should be rather based on the documents content.

5. Text Classification

Table 5: Summary of research works related to Document Classification. Accuracy corresponds to the word recognition rate.

Work	Description	Input	Dataset	Accuracy
Model-based approaches				
[5]	Sub-word recognition model based on CNN	Original images	Three test datasets, each one has 22,218 sub-words belongs 68 classes	83.16%
[4]	Same approach as in [5] + Study of data augmentation and synthesizing techniques	Original images	Three test datasets, each one has 4,124 sub-words belonging to 39 classes	97.82%
Model-free approaches				
[15]	Hierarchical representation of sub-words and matching based on DTW distance	N/A —	40, 000 sub-words of unnamed dataset	N/A

Given an input document image, it is important to understand and recognize its content. In the context of historical documents, it is often difficult to recognize each character separately. Two text classification tasks are investigated in the state of the art according to the input type:

• **Word recognition:** mostly, the system takes as input a word/sub-word image and aims to classify the image into one reference class. Each reference class includes one semantic word with different shapes.

• **Text alignment:** the system takes as input two document images and check whether they have the same content. Mostly, the two documents are one text image and its potential transcript.

5.1. Word Recognition

Most of the time, document analysis (section 3) and word recognition have performed successively. Text analysis, as detailed previously, is considered as a prepossessing step where the main-text is located and lines are extracted. Then, other tasks, such as word recognition, can be applied .

As explained in section 2, processing historical Arabic documents faces different challenges in spite of the conception of new methods like the Optical Character Recognition to modern printed documents. To overcome these challenges, researchers segment historical documents into words or sub-words instead of characters. Then, these words (sub-words) are recognized by matching them with known words (sub-words) in the reference dataset; this is called word recognition. Despite the importance of this task, only few solutions are proposed in the literature, and can be roughly regrouped into two classes: i) model-based approaches and ii) model-free approaches.

As *model-based approaches*, a Deep Learning-based solution was proposed in [5] to recognize Arabic sub-words. As the amount of available data was limited, the authors applied data augmentation techniques to generate more image types. They first generated various printed forms of sub-words using the "$Naskh font$", a widely used font to write Arabic scripts. Second, they augmented the number of images by generating synthetic images [54]. In [4], the same authors assessed the benefits of extending the training set with the augmented data on the performance of Historical Arabic sub-words recognition. They realized several experiments to better represent the training dataset and improve the sub-word recognition performance. The investigations led to two conclusions: i) the use of ten pages of a manuscript for training and extending them is sufficient for successful sub-word recognition on the whole manuscript and ii) the combination of different training sets can improve sub-word recognition performance on the whole manuscript.

As *model-free approaches*, the authors proposed in [15] a hierarchical scheme for Arabic text recognition. The main idea is to generate a hierarchical representation for the shapes of sub-words. The top levels of the hierarchy include the coarse representations (i.e global representations of sub-words) and the low levels include the fine representations. The shapes of a given layer are simplified via a Skeleton algorithm [20] and then clustered to form the next level. The clustering step is based on the Dynamic Time Warping (DTW) distance to measure the similarity between skeleton graphs. Given a query shape, the lexicon is reduced by traversing the hierarchy in a top-down fashion and by skipping

the less promising clusters.

It is important to mention that word classification has been widely studied for non-historical and Arabic scripts, and several state-of-the-art studies were published [75, 97].

Regarding the model-based category, the authors proposed in [6, 50] two segmentation-free approaches (based on sliding window). In [6], each word image is described as a sequence of statistical features such as pixel densities. Then, the extracted features are fed to a Hidden Markov Model (HMM) in the matching step. In [50], HMM is used to model a multi-stream input. The authors proposed in[78] to use a CNN to estimate the $n-gram$ frequency profile of a word image. Then, the Canonical Correlation Analysis is used to match the estimated profile to the reference dataset. In [9], a CNN is used for feature extraction. Then, in the matching step, a HMM is used to measure a similarity between two word features. In [38], the authors investigate a Deep Belief Neural Network (DBNN) for Arabic word recognition.

Regarding the model-free category, the authors of [100] proposed a lexicon reduction as a preprocessing step for word recognition. It consists of detecting dot descriptors. Then, a convolutional neural network is used to verify these dot candidates. Finally, the number and position of dots are used to eliminate unlikely lexicon candidates. In [85], the authors proposed a segmentation-free approach using holistic features related to word shape and geometry. Then, a set of filters is applied to reduce the search space. Finally, the DTW distance is used to match words.

It is important to mention that researchers have been recently interested nowadays in Arabic text recognition from video sequences. They exploit the benefits of temporal layers of deep learning models to describe text in videos. The authors of [104] proposed the use of recurrent neural networks by coupling a Long Short-Term Memory (LSTM) and a temporal classification layer. The work of [69] proposed the integration of two deep neural networks. First a Convolutional Neural Network is applied to extract features from the document image. Second, a Bidirectional LSTM followed by a Connectionist Temporal Classification layer (CTC) is used for sequence labelling.

Discussion

In the literature, the word recognition task is formulated as a classification problem, and mostly, model-based solutions are proposed. Moreover, the researchers are often driven to overcome other challenges when dealing with the word recognition task, namely i) extending the size of the training dataset and ii) recognizing the whole manuscript using only a part of the manuscript in training.

Note that the word recognition task may be used for writer classification (section 4). For an unknown manuscript, each sub-word may be classified to one reference class associated with a known writer. The author of the query manuscript is then claimed as the writer associated with a maximum of reference words.

Although deep learning based approaches have mostly been applied to non-historical documents, it is expected that they will become mainstream in the historical word recognition context.

5.2. Text Alignment

Text alignment is one of the APHAD important tasks. It aims at determining the similarities and differences between two given manuscripts, mostly, two versions of the same manuscript. These dissimilarities between versions of the same manuscript can result from: *i*) details and notes added by readers on the original manuscript, *ii*) scribes inserted, replaced or omitted to adapt the content to geographical regions, etc.

Why do we need to align manuscripts? To answer this question, recall that in most HAD datasets, the documents are saved as images, which makes searching and indexing the document a difficult task. In some cases, an ASCII transcription is provided together with the document's image, and the alignment task would be the task of mapping each word from the transcript to its corresponding word image in the original document's image. The alignment task allows to *i*) accelerate searching and indexing in the document images and *ii*) generate an automatic ground truth for a document image, that can be used to evaluate the recognition algorithms (or other APHAD tasks).

Text alignment algorithms can be regrouped into two categories: 1) model-based alignment and 2) model-free alignment. In the model-based alignment approaches, [56], a *prior* model is learned from a dataset of annotated manuscripts. In model-free alignment solutions, original document images and transcript images are matched without any *prior* information by estimating a mapping function (i.e alignment function) [32, 80, 16].

