Segmentation Accuracy for Offline Arabic Handwritten Recognition Based on Bounding Box Algorithm

Ismail. A. Humied
Assistant Professor of Computer Science, Faculty of Police, Policy Academic, Ministry of interior, Sana'a, Yemen

ABSTRACT
Character segmentation plays an important role in the Arabic optical character recognition (OCR) system, because the letters incorrectly segmented perform to unrecognized character. Accuracy of character recognition depends mainly on the segmentation algorithm used. The domain of off-line handwriting in the Arabic script presents unique technical challenges and has been addressed more recently than other domains. Many different segmentation algorithms for off-line Arabic handwriting recognition have been proposed and applied to various types of word images. This paper provides modify segmentation algorithm based on bounding box to improve segmentation accuracy using two main stages: preprocessing stage and segmentation stage. In preprocessing stage, used a set of methods such as noise removal, binarization, skew correction, thinning and slant correction, which retains shape of the character. In segmentation stage, the modify bounding box algorithm is done. In this algorithm a distance analysis use on bounding boxes of two connected components (CCs): main (CCs), auxiliary (CCs). The modified algorithm is presented and taking place according to three cases. Cut points also determined using structural features for segmentation character. The modified bounding box algorithm has been successfully tested on 450 word images of Arabic handwritten words. The results were very promising, indicating the efficiency of the suggested approach.

Keywords:
Arabic OCR, Off-line Handwriting Segmentation; Connected Components, Pattern Recognition.

1. Introduction

Optical character recognition technology has grown from the simple character recognition tools into widely used and specialized technologies, capable of enhancing numerous business processes and the researches on the recognition of handwritten letters have obtained increasing attention in recent years. The handwritten recognition is generally considered a difficult task because of the differences of handwritings and of the irregularity of the writing of the same writer [1].

Although there are many researches on OCR handwritten in more than 50 years of age, there are still many open issues that must be resolved. System for off-line Arabic handwriting recognition still faces most challenges. Due to the character nature of the Arabic language, most of published works are based on recognition of a whole word without segmentation [2]. On the other hand, the complete dominance of the Internet as the main source of knowledge enforces a proper conversion of knowledge into an editable and searchable format, so that such priceless knowledge can be not only preserved, but also mined for information [3]. Nowadays, OCR systems built upon segmentation-free algorithms are put successfully into service in a number of application areas such as automatic reading of postal addresses and bank checks, processing documents such as forms [4]. Given the importance and the difficulty of the segmentation problem, solving such problem would be a great achievement in the field of OCR applications [5]. Hence, this article contribution is modified bounding box segmentation algorithm to improve segmentation accuracy.

The rest of the paper is organized as follows. Sect.2, give a brief review of related literature. Sect.3, describe the concept of OCR and its phases. In Sect.4 describes Arabic writing characteristics, preprocessing, methodology used in the segmentation of off-line handwritten Arabic character research, the modified for bounding box segmentation algorithm, and cut points. Simulation results and discussions are presented in Sect. 5. Finally, concludes works is provided in section 6.

2. Previous works

In the literature, multiple research works reported the segmentation use of several off-line Arabic words. One of algorithms based segmentation of Arabic handwritten word proposed by Lorigo and Govindaraju in 2005. It proposed a new algorithm for Arabic handwritten word for segmentation into sub word. This algorithm used to over-segment the words and derivative info nearby the baseline, an imaginary horizontally line, location [6]. Xiu, P et al. in 2006 propose a probabilistic model segmentation algorithm, which is performed contour based over segmentation in the image of the text, and as a result, the production of a group called grapheme [7]. Abdulla, S.et al. in 2008 proposed algorithm that begins in the assessment of strokes, where a stroke is the curve between any two structure points (end points or branch points), or with curved segments in words, extracts the upper contour of the image of the word smoothed. Then

Manuscript received September 5, 2016
Manuscript revised September 20, 2016

using the chain- code representation of the upper contour, and are paired adjacent points is the slope of the line connecting each pair calculated [8]. AlKhateeb, et al. in 2009 proposed technique for segmentation word; the method words are extracted and detected in the Arabic handwritten. The technique is based on the distances between words and also sub-words. The measure distances between connected components and analyzed to determine the optimal threshold for the segmentation word [9]. Al Hamad and Abu Zitar in 2010 proposed segmentation algorithm and strategy validation for the Arabic words handwritten. The described technique to segmenting a word to its priorities is used ANN to validate points segmentation on the basis of certain features. Technique works in three phases. Is obtained by an over segmentation on the histogram vertical modified of the word thinning. First, the segmentation is using a heuristic algorithm to segmentation of the Arabic word to the primitives, then, the extraction of the structural features of the characters using the modified direction features method. Then, conversion of these features in the character of the ANN training and testing into validates the point's segmentation [10]. Elzobi, M., et al. in 2011 suggested segmentation algorithm has two phases. First algorithm starts by processing stage, it considered issues such as skew words and slant correct. Secondly segmentation stage by detecting and solving sub words overlapping, and then is applied segmentation topographical features through a set of rules heuristic [11]. Lawgali et al. in 2011 proposed algorithm exploited the segmentation points that occur through end of a letter and start of the after letter, also are located in the baseline of the region surrounding to present a segmentation algorithm of Arabic handwritten words. In this segmentation algorithm starts with segmenting the word into sub words and the baseline of every sub word is computed. And then, deletes all the descended sub- words that have a beginning point below in the baseline. In this algorithm used vertical projection for find the candidate points for the segmentation [12]. Eraqi & Abdelazeem in 2012 proposed technique combined the neighborhood geometric characteristics and the local writing direction information to propose a new efficient explicit method for offline handwriting Arabic segmentation which segmented the text into graphemes [13]. Samoud et al. in 2012 proposed two combining methods for segmenting Arabic handwritten word into characters. The one method was on the basis of the analysis of the contour minima and maxima and the projection. The two methods were on the basis of Hough Transform and also Mathematical Morphology operators [14]. Al Hamad.H in 2013 proposed fusion equations for improving the segmentation of Arabic word, this method has two phases. In the one phase the method applied Arabic Heuristic Segment to place the prospective points of segmentation in the each parts of the word image. In the two phases the method applies neural-based segmentation technique to examine all prospective points of segmentation and identify the invalid ones [15]. Osman.Y. in 2013 proposed algorithm for segmentation of word Arabic handwritten. The idea of this algorithm is to segment the image to the lines and sub words. After that, keep track of all sub word, and the contour of every sub word. In this algorithm detects the finer points where the contour condition of a horizontal line is changed to another state of the vertical line [16]. Elnagar,A & Bentrcia, R. in 2015 proposed effective segmentation Arabic words handwriting method. In this method, using a multi agent technique to segment words and relied on recognition to verify the validity of the candidate segmentation points. The method use of artificial neural network along with the compilation of rules lead to good treatment of the problem of excessive segmentation of the handwritings are Arab. This was due to a proxy resolution, which shall take the appropriate decisions to determine the candidate segmentation points. The segments pass led to the identification that will invoke and apply the rules and agent pool on the unrecognized slides before passing to recognize again [17].

