Technological Communication

Biosci. Biotech. Res. Comm. 12(2): 283-296 (2019)

BBRC
Bioscience Biotechnology Research Communications

Arabic calligraphy, typewritten and handwritten using optical character recognition (OCR) system

Hassanin M. Al-Barhamtoshy[1], Kamal M. Jambi[1], Hany Ahmed[2], Shaimaa Mohamed[3], Sherif M. Abdo[2] and Mohsen. A. Rashwan[3]

[1]*Department of Information Technology, Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*
[2]*Faculty of Computers and Information Systems, Cairo University, Egypt*
[3]*Electronics and Communication Department, Cairo University, Egypt*

ABSTRACT

This paper describes an Omni OCR system for recognizing typewritten and handwritten Arabic texts documents. The proposed system of the Arabic OCR system can be classified into four main phases. The first phase is the pre-processing phase; it focuses on binarizing, skewing treatment, framing, and noise removing from the prepared documents (dataset). The second phase aims to segment the preprocessed documents into lines and words. Two main tasks are pointed during this phase: language model with the used Arabic dictionary, and the detection of segmented lines and segmented words. The third phase is features extraction phase; it is used to extract features for each segmented line/word according to the used language model. Finally, the classifier or the recognizer will be used to recognize each word/line into a text stream. Therefore, scientific evaluation of the four phases will be applied to measure the accuracy of the Arabic OCR system. The recognition approachis based on Hidden Markov Models (HMM) with the prepared datasets and software development tool are discussed and introduced. State of the art OCR's recognition systems are now capable to perform accuracy of 70% for unconstrained Arabic texts. However, this outline is still far away from what is required in a lot of useful applications.In other words, this paper describes a proposed approach based on language model with ligature and overlap characters for the pro-posed Arabic OCR. Therefore, a posterior word-based approach is used with tri-gram model to recognize the Arabic text. Features are extracted from images of words and generated pattern using the proposed solution. We test our proposed OCR system in different categories of Arabic documents: early printed or typewritten, printed, historical and calligraphy documents. The test bed of our system gives 12.5%-character error rate compared to the best OCR of other systems.

KEY WORDS: ARABIC OCR, SEGMENTATION, FEATURE EXTRACTION, CALLIGRAPHY, TYPEWRITTEN, HANDWRITTEN, HMM

ARTICLE INFORMATION:

Corresponding Author: Hassanin M. Al-Barhamtoshy
Received 6th April, 2019
Accepted after revision 29th June, 2019
BBRC Print ISSN: 0974-6455
Online ISSN: 2321-4007 CODEN: USA BBRCBA
Thomson Reuters ISI ESC / Clarivate Analytics USA
Crossref Clarivate Analytics
NAAS Journal Score 2019: 4.31 SJIF: 4.196

Online Contents Available at: http//www.bbrc.in/
DOI: 10.21786/bbrc/12.2/11

INTRODUCTION

Currently, OCR systems achieve important role in document analysis and content retrieving with high accuracy. Though, Arabic document analysis and Arabic retrieving systems have more challenges (Stahlberg & Vogel, 2016). The document outline analysis is the method of classifying and labeling the zones of the image documents into a segments of text documents. Somewhat OCR system involves the segmentation of text regions from non-textual ones. Nevertheless, Arabic text regions play diverse its logical parts inside the manuscript (title, subtitle, notes, cross-references, etc.) and this kind of semantic labeling is the opportunity of the layout analysis. Figure 1 shows sample of the challenges in Arabic scripts. Due to these properties the current performance of state of art Arabic OCR systems is much lagging compared with the performance of other Latin-based OCR systems.

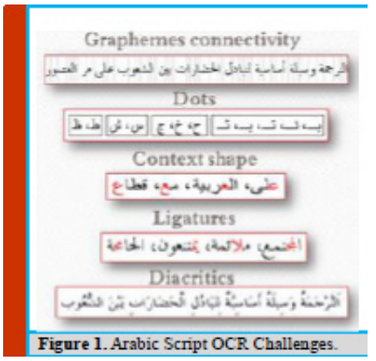


Figure 1. Arabic Script OCR Challenges.

Recent submissions that used Arabic text segmentation and Arabic OCR systems are discussed in (Stahlberg & Vogel, 2016). Other published applications to recognize Arabic handwritten is illustrated in (Cao & Natarajan, 2014). A wide range of document analysis, layouts processing to locate text blocks and none text blocks and separating between them have been presented in (Nobile & Suen, 2014). Script identification has been a real challenge in OCR and information retrieval systems (Pal & Dash, 2014). The most state of the art papers are published into the OCR machine printed character domain.Moreover, segmentation process leads to errors more than the other processes in document analysis and processing (Cao & Natarajan, 2014).

Text classification addresses the problem of document analysis into "classifying modern machine printed text, handwritten text and historical typewritten text from degraded noisy document" (Zha, et al., 2014). Therefore, text classification approach based on iVector is proposed in (Zha, et al., 2014). The text line is classified using SVM in iVector space. An OCR for multilingual documents (Amazigh-Frensh) has been proposed in (El-Gajoui, et al., 2015). Hence, Amazigh writing transcription methods are employed using Latin or Arabic alphabet. Acomprehensive survey on Arabic cursive scene text recognition, and the text having variations infont styles, size, alignment, orientation, reflection, illumination change, blurriness and complexbackground had been illustrated in (Bin Ahmed, et al., 2019). Arabic text recognition system is presented using deep learning architecture, and text localization and feature extraction are also presented. Then, it injects such feature vectors to the HMM. Holistic Arabic printed word recognizer is introduced along with discrete Markov classifier, HMM toolkit (HTK), and discrete cosine transform (DCT). Five fonts are used, having size of 14 points with plain style. Additional details, technical points and paper analysis are presented in (Nashwan, et al., 2017).

The documents to be studied within the datasets are composed of different fields under different formats styles (document styles, sizes, colors ...). Figure 2 shows samples of documents with different fonts, styles, colors ...). Accordingly, different manipulation with respect to syntax, styles, and colors during the analytic processes are needed of the proposed system.This paper is organized as the following description. Section 2 introduces an overview of multi classification of OCR system and related definitions. Section 3 gives the proposed solution of the Arabic OCR system with preprocessing operations (binarization, skewing, frame removing, and segmentation modules). Section 4 provides a language model and a lexicon building with Arabic OCR dataset description. Section 5 introduces features extraction and HMM decoding description. Performance evaluation results will be discussed in section 6. Section 7 summarizes conclusion and future works.