As model-based alignment approach, the authors of [56] formulated the alignment task as a classification problem, using a Siamese Neural Network. The approach aims to automatically identify whether a pair of images have the same content, without recognizing the text. It consists of four major steps: 1) Data preparation, 2) Data generation, 3) Model creation and 4) Text alignment. A neural network

Figure 4: Alignment approaches categorization

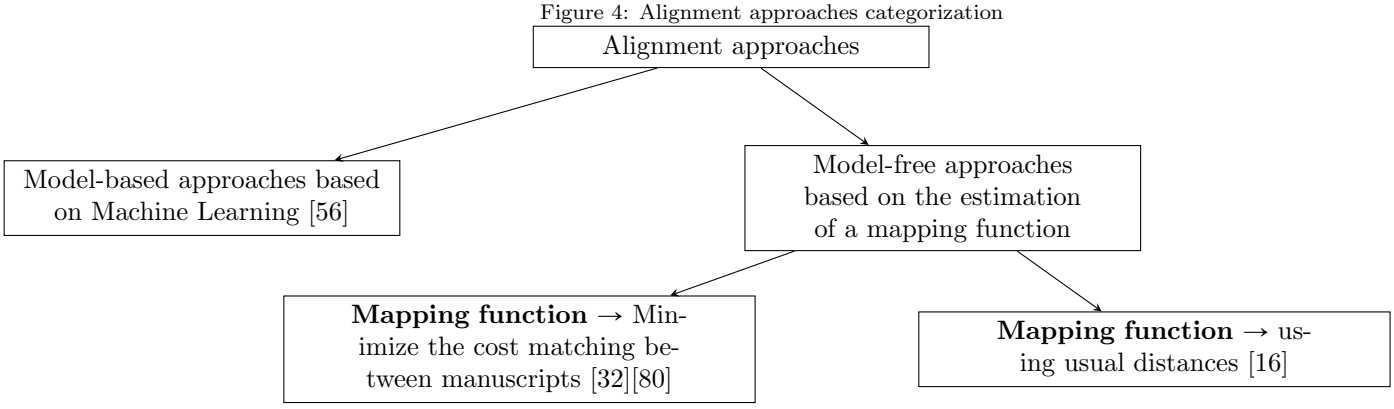


Table 6: Summary of research works related to Document Alignment

Work	Description	Input	Dataset	Accuracy
Model-free approaches				
[16]	Estimate a mapping function that minimize the matching cost between manuscripts	Not mentioned, but it can be gray or binary format	Not mentioned	N/A
[80]	Estimate a mapping function that minimize the matching cost between manuscripts	Binary format	6 pages (130 lines)	73.89%
[32]	Estimate a mapping function to measure similarly between manuscripts based on the DTW distance	Binary format	6 pages (130 lines)	76.41%
Model-based approaches				
[56]	Use the Siamese Network to match manuscripts	Original format	72 pages from the VML dataset	97.36%

needs a large annotated dataset as input to create a reliable alignment model. To have such a dataset, several pages of the original database were annotated using the WebGT tool [24]. Besides, to enrich the original dataset, data augmentation techniques were employed: 1) first, minor distortion and affine transformation were applied, 2) second, synthesis images (generated from the original ones) were added using the algorithm in [54]. After data preparation, two sets of image pairs were generated: true-pair and false-pair. The true-pair set corresponds to the images containing the same text. The false-pair set corresponds to the images containing different texts. These pairs were used as input of the Siamese Neural Network. Finally, the text alignment algorithm uses a window of a fixed size. In fact, given w_1 resp. w_2 two windows of two manuscripts, composed of a set of sub-words ${sb_1}^1, ..., {sb_n}^1$, resp. ${sb_1}^2, ..., {sb_n}^2$, all pair of sub-words are passed into the Siamese Neural Network, and only true-pair images are retained. Then, using the Hungarian method [60], the most optimal fit of w_1 and w_2 is retained.

In the model-free approaches category, the alignment task is formulated as *the problem of estimating a mapping function*. This function can be estimated using : 1) usual distances such as the Dynamic Time Warping distance as in [16] or 2) minimization algorithms as in [79, 32].

In [16], the mapping function was estimated using usual distances. To align two manuscripts, each one is considered as a long row of components, then the components are compared using distances. The approach consists of two major steps: 1) extracting the features from the bounding box of each component and 2) matching the components of the two manuscripts using the DTW distance to compute the optimal path.

In [79, 32], the alignment task is formulated as *a minimization problem:* given a set of m ordered words $T = \{w_1, w_2, ..., w_m\}$ in the transcript line and a set of n connected components in the original line $C = \{c_1, c_2, ..., c_n\}$, let $I_{C_k, C_{k+1}, ..., C_i}$ be the image defined by the connected component $C_k, C_{k+1}, ..., C_i$ and I_{w_j} be the image of the word w_j. The alignment algorithm aims to estimate a mapping function that assigns each connected component to a word. To accomplish this, two solutions based on minimization

techniques were proposed: one uses dynamic programming [79] and the second minimizes an energy function [32].

In [79], the authors proposed to use dynamic programming to match words in the transcript line to groups of connected components in the original line. To do this, a cost function $T(i,j)$ was defined. It reflects the cost of aligning connected components $\{c_1, c_2, ..., c_i\}$ with the words $\{w_1, w_2, ..., w_j\}$. $T(i,j)$ is equal to the minimum cost, over all $k < i$, of aligning $C_1, C_2, ..., C_k$ with $w_1, ..., w_{j-1}$ plus the distance between $I_{C_{k+1},...,C_i}$ and I_{w_j}. To measure similarity between images, handcrafted features, namely Histogram of Gradient Oriented [34], are used. Based on the matching distance value, the original and manuscript lines are considered to have or not the same content.

In [32], the mapping function f is estimated by minimizing its energy $E(f)$ defined as the sum of two terms: `data_costs` and `smoothness_term`. The `data_costs` reflects the cost of assigning the component c to the word w_c. In fact, the authors used the algorithm proposed in [32] to find the initial alignment, then, it is refined by adding constraints. The `smoothness_term` reflects the coherence of the labels w_c and $w_{c\prime}$ with the spatial relation of the components c and $c\prime$.

Discussion

It is important to mention that the works [80, 32] were validated on Hebrew data-sets. We nonetheless included them in this survey for two reasons: first for the specificity of the proposed approach based on energy minimization algorithm and second because Hebrew and Arabic scripts share the same cursive aspect.

In the first years, the alignment task was formulated as a minimization problem. This kind of approaches need to define the objective function and initialize its parameters. This can be challenging since the objective function should include, in addition to the matching cost, other constrains such as the spatial coherence between adjacent components, etc. Recently, model-based approaches based on Deep Learning were proposed to align the documents. Thanks to the potential of Deep Learning, such approaches are promising, but require a large data to train the model.