The note in this section is that the researchers have proposed several algorithms for segmentation of words. Research workers typically use the easy algorithm to using horizontal and vertical projections of the word picture and search for minima’s to segment characters from words. The algorithm of Arabic Heuristic segmentation is used to segment a word into primitives. Subsequently, the features of the structural characters are extracted through the use of the Modified Direction Features technique, and there algorithm begins with segmenting the word into sub-words or connected components and then the baseline of every sub-word is computed. A variant of that is to make use of the projection of a segment across the baseline to prevent the problems of overlapping characters and holes. Some researchers use the minima of the upper profiles of words. A lot of the algorithms presume that the characters are connected at baseline. Other methods use the upper contour rather of projections. Many researchers over segment the text and finalize the segmentation after recognition by combining segments until characters are shaped. In this event they use all potential mixes of consequent segments. Yet other algorithms thin the word or make use of the skeleton of the word to simplify the segmentation. From a previous works bounding box segmentation algorithm will modify to improve segmentation accuracy for handwritten Arabic text recognition.

3. Optical Character Recognition

Optical character recognition (OCR) is a research field in pattern recognition, artificial intelligence and computer vision. It is used widely as a form of data entry from paper data printed records, both passports, invoices, bank statements, receipts computerized, business cards, mail, publications of static data, or any appropriate documents. It is common to digitize printed text the way so they can be electronically adjusted, inspected or stored more compact, offer on the Internet, and is used in operations, such as automatic machine translation, and text-to-speech, the fundamental data and text mining. It uses OCR also by some of the archives as a means to transform the massive amounts of handwritten to search digital forms of historical documents, easy access. In general, Recognize Handwritten Letters Systems are divided, according to preliminary data (image) acquisition, to the main systems; on-line and off-line systems. There are many phases in OCR systems performed one by one to carry out the whole task [4]. The phases of OCR systems include the following phases: data acquisition, pre-processing, segmentation, feature extraction, classification, and post processing, these phases are as follows:

3.1 Data Acquisition

Any OCR takes as input data in two directions, either online or offline system. In handwriting recognition on the online, when use a special pen to write on a digital tablet, it is also that image is stored in digital form. When a handwritten word image scanned, it is converted into a digital image. The words images that experimented on are gray scale images, taken from an under construction database; conventional flatbed scanner is used to extract the text with 350 dpi resolution. The form test includes 450 words. The set of words that include all the forms of the characters in all positions in Arabic are used. The images saved in PNG graphic file format rather than other format, for example, TIFF, BMP, JPEG, or GIF, since PNG files are relatively smaller in size with no loss in quality.

3.2 Pre-processing

The importance of preprocessing phase of character recognition system lies in its ability to address some of the problems that may occur as a result of certain factors. The use of pre-processing techniques can enhance the image of its preparation for the next phase in the character recognition system. The preprocessing is a collection of processes applied on the digital image, for enhancing and smoothing image to take additional steps of character recognition simple and accurate. This includes processing of noise removal reduction, binarization, skew correction, skeletonization (thinning) the threshold image and slant correction.

Skew is the tilt in the image that occurs during scanning, if the paper is not fed straight into the scanner. Skeletonization removes the width of the image from much pixel width to a single pixel width [18]. The preprocessed image is used as input in further phase after removing the above mentioned imperfections. In order to achieve the highest recognition rates, it is necessary to have an effective preprocessing phase, and therefore; using the effective processing algorithms makes the OCR system power mainly through precise image enhancement, and noise removal system, the threshold image, thinning, skew and slant correction, as describe it in detail in section 4.

3.3 Segmentation

Segmentation is a very important stage in any recognition system. Segmentation includes the separation of text to lines, lines to word and also word into characters. Handwritten text has a lot of problems, such as touching of the characters, leading to segmentation inappropriate and errors in the segmentation can reduce the rate of recognition. Therefore, efforts should be made to develop good segmentation techniques. Two techniques were applied to divide the printing and Arabic words handwritten machine to segment characters: explicit segmentations and implicit segmentations. Explicit segmentation: words are externally segmented into pseudo-letters, which are individually recognized. Implicit segmentation: usually the design of this type of segmentation with the rules that tries to identify all points of segmentation image for the segment words directly to the letters. Implicit segmentation operation is performed by several methods such as region based segmentation, the edge-based segmentation, threshold based segmentation, clustering technique and, bounding box algorithm [19]. In this paper the segmentation improved using bounding box algorithm.

3.4 Feature Extraction

Feature extraction is to extract useful information from the text that can be used for the recognition purpose, therefore it is very important to determine the features meaningful. The feature extraction is done before recognition of any character. Recognition accuracy OCR system on the characters directly depends also on the feature extraction precision accuracy. As characters handwritten vary greatly in slant and size, so efforts should be made to determine the slant and size invariant features. The key goal of the feature extraction would be to map the input picture onto points in a feature area for the classification and recognition stage. Features may be separated into statistical and structural features. The

statistical features are extracted in the statistical distribution of pixels which describe the feature measurements of the input picture pattern. The structural features include the geometrical and topological features of an input picture [20]. In this paper use topological features to extract useful information from the text image that can be used for the recognition purpose.

3.5 Classification

Classification phase is the phase of making major decisions for any OCR system. It classifies unknown character into different classes based on the extracted features. A class is a feature space or region in which the particular character falls. The different algorithms are used to classify characters pixel-based, statistical, structural and neural network. Typical character classification systems typical of many of the features of each character picture on the basis of similarity of feature vectors to the character class, trying to classify. There are various character classifier structures of isolated handwritten character classification, such as simple linear classifiers, two-phase tree classifiers, and hierarchical classifiers. According to the results of tests on the handwritten characters which combine multiple classifiers is an effective way to produce works extremely reliable decision classifiers [21].

3.6 Post-processing

Post-processing system is the main stage to correct segmentation and classification errors without human intervention. Recognize some of the characters that cannot be properly segmented in a speech during a post-treatment, and the word can also be interpreted as a whole. And can classification process output go through a phase error detection and correction. Post processing include dictionary look up and apply of language-specific information on words unrecognized. Known from the lexical knowledge of contextual post-processing application compares the dictionary on the basis of top to bottom and statistical algorithm bottom to top. Finally, the post-processing of the context of the results of OCR can also take into account the knowledge of the context of words [22].