Figure 2. Samples of Arabic documents

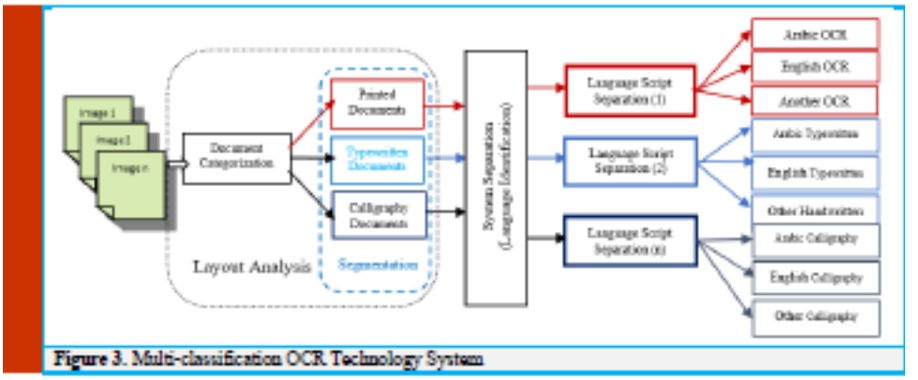


Figure 3. Multi-classification OCR Technology System

Analyzing the layout of documents is difficult, particularly in typewritten, handwritten, historical images, and manuscript notations. Any documents can be classified based on its category; either printed, typewritten, handwritten, or scripts and manuscripts documents. The documents require dedicated preprocessing modules to deal with some common properties, structure, clearing irrelevant objects or noises, language direction...etc. Therefore, the documents require pre-processing procedures because of some familiar properties such identifying unrelated objects or noises, language direction ...etc. An OCR system takes into consideration the different categories of such documents. Figure 3 illustrates a flow diagram of multi-classification that used in OCR technology.

Generally developing an OCR system framework for multi-script language documents is more difficult than a single-script OCR, due to features associated with the language models, structures, properties, styles, and the nature of writing. The first module in the layout analysis is to organize text and non-text regions. Therefore, given a document image, then converts it into a decomposition of smaller regions. Thus, these regions are classified as text or non-text elements. Such text regions are fed to the second module of the OCR system, in order to declare category of the document (Printed, calligraphy ... or typewritten). In this proposal, we are dealing with heterogeneous large-scale documents with wide varying structured category. Furthermore, there could be multi-page document with different languages. Accordingly, the language domain will be identified within the language script specification module.

The goal is to improve the OCR accuracy by creating auto selection of the OCR system. To enable this goal, the layout module aims to:

1. Identifying the document image category.
2. Detecting and segmenting text and non-text regions.

A script is a visual representation or an organized arrangement of distinct graphic characters in precise patterns known as the alphabet of a language Grapheme, allograph and glyph are needed to study the script of a language, and they contribute to the script formation, as well as designing fonts Any OCR can be carried out using one of the following steps:

- Building a general OCR that recognizes all words and characters of the alphabets in all possible languages.
- Building language separation module to identify each single script with related OCR engine.

Arabic historical documents are classified as one of the most important documents that includes historical, political, and ancient information over the world archives. Figure 4 shows two examples of Arabic historical documents and one multi script (Arabic/English) with different kinds of orientation.

Figure 4. Examples of Calligraphy and Handwritten Ancient Arabic Documents

1. Proposed Methods and Preprocessing Block Diagram

Several commercial systems have been developed based on images analysis, segments determination and features extraction techniques. Most of such documents contains a lot of challenges; such as documents layout forms, physical structure, low quality due to aging, etc. this in addition to; the Arabic documents challenges; such as language writing orientation, Arabic dots, and Arabic diacritics. The data flow diagram (DFD) of the preprocessing stage of document analysis is shown in Figure 5.

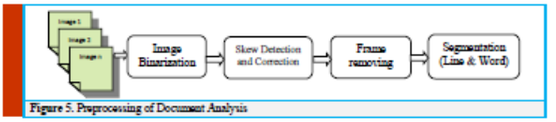


Figure 5. Preprocessing of Document Analysis

It starts with documents binarization (Al-Barhamtoshy, et al., 2019).

2.1 Binarization

The main goal of binarization is separating the foreground text from the background to enable and identify useful feature for character recognition (Al-Barhamtoshy, et al., 2019). According to document degradation, low contrast, shadows, background intensity, smear, etc.; we need further processing for these challenging tasks. Several algorithms are used in binarization for modern documents, the most commonly used algorithmsare the Otsu algorithm, the Niblack algorithm, and the Sauvola algorithm(Hadjadj, et al., 2016). In the binarization development process, Sauvola algorithm will be used with adaptation method to work with the Arabic documents. The proposed solution gives good results for the documents, as shown in figure 6.

Figure 6. Original Colored Documents and Binarized Documents (after binarization)

Local binarization computes a threshold t(i,j) for each pixel according to the following formula:

$$F2 = \frac{\sum_{i=1}^{i=H} i.r(i)}{\sum_{i=1}^{H} r(i)}$$

The threshold t(i , j) is calculated after getting the mean value of m(i , j) and standard deviation S(i , j) of the pixel intensity in window (w , w) centered around the pixel (i , j).

$$Fi = \frac{100 * Fi}{H} \quad i = 1,2,4,5,8$$

where R =128 for grayscale, and the default value of k= 0.34 gives best results.

2.2 Skew Detection and Correction

There are many methods used in documents skew detection. For example, projection profiles (El-Gajoui, et al., 2015), Hough transform, nearest-neighboring, segmentation (Al-Barhamtoshy & Rashwan, 2014), and coloring documents are used. In the projection profiles, horizontal and objective functions are computed to detect the skew rotating angle in the image. Such horizontal profile has consecutive local maxima and minima as the document is rotated to the computed angle. The objective function is used to minimize the searching process. Another technique is used by dividing the document into vertical slices, and then compute the horizontal projection profile for each slice. Also, wavelet decomposition can be used with inaccurate results. Hough transform can detect skew angle and estimate the lines in the documents and obtain their angles within time complexity (slow speed and it takes lot of memory).

In (Subrahmanyam, et al., 2018) a skew estimation is obtained based on image borderlines by using a run-length algorithm with large connected components in the whole documents. The proposed de-skewing algorithm presented by, first it segments the imaged document, then it detects the skewing angle using the segmented objects, but when it tries to obtain the skewing angle from page borders, it faced a problem that some pages does not contain any borders and also some borders are not continuous therefore, it will be hard to be extracted. This leads us to try another technique in skew detection, first we assume that the skewing angle ranges from -5 to 5, all borders and small components are removed, then the page is segmented into lines using a histogram technique, curve fitting is then used to obtain the skew angle for each line, and finally the average skew angle is calculated and rotate the whole page by this skew angle. The main advantage of this method is its ability to detect the skewing angle even if it is very small. Figure 7 shows the block diagram of the de-skewing algorithm.In our case, we use connected component to remove the marginal noise to perform documents cleaning. Then vertical histogram profile is used to obtain a rough estimation for text lines, and then we use a curve fitting to obtain the skew angle.