In conclusion, we think that using model-based approaches and extending the size of training dataset with data augmentation techniques is a promising approach to align documents.

6. Data Retrieval

6.1. Problem description

Data Retrieval (DR) consists of searching/retrieving a specific information from a query document. The need for such application is increasing with the increasing number of documents available in the datasets. In many data retrieval systems focusing on *non* historical documents, document images are first converted to their text format (i.e. Electronic Representations) using the Optical Character

Recognition algorithm, then DR techniques are applied on the text format to retrieve information. However, regarding the low quality of historical documents, converting the document images to their text format is still a challenging and complex task that is often avoided. In fact, it is generally more advantageous to explore the documents at the image level rather than the text format level.

Mostly, two types of data retrieval queries are considered in the sate of the art, and are often tackled together: *i*) *Information Search* and *ii*) *Keyword Spotting*.

• An **Information Search** request asks whether the document image contains a query keyword and the answer is "yes" or "no.

• A **Word Spotting** system takes as input a document images and a keyword image and aims to locate the keyword occurrences in the document images without recognizing its content. It is different from the *word recognition* task where the content of the word image is recognized.

According to these definitions, *information search task* is a special case of the *word spotting task*. Mostly, researchers focus on the word spotting task.

A data retrieval system is based on image matching; and depending on the way the query and reference images are matched, data retrieval approaches can be grouped into two categorizes: model-based approaches and model-free approaches. In both categories, depending on the way the keyword occurrences are fetched, the approaches can be either: *i*) segmentation-based or *ii*) segmentation-free.

• In **segmentation-based approaches**, the input document is segmented into words that are compared each to the query keyword.

• In **segmentation-free approaches**, a sliding window technique is used and each patch is compared to the query keyword.

6.2. Survey of existing approaches

In this section, an survey of existing solutions is presented: *i*) Model-based approaches and *ii*) Model-free approaches.

6.2.1. Model-based approaches

Most of model-based approaches are segmentation based. Given the input document images and the keyword to spot, the input documents are first segmented (eg. using a line and connected components detection techniques such as the technique proposed in [61]). Second, the keyword and connected components are matched using a *prior* learned model.

In [36, 58], the authors formulate the data retrieval process as a classification task composed of four major steps: *i*) Text line segmentation, *ii*) Feature extraction, *iii*) Model training and *iv*) Matching. In

Table 7: Summary of related works to word Searching / Indexation task. Along the accuracy, the metric, when available, is mentioned.

Work	Features	Classifier	Segmentation	Dataset	Accuracy
Model-based approaches					
[36, 58]	HOG	SVM	Sliding window	Ibn-Sina	85.63% (mAP)
[84]	Geometric shape features	Hidden Markov Model	• Line Extraction • Components labeling	20 pages (≈ 4000 words)	66.00%
[22]	Siamese Neural Network		Manual segmentation	VML	62.00% (mAP)
Model-free approaches					
[53]	• Radial descriptor • BoW representation	Cosine and χ^2 distances	Word-parts segmentation	20 pages (916 word-parts)	75.52% (Recall)
[55]	Graph representation based on Radial Descriptor	Graph Matching	Manual segmentation	VML	84.30% (mAP)
[81]	BoW representation based on SIFT descriptors	Hash table	Segmentation-free	12 pages (Cairo Genizah)	60.00% (mAP)
[10, 11]	Generalized Hough Transform	Hash table	Slide window	23 pages (3312 words)	N/A
[105]	Shape features	DTW	Run Length Smoothing Algorithm	40 pages (1616 words)	96.04% (F-Score)
[39]	Binary images pixels	Normalized Cross Correlation	Segmentation-free	HADARA80P	81.00% (Recall)
[84]	Geometric shape features	DTW	Line Extraction & Components labeling	20 pages (≈ 4000 words)	81.00%

the first step, an algorithm based on partial projection histogram [102] is used to split the text lines of the input document. Next, a SVM-based model is learned using *positive* and *negative* image sets. The positive set is produced by slightly moving the window around the query region. The negative region is obtained by taking a sampler random region. For both sets, images are characterized by a HOG vector (Histograms of Oriented Gradients [34]). Finally, in the matching step, the window slide technique is used to compare the query keyword to the document regions using the pre-trained SVM model. The regions with high SVM output score are retained as similar to the keyword.

In [84, 87], the search for a keyword is performed by searching its word-parts, including the additional strokes, in the right order. The process is performed in three steps: 1) Line extraction and Component labeling, 2) Features extraction 3) and Matching. First, input documents are segmented into lines, then the lower and upper base lines are used to extract the various components (a component can be a word or a part-of-word) [86]. Second, from each component, a set of features related to its geometry are

extracted. In fact, each component is defined by a polygon with k vertices, then, geometry features, related to angles and length between the different vertices are extracted. Finally, these features are used as inputs to the matching step. Two classifiers were investigated: *Hidden Markov Models* (HMM) and *Dynamic Time Warping* (DTW). HMM is a generative probabilistic model that models a set of temporal observations [82]. Dynamic Time Warping is an unsupervised algorithm for measuring similarity between two sequences which may vary in length [88]. In [87], the authors evaluated their approach in a larger historical dataset. Moreover, the solution proposed in [84] was enhanced by slightly modifying the DTW to include different costs for substitution, insertion and deletion of segments from the compared sequences.

Recently, a Convolutional Siamese Network-based approach was proposed in [22]. The network employs two identical Convolutional Networks to measure similarity between two input word images. Once the network is trained, it can be used to match input images.

6.2.2. Model-free approaches

In model-free approaches, image features are matched without any *prior* informations. Matching techniques used in the literature can be classified into two categories: *i*) Data structure-based matching and *ii*) Direct features matching.

• **Data structure-based matching:** the reference dataset is transformed into a data structure. It can be a hash table, a dictionary or other data structure.

• **Direct features matching:** features are matched directly without any *prior* information or data structure.

In the data structure-based matching category, a dictionary-based approach was proposed in [53]. The approach is segmentation-based as the input document is segmented into word-parts. Next, two steps are applied: *i*) features extraction based on Radial descriptor and *ii*) features matching. Each word-part image is described by a set of local features representing the dominant features. These features are associated to the pixels having a high variance according to their response to Radial descriptor [53]. Then, each word-part is represented by a Bag of Word (BoW) [93] descriptor. The descriptor is based on dictionary representation. The dictionary is constructed by clustering all reference dataset features into k classes and a pattern is represented by its occurrence probability histogram according to the constructed dictionary. The matching step is based on usual distance metrics such as the cosine distance and the χ^2 distance.