4. Offline Handwritten Arabic Character Segmentation

Firstly, in this section describe Arabic writing characteristics to help full understand methodology used in the segmentation of off-line handwritten Arabic character. After that describe, using example, most methods in pre-processing phase of OCR systems as mentioned in the previous section; such that: noise removal, binarization, skew correction, thinning and slant correction. Finally explain segmentation phase to segmentation of off-line handwritten Arabic character.

4.1 Arabic Writing Characteristics

The actual Arabic alphabet contains of 28 characters and contains numerous characteristics. Arabic writing process differs than the English language; Arabic is written from right to left and it is cursive in general. The alphabet set can broaden to 84 different forms based on the location of the character in addition to the style of writing (Nasekh, Roqa'a, Farisi and others). Table 1 shows the different forms of Arabic characters depending on their location within the word and from the 28 basic Arabic characters, six may be linked in the right part : dal (د), raa (ر) waw (و), alef (ا), thal (ذ) , and zay (ز), just as the other 22 can be linked from many sides. Most of these six characters include only two forms, the stand alone form and the last form. Although other characters can come in any of four forms: the beginning, the middle, the last, and the stand alone form. Therefore, the Arabic word might contain one or more connected components [19].

The secondary components (dots) perform an important role in Arabic characters. The form of many characters is similar but the difference occurs with number and position of dots, which could take place either above or below the characters, like (ب, ت, ث). Two characters in the alphabet have three dots; three have two dots and ten have one dot. Dots can take place as two distinctive dots or could be connected in a line in handwritten word. The difficulty in recognizing the secondary components comes due to quickly writing, as writers draw them connected to the main body.

In Arabic, small marks like a "'hamza", may be located above or below five distinct characters or can appear as isolated characters. In addition, besides the alphabet there are what so-called Diacritic Symbols, which are used to indicate vowels, and written in a very small size (compared to letters size) above or below a letter (e.g.) , □).

The cursive nature of Arabic text means that characters of a word are connected through an imaginary horizontally line known as baseline. Arabic writing is proven to be cursive additionally in printed type. On the other hand, that is not the same as cursive handwriting of the English language in that several characters may be connected in one part only. Several Arabic characters have a loop, like (ص, ف, و) [23].

Table 1. The different forms of Arabic characters depending on their location within the word.

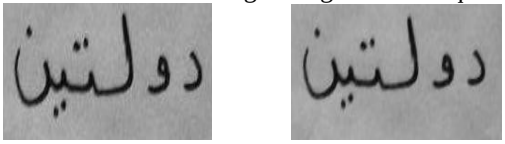
Name	Isolated	First	Middle	Last
Alif	ا	ا	ا	ا
Baa	ب	بـ	ـبـ	ـب
Taa	ت	تـ	ـتـ	ـت
Thaa	ث	ثـ	ـثـ	ـث
Geem	ج	جـ	ـجـ	ـج
Hha	ح	حـ	ـحـ	ـح
Kha	خ	خـ	ـخـ	ـخ
Dal	د	د	ـد	ـد
Thal	ذ	ذ	ـذ	ـذ
Raa	ر	ر	ـر	ـر
Zain	ز	ز	ـز	ـز
Seen	س	سـ	ـسـ	ـس
Sheen	ش	شـ	ـشـ	ـش
Saad	ص	صـ	ـصـ	ـص
Dhad	ض	ضـ	ـضـ	ـض
Tta	ط	طـ	ـطـ	ـط
Zha	ظ	ظـ	ـظـ	ـظ
Ain	ع	عـ	ـعـ	ـع
Ghain	غ	غـ	ـغـ	ـغ
Faa	ف	فـ	ـفـ	ـف
Gaf	ق	قـ	ـقـ	ـق
Kaf	ك	كـ	ـكـ	ـك
Lam	ل	لـ	ـلـ	ـل
Meem	م	مـ	ـمـ	ـم
Noon	ن	نـ	ـنـ	ـن
Haa	ه	هـ	ـهـ	ـه
Waw	و	و	ـو	ـو
Yaa	ي	يـ	ـيـ	ـي

4.2 Pre-processing

The purpose of this phase is enhancing the readability of text image and removing the details that do not have the discriminatory power in the recognition process. The pre-processing is a series of operations performed on the scanned input text image. It essentially enhances the resulting image to suitable for segmentation, which includes noise removal, binarization, skew correction, and thinning and slant correction, as description in the following [16, 21, 24]:

4.2.1 Noise Removal

The spatial noise descriptor which shall be concerned is "salt and pepper" of structural features in the noise component of the model. The "salt and pepper" noise model is the most common in OCR system found in image processing applications. Median filtering technique is non-linear helpful to remove noise from pictures, it's especially efficient to removing the "salt and pepper" noise [25]. In this research used median filtering of size 3X3 to remove noise from text image. Fig. 1 shows the noise removal in a word image using this technique.



(a) (b)

Figure 1: Noise removal in a word image using median filtering technique with size 3X3 (a) Grey scale word image with noise. (b) Image (a) without noise.

4.2.2 Binarization

The global thresholding used to converting image into binary image which iteratively determine all possible threshold values and find out there variance [3]. The output binary image has values of 0 as the front pixels (black) for all pixels in the input image and 1 as the background pixels (white) for all other pixels [10]. Fig. 2 shows the binary word image of Fig. 1 (b).

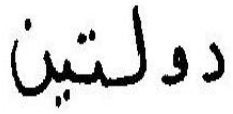


Figure 2: The binary word image of Fig. 1 (b).

4.2.3 Skew correction and baseline estimation

Skew correction is based on the estimation of a fitting line used during the writing process. In this paper, use the algorithm estimate linear regression of this line is the use made by linear regression of local minima of the word image skeleton (*LMR*) [26]. Benefiting from the fact that most of the local minima (*LM*) points are usually occurring on near of the baseline; the problem of finding the baseline can be reduced to a linear fitting problem of local minima points. However, this point contains points from the descending letter and the baseline estimate. A consequence, those spurious points have to be filtered prior to the baseline estimation. The method is performed it is based on a linear regression on the remaining points to estimate the skew of the baseline image of the word, then the skew correction using the rotation. The baseline detection algorithm is dependent on a two-step linear regression:

First step begins through the fitting line of local minima points calculated according to the following equations:

$$y = a + bx \quad (1)$$

Where a and b are coefficients calculated as follow:

$$a = \bar{y} - b\bar{x} \quad (2)$$

$$b = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2} \quad (3)$$

Where $\bar{x}$ and $\bar{y}$ are the statistical means of x and y coordinates, respectively. The slope angle α of inclination processed line is calculated according to the following equation:

$$\alpha = \arctan(b) \quad (4)$$

Second step to compute baseline using a give the θ limited area. First, discretize the θ and the parameters ρ then each additional point (x_i, y_i) at area of the image; calculated ρ′ as stated in Eq. (5):

$$\rho' = x_i \sin\theta' + y_i \cos\theta' \quad \forall\, \theta' \in [\alpha - \varepsilon, \alpha + \varepsilon] \qquad (5)$$