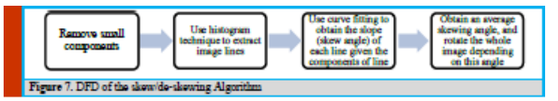


Figure 7. DFD of the skew/de-skewing Algorithm

Figure 8 display two documents before and after the skewing algorithm. A straight forward approach to correct skewing angle, we use the following formula:X = x cos θ + y sin θ, and Y = y sin θ – x cos θ. Where X and Y are the new point from the original (x , y) point.

2.3 Frame Detection and Removing

This section focuses on removing frames or border lines; especially in old documents. A rule lines are removed, and broken characters are reconstructed. In (Arvind, et al., 2008) line detection is done by examining the peak of connected component to obtain the page borders which is the most common and easy method to be implemented. Eliminating borderline noise is done by

Figure 8. documents before and after skewing

cleaningavailable connected components based on their size and aspect ratio. Projection profile is used to detect the location of border objects, analyze them and then remove them.In the current algorithm;the connected components methodis used to remove the page frame, the average width and height of all components are computed, then searching for components with width greater than *x**average width, and height greater than *x**average height, then exclude those components from the original image, where *x* is a constant value multiplied by the average width or average height, the value of *x* is obtained experimentally (threshold), and the best result was when x equal to 6 as shown in figure 9.

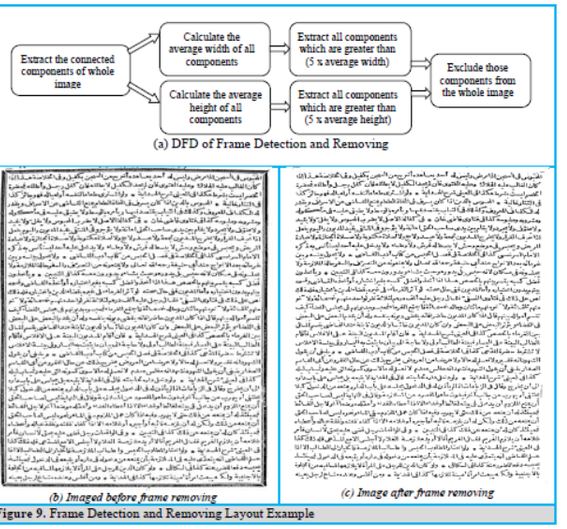


Figure 9. Frame Detection and Removing Layout Example

In Arabic typewritten and handwritten texts, usually lot of overlapped components can exist, and it is causing segmentation problems. Figure 10 includes samples of these overlapped components of Figure 9(a). The following algorithm is used to separate the overlapped components:

1. Obtain image's connected components.
2. Calculate average height, and width of all components.
3. Obtain components that are greater than (thresh1 *average height), and components that are greater than (thresh 2*average width), by trying different values thresh1 = 5, and thresh 2 = 10.

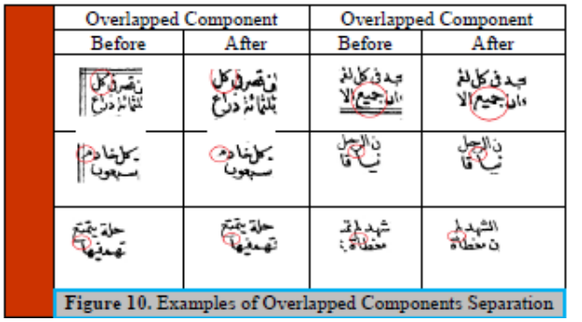


Figure 10. Examples of Overlapped Components Separation

4. Obtain the histogram for each component, and if peak exceeds a certain threshold (experimentally 0.7) remove the region around this peak.
5. IF there are no peaks in a component with a large height, split the component into two components from the middle.

2.4 Segmentation

Line segmentation is a technique to extract lines from a scanned document (Gatos, et al., 2006). Line segmentation techniques basicallycategorized into fourdifferent categories: Hough Transform (Gatos, et al., 2006), projection profiles, smearing, and segmentation based on connected components. In our case, we first use vertical projection profile but if there were some overlapped characters between lines, we try to segment lines using connected components techniques and this leads to segment lines with overlapped characters.

In the segmentation algorithm, three methodsof smearing, projection profiles, and connected component methods are used. The problem in smearing method was the overlapping between lines and each other's, in early printed Arabic books. Therefore, when smearing is done two lines may be connected to each other, and this will lead to the segmentation of two lines as one line. Projection profile methods give us better results than smearing methods, but this leaves us with the problem of having very small components in extracted lines as results from upper or lower components in other lines. That is the reason we move to these connected component methods. In the proposed segmented case; first vertical projection profile is invoked; but there were some overlapped characters between lines, thereforethe segmented lines using connected components techniques are invokedand this leads to segment lines with less overlapped characters, as shown in Figure 11. The output of the line segmentation are shown in figure 12 after frame removing in figure 10.

The connected components of the document are proposed to obtain a sequence of words to realize word indexing. Features victors of the words will be computed and stored in the query indexed database.

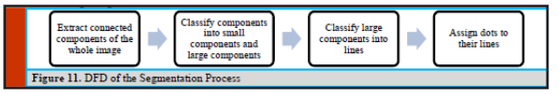


Figure 11. DFD of the Segmentation Process

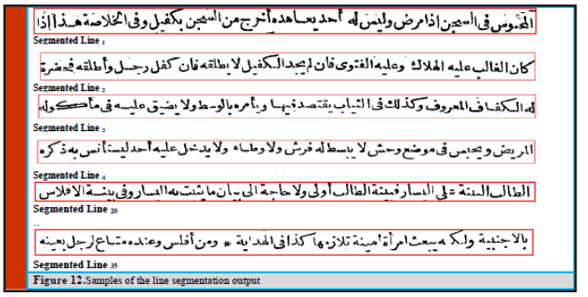


Figure 12.Samples of the line segmentation output

Line segmentation is done by following Algorithm steps (Figure 13):

1. Use horizontal projection profile to obtain an initial position for the separated lines, and baselines.
2. Divide the image into smaller regions, by taking the region between each two successive base lines, for each region do the following steps:

a. Obtain energy map.
b. Remove small regions from energy map (like removing dots and diacritics).
c. Fill the horizontal edges of the image with high values in order not to be included in path.
d. Find seam with minimum energy following the given steps

i. Set initial point in a path by using the approximated baseline.
ii. Obtaining the next step in the path, by searching the row with minimum value in a window with size equal to 2/5 of the image height.
iii. Translate the previous step to be as far as possible from the two components that lie above and below it.

The segmentation module is the most important phase in documents analysis and OCR recognition systems. Three types of segmentation are described, two are existing before: (1) Global and local approches, and the third is new: line segmentation approach.

Figure 13. Arabic documents (Before and After Segmentation process)

A. Global or Word Approach

It aims to segment the whole words (or Piece of A Word: PAWs) of the entire document, and keep track of the index/references of their location in the text.

B. Local or Letter Approach

It allows to segment words' images into letters, graphemes, and/or legitures. This kind of methodology may derive enhanced recognition percentage with slow in implemntition task.