The same authors proposed in [55] a graph structure-based approach. The main idea is to take into account the position of features in the image. Instead of the Bag of Word representation, the authors proposed to encode a word-part image as a graph. The nodes of the graph are the feature points, and

adjacent points are connected by an edge to form a planar graph. Then, iteratively, the graph size is reduced by merging adjacent nodes. The matching step is based on graph matching technique that consist on optimizing the distance between the feature points and the structure of the graphs.
In [81], the word spotting task was solved using a hash table. Given an input document image and a query keyword q, first, a data structure based on hash tables is pre-computed, then candidates similar to q are retained from the data structure. The approach consists of three steps: 1) feature extraction, 2) data structure construction and 3) word spotting. The feature extraction is performed in an off-line stage where the input images are subdivided into overlapping patches. Each patch is described by the Bag of Word descriptors based on SIFT [68] Interest Point descriptor. Second, the data structure is constructed by hashing all patches descriptors into several hash tables, using the Kernelized Locality Sensitives Hashing (KLSH) technique [62]. The goal of the hash function is to affect a group of words, with high probability to be similar to q, to the same index in the hash table. Finally, to retrieve a patch similar to the query keyword, its BoW descriptor is first computed and then compared to the hash table entries.
Again based-on the hash table, in [11, 10], the authors proposed the use of the Generalized Hough Transform (GHT) [21]. The word spotting task was formulated as a problem of finding GHT transformation parameters. Given a query keyword image (described by its GHT transformation parameters), the goal is to find the transformation's parameters that map the query to the reference images. The approach consists of two steps: *i*) Hash table construction and *ii*) Spotting. First, word's GHT transformation parameters of the reference dataset are saved in a hash table, then at the spotting step, the query parameters are compared to reference images using usual distances.

In the direct features matching category, a DTW-based approach is proposed in [105]. It consists of three steps: *i*) word segmentation, *ii*) features extraction and *iii*) word matching. The system takes as input a binary image and segments it into words via the "Run Length Smoothing Algorithm (RLSA)" [99]. At the features extraction step, each word is described by its vertical/horizontal histogram and upper/lower word profile. Finally, the DTW distance is used to match two words.
A Template Matching-based approach was proposed in [39]. First, the input documents are pre-processed for separating text from background and then binarized. Second, Normalized Cross Correlation (NCC) [101] is used to determine the position of the query word image in the input documents. It consists of calculating a similarity measure independent of the illumination conditions.

In this survey, it is important to mention the work of [54] where the authors proposed a framework to assist the word spotting task. The objective was to generate a data-structure that guides the synthesis of new samples. It can be seen as a data augmentation technique to generate new synthetic samples. The framework consists of two steps: *i*) Letter Connectivity Map (LCM) Construction and *ii*) Synthesizing words. To construct the LCM, multiple instances of each letter various shapes are

needed (since an Arabic letter shape varies by its position in the word). The LCM is constructed from the training dataset. Once the letter connectivity map is available, the framework can synthesize the pictorial representation of any Arabic word or sentence from their text representation. The synthesized words can be used in word spotting task and many other historical document applications to enrich the training dataset.

6.3. Discussion

Compared to the others APHAD tasks, the word spotting was well investigated in the literature. In fact, retrieving information from documents is very useful for users and specially, for scientists, historians, etc.

The performance of the word spotting process depends on other preliminary tasks such as word segmentation. To reduce the impact of such tasks, the researchers propose to use the sliding window technique or a segmentation-free approach. In our opinion, word segmentation and word spotting tasks are independent and should be investigated separately. Moreover, to properly compare the performance of two word spotting approaches, the same segmentation method should be used.

Note that the word spotting task becomes more interesting when the keyword searching is performed on a large dataset. As shown in Tab.7, the used datasets have a small size. It would be more interesting to evaluate the word spotting task performance on larger and more various datasets (printed, handwritten, degraded, low quality, low resolution, etc).

Moreover, the absence of large annotated datasets limits the use of advanced machine learning methods such as Deep Learning. To overcome this problem, recent research works [54] aim to augment the size of the training dataset using synthesis techniques. In this context, the data augmentation step should be investigated further.

7. Datasets & Softwares

7.1. Datasets

To evaluate computer vision algorithms, researchers need datasets. In the literature, only few historic Arabic document datasets exist. Existing datasets can be roughly regrouped into two categories: *i*) Degraded datasets [91, 92] and *ii*) Not-degraded datasets [3, 74, 52, 71, 59]. In this section, we present the characteristics of each dataset and the APHAD applications that can be evaluated using that datasets. Figure 5 shows samples from datasets.

7.1.1. WAHD Dataset

It is an Israeli dataset [3], published in 2017 to evaluate writer classification task . It is composed of $N = 353$ different manuscripts collected from two sources: *i*) the Islamic Heritage Project (IHP) (333 manuscripts) and *ii*) the National Library in Jerusalem (NLJ) (20 manuscripts).

Figure 5: Samples from the available datasets, from left to right: VML-HD, WAHD, MHDID, Ibn-Sina

Table 8: Summary of Datasets of Arabic Historical Documents.

Reference	Name	#writers	#manuscripts	#pages	Applications	Year
[3]	WAHD: IHP subset	302	333	36969	Writer classification	2017
[3]	WAHD: NLJ subset	less than 20	20	7007		2017
[74]	HADARA80P	1	1	80	Word spotting	2014
[52]	VML-HD	5	5	680		2017
[71]	IBN-SINA	1	1	51		2010
[59]	BADAM	less than 42	42	400	Document analysis	2019
[91]	MHDID	less than 130	less than 130	335	Quality assessment	2018
[92]	VDIQA	–	–	177		2018

• The 333 IHP manuscripts were written by 302 different writers: 23 of them are known writers and the remaining writers are unknown. Each scribe wrote one manuscript (single writer) or more (multiple writer). In more detail:

– Eleven scribes are multiple writers. They wrote 42 manuscripts consisting of 2313 pages.

– The remaining twelve scribes are single writers and wrote 2108 pages.

The remaining 279 manuscripts, written by unknown writers, were considered, all, as single writers. These manuscripts composed of 32,548 pages.

• The NLJ manuscripts were photographed using a high-quality camera from 1 meter distance. The NLJ dataset is composed of 20 books of 7007 pages. These books have been written between the 14^{th} and the 20^{th} century.

7.1.2. VML-HD Dataset

It is an Israeli dataset [52], published in 2017 for word spotting and word recognition tasks. The dataset is composed of five manuscripts; each manuscript had been written by one distinct scribe, from the year 1088 to 1451. The dataset contains 680 pages fully annotated on the sub-word level. It

contains a total of 121,636 sub-word appearances from 1,731 classes. The variation in a word may be any combination of noise, writing style and font size.