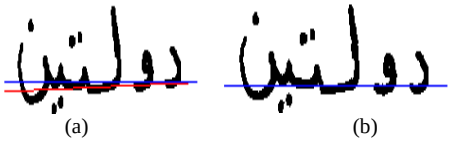
where ε is constant that uses to offset the random error that can be produced in the first step. Experimentally, it found that $\varepsilon = 10^{o}$ gives most accurate results. Next, each point in the image space will vote for bins that could have generated it in the though accumulator *A*, and votes will be accumulated in *A* according to Eq. (6).

$$A(\rho', \theta') = A(\rho', \theta') + 1 \qquad (6)$$

Finally, it will be considered ρ' and the θ' with the maximum number (global maxima) of votes will be considered as the parameters of the word baseline as shown in the following equation:

$$\arg\max_{p',\theta'} A(\rho', \theta') \qquad (7)$$

Fig. 3 shows an example of the results, where Fig. 3(a) is the binary word image and its baseline estimation. Fig. 3(b) is the skew corrected image with *LMs*: skew angle = -1.749.



(a) (b)

Figure 3 skew correction and baseline estimation, (a) Binary word image and its baseline estimation (b) Skew corrected using linear regression.

4.2.4 Thinning

Thinning is a process to reduce the foreground regions in the binary image of the remains to the skeleton that keeps largely on the extent of the contact in the original region while throwing more than the original foreground. Commonly used in pattern recognition, digital image processing and image analysis. The thinning process is applying to enhanced images words. An effectively skeleton algorithm has been proven in a wide range of applications for image processing including the OCR. Skeleton algorithm will find a single pixel thick representation showing centerlines of the text. Generally, skeletonization algorithm to be effective, it should ideally data compression and retaining the important features of this pattern. For the case of handwritten Arabic it is hard to find a robust and useful skeleton algorithm that retains the significant feature of the pattern due to the variety of handwritten Arabic writing styles. This paper has been used the thinning algorithm that is based on the Zhang-Suen's thinning algorithm [27]. The Zhang-Suen's thinning algorithm for extracting the skeleton of a picture consists of removing all the contour points of the picture except those points that belong to the skeleton. In order to preserve the connectivity of the skeleton, it divides iteration into two sub iterations. In the first sub iteration, the contour point P_1 is deleted from the digital pattern if it satisfies the following conditions:

a) $2 <= N(P_1) <= 6$. b) $S(P_1) = 1$.

c) $P_2 * P_4 * P_6 = 0$. d) $P_4 * P_6 * P_8 = 0$.

Where $S(P_1)$ is the number of 01 patterns in the ordered sequence of P_2, P_3, … P_8, P_9 and $N(P_1)$ is the number of nonzero neighbors of P_1, that is,

$$N(P_1) = P_2 + P_3 + \cdots + P_8 + P_9.$$

In the second sub iteration, conditions (a) and (b) remain the same, but conditions (c) and (d) are changed to

c') $P_2 * P_4 * P_8 = 0$. d') $P_2 * P_6 * P_8 = 0$.

Is executed one step to each pixel in the binary area under consideration, In the event of one or more violations of the requirements (a) to (d), does not change the value of the points in question. If the all of the requirements and a point of has developed for deletion are met. And it is important to note, that the point is not deleted even address all the points. This prevents changing the structure of the data during the implementation of the algorithm. After they have been applied one step to each pixel, and the ones that have been flagged are deleted. Then apply a two-step to the resulting data at exactly the same manner as a single step. Fig.4 shows the thinned word image.

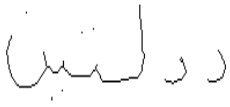


Figure 4: The thinned word image of Fig. 3(b)

4.2.5 Slant correction

Handwritten word is usually characterized by slanted characters. The slanted characters slope either from left to right or versa. Different deviations may appear not only within a word but also within a single character. The slant correction does not affect the connectivity of the word and the resulting characters are natural. Slant is an individual variation in handwritten words to lessen the consequences of this variation the slant angle must be detected shear normalization has to be done efficiently on the contour level. Among the measurable variables of various handwriting forms is the slant angle between longest strokes in a character along with the vertical direction. Slant correction can be used to correct all characters into a regular form. Coordinates of the beginning and end-points

of each line component provide the slant angle. The algorithm used by which projection profiles are computed for a number of angles from the vertical direction [28].

In this paper use for slant correction technique the vertical projection histogram [29]. The histogram of the word that is written in a row would be a distinct peaks larger and more. Therefore, the chart can looked of the word in different shear angles and take the one with the highest peaks. It does for the angles between -45 and 45 degrees, which is the most common form of slant angles in a regular writing. Every angle, the histogram vertical calculated, application function scale, which measures the height of the peaks. The angle with the highest winning measure and will be used as the shear angle. To save time, go through the first set great strides, say 5 degrees. Each of these angles has to identify those with the highest standards, and looking about each of these with smaller steps.

The technique projection profile based technique are calculated on the image of the horizontal gradient in different shear angles in the range [± 45], and used to estimate the slant angle. For the first time, the focus will be on the image of the horizontal gradient to calculate vertical strokes at the expense of those horizontal and second, will reduce the cost of the expense, due to the need to address the relatively less pixels. It is determined extremist points in the horizontal extent to contour the Arabic word handwritten. It is presumed that the amount of the absolute differences involving the coordinates x of the left (or right) end points from five successive runs vertically with the present range of being one in the center to be an intrinsic element of the slope of the endpoint.

$$x' = x - y.\tan(\alpha), \qquad y' = y \tag{8}$$

Where $\alpha \in [\pm 45]$: is the shearing angle.

For each sheared image, vertical histogram H is calculate as stated in Eq. 9

$$H(\acute{x}_L; \alpha) = \sum_{k=0}^{\infty} \acute{i}\,(x'_L, \acute{y}_k) \tag{9}$$

And apply a variation analysis for every histogram profile according to Eq. 10

$$V(\alpha) = \sum_{L=0}^{\infty} [H(\acute{x}_L\,; \alpha) - H(\acute{x}_L + 1; \alpha)]^2 \tag{10}$$

Where the sheared angle is the angle associated with maximum variation according to Eq. 11

$$\acute{\alpha} = \arg\max_{\alpha} V(\alpha) \tag{11}$$

Fig. 5 shows a slant corrected, Fig 5(a - c) shows a binary word image of Fig. 2, the thinned word image of Fig. 4 and the slant corrected version of Fig. 4, respectively; and its corresponding vertical projection profiles.