C. Line or (n-grams) Approach

This approach is based on the whole line segments (number of words). It trys to use linguistic knowledge such as dictionary, lexicon, and/or language model with word probabilities. Such dictionary, lexicon and language model are generated before or during training phase of the OCR systems. Therefore, it trys to select and identify the correct segment hypothesis.

3. The Language Model and Lexicon Building Results

A statistical language model will be used based on probability distribution and function of sequences of likelihood word of different phrases. Consequently, an N-gram can be defined as an overlappingsequence of N Arabic words. For the line$_2$ (in Figure 12), the following sets of N-grams can be attained.

1- Uni- Grams (unigrams): فحضرة ,واطلقه ,رجل ,كفل ,فان ,يطلقه ,لا ,التكفيل ,يجد ,لم ,فان ,القنوى ,و عليه ,الهلاك ,عليه ,الغالب ,كان
2- Two-Grams (bigrams): التكفيل ,يجد التكفيل ,لم يجد ,فان لم ,القنوى فان ,و عليه القنوى ,الهلاك و عليه ,عليه الهلاك ,الغالب عليه ,كان الغالب
واطلقه فحضرة ,رجل واطلقه ,كفل رجل ,فان كفل ,يطلقه فان ,لا يطلقه ,لا
3- Tri-Grams (trigrams): القنوى فان لم ,و عليه القنوى فان ,الهلاك و عليه القنوى ,عليه الهلاك و عليه ,الغالب عليه الهلاك ,كان الغالب الهلاك
رجل واطلقه فحضرة ,كفل رجل واطلقه ,فان كفل رجل ,يطلقه فان التكفيل لا يطلقه ,لا يطلقه يجد التكفيل لا ,لم يجد التكفيل ,فان لم يجد
...
4- N-grams: The complete line:
كانالغالبعليهالهلاكو عليهالقنوىفانلميجدالتكفيليطلقهفانكفلرجلواطلقهفحضرة

Therefore, n-grams is said to include order of n by applying a sequence task of probability to arrange the desired probability of a line. The rule of the segmented line2 can be calculated according to the following:

$$P(\text{Linei}) = P(W1) * P(W2|W1) * P(W3|W2W1) * ..... P(Wn|Wn-1 ....W1)$$

Accordingly, Arabic corpora and lexicon are used to support the proposed language model.The proposed model is based on features extraction for each line/word during training or classification module. This module focuses on HMM based classifier with full language model data. Figure 14 displays the DFD of the proposed system for our syntactic classifier.

Once the Arabic imaged – documents is scanned, the pre-processing stage takes place (as discussed before). Therefore, the segmented lines/words are extracted from the imaged– document using appropriate algorithm. From each segmented lines/words, many features will be selected and extracted by utilizing the vocabulary of syntactic language model data base. The language model can reduce the computational overheadand enhance the recognition rate.

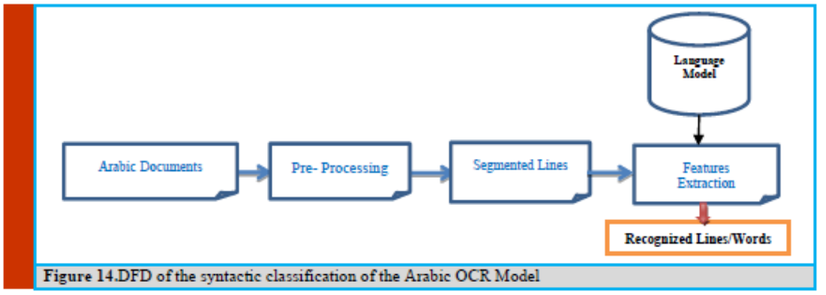


Figure 14.DFD of the syntactic classification of the Arabic OCR Model

3.1 Lexicon building

A dictionary of a unique 800k words has been built from very large corpus. The data of corpus has been collected from different resources to guarantee occurrence of words in different fields of life. The same corpus has been used to build our language model. Our implemented language model contains up to 3-words (tri-gram).

The character shape in Arabic is context sensitive, that is, depending on its position within a word (isolated, start, middle, or end). Also, Arabic characters are rich in diacritic marks and delayed strokes (dots, Shadda, Hamza, etc.). By tracing our training data, a total of 341 character models has been generated by taking into consideration the fact that an Arabic character may have different shapes according to its position in a word. Another model for space has been added to solve the problem of intra and inter spaces in the line. A total of 342 models have been generated and trained with the embedded training data.

A dictionary of all the different unique words in our database has been constructed. Each word in the dictionary is described by its related ligatures and characters as shown in table 1.

Table 1Dictionary Description with different forms and ligatures

Word-ID	Image of Word	Segments of Characters	Ligature
1	محمد	م ح م د	
1	محمد	محـ ـمـ ـد	محـ
1	محمد	مح ـمـ ـد	مح
2	عجم	عـ جـ م	
2	عجم	عجـ م	عجـ
3	..	..	..
..	..	..	..

3.2 Arabic OCR Dataset Description

In this section, we present Arabic OCR dataset that is used in this research. The proposed Arabic OCR dataset was developed by RDI and the Arabic OCR team project, Faculty of Computing and Information Technology at KAU in Saudi Arabia. It is made by helping staff of RDI and Arabic Language Technology Center (ALTEC) dataset. The creation of this dataset is a result of cooperation to the three team works; Arabic OCR system team, RDI team and ALTEC team.

The dataset is required to consist mainly of images (one page), and the corresponding formal description of that image (XML transcription file). The number of images to be produced is anticipated to be in the order of same number of images, with the corresponding transcription files. These are mainly using two streams; the first will be generated using word lists and the second will be generated using a collection of files and theses documents.

The production of the required dataset is carried out according to the following descriptions:

1. Fonts include: (a) Simplified Arabic, (b) Arabic Transparent, and (c) Traditional Arabic.
2. Sizes: Each font is required and produced for sizes of: 10, 12, 14, 16, 18, 20, and 22.
3. Books and Thesis Documents:

   It is required to select 1500 pages from different scanned documents (average of 10 pages from each book for copyright constraints which gives approximately 150 titles). The titlesmust be chosen to cover uniformly the past 50 years.
   In addition, 1000 pages from theses (in Arabic) have been selected as well which is also covered uniformly the past 50 years. Books came from at least 15 different categories based on the fonts and sizes used. The used books are classified manually and approved by ALTEC. Theses come from 10 different categories based on the used fonts and sizes. The typo thesisis classified manually and approved by the team members.
4. Documents Production:
   The documents production stage has two steps: the first is the printing step and the second is the scanning step. As for the books and theses, we go directly to the scanning step.
   In printing step, the produced output files are printed then undergo different processes to add noise to the produced documents. At the end of this step, the following document versions have been produced:

a. A Clean Version: the clean version is the first print out from the created files. Printing is done using a

different printer for every document set. In addition, the original document produced by typewriter is to be considered as a clean version.

b. A Copy Version: the clean version is photocopied using different photocopying machines.

c. Take a shot of the clean version using Digital Cameras and Mobile Cameras. (10 digital cameras and 10 mobile cameras are used). In this case, no scanning is required since we get the ".tif" images directly. All cameras have at least 5M-pixel of resolution, and the distance to the documents is 50cm. There is a 50% of the imaging with separate cameras, and 50% with mobile cameras. The produced (.tif) images of this step is not undergo any further processing.