7.1.3. HADARA80P dataset

It was proposed in 2014 by a Germany Institute [74]. It is composed of 80 pages of the historical book "Taaun". These pages correspond to the book cover and the first 79 first pages. In addition to the images, XML files containing the ground truth are available. The HADARA80P dataset contains 16,720 annotated words. It also includes a selection of 25 pre-defined keywords. The number of occurrences of keywords full match in the dataset is between 5 occurrences and 349 occurrences.

7.1.4. IBN-SINA Dataset

This dataset [71] was proposed, in 2010, by a Canadian University and provided by the Institute of Islamic Studies (IIS). The dataset contains 51 folio of one manuscript, the "Kitab Kashf al-tamwihat sharh al-Tanbihat". The dataset was manually annotated on sub-words level, to extract 20,722 shapes (connected components). The shapes can correspond to words, sub-words or letters. Based on clustering algorithms, the dataset can include more than 1,000 basis connected components (keywords). The dataset can be used for word recognition/spotting tasks.

7.1.5. BADAM Dataset

The public BADAM dataset [59] contains 42 manuscripts from four digital collections of the Arabic and Persian languages. It contains 400 annotated pages from different domains and time periods. From each manuscript, 10 pages were chosen, except for four shorter manuscripts containing only 3 to 7 pages.

7.1.6. MHDID Dataset: Multi-distortion Historical Document Image Database

The MHDID dataset [91] contains 335 historical document images with resolution of 1024x1280 pixels. It includes various distortion types that can be roughly classified into four categories: *i*) Paper translucency (88 images), *ii*) Stain (113 images), *iii*) Readers annotations (61 images) and *iv*) Worn holes (73 images) [66]. In more detail:

- **Paper translucency:** is an internal degradation and appears when a document is written on both sides of translucent paper.
- **Stains:** are external noises that may occur during the printing process.
- **Readers annotation:** sometimes readers add notes and highlight sentences in the document. Readers annotation can produce noise.
- **Worm holes:** are the dig tunnels in old documents.

For information, the dataset was selected from the library of Qatar University. The library includes, in total, around 7,000 degraded documents with at least two types of degradations per image; it is collected from 130 different books edited during the 1^{st} to the 14^{th} islamic centuries period.

7.1.7. VDIQA Dataset: Visual Document Image quality assessment

The VDIQA dataset [92] contains 177 historical document images with resolution of 1024x1280 pixels. It includes various distortion types. In fact, each image contains at least one or more types of the following physical noises common in historical document images: paper deterioration, bleed-through, show-through, alien ink, ink-smear, ink-noise and faded ink. The dataset was collected from the library of Qatar University and documents were edited between the 1^{st} and the 14^{th} islamic centuries period.

7.1.8. Summary of Datasets

As shown in Tab.8, only one dataset, composed of two subsets, was proposed to evaluate Writer Classification algorithms. Moreover, only 8% of manuscript writer's are known. In fact, the need to have more data still a challenge. Regarding word spotting tasks, three datasets were proposed; two of them have less than 100 pages. In fact, despite the availability of large digital books, pages should be annotated. Since this task is mostly done manually; having a large annotated datasets still a challenge for researchers.

Concerning the degraded datasets, the two datasets, MHDID and VDIQA, was proposed essentially to evaluate and assess the quality of degraded documents. The major difference between the two datasets are: *i*) MHDID have twice the number of pages in VDIQA and *ii*) contrary to MHDID, we have any idea about the types of distortion in each image of VDIQA.

7.2. Softwares

In the literature, some softwares are proposed to process Historical Arabic Documents and aims essentially to facilitate the use of documents. These softwares are essentially related to document annotation, document image preprocessing, information search/retrieval and word recognition.

Regarding annotation tools, in [24], an interactive Web-based system (WebGT) is proposed to generate ground truth. Given an input document image, users can annotate the most basic elements and group them to create higher hierarchy elements. The system supports two file formats: XML format and Comma Separated Values (CSV) format. The tool is free for use. In [57], an interactive annotation tool is proposed. It takes as input scanned document and outputs XML file that contains the annotation information for the respective words. The tool includes three major functionalities such as: *i*) Image binarization, *ii*) Word segmentation and *iii*) Annotation. For the binarization task, authors integrate existing algorithms such as the Otsu algorithm [73]. Concerning the implementation of line and word segmentations, authors include the approach proposed in [47]. To correct segmentation, the interface allows user to rectify word segmentation errors and add their annotations. Finally, the system enables the user to store the annotated text and the corresponding word locations in XML files.

Regarding image processing softwares, a tool is proposed in [25]. It includes preprocessing and analysis tasks. As preprocessing tasks, it integrates binarization, skew correction [94] and Back-

ground/Foreground separation [25] . Concerning analysis tasks, it includes text/graphic segmentation [26] and text line segmentation [103]. In both cases, authors mostly integrate existing approaches. The tool is not given to the public. In [13], a framework is proposed to simplify the reading of historical Arabic documents with complex layout. It includes algorithms for text localization, classification and dewarping. Authors integrate existing algorithms for each task. For text localization, the coarse-fine approach of [14] is extended to separate the side-text from the main text. For line detection, the approach proposed in [32] is integrated into the tool. For text dewarping, a simple approach based on affine transformations is proposed. Similar to [25], the tool is not available.

As searching tool, a search engine of ancient Arabic manuscripts based on meta-data and XML annotations is proposed in [70]. The search functionality is done via a intuitive user interface taking as input the handwritten transcribed documents and the indexed images corresponding to users' queries. Only the abstract of this work is available in the publisher website; details are not provided.

As recognition tools, a software for Optical Character Recognition (OCR) is proposed in [95]. It includes the kaldi-based recognizer approach [77]. The tool takes as input documents images and outputs XML files contains transcriptions and layout information. The tool is not available for the public.

Discussions

There are few software proposed to manipulate historical Arabic documents and exploit its contents. Mostly, authors collect theirs approaches, proposed for different tasks, in one tool by adding a graphical interface to manage inputs/outputs. These softwares are essentially related to document annotation and preprocessing. Most of these software are private and not available for users. This non-availability of tools not allows us to confirm their performances on recent HAD datasets. The annotation tool, WebGT, is free and widely used by several researchers to generate ground truth.

8. Conclusion

This paper presented a survey on existing approaches to analyze and understand the content of historical Arabic documents. Four major applications were identified: *i*) Document analysis, *ii*) Writer classification, *iii*) Document classification and *iv*) Data retrieval. For each task, a review of existing solutions was presented and recommendations were suggested. Moreover, the public datasets used to evaluate existing approaches were summarized.

our future extensions of this work is to evaluate tasks on datasets include degraded documents. Moreover, our research direction is to investigate the data augmentation step that facilitate the use of advanced machine learning techniques such as Deep Learning.