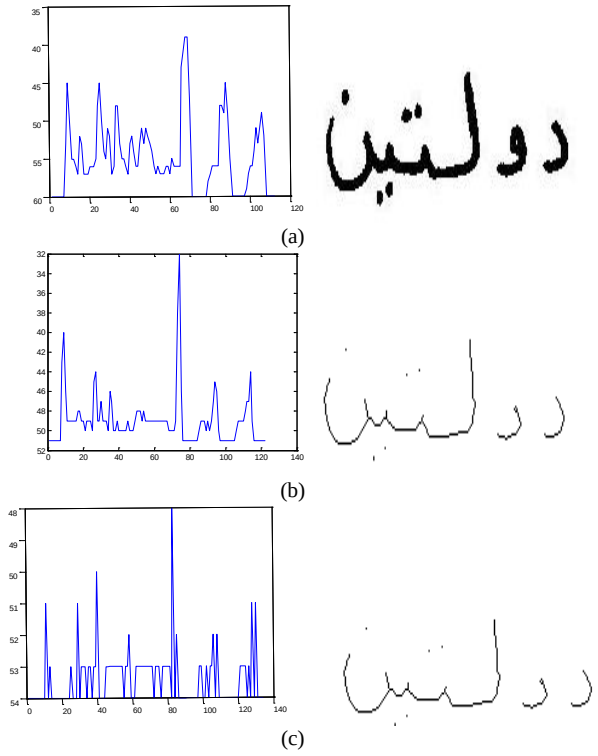


Figure 5: Slant correction (a) A binary word image (Fig.2), (b) The thinned word image (Fig. 4), (c) The slant corrected version of Fig. 4. The corresponding vertical projection histograms are in the left of Fig.

4.3 Segmentation

Following the preprocessing phase, character recognition systems perform segmentation, for the text to be recognized. Generally, segmentation of a binary text is dependent on re-grouping of the connected components *CC*s. Arabic writing is text therefore which words are divide by spaces. While, a word might include many *CC*s that are parts of the word including one or even more connected character. The *CC*s for the word must be determined. The objective of the *CC*s phase is to form minimum sized rectangles about all the connected objects in the image. The method used to acquire the *CC*s is an iterative process which checks any black pixels for connectivity with another. Bounding rectangles are extended to enclose any grouping of connected black pixels. In this paper, the 8 neighbors are used for extracting the connecting components by scanning the image pixel by pixel checking for pixel connectivity and improve segmentation using bounding box algorithm as mentioned above.

Firstly, let P describes to foreground pixel in the skeleton word $g(x, y)$, then allow the 8-neighborhood set of P. Second, by analyze every $P \in g(x, y)$, a set of feature points are identified, that call the main feature points (*MFP*s). Third, determine bounding box segmentation algorithm. Finally, generate a new set called the cut point (CP). As describe in the following sub-sections:

4.3.1 The Main Feature Points

Structural and statistical features are the most commonly used features of the character recognition. Choose the type of features and extraction of the characters is a very critical step. Feature extraction transfer of Two-dimensional image into a set of vectors that are letter input representation by a set of numerical values to pass to determine the recognition. Since the words are represented in the system by the skeletal pattern so most of the topological features are suitable for this representation [30]. The topological features were chosen are: loop point, branch point, dot point and end point, which all operate on a skeletonized word and describe in more detail in the following:

(i) Loop Points (*LP*)

First convert the starting locations linear indicators because the linear indicators may be used to extract the pixel values of all of the locations and then select eight neighbors $N_8(P)$ compensation expense of all the neighbors of a group of pixels in one. We're considering in finding the north, east, south, and west neighbors all these pixels and then add all the neighbors compensate for each linear index now carry out *flood-fill* algorithm [31]. The *flood-fill* (*ff*) algorithm on a binary image, you specify a background pixel as a starting-point, and flood-fill changes associated background pixels (1) to foreground pixels (0), stopping when it reaches object boundaries. The boundaries are determined according to the type of neighborhood you specify.

$$LP = \{P \mid P \in ff(i(x, y))\} \quad (12)$$

(ii) Branch Points (*BP*)

The branch points in the skeleton with 3 to 4 neighbors, the *BP* is determined by examining each with 3 to 4 pixels in the bitmap skeletons. The consequence of the skeleton, if the total of eight neighbors $N_8(P)$ with 3 or 4 pixels, this is the Branch Points.

$$BP = \{P \mid N_8(P) = 3 \vee P \mid N_8(P) = 4\} \quad (13)$$

(iii) Dots Points (*DP*)

Points are determined whether higher or lower than short strokes significantly, and isolated that occur on above or below the half line of potential as points. It must determine the number of points and its location relative to the main skeleton structure of the character in every part. And it must be done to determine the number of points can be one, two, or three, can also be above or below the main skeleton of the structure of the character. All pieces are accounted for the first points that follow each track of each endpoint. If you reach the end point along the track to another track procedure finds less than the threshold point. If the path pixels more than one, and joined the halfway point of the feature. This is then added to the connected components (*CC*s) are cleared endpoint at this point feature (one point). Contour, if the width of the point is twice the height of a point, then the line is considered to be a few points. Therefor the dot points (*DP*) is the union of the set of all isolated pixels, and the set of pixels that belong to *CC*s that are less in size than an adaptive threshold T proportional to the estimated character size calculated upon the thinned text image.

$$DP = \{P \mid N_8(P) = 0\} \cup \{P \mid P \in CC \wedge size(CC) < T\} \quad (14)$$

(iv) End Points (*EP*)

The endpoint is the beginning or end of a word segment. The end point in the skeleton with only one neighbor, which also marks the completion of the strokes, the endpoint is determined by examining each individual one pixel in the bitmap skeletons. The consequence of the skeleton, one end points of the total eight neighborhoods a one pixel. Therefore, if the total of eight neighbors $N_8(P)$ one, this is the end point.

$$EP = \{P \mid N_8(P) = 1\} \quad (15)$$

Fig. 6 shows a thinned word image with all possible *MFP*s that will be utilized to guide the characters segmentation process.

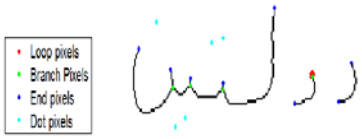


Figure 6: Thinned word image in Fig. 5 with all possible MFPs.

The rest of the sub-sections will be details the modified for bounding box segmentation algorithm and then presents cut points of character to segmentation words.

4.3.2 Modified for bounding box segmentation algorithm

One of the most important methods to improve segmentation accuracy is enhance nonoverlapping connected components (CCs) using bounding box segmentation algorithm. First, find the word baseline as stated above. Then upon finding the baseline, differentiate between two types of CCs. The first is call main (CCs),

which are all (CCs) that intersecting with the baseline “y” coordinate. The second are call auxiliary (CCs), which are all (CCs) that are not intersecting the baseline “y” coordinate. After identifying main CCs and auxiliaries CCs, CCs bounding boxes computed along the y-axis [32]. Fig. 7(a) shows a simple example of CCs of word image, where the main CCs are 2, 3 and 4, and the auxiliaries are 1, 5 and 6; and the horizontal red line representing the baseline.