In the scanning and digitizing step, the documents produced by the printing step (a and b above) are scanned using a different scanner for every set of documents and saved in (.tif) format. The scanning is done using the following resolutions: 200, 300 and 600 dpi. As for the books and theses, a Book Digitizer is preferably used to produce three resolution versions of each page: 200, 300, and 600 dpi. As well, 300 pages of the books and 300 pages of the thesis's pages are also captured by digital or mobile cameras.

3.3 Calligraphy, Typewritten and Handwritten DatasetsPreparation

Designing general OCR engine for multiple documents is very difficult than a single script OCR engine. This is because existence of many of different features, properties, fonts, styles and nature of language writing for each language script are needed.

Definition 1. (Multi-Dimensional Language Documents) A multi-dimensional documents set D is a set of documents D_1, D_2 ... D_n, such that each language script L_i contains a set of f features denoted by d_i^f.

Definition 2. (Language Script L_i) An OCR of language length n documents contains w words (w_{i1} ... w_{in}). Each word contains numeric features at each font (F_1... F_f), together with a set of m sizes (S_1 ... S_m), such that the word W_{ij} is associated with font F_i and size S_i.

Definition 3. (Document Data Clustering). Given a dataset DS, then decompose its words into sets W_1 ... W_k, such that W_i can be represented as rows with set of features d_i^f and each feature is "similar" to one another, according to their fonts and sizes.

Definition 4. (Document Data Classification). Given an n x d training data set DS, and a class label value in {1...k} associated with each of the n in DS. Create a training model M, which is evolved to predict the class label of dimensional record Y & D.

The Arabic calligraphy, typewritten and handwritten texts have many of documents' varieties, as follows:

Height. Measured as the letter zones; the distance between the upper, the middle and the lower zones.

Width. Defined as the length of the connecting strokes and therefore it examines the breadth of the letters, the distances between letters, words, lines, and the slant of the writing.

Spacing. It examines the margins and the spacing between lines, between words and the balance between the height and the width of the letters.

Depth. Ittries to determine the direction of the stroke from the width of the upstrokes and the downstrokes.

Curved writing. In Arabic typewritten and handwriting, some of letters composed from oval shapes(loops: formations of circle), or other Arabic letters are comprised of parts of a circle.

Overly Embellished Writing. It likes all extremes in handwriting, a facade or compensation for inner weakness.

Pieces of Arabic Word (PAW) is defined as sub-words, ligatures, and diacritics are two issues should be handled.

Writing Direction. Arabic printed, typewritten or handwritten writing directions are from right to left direction. To determine the size of Arabic typewritten andhandwritten text, the system initially looks at the middle zone letters. If the height was within 1/8" (3mm) then the size will be normal, if it was greater than (or less than) 1/8" then the word should be normalized.

The proposed dataset contains images of calligraphy, handwritten Arabic text which can be used to train and test calligraphy, printed and handwritten text recognizers and to perform OCR recognition and verification experiments.

The dataset was first prepared at the ICDAR. Using this dataset an HMM based recognition system for printed documents was developed and published at the ICPR 2000. The segmentation scheme used in the second version of the dataset is documented in (Nobile & Suen, 2014) and has been published in the ICPR 2002. The Altec- is described in (Pal & Dash, 2014). We use the dataset extensively in our own research, see publications for further details.

3.4 Training Documents

Training documents preparation is the process of segmenting and extracting boxes of objected-words from a large set of Arabic documents and storingeach objected-

word as a separate labeled image. This label identifies the text it contains, and this makes the usability of this dataset much easier.Extracting training documents can be automated through a smart computer tool, in addition toa manual segment selection usingthe developed tool.

The manual technique of the segment selection is done using a developed computer tool,with the helping of an expert user that produces objects of the lines/words with its coordinates. These coordinates are defined as (x_i , y_i , x_j , y_j) where (x_i , y_i) are the coordinates of the upper left corner in the bound box of the objected-word (segmented word), and (x_j , y_j) are the coordinates of the lower right corner in the bounding box of the objected-lines/word. This proposed development tool makes the manual efforts to be more efficient in time and accuracy than the ordinary methods. In the first training dataset, there are 28 used typewritten books, the overall of the selected objected-words from these typewritten books includes 84,000 words.

Therefore, the automated tool generates two types of outcomes: (1) Extracting segment regions; and (2) Generating text file description. The resulting segmented regions and text labeling with its segmented coordination are used to build a classification model. The automated tool generates object-segments or word-segments coordinates using x_i, y_i and x_j, y_j points as a region of word/line picture. Then, the automated tool stores such pictures in the name of pictures' words. Each segment is manually labeled for its text by annotator peoples.

Many processing algorithms need considerable tuning to address the special features of Arabic documents. The following section describes these modifications.

Now, we have two streams of picture files (automated segmented regions: asr) and labeled text files (txt). The learning algorithm uses such two streams to extract visual features from the two input streams. Accordingly, for each documentfile, the output of this step creates two generated files as illustrated in Fig 13.

1. ASR file: Extract the segmented region with its coordinates; that includes all keyword box locations (x_i, y_i, width, height).
2. TXT file: Generate textile labels of these segments that represent the segmented region itself (keywords text; it may be a single word or compound words).

Accordingly, an overview of indexing and querying data files for each documentdataset are formulated. The first step of the indexing or querying is the computation of the word representation in this section. The documents indexing is done in advance to allow a quick examination.

Also, each lines file is segmented and described with each document and contains the segmented line coordinates and its labeled text description, see Figure 16.

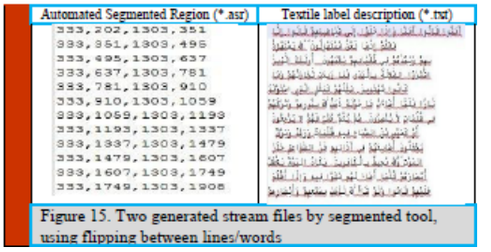


Figure 15. Two generated stream files by segmented tool, using flipping between lines/words


Figure 16. Two generated pages by segmented tool, using flipping between lines/words

The next process; each word file is related with each document and includes the coordinates of such words in the page, the positions (left upper and lower right corners) and their labeled text, as the <pawposition = "x_1, y_1 , x_2, y_2">Arabic Word</paw>". Each segmented Arabic word or sub-word (PAW) is fully described using XML standard format. The labeling of the segmented word, or sub-word contains tags that represent ground truth information about sequence of words/PAWs. Figure 17 gives generated example about information of document, word text with PAW segment locations.