References

[Abe] Juma al-majid center for culture and heritage. `http://www.almajidcenter.org`. Accessed: 2018-11-02.

[2] Abdalhaleem, A., Barakat, B. K., and El-Sana, J. (2018). Case study: Fine writing style classification using siamese neural network. In 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pages 62–66.

[3] Abdelhaleem, A., Droby, A., Asi, A., Kassis, M., Asam, R. A., and El-Sana, J. (2017). WAHD: A database for writer identification of arabic historical documents. In 1st International Workshop on Arabic Script Analysis and Recognition, ASAR 2017, Nancy, France, April 3-5, 2017, pages 64–68.

[4] Alaasam, R., Barakat, B. K., and El-Sana, J. (2018). Synthesizing versus augmentation for arabic word recognition with convolutional neural networks. In 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pages 114–118.

[5] Alaasam, R., Kurar, B., Kassis, M., and El-Sana, J. (2017). Experiment study on utilizing convolutional neural networks to recognize historical arabic handwritten text. In 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), pages 124–128.

[6] AlKhateeb, J. H., Ren, J., Jiang, J., and Al-Muhtaseb, H. (2011). Offline handwritten arabic cursive text recognition using hidden markov models and re-ranking. Pattern Recognition Letters, 32(8):1081 – 1088.

[7] Althobaiti, H. and Lu, C. (2017). A survey on arabic optical character recognition and an isolated handwritten arabic character recognition algorithm using encoded freeman chain code. In 2017 51st Annual Conference on Information Sciences and Systems (CISS), pages 1–6.

[8] Amin, K. M., Elfattah, M. A., Hassanien, A. E., and Schaefer, G. (2014). A binarization algorithm for historical arabic manuscript images using a neutrosophic approach. In 2014 9th International Conference on Computer Engineering Systems (ICCES), pages 266–270.

[9] Amrouch, M. and Rabi, M. (2018). Deep neural networks features for arabic handwriting recognition. In Ezziyyani, M., Bahaj, M., and Khoukhi, F., editors, Advanced Information Technology, Services and Systems, pages 138–149, Cham. Springer International Publishing.

[10] Aouadi, N. and Echi, A. K. (2014). Prior segmentation of old arabic manuscripts by separator word spotting. In 2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR), pages 31–36.

[11] Aouadi, N. and Kacem, A. (2011). Word Spotting for Arabic Handwritten Historical Document Retrieval using Generalized Hough Transform.

[12] Asi, A., Abdalhaleem, A., Fecker, D., Märgner, V., and El-Sana, J. (2017). On writer identification for arabic historical manuscripts. Int. J. Doc. Anal. Recognit., 20(3):173–187.

[13] Asi, A., Cohen, R., Kedem, K., and El-Sana, J. (2015). Simplifying the reading of historical manuscripts. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pages 826–830.

[14] Asi, A., Cohen, R., Kedem, K., El-Sana, J., and Dinstein, I. (2014). A coarse-to-fine approach for layout analysis of ancient manuscripts. In 14th International Conference on Frontiers in Handwriting Recognition, ICFHR 2014, Crete, Greece, September 1-4, 2014, pages 140–145.

[15] Asi, A., El-Sana, J., and Mrgner, V. (2012). Hierarchical scheme for arabic text recognition. In 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), pages 1266–1271.

[16] Asi, A., Rabaev, I., Kedem, K., and El-Sana, J. (2011a). User-assisted alignment of arabic historical manuscripts. In Proceedings of the 2011 Workshop on Historical Document Imaging and Processing, HIP@ICDAR 2011, Beijing, China, September 16-17, 2011, pages 22–28.

[17] Asi, A., Saabni, R., and El-Sana, J. (2011b). Text line segmentation for gray scale historical document images. In Proceedings of the 2011 Workshop on Historical Document Imaging and Processing, HIP '11, pages 120–126, New York, NY, USA. ACM.

[18] Avidan, S. and Shamir, A. (2007). Seam carving for content-aware image resizing. ACM Trans. Graph., 26(3).

[19] Awaida, S. M. (2015). Text independent writer identification of arabic manuscripts and the effects of writers increase. In International Conference on Computer Vision and Image Analysis Applications, pages 1–4.

[20] Bai, X., Latecki, L. J., and Liu, W. (2007). Skeleton pruning by contour partitioning with discrete curve evolution. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(3):449–462.

[21] Ballard, D. (1981). Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition, 13(2):111 – 122.

[22] Barakat, B. K., Alasam, R., and El-Sana, J. (2018). Word spotting using convolutional siamese network. In 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), pages 229–234.

[23] Barakat, B. K. and El-Sana, J. (2018). Binarization free layout analysis for arabic historical documents using fully convolutional networks. In 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pages 151–155.

[24] Biller, O., Asi, A., Kedem, K., El-Sana, J., and Dinstein, I. (2013). Webgt: An interactive web-based system for historical document ground truth generation. In 2013 12th International Conference on Document Analysis and Recognition, pages 305–308.

[25] Boussellaa, W., Zahour, A., Taconet, B., Alimi, A., and Benabdelhafid, A. (2007). Praad: Pre-processing and analysis tool for arabic ancient documents. In Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), volume 2, pages 1058–1062.

[26] Boussellaa, W., Zahour, A., Taconet, B., Benabdelhafid, A., and Alimi, A. (2006). Segmentation texte /graphique : Application au manuscrits Arabes Anciens. pages 139–144. SDN06.

[27] Breuel, T. and Shafait, F. (2010). Automlp: Simple, effective, fully automated learning rate and size adjustment. In The Learning Workshop. Online. Extended Abstract.

[28] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a "siamese" time delay neural network. In Proceedings of the 6th International Conference on Neural Information Processing Systems, NIPS'93, pages 737–744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[29] Bukhari, S. S., Breuel, T. M., Asi, A., and El-Sana, J. (2012). Layout analysis for arabic historical document images using machine learning. In 2012 International Conference on Frontiers in Handwriting Recognition, pages 639–644.

[30] Bulacu, M., Schomaker, L., and Brink, A. (2007). Text-independent writer identification and verification on offline arabic handwriting. In Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), volume 2, pages 769–773.

[31] Cohen, R., Dinstein, I., El-Sana, J., and Kedem, K. (2014). Using scale-space anisotropic smoothing for text line extraction in historical documents. In Image Analysis and Recognition - 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part I, pages 349–358.

[32] Cohen, R., Rabaev, I., El-Sana, J., Kedem, K., and Dinstein, I. (2015). Aligning transcript of historical documents using energy minimization. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pages 266–270.

[33] Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. 13:21– 27.

