

An approach to analysis of Arabic text documents into text lines, words, and characters

Hakim A. Abdo^{1,2}, Ahmed Abdu³, Ramesh R. Manza¹, Shobha Bawiskar⁴

¹Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India

²Department of Computer Science, Hodeidah University, Al-Hudaydah, Yemen

³Department of Software Engineering, Northwestern Polytechnical University, Xi'an, China

⁴Department of Digital and Cyber Forensics, Government Institute of Forensic Science, Aurangabad, India

Article Info

Article history:

Received Dec 4, 2021

Revised Feb 13, 2022

Accepted Mar 11, 2022

Keywords:

Arabic text

Geometric characteristics

Handwritten

Projection

Segmentation

ABSTRACT

Text line extraction from a text document image and segmenting it into isolate words and segmenting these words into individual characters are considered as one of the most critical processes in optical character recognition (OCR) systems development and turning the document into a searchable electronic representation, this paper presents a new approach to analyze the Arabic text documents, the proposed approach contains four steps, preprocessing, text line segmentation, word segmentation, character segmentation. The horizontal projection method are used to detect and extract the text line from preprocessed text documents image, in word segmentation step. The space threshold are computed to determine the spaces among connected components in text line as within-word space or between-words space for segmenting the text line into isolate words, finally thinning method applied to find the skeleton of segmented word and analyses geometric characteristics of the characters to detect ligatures and characters. The proposed approach was tested and evaluated on a set of 115 text images, this set contains images from the King Fahd University of Petroleum and Minerals (KFUPM) handwritten Arabic text (KHATT) database and some images produced by the authors. The experiment results are extremely encouraging, with a success rate of 98.6% for lines segmentation, 96% for words segmentation, and 87.1% for characters segmentation.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Shobha Bawiskar

Department of Digital and Cyber Forensics, Government Institute of Forensic Science

Nipatniranjan Nagar, Caves Road, Babasaheb Ambedkar Marathwada University Campus

Aurangabad Maharashtra 431004, India

Email: shobha_bawiskar@yahoo.co.in

1. INTRODUCTION

Text-line, word, and character segmentation is the technique by which the fundamental elements in a text document image are localized and extracted. Segmentation is a critical stage for handwriting and printed recognition, it is the most important step in in online and offline character recognition [1], [2]. It is the most important and most challenging phase in optical character recognition (OCR). In order to recognize words or characters, the OCR systems must first break the text into lines, then segment the lines into words, and then word into characters [3], [4]. The bad segmentation method causes misrecognition or rejection [5].

Segmentation of text lines includes both detection and extraction of text lines. Detection of text lines commonly locates text line patterns, while text line extraction assigns pixels to text lines with precision. The

text line includes a sequence of, and words are usually made up of a number of sub-words (characters, related components) that are spaced by spaces. In Arabic handwriting, spaces are divided into two types: within-word space, which is the space between sub-words of the same word, and between-word space, which is the gap between two consecutive words.

The spaces in Arabic handwriting do not adhere to any rules because each individual has his or her own unique writing style, and thus each writer has his or her unique way of generating gaps in between words. Extraction of word consists of identifying between-word spaces. When there is not enough space between-word in Arabic handwriting, separation becomes difficult. The majority of the approaches proposed in the literature for extraction of words involve measuring a threshold to characterize the gaps between the words (between-word space) and between the linked components of the same word (within-word space) [6].

Detecting text-lines, words, and characters in Arabic documents remains a challenge. The Arabic documents are regarded to be more sophisticated than other manuscripts written in other languages. This intricacy arises initially from handwritten text features which may vary in writing style, size, orientation, alignment, and where consecutive text-lines might be touched or overlapped, and the second intricacy from the Arabic writing nature: cursiveness of the text, character intersecting, diacritics, diversity of calligraphy, words are frequently split into letters and sub-words, and the spaces between them are varied.

2. RELATED WORK

Segmentation of offline characters is a crucial step before feature extraction and therefore character recognition. In the literature, a variety of text-line and word extraction techniques have been proposed, for text-line segmentation, Kumar *et al.* [7] has developed an approach based on graphs for handwritten text lines extraction, the approach is highly resistant to variances in font size and to non-uniform asymmetry. Shi *et al.* [8] proposed a technique based on a directional filter and a local connectivity map of generalized adaptive. It works well for varying, touching, or crossing text lines.

Barakat *et al.* [9] suggested an unsupervised method for extracting text lines, which was driven by the relative variation in text lines and space between text lines. The number of foreground pixels over text lines differs significantly from the number of foreground pixels over text line gaps. A Siamese convolutional network is used in this technique to predict whether two given document picture patches are similar or distinct, based on the number of foreground pixels in the patches. Alghamdi *et al.* [10] utilized projection of horizontal for segmentation of the historical document image into text lines, this approach transforms the image from two dimensions into one dimension by computing the pixels of all rows. Finally, the quantity of the minimum pixel is used for cropping all lines in the historical document.

Arvanitopoulos and Susstrunk [11] utilized a seam carving technique, which is a top-down method. They use a projection profile matching method to calculate medial seams on the text lines initially. Then, using a modified version of the seam carving method, they calculate the separating seam. On the Arabic dataset they utilized, their approach yielded positive results (99.9%). Ouwayed and Belaïd [12] used analysis of morphology to determine the Arabic words last letters; the suggested method was tested on overlapping texts and found to be highly efficient.