4. Proposed Model Discussion

In this section, we introduce our extracted features technique. Most of these features are well known and used. The input image in binary after the pre-processing stage is used to extract a feature vector. The background pixels of the image are labeled by logic "0" and the foreground pixels by logic "1".A sliding window from right to left is used. The window is of one column width "W" -without any overlap- and the image's height "H".The

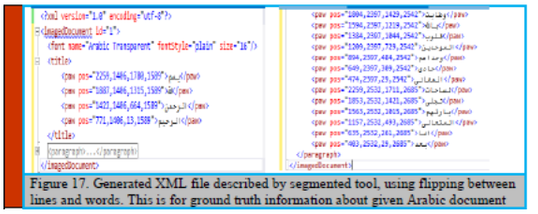


Figure 17. Generated XML file described by segmented tool, using flipping between lines and words. This is for ground truth information about given Arabic document

following three features represent the distribution of pixels per column in the given image.

a) Number of "1" pixels in each column were counted to from the first feature (F1)
b) The gravitational center of each column was calculated to form the second feature (F2), as shown in the following formula:

$$F2 = \frac{\sum_{i=1}^{i=H} i.r(i)}{\sum_{i=1}^{H} r(i)}$$

Where, r (i) is the number of "1" pixels in the i^{th} row of a column.

a) The second order of moments was used to form the third features (F3). The second order of moments was used to represents the variance of each column in the image.
b) The following four features represent the style of characters and ligatures.
c) Position of the upper contour in the column per image (F4).
d) Position of the lower contour in the column per image (F5).
e) Orientation of the upper contour in the column per image (F6).
f) Orientation of the lower contour in the column per image (F7).
g) Number of "1" Pixels between the lower and upper contours (F8).

The previous eight features used before with Latin and gave a promising accuracy. To make this feature suitable for Arabic, we normalized some of them to be independent of the height of the image. (F1, F2, F4, F5, F8) were normalized by the following formula.

$$Fi = \frac{100 * Fi}{H} \quad i = 1,2,4,5,8$$

Delta and acceleration were calculated and then concatenated with the extracted features to form feature vector of length 24. A short script is used to read the previous output files that contain the text and the locations' points of the bounding boxes of each objected-wordand output the objected-word as a separate labeled image file.

Multi-match process is proposed to retrieve similar matched features of the words, based on word spotting. The objective of the matching is to filter number of words to be measured with the searchable word query. Accordingly, a set of bound features is measured as adequate for matching, such as ratio of their words' segments sizes. If such a ratio does not lie within a specific threshold related to the size of the searchable word query, therefore, don't consider this word as candidate. Then, compute the similarity between searchable word queriesstored features of the candidate set of the comfortable content. Therefore, word spotting with vector of features was computed and used in a word holistic approach.

The over mentioned features' vectors are used to compare words with the given in word query, in case of retrieving the relevant words and documents.

Figure 18 illustrates the DFD of the proposed matching module, consideredthe distance that is used to compare two words.

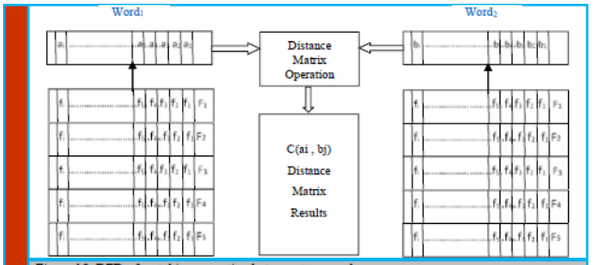


Figure 18. DFD of matching operation between two words

4.1. HMM Classifier/Decoding Module

In this section, we present the decoding stage in a general way and describe what happens to the test image from the moment it enters the system until the output text is resulted from the system. First, the test image goes through pre-processing stages that binarize the image, remove the noise, remove frame (if exist), resolve skewing, and segment the image into lines (Figure 19).

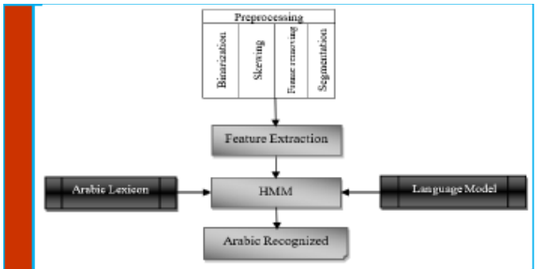


Figure 19. Decoding Processing module

Then, the pre-processed lines of the given image go through the feature extraction stage to extract a set of robust features by using frames from right to left. The different blocks of the architecture are going to be explained in detail in the following sections of this paper.

4.1 Hidden Markov Model

A hidden Markov model (HMM) represents stochastic process to generate sequence of output over period. Officially, an HMM can be modeled by 4 parameters (λ = {S, A, Π, B}), where S = {S_1, S_2, ..., S_n} represents set of states, A = (a_{ij}) is the state transition probability matrix,

Π signify the start probability vector; $(\pi_i \equiv P(x_0 = i))$, and B = $\{b_1, b_2, ..., b_n\}$ stand for emission probabilities of states (Figure 20).

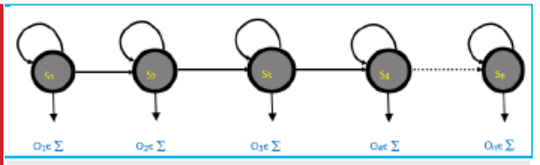


FIGURE 20. HMM State topology for Arabic OCR System

Where Π is the initial state probability vector with the following descriptions: $0 \le \pi_i \le 1 \quad \sum_{i=1}^{N} \pi_i = 1$; The transition probability matrix A is described as:

$$a_{ij} \equiv P(x_{t+1} = j \mid x_t = i) \quad , \quad 0 \le a_{ij} \le 1 \quad , \text{and} \quad \sum_{j=1}^{N} a_{ij} = 1$$

The Baum-Welch algorithm is used as a training method. It considers the probability P(O| λ) that a known sequence O is produced by the HMM λ and calculates the latest model λ which likely to have produced the given sequence. A detailed description of the Baum-Welch algorithm is given in ([1]). A modified algorithm is applied during training phase. Hence, the text line HMMS are accumulated using character of HMMs or whole-word HMMs. The HMMs characters are coupled to form word HMMs according to the language model dictionary (Figure 21). The recognizer is restricted to those words are existing in the language model dictionary. Therefore, thus, a model for word sequences is attained by the word models concatenating.

In Arabic OCR system, a segmented textual line object is handled from right-to-left signal and modeled as a sequence of extracted feature represented in vectors sampled at a fixed slice rate.Each segment is divided into several slides, and at each slide, object features are computed. Accordingly, histogram features are obtained by disintegrating the object signal at the pixel into gradient, structural and concavity features.