[34] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 886–893.

[35] El-etriby, S. S. and Amin, K. M. (2010). Detection and correction of deformed historical arabic manuscripts. In Computer and Communication Engineering (ICCCE), 2010 International Conference on, pages 1–6.

[36] Elfakir, Y., Khaissidi, G., Mrabti, M., and Chenouni, D. (2015). Article: Handwritten arabic documents indexation using hog feature. International Journal of Computer Applications, 126(9):14–18. Published by Foundation of Computer Science (FCS), NY, USA.

[37] Elfattah, M. A., Hassanien, A. E., Mostafa, A., Ali, A. F., Amin, K. M., and Mohamed, S. (2015). Artificial bee colony optimizer for historical arabic manuscript images binarization. In 2015 11th International Computer Engineering Conference (ICENCO), pages 251–255.

[38] Elleuch, M., Tagougui, N., and Kherallah, M. (2015). Deep learning for feature extraction of arabic handwritten script. In Azzopardi, G. and Petkov, N., editors, Computer Analysis of Images and Patterns, pages 371–382, Cham. Springer International Publishing.

[39] Faisal, T. and AlMaadeed, S. (2017). Enabling indexing and retrieval of historical arabic manuscripts through template matching based word spotting. In 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), pages 57–63.

[40] Fecker, D., Asi, A., Pantke, W., Mrgner, V., El-Sana, J., and Fingscheidt, T. (2014a). Document writer analysis with rejection for historical arabic manuscripts. In 2014 14th International Conference on Frontiers in Handwriting Recognition, pages 743–748.

[41] Fecker, D., Asit, A., Mrgner, V., El-Sana, J., and Fingscheidt, T. (2014b). Writer identification for historical arabic documents. In 2014 22nd International Conference on Pattern Recognition, pages 3050–3055.

[42] Fogel, I. and Sagi, D. (1989). Gabor filters as texture discriminator. Biol. Cybern., 61(2):103–113.

[43] Gatos, B., Stamatopoulos, N., and Louloudis, G. (2011). Icdar2009 handwriting segmentation contest. Int. J. Doc. Anal. Recognit., 14(1):25–33.

[44] Giotis, A. P., Sfikas, G., Gatos, B., and Nikou, C. (2017). A survey of document image word spotting techniques. Pattern Recogn., 68(C):310–332.

[45] Guo, Y., Cheng, H., Tian, J., and Zhang, Y. (2009). A novel approach to speckle reduction in ultrasound imaging. Ultrasound in Medicine & Biology, 35(4):628 – 640.

[46] Hassanien, A. E., Elfattah, M. A., Aboulenin, S., Schaefer, G., Zhu, S. Y., and Korovin, I. (2016). Historic handwritten manuscript binarisation using whale optimisation. In 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 003842–003846.

[47] Hassane, A. (2013). A robust method for line and word segmentation in handwritten text. Qatar Foundation Annual Research Forum Proceedings, page ICTP 057.

[48] Hussain, R., Raza, A., Siddiqi, I., Khurshid, K., and Djeddi, C. (2015). A comprehensive survey of handwritten document benchmarks: structure, usage and evaluation. EURASIP Journal on Image and Video Processing, 2015(1):46.

[49] Indian, A. and Bhatia, K. (2017). A survey of offline handwritten hindi character recognition. In 2017 3rd International Conference on Advances in Computing,Communication Automation (ICACCA) (Fall), pages 1–6.

[50] Jayech, K., Mahjoub, M. A., and Amara, N. E. B. (2016). Synchronous multi-stream hidden markov model for offline arabic handwriting recognition without explicit segmentation. Neurocomputing, 214:958–971.

[51] Karaboga, D. and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Journal of Global Optimization, 39(3):459–471.

[52] Kassis, M., Abdalhaleem, A., Droby, A., Alaasam, R., and El-Sana, J. (2017a). Vml-hd: The historical arabic documents dataset for recognition systems. In 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), pages 11–14.

[53] Kassis, M. and El-Sana, J. (2014). Word spotting using radial descriptor. In 2014 14th International Conference on Frontiers in Handwriting Recognition, pages 387–392.

[54] Kassis, M. and El-Sana, J. (2016). Scribble based interactive page layout segmentation using gabor filter. In 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 13–18.

[55] Kassis, M. and El-Sana, J. (2016). Word spotting using radial descriptor graph. In 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 31–35.

[56] Kassis, M., Nassour, J., and El-Sana, J. (2017b). Alignment of historical handwritten manuscripts using siamese neural network. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 01, pages 293–298.

[57] Khader, H., Al-Marridi, A., Alpona, H., Kunhoth, S., Hassaine, A., and Al-maadeed, S. (2014). An interactive annotation tool for indexing historical manuscripts. In 2014 World Symposium on Computer Applications Research (WSCAR), pages 1–4.

[58] Khaissidi, G., Elfakir, Y., Mrabti, M., El-Yacoubi, M. A., Chenouni, D., and Lakhliai, Z. (2016). Segmentation-free word spotting for handwritten arabic documents. IJIMAI, 4(1):6–10.

[59] Kiessling, B., Ezra, D. S. B., and Miller, M. T. (2019). BADAM: A public dataset for baseline detection in arabic-script manuscripts. CoRR, abs/1907.04041.

[60] Knuth, D. E. (1993). The Stanford GraphBase - a platform for combinatorial computing. ACM.

[61] Kolcz, A., Alspector, J., Augusteijn, M., Carlson, R., and Viorel Popescu, G. (2000). A line-oriented approach to word spotting in handwritten documents. Pattern Analysis & Applications, 3(2):153–168.

[62] Kulis, B. and Grauman, K. (2012). Kernelized locality-sensitive hashing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(6):1092–1104.

[63] Levi, G. and Montanari, U. (1970). A grey-weighted skeleton. Information and Control, 17(1):62 – 91.

[64] Likforman-Sulem, L., Zahour, A., and Taconet, B. (2007). Text line segmentation of historical documents: A survey. Int. J. Doc. Anal. Recognit., 9(2):123–138.

[65] Lillholm, M. and Griffin, L. (2008). Novel image feature alphabets for object recognition. In 2008 19th International Conference on Pattern Recognition, pages 1–4.

[66] Lins, R. D. (2009). A taxonomy for noise in images of paper documents - the physical noises. In Kamel, M. and Campilho, A., editors, Image Analysis and Recognition, pages 844–854, Berlin, Heidelberg. Springer Berlin Heidelberg.

[67] Lorigo, L. M. and Govindaraju, V. (2006). Offline arabic handwriting recognition: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(5):712–724.

[68] Lowe, D. G. (2001). Local feature view clustering for 3d object recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pages 682–688.