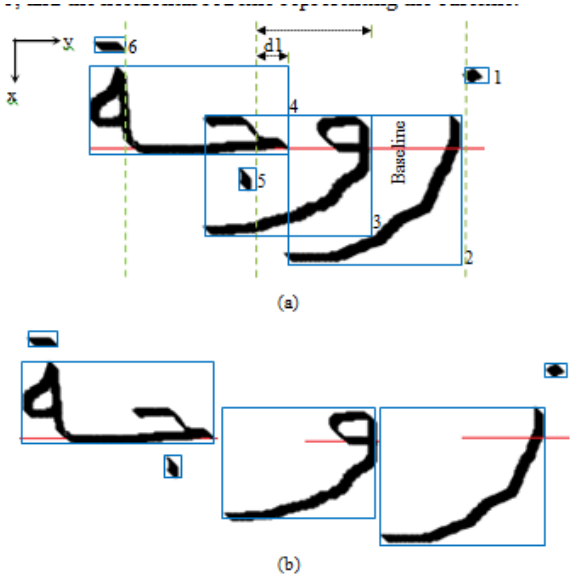


Figure 7: (a) Overlapped CCs within a given word image, (b) nonoverlapping CCs of (a).

Basely conduct the distance analysis on their bounding boxes along the y-axis, in order to identify the baseline overlapped main CCs and their corresponding overlapping distances according to the fact that the Arabic text is written from right to left. The right border of the bounding box is computed to be the farthest right border among all bounding box elements. In Fig. 7(a), for example, main CCs that are overlapping are (2, 3) and (3, 4). Another distance analysis is performed against the auxiliary (CCs), so each can be assigned to its corresponding main CC according to a collection of columns (Ci) which each column is intersecting the box of the auxiliary CC. Fig. 7(a) shows three columns (green dashed line) of Ci where i, auxiliary CC number, equal to 1, 5 and 6. There are three cases as following:

(i) If Ci are not intersecting any main CCs, the auxiliary CC is assigned to the direct next main CC on the left. This is due to the fact that Arabic text is written from right to left, and writers are usually writing main CC first then auxiliaries. For example, In Fig. 7, auxiliary CC number “1” will be assigned to the main CC number “2”.

(ii) If Ci are intersecting only a given main CC, then assign the auxiliary CC to this main. So for example in Fig. 7(a), auxiliary CC number “6” will be assigned to the main CC number “4”.

(iii) If two or more main CCs are intersecting one at least of Ci the absolute distance along y-axis will calculate, between right bounding box of the auxiliary CC and right bounding box of the intersecting main CCs; the one with minimal distance wins the auxiliary, like in case C5 of auxiliary CC number “5” that is intersecting both main CCs “3” and “4”. The auxiliary CC number “5” will be assigned to the main CC number “4” because the absolute distance along y-axis, between the right bounding box of auxiliary CC number “5” and the right bounding box of main CC number “4” (d1) less than the absolute distance along y-axis, between the right bounding box of auxiliary CC of number “5” and the right bounding box of main CC number “3” (d2); {d1 < d2}.

Even though the aforementioned rules resolve almost all the cases, there are some extreme cases where auxiliary CCs in wrong position of another suitable main CCs. Such that, if write character د (Dal) before character ز (Zain) in Fig. 7(a). The column Ci is intersecting main CC د (Dal) and it is suitable for this auxiliary CC, and then assigns the auxiliary CC to this main according to case (ii). Those problems can be solved in subsequent recognition phases like in the post-processing phase, for example, where the recognition results can be corrected against lexicons using different text retrieval techniques.

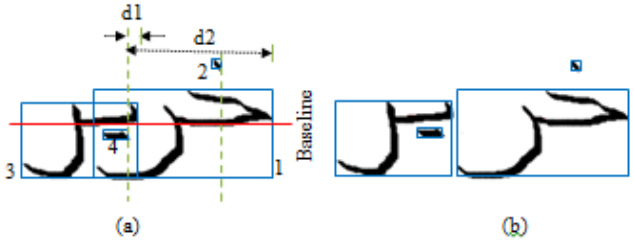


Figure 8: (a) Overlapped connected components within a given word image, (b) nonoverlapping connected components.

Figure 8 shows another example overlapped CCs. Fig. 8 (a) shows that main CCs that are overlapping are 1 and 3, the auxiliaries CCs are 2 and 4; and the horizontal red line representing the baseline. Apply modified bounding box segmentation algorithm on this figure. Therefore auxiliary CC number “2” will be assigned to the main CC number “1” according to case (ii) at above, and the auxiliary CC number “4” will be assigned to the main CC number “3” because d1 < d2 as discuses above in case (iii).

Finally, a distance analysis is preformed against the new sub-word borders and the nonoverlapping CCs done by shifting away the overlapped CCs. Figs. 7(b) and 8(b) show the overlapping free version of Figs. 7(a) and 8(a).

Figure 9: Bounded by a rectangle in the thinned word image of Fig.6.

4.3.3 Cut points of character

After perform bounding box segmentation, the character represent in segmentation. The segmentation algorithm presented in [32] is adopted on the basis of segmentation algorithm. Firstly, generate a new set called cut points (CP) and the Arabic characters have their boundaries in column C with the minimum number of pixels. Cut-point set is a collection of columns that will be indicators columns, where each column is in a word thinning column indicators, which only contain a single pixel. The next step is to exclude from the candidates set all columns that are intersecting with any MFP. After excluding some points from the cut points will notice that alleviate the number of points of the cut points to be the character segmentation properly, so there is no more than a segmented character after the exclusion of points of pieces. Moreover, to mark the start and/or the end of all letters, insert segmentation candidates direct before and after each main CC. Finally, each set of pixels between every two segmentation candidates in the binary image are assumed to represent a letter in the word.

To extract the most possible accurate letter image and to eliminate isolated pixels belonging to neighboring letters that may appear as a result of the crop process, the reconstruction module perform according to [33], in which use those sets of pixels as masks, and their counterparts in the thinned image as a marker. Then, constructed the letter image and save it, as the result of the segmentation. Fig. 10 illustrates the final segmentation results of the word image.

(a) (b)

Figure 10: (a) the segmented characters borders on the thinned image of Fig.6. (b) The segmented characters.

5. Experimental Results

The experimented of the modified for bounding box segmentation algorithm to improve segmentation methodology on 450 word images; Figures 11, 12 and 13 illustrate some of the results. Figure 11 illustrates cases of segmentation of word images without overlapping of main CC or auxiliary CC, it represent 87% of all cases. Partial overlapping cases of auxiliary CCs or main CCs reported in 13% of all cases, as illustrates in Fig. 12 and 13.