A character object is divided into patches; three features (gradient, structure, and concavity) are employed and calculated. The gradient features are calculated by using Sobel gradient operators to the whole character object. Then, the gradient at each pixel is quantized and transformed to 12 directions. Therefore, 12 features are calculated according to the following equation:

Feature $_{\text{Gradient(i)}}$ = (No of pixels in the patch with gradient degree i) / (Total no. in the patch)

The second step is related to derive structure features from gradient features. Therefore, the 12 assignments of directions are clustered in 12 classes.

1 http://www.lasorsa.com/wp-content/uploads/2015/02/Handwriting-Analysis.pdf

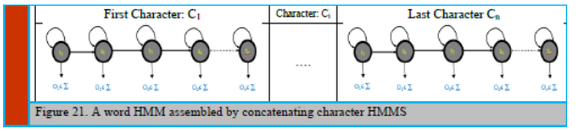


Figure 21. A word HMM assembled by concatenating character HMMS

This clustering represents the local structure central pixel. Therefore, every structure feature is defined for each patch as following equation:

Feature Structure$_{(i)}$ = (No of black pixels of class i in the patch) / (Total number in the patch)

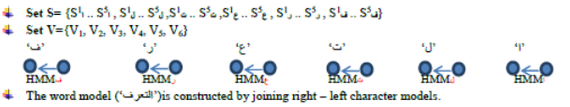
The next step computes the concavity features by using two directions scanning (from one side to the opposite side). The scanning is employed row by row (from right to left), then column by column (from up to down). During the scanning, preserve path of all the scanned white pixels until the first black one is reached and end. Figure 3 displays labeled white pixels during scanning from right to left and from top to down. At this stage, histogram features are computed in the same way, using the following equation:

Feature$_{\text{concavity(i)}}$ = (No. of scanning white pixels in the patch at i) / (Total no. in the patch)

Also, the hole feature is calculated as:

Feature $_{\text{hole(i)}}$ = (Number of scanned white pixels in the patch at i) / (Total no in the patch)

Example of imaged - word (التعرف) generation of a 6-characteralphabets ('ا', 'ل', 'ت', 'ع', 'ر', 'ف')



The hidden Markov model is a double stochastic process which can efficiently model the generation of sequential data. The HMM used in this paper is a continuous HMM with one HMM for each ligature or character. In this paper, we use the same HMM classifier without modification as simple mentioned in HTK Speech Recognition Toolkit (Rabiner, 1989)]. However, we implement our own parameters of the HMM. We allowed transition to the current and to the next state only. HTK models the feature vector with a mixture of Gaussians. It uses the Viterbi algorithm in the recognition phase, which searches for the most likely sequence of a character given the input feature vector.

In the training phase, an iterative optimization of the model with respect to the training data is performed, and we used Baum–Welch algorithm, a variant of the expectation maximization (EM) algorithm, for optimization of the HMM model depending on the training data.

To achieve high recognition rates, the character HMMs have to be fitto the problem. In particular, the

number of states, the possible transitions, and the type of the output probability distributions have to be carefully chosen.

HTK is principally concerned with continuous density models in which each observation probability distribution is represented by a mixture Gaussian density. In this case, for state j, the probability $b_j(O_t)$ of generating observation O_t is given by:

$$b_j(O_t) = \prod_{s=1}^{S} [\sum_{m=1}^{M_s} C_{jsm} \psi(o_{st}; \mu_{jsm}, \Sigma_{jsm})]^{\gamma_s}$$

where M_{js} is the number of mixture components in state j for stream s (1-stream used), the exponent γ s is a stream weight and its default value is one, C_{jsm} is the weight of the mth component, and $\psi(o;\mu,\Sigma)$ is a multivariate Gaussian with mean vector μ and covariance Σ that is:

$$\psi(o;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} e^{-\frac{1}{2}(o-\mu)^T\Sigma^{-1}(o-\mu)}$$

where n is the dimensionality of 0.64 mixtures have been chosen to give a robust model for each character and high recognition rate.

4.3 Testing and Evaluation

To evaluate the preprocessing module, RDI and ALTEC-dataset are used. The used dataset consists from documents of Arabic typewritten books, and documents of Arabic handwritten documents. Such experimental test validates each individual module in the preprocessing phase, as well as for testing the Arabic OCR project. We need 2 subsets:

(A) 200 selected documents taken from Arabic typewritten books.
(B) 200 documents gathered from Arabic handwritten documents.

Datasets of any documents are very important part to measure the accuracy of system. The accuracy achievement is very important to measure the performance of the recovery model. Table 2 illustrates the present visibly and available documents datasets that will be used in the current paper.

Table 2. Datasetsfor Current Research

Dataset Name	Nature / Type Style	Content /Size	Classifier
RDI	Calligraphy documents	200 images	HMM
ALTEC	Printed documents	200 images	HMM

The collected dataset includes 200 Arabic typewritten and 200 handwritten documents. Each document from typewritten dataset contains 35-40 Arabic text lines. For training, we select 15 documents from each book, and for testing we select 5 documents from each book.

The percentage value of the accuracy of the proposed Arabic OCR for typewritten text can be calculated by the following equation:

$$PercentAccuracy = \frac{N - D - S - I}{N} x\ 100$$

N: represents the total number of words in the reference file.
D: stands for the deleted words in the resulted file.
S: represents the substituted words in the resulted file.
I: is the inserted words in the resulted file.

The proposed line segmentation algorithm takes its input as typewritten and handwritten documents, and produces segmented text waved line images, as illustrated in Figure 11. The proposed algorithm is tested on 40 different documents with 1480 lines (35-40 text lines for each image). Table 3 summarizes detail results of lines segmentation.

Table 3. The segmentation accuracy of the calligraphy books.

Book	No. of segmented lines	Accuracy
1	37	99.94
2	35	99.95
3	38	99.96
--	--	--
39	35	99.93
40	36	99.92
Total	1480	99.94 %

The following table (Table 4) summarizes the percent accuracy of the recognition processof the used typewritten books.

Table 4. The accuracy percentage relative to the calligraphy books.

Book	Accuracy	Book	Accuracy	Book	Accuracy	Book	Accuracy
1	82%	11	78%	21	68%	31	67%
2	79%	12	77%	22	70%	32	67%
3	78%	13	76%	23	70%	33	65%
4	80%	14	74%	24	70%	34	63%
5	78%	15	76%	25	64%	35	65%
6	79%	16	74%	26	64%	36	66%
7	79%	17	70%	27	69%	37	65%
8	76%	18	73%	28	60%	38	65%
9	75%	19	74%	29	68%	39	78%
10	71%	20	72%	30	67%	40	81%
Average	72.00 %						

During *calligraphy* testing Dynamic Time Warping (DTW) is a good solution to be used in image matching. A flexible dynamic matching is performed when the two feature vectors are compared using DTW. Consider the two words A and B of widths i and j. The two vector sequences A={a_1 ... a_i} and B= {b_1 ... b_j} are representing the two words, where a_i and b_j are two vectors in a distance matrix vector space with 5 dimensional space features. In addition, a matrix C of dimension i x j is created where every element C(i,j) represents matching cost between sequence a_i and b_j according to the Euclidean distance:

$$C(a_i, b_j) = \sqrt{\sum_{n=1}^{k} (a_{i,k} - b_{j,k})^2}$$; where k is the number of features.