[69] Maalej, R. and Kherallah, M. (2018). Convolutional neural network and blstm for offline arabic handwriting recognition. In 2018 International Arab Conference on Information Technology (ACIT), pages 1–6.

[70] Makhfi, N. E., Bannay, O. E., Benslimane, R., and Rais, N. (2011). Search engine of ancient arabic manuscripts based on metadata and xml annotations. In 2011 Colloquium in Information Science and Technology, pages 10–10.

[71] Moghaddam, R. F., Cheriet, M., Adankon, M. M., Filonenko, K., and Wisnovsky, R. (2010). IBN SINA: a database for research on processing and understanding of Arabic manuscripts images. In DAS '10: Proceedings of the 8th IAPR International Workshop on Document Analysis Systems, pages 11–18, New York, NY, USA. ACM.

[72] Naegel, B. and Wendling, L. (2010). A document binarization method based on connected operators. Pattern Recogn. Lett., 31(11):1251–1259.

[73] Otsu, N. (1979). A Threshold Selection Method from Gray-level Histograms. IEEE Transactions on Systems, Man and Cybernetics, 9(1):62–66.

[74] Pantke, W., Dennhardt, M., Fecker, D., Mrgner, V., and Fingscheidt, T. (2014). An historical handwritten arabic dataset for segmentation-free word spotting - hadara80p. In 2014 14th International Conference on Frontiers in Handwriting Recognition, pages 15–20.

[75] Parvez, M. T. and Mahmoud, S. A. (2013). Offline arabic handwritten text recognition: A survey. ACM Comput. Surv., 45(2):23:1–23:35.

[76] Pechwitz, M., Maddouri, S. S., Mrgner, V., Ellouze, N., and Amiri, H. (2002). Ifn/enit - database of handwritten arabic words. In In Proc. of CIFED 2002, pages 129–136.

[77] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

[78] Poznanski, A. and Wolf, L. (2016). Cnn-n-gram for handwritingword recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2305–2314.

[79] Rabaev, I., Biller, O., El-Sana, J., Kedem, K., and Dinstein, I. (2013). Text line detection in corrupted and damaged historical manuscripts. In 2013 12th International Conference on Document Analysis and Recognition, pages 812–816.

[80] Rabaev, I., Cohen, R., El-Sana, J., and Kedem, K. (2015). Aligning transcript of historical documents using dynamic programming. In Document Recognition and Retrieval XXII, San Francisco, California, USA, February 11-12, 2015., page 94020I.

[81] Rabaev, I., Dinstein, I., El-Sana, J., and Kedem, K. (2014). Segmentation-free keyword retrieval in historical document images. In Campilho, A. and Kamel, M., editors, Image Analysis and Recognition, pages 369–378, Cham. Springer International Publishing.

[82] Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286.

[83] Saabni, R., Asi, A., and El-Sana, J. (2014). Text line extraction for historical document images. Pattern Recognition Letters, 35:23 – 33. Frontiers in Handwriting Processing.

[84] Saabni, R. and El-sana, J. (2008). Keyword searching for arabic handwritten documents. In In The 11th International Conference on Frontiers in Handwriting recognition (ICFHR2008), Montreal, pages 716–722.

[85] Saabni, R. and El-Sana, J. (2009). Hierarchical on-line arabic handwriting recognition. In 2009 10th International Conference on Document Analysis and Recognition, pages 867–871.

[86] Saabni, R. and El-Sana, J. (2011). Language-independent text lines extraction using seam carving. In 2011 International Conference on Document Analysis and Recognition, pages 563–568.

[87] Saabni, R. and El-Sana, J. (2013). Keywords image retrieval in historical handwritten arabic documents. J. Electronic Imaging, 22(1):013016.

[88] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 26(1):43–49.

[89] Sauvola, J. and Pietikinen, M. (2000). Adaptive document image binarization. Pattern Recognition, 33(2):225 – 236.

[90] Schantz, H. F. (1982). History of OCR, Optical Character Recognition. Recognition Technologies Users Association.

[91] Shahkolaei, A., Beghdadi, A., Al-maadeed, S., and Cheriet, M. (2018a). Mhdid: A multi-distortion historical document image database. In 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pages 156–160.

[92] Shahkolaei, A., Nafchi, H. Z., Al-Maadeed, S., and Cheriet, M. (2018b). Subjective and objective quality assessment of degraded document images. Journal of Cultural Heritage, 30:199 – 209.

[93] Sivic, J. and Zisserman, A. (2003). Video google: a text retrieval approach to object matching in videos. In Proceedings of the 9th IEEE International Conference on Computer Vision, volume 2, pages 1470–1477.

[94] Srihari, S. N. and Govindaraju, V. (1989). Analysis of textual images using the hough transform. Machine Vision and Applications, 2(3):141–153.

[95] Stahlberg, F. and Vogel, S. (2016). Qatip – an optical character recognition system for arabic heritage collections in libraries. In 2016 12th IAPR Workshop on Document Analysis Systems (DAS), pages 168–173.

[96] Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., and Alaei, A. (2013). Icdar 2013 handwriting segmentation contest. In 2013 12th International Conference on Document Analysis and Recognition, pages 1402–1406.

[97] Tagougui, N., Kherallah, M., and Alimi, A. M. (2013). Online arabic handwriting recognition: a survey. International Journal on Document Analysis and Recognition (IJDAR), 16(3):209–226.

[98] Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. Springer-Verlag, Berlin, Heidelberg.

[99] Wong, K. Y., Casey, R. G., and Wahl, F. M. (1982). Document analysis system. IBM Journal of Research and Development, 26(6):647–656.

[100] Wshah, S., Govindaraju, V., Cheng, Y., and Li, H. (2010). A novel lexicon reduction method for arabic handwriting recognition. In 2010 20th International Conference on Pattern Recognition, pages 2865–2868.

[101] Yoo, J.-C. and Han, T. H. (2009). Fast normalized cross-correlation. Circuits, Systems and Signal Processing, 28(6):819.

[102] Zahour, A., Taconet, B., Mercy, P., and Ramdane, S. (2001). Arabic hand-written text-line extraction. In Proceedings of Sixth International Conference on Document Analysis and Recognition, pages 281–285.

[103] Zahour, A., Taconet, B., and Ramdane, S. (2004). Contribution à la segmentation de textes manuscrits anciens.

[104] Zayene, O., Touj, S. M., Hennebert, J., Ingold, R., and Amara, N. E. B. (2018). Multi-dimensional long short-term memory networks for artificial arabic text recognition in news video. IET Computer Vision, 12(5):710–719.

[105] Zirari, F., Ennaji, A., Nicolas, S., and Mammass, D. (2013). A methodology to spot words in historical arabic documents. In AICCSA, pages 1–4. IEEE Computer Society.