The partial overlapping cases between auxiliary CCs and main CCs represent 2.4% of all cases for partial overlapping. It generated when auxiliary CCs in wrong position of another suitable main CCs as discuses in subsection 4.3.2. Fig. 12 illustrates such case: the auxiliary CCs (dot) in characters ي on left column and ن on right column.

The other 10.6% of the partial overlapping cases in main CCs happen when MFPs occur inside the character instead of on its borders and position of the character in the end of word, this leading to call an over-segmentation. That is because part of a stroke is regarded as character representative, which in fact is not. This problem is specific to characters س and ش (SIEN and SHIEN). The first row in Fig. 13 illustrates such case. The second row in Fig. 13 illustrates another case of partial overlapping happen when ـكـ (KAF) occurs in the middle of two connected characters, ك having upper part vertically overlapping the previous character on the right. Also, it appears occasionally incase that MFPs cease to exist between two consecutive characters, leading them to being considered as a representative of one character. This problem is called under-segmentation and it is specific for cases, when the second character to left is connected ا (ALF) or connected ل (LAM) with sheared distortion angle to the left. The last row in Fig. 13 illustrates such case. Those problems may be solved by expanding the MFP set to contain more features points like local minima points and then accordingly modify and add heuristic rules.

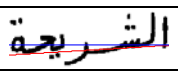
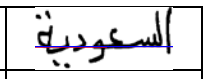
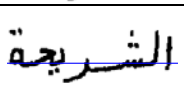
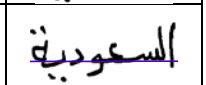
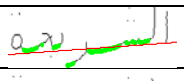
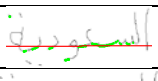
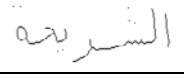
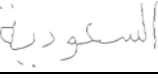
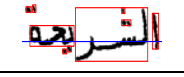
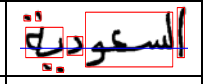
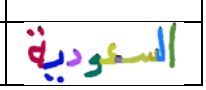
Original Image for *LM*s		
Corrected binary image with *LM*s skew angle		
image after thinning		
Corrected image with thinning slant		
bounded by a rectangle in the *CC*s		
The segmented characters borders on the thinned image		
The segmented characters		

Figure 11: Examples of segmentation of word images without overlapping of main CCs or auxiliary CCs (The skew angle = -3.096 in the left word and -0.142 in the right).

The note through experiments that an algorithm acceptable characters very effectively by identifying algorithm for bounding box segmentation and cut points, which in turn uses a set of rules that enable them to correct segment characters rate since been identified loop, branch, dot and end points. There are some cases are not selected dot points because some of the people writing the points tangled way.

The effort was to enhance the current status of off-line handwritten Arabic character segmentation. Although every of the algorithms summarized in section 2 have their own downsides and superiorities, the offered segmentation results of different systems seem quite successful. It is extremely difficult to make a judgment about the success of the results of segmentation systems, especially in terms of segmentation rates, because of different databases, constraints and sample rates. For words that are handwritten under poor conditions or for freestyle hand writing, there is still an intensive demand in virtually all the phases of the handwritten Arabic character segmentation research.

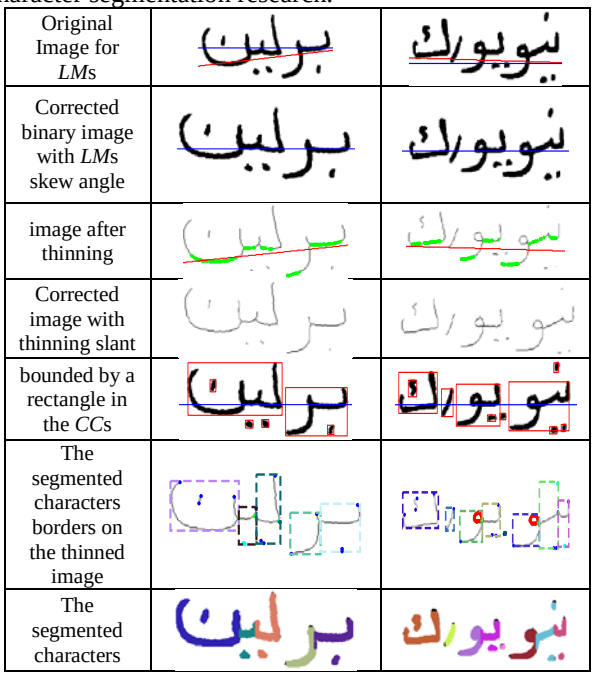
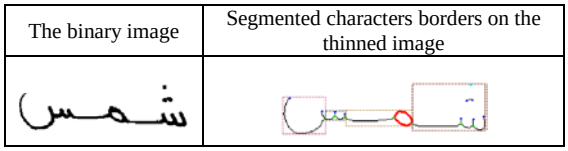


Figure 12: Examples of segmentation of word images with partial overlapping between auxiliary CC and main CCs (The skew angle = -0.32 in the left word and 1.187 in the right).



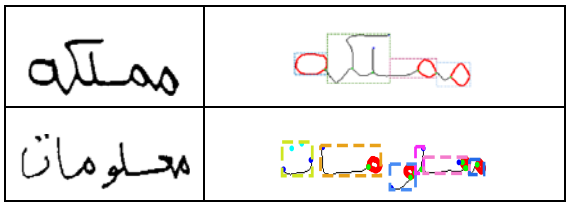


Figure 13: Examples of segmentation of word images with partial overlapping of main CCs.

6. Conclusion

Characters segmentation is an essential phase in handwriting-recognition system. There is not any universally accepted solution in automated handwritten document recognition techniques. In this paper the modified bounding box segmentation algorithm performed to improve segmentation word into characters. This approach is based on a distance analysis on bounding boxes of two CCs: main (CCs), auxiliary (CCs). The modified for bounding box segmentation algorithm is presented and taking place according to three cases. Cut points also determined using structural features for segmentation character. The proposed modified bounding box segmentation algorithm has been successfully tested on 450 word images of Arabic handwritten words. The results were very promising, indicating the efficiency of the suggested approach. However, this technique can be conducted in automated manner to segmentation several off-line Arabic words.

References

[1] Elzobi, M., Al-Hamadi, A., Dinges, L., Michaelis, B.: A structural features based segmentation for off-line handwritten Arabic text. In: 2010 5th International Symposium on I/V Communications and Mobile Network (ISVC), pp. 1–4. Rabat, Morocco (2010).

[2] Belaïd, A., Choisy, C.: Human reading based strategies for offline arabic word recognition. In: Proceedings of the 2006 Conference on Arabic and Chinese Handwriting Recognition, SACH'06, pp. 36–56. Springer-Verlag, Berlin, Heidelberg (2008).