Due to Arabic text peculiarities: font, style andsize variations, writing direction in Arabic handwritten text,

an adopted routine is used to handle these variations. The proposed routine uses set of thresholds of distance normalization to adapt the well-known variations. Therefore, divide the C(i, j) by average width of the two words:

$$\text{Distance}(\text{word}_a, \text{word}_b) = \{C(i, j)/(i+j)/2\}$$

Another factor of normalization should be taken into consideration is the size of the words to be compared and can be defined as the number of processes concerned in the best possible matching. In this case, the normalized matching cost of the two words is given by the normalized word distance = W ($Word_{Query}$, $Word_{testing}$) / N. Table 5 illustrates an example of matching words "السماء" and "أبواب السماء" with respect to the pre trained dataset.

Table 5. Fetching Similarity between Query and Pre-trained imaged words

Query Word	Pre-Trained Stored Dataset				Final Cost
	Word 1	Word 2	Word 3	Word n	
السماء	0.94	0.89	...	0. 17	1
ابواب السماء	0.52	0.45	0.92	0.16	i
..	..	..	..	..	n

When searching about word, it will be fed into the Arabic OCR system, the stored features of the dataset words are compared with the given word (query word). We can perform word by example via selecting ona word in a page of the displayed documents.

CONCLUSION

This paper introduced an Omni Arabic OCR system to analyze, segment and recognize the Arabic calligraphy and typewritten documents. Consequently, the paper described the preprocessing, the segmentation, and the dataset and language model preparations modules. The line and word segmentations are the most challenging algorithms, and they are essential requirement in OCR system. In this paper, typewritten and handwritten documents are segmented with accuracy 99.4% and 98.1 respectively in line segmentation, and with 95.3% and 90.2% in word segmentation.

The system used the modified HMM classifier to extract features and therefore recognize the input documents into Arabic text. The proposed Arabic OCR system is tested and achieved accuracy not less than 72 % for typewritten and handwritten.

The experimental test with the classification algorithm uses a dictionary of 800,000 classical Arabic words without redundancy, which has been built from many Arabic corpora. Two types of datasets are used during testing the OCR system, the first test includes the typewritten imaged documents; it includes 40 typewritten documents, the classification accuracy is in average equal72%.The second dataset includes 30 documents from historical handwritten books; the recognition accuracy was 63.52%.

Accordingly, enoughgreat of training dataset is prepared to be used in the proposed system.Dueto HMMs drawbacks (Cao & Natarajan, 2014), Artificial Neural Network (ANN) will be used specially in early printed and handwritten retrieval documents. Especially Convolutional Neural Network (CNN) accepts feature vector and makes useful information to estimate and expect the output label. Therefore, to overcome some limitation pre-segmented objected-word is needed for each position in the input stream sequence of segmented-words. Additional designed output layer for the CNN also, will beimplemented to map the input word sequence to related label text sequence without need of pre-segmented objected-words.

In future work, extended work may include other approaches and algorithms to segment and recognize other categories of Arabic documents such as handwritten and historical. In addition, the language model and the statistical lexicon/corpus will be extended with probabilistic and statistics rules. The lexicon/corpus of the language model will include different types, categories and domains of Arabic documents and scripts. Therefore, this extension may improve recognition results significantly. Also, we plan to implement our Arabic OCR prototype system as a final product. This are, in addition to, Arabic OCR web services that will be implemented.

ACKNOWLEDGMENT

The teamwork of the *"Arabic Printed OCR System"* project was funded and supported; by the NSTIP strategic technologies program in the Kingdom of Saudi Arabia-project no. (11-INF-1997-03). In addition, the authors acknowledge with thanks Science and Technology Unit, King Abdulaziz University for technical support.

REFERENCES

Al-Barhamtoshy, H. et al., 2019. An OCR System for Arabic Calligraphy Documents,. International Journal of Engineering & Technology, pp. 9-16.

Al-Barhamtoshy, H. & Rashwan, M., 2014. Arabic OCR Segmented-based System. Lif Science Journal, pp. 1273-1283.

Arvind, K., Kumar, J. & Ramakrishnan, A., 2008. Line removal and restoration of handwritten strokes. Sivakasi, Tamil Nadu, India, IEEE, pp. 208-214.

Bin Ahmed, S., Naz, S., Razzak, M. & Yusof, R., 2019. Arabic Cursive Text Recognition from Natural. Applied Science, pp. 1-27.

Cao, H. & Natarajan, P., 2014. Machine-Printed Character Recognition. In: Handbook of Document Image. London: Springer, pp. 331-358.

El-Gajoui, K., Ataa-Allah, F. & M., O., 2015. Diacritical Language OCR Based on Neural Network: Case of Amazigh Language. Procedia Computer Science, pp. 298-305.

Gatos, B. et al., 2006. A Block-Based Hough Transform Mapping for Text Line Detection in Handwritten Documents. LaBaule, s.n., pp. 73-131.

Hadjadj, Z., Meziane, A., Cherfa, Y. & et, a., 2016. ISauvola: Improved Sauvola's Algorithm for Document Image Binarization. Switzerland, Springer International Publishing Switzerland 2016, p. 737–745.

Nashwan, F. et al., 2017. A Holistic Technique for an Arabic OCR System. Journal of Imaging, pp. 1-11.

Nobile, N. & Suen, C. Y., 2014. Text Segmentation for Document Recognition. In: Hand Book of Document Image Processing and Recognition. London: Springer , pp. 259-290.

Otsu, N., 1979. A threshold selection method from gray-level histograms. IEE trans, 1(1), pp. 62-66.

Pal, U. & Dash, N., 2014. Language, Script, and Font Recognition. In: Handbook of Document Image Processing and Recognition. London: Springer, pp. 291-330.

Rabiner, L., 1989. A tutorial on hidden Markov models and selected applications in speech recognition.. USA, IEEE, pp. 257-286.

Stahlberg, F. & Vogel, S., 2016. QATIP - An Optical Character Recognition System for Arabic Heritage Collections in Libraries. Santorini, Greece, DOI: 10.1109/DAS.2016.81.

Subrahmanyam, M., Kumar, V. & Reddy, B., 2018. A New Algorithm for Skew Detection of Telugu. I.J. Image, Graphics and Signal Processing, pp. 47-58.

Zha, S. et al., 2014. Text Classification via iVector Based Feature Representation. Tours – Loire Valley, France, DAS, pp. 151-155.