[3] Al Aghbari, Z., Brook, S.: Hahmanuscripts: a holistic paradigm for classifying and retrieving historical arabic handwritten documents. Expert Syst. Appl. 36(8), pp. 10942–10951 (2009).

[4] Lavrenko, V., Rath, T.M., Manmatha, R.: Holisticword recognition for handwritten historical documents. In: Proceedings of the First International Workshop on Document Image Analysis for Libraries, pp. 278–287. ACM, New York (2004).

[5] Blumenstein, M.: Cursive character segmentation using neural network techniques. In: Marinai, S., Fujisawa, H. (eds.) Machine Learning in Document Analysis and Recognition, vol. 90 of Studies in Computational Intelligence, pp. 259–275. Springer, Berlin (2008).

[6] Lorigo, L., and Govindaraju, V.: Segmentation and Pre-Recognition of Arabic Handwriting. In Proceedings of the

Eighth International Conference on Document Analysis and Recognition, vol. 2, pp. 605-609 (2005).
[7] Xiu, P., Peng, L., Ding, X., and Wang, H.: Offline Handwritten Arabic Character Segmentation with Probabilistic Model. Document Analysis Systems. VII, pp. 402-412(2006).
[8] Abdulla, S., Al-Nassiri, A., and Salam, R.A.: Offline Arabic Handwriting Word Segmentation Using Rotational Invariant Segments Features. The International Arab Journal of Information Technology. vol. 5, no. 2 . pp. 200-208(2008).
[9] AlKhateeb,J.H.,Jiang ,J., Ren, J., & Ipson, S.: Component-based Segmentation of words from handwritten Arabic text. International Journal of Computer Systems Science and Engineering, 5(1) (2009).
[10] Al-Hamad H.A., Zitar R. A.: Development of an efficient neural -Based Segmentation Technique for Arabic Handwriting Recognition. Pattern Recognition, vol. 43, no. 8, pp. 2773–2798(2010).
[11] Elzobi, M., Al-Hamadi, A., Al Aghbari, Z.: Off-line Handwritten Arabic Words Segmentation Based on Structural Features and Connected Components Analysis. In I/V Communications and Mobile Network (ISVC) (2011).
[12] Lawgali, A., Bouridane, A., Angelova, M., and Ghassemlooy, Z.: Automatic segmentation for Arabic characters in handwriting documents. In Image Processing (ICIP), 18th International Conference on IEEE, pp. 3529-3532. IEEE (2011).
[13] Eraqi, H., M., and Abdelazeem. S.: A new Efficient Graphemes Segmentation Technique for Offline Arabic Handwriting. Frontiers in Handwriting Recognition (ICFHR), International Conference on. IEEE, 2012.
[14] Samoud, F.B., Maddouri, S.S., and Amiri, H.: Three Evaluation Criteria's towards a Comparison of Two Characters Segmentation Methods for Handwritten Arabic Script. Frontiers in Handwriting Recognition (ICFHR), International Conference on IEEE (2012).
[15] Al Hamad, Husam A.: Neural-Based Segmentation Technique for Arabic Handwriting Scripts. 21st International Conference on Computer Graphics, Visualization and Computer Vision, WSCG (2013).
[16] Osman, Y.: Segmentation Algorithm for Arabic Handwritten Text based on Contour Analysis. International Conference on computing, Electrical and Engineering (ICCEEE) (2013).
[17] Elnagar, A., and Bentrcia, R.: A Recognition-Based Approach to Segmenting Arabic Handwritten Text. Journal of Intelligent Learning Systems and Applications, 7, pp. 93-103 (2015).
[18] MELHI, M., H.: Off-Line Arabic Cursive Handwriting Recognition Using Artificial Neural Networks; PhD thesis. Department of Cybernetics, Internet and Virtual Systems. Bradford, University Bradford (2001).
[19] PLAMONDON, R., and SRIHARI, S. N.: Online and off-line handwriting recognition: a comprehensive survey. Pattern Analysis and Machine Intelligence, IEEE Transactions on, pp. 22, 63-84 (2000).
[20] W. M. Newman and R. F. Sproull: Principles of Interactive Computer Graphics. Sec 17.2. 2nd edition, McGraw Hill (1989).
[21] Ali, A., Shaout, A., Elhafiz, M.: Two stage classifier for Arabic Handwritten Character Recognition, International Journal of Advanced Research in Computer and Communication Engineering, pp 646- 650 (2015).
[22] Bassil, Y., Alwani, M.: Ocr Post-Processing Error Correction Algorithm Using Google's Online Spelling Suggestion, Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, No. 1 (2012).
[23] Zeki, A.M.: The segmentation problem in Arabic character recognition: The state of the art. First International Conference on Information and Communication Technologies, ICICT, pp. 11–26 (2005).
[24] Farooq, F., Govindaraju, V., and Perrone, M.: Pre-processing Methods for Handwritten Arabic Documents", In Eighth International Conference on Document Analysis and Recognition, vol. 1, pp. 267–271 (2005).
[25] Gonzalez, R., and Woods, R., Digital Image Processing (3rd Edition), Prentice Hall, August (2008).
[26] Boubaker, H., Kherallah, M., Alimi, A.M.: New Algorithm of Straight or Curved Baseline Detection for Short Arabic Handwritten Writing. 10th International Conference on Document Analysis and Recognition. ICDAR '09, pp. 778-782. Washington, DC, USA. IEEE Computer Society (2009).
[27] ZHANG, T., and SUEN C.: A Fast Parallel Algorithm for Thinning Digital Patterns, Communications of the ACM, Volume 27 Number 3, pp 236-239 (1984).
[28] Bunke.H. and Wang. P.S.P.: Handbook of Character Recognition and Document Image Analysis. Chapter Image Processing Methods for Document Image Analysis, pp. 15, 19. World Scientic (1997).
[29] Slavik, P., Govindaraju, V. (eds.): Equivalence of different methods for slant and skew corrections in word recognition applications. IEEE Trans. Pattern Anal. Mach. Intell. 23(3), pp. 323–326 (2001).
[30] Al Aghbari, Z.: HAH manuscripts: Aholistic paradigm for classifying and retrieving historical Arabic handwritten documents, Expert Systems with Applications, Vol 36, pp. 10943- 10951 (2009).
[31] El-Abed, H., and Margner, V.: Comparison of Different Preprocessing and Feature Extraction Methods for Offline Recognition of Handwritten Arabic Words. In Ninth International Conference on Document Analysis and Recognition, vol. 2, pp. 974-978 (2007).
[32] Elzobi, M., Al-Hamadi, A., Al Aghbari, Z., and Dings, L.: IESK-ArDB: a database for handwritten Arabic and an optimized topological segmentation approach, In International Journal on Document Analysis and Recognition (IJDAR) (2012).
[33] Vincent, L.: Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. IEEE Trans. Image Process. 2, pp. 176–201 (1993).