For extraction of words, in literature, most of the approaches given employ a certain measure of the distance between connecting successive elements and establish a threshold for classifying the gaps between words (between-words space) and linked constituents (between-words space) from the same word (within-word space) [13]. To establish an appropriate threshold [14] suggested an approach based on the magnitude of the gaps, the method classified the gaps among connected components into three sets.

For extraction of character, in art literature, there are many techniques are used, the segmentation of explicit methods segments a word image into a number of tiny components, whereas the segmentation of implied method combines the segmentation and recognition stages by segmenting words into characters and recognizing them at the same time. The character segmentation algorithms based on the techniques used can be categorized into projection profile-based methods, character skeleton-based methods, contour tracing-based methods, template matching-based methods, neural network (NN) based methods, hidden markov (HM) models-based methods, line adjacency graph-based methods, morphological operations-based methods, and recognition-based segmentation methods [15].

Projection profile based methods are based on the fact that the connecting stroke among successive letters is thinner than the letter itself. The projection of horizontal is used to separate lines and identify text baseline, while the projection of vertical is used to segment words, sub-words, and characters. These methods compute the projections of vertical and horizontal method is use in Alghamdi *et al.* [10] at the first, the pre-processing is done for cleaning the historical document image. The proposed method utilized projection of Horizontal for segmentation of the historical document image into text lines, and projection of vertical for the segment the text lines into characters, and then the erosion followed by a dilation are applied for each text line. Finally, the density is calculating for all columns and segment the characters. Anwar *et al.* [16] the

method looks for possible segmentation locations in a segmented word image based on the fact that the connecting stroke among successive letters is thinner than the letter itself. Character skeleton is another technique is used for segmentation the word into characters, in the recognition of character, the skeleton of a shape contains all of the necessary information.

In general, a number of approaches have been reported to extract skeletons in the literature, the approaches proposed specifically for Arabic are [17] presented a novel skeletonization technique based on clustering the character image for solitary Arabic letters. After that, the skeleton was created by locating the neighboring matrix of various clusters. They finished by eliminating unimportant vertices from the skeleton. To cluster the Arabic letter, Altuwaijri and Bayoumi [18] used a self-organizing neural network. The skeleton was created by plotting the cluster centers and linking neighboring clusters in a straight-line succession. Cowell and Hussain [19] utilized an iterative mitigation method with post-processing to create thin shapes of segmented Arabic letters. They also discussed the issues of thinning Arabic letters from poor image quality.

Contour tracing method used in Osman [20] the proposed algorithm divides the acquired image into text lines and connected components (sub words). Then, the contour of every connected component is traced. Then, the exact points extract depending on changing of the contour state from the line of vertical to a horizontal line or vice versa. The last step is these points' coordinates are then used as the separation points. The method in Wshah *et al.* [21] depend on the idea that every linked letter inside sub-word has junction points in the letter skeleton, this method analysis skeleton of a sub-word image to determines the junction points, in order to determine the shortest path to the extracted junction points the contour of image is analyzed. Finally, the points of segmentation are the first three lower peaks of the distance map between the intersection points and the chain code. Mohammad *et al.* [22] the contour segmentation method is used, the proposed method is composite of four phases, the method inputs are the binary image of word/sub-word, the first step is to extract the connected component contour of the word/sub-word, from extracted contour the points of start and end are determined by contour tracing as up-contour, and then the identify the splitting point from extracted up-contour by using pixels values, and finally, the post-processing phase tries to identify each portion in order to evaluate if it needs to be combined or a separate character. In Omidyeganeh *et al.* [23] the method collects information on the word's general shape by identifying the contour of word, which depicts the pixels that make up the word's outer shape. It uses a representation of the word shape (contour) based on the fact that each letter has a high contour followed by a flat or low contour, with the segmentation points identified before the contour starts rising, to find the potential segmentation point. Neural network used in Radwan *et al.* [24] a model consists of multi-channel NN as input layers, the input of these layers are three windows. The contains a sliding window as a middle channel, as well as next and prior windows for further context. The suggested model predicts that the present window is probably a segmentation area. The channel on the left is responsible to learn the characters' right parts, while the channel on the right is responsible for learns the characters left parts, and the middle channel is responsible for learns the region in among. An output layer learns the relationships among every channel's unique property in order to determine the correlation among them.

Graph approach is used in Elgammal *et al.* [25] for Arabic character segmentation, the suggested method is based on the morphological relationship among the base-line and the line neighboring graph (LAG) text representation. Hidden Markov method (HMM) is the suggested method contains two modules: the trainer module is responsible to prepare and train HMM, this module trained on printed Arabic text and the isolation module is responsible to segment input images into letters [26].

Connected component-based method is used in Alirezaee *et al.* [27] the text lines are segmented into a linked block (word/sub-word or characters). Before segmentation of the text document image, the preprocessing was applied for cleaning the document. Then, morphological linked block extraction is utilized for extraction of the line block utilizing a specific formula. Finally, in post-processing applied specific conditionals and formulas to text elements extraction. Also this technique with a clustering algorithm was utilized in Ouwayed and Belaid [12]. Firstly, the picture is transformed into a black and white image, then this binary image going to segment into linked blocks. Some points in the center of a linked block that could be used as a feed to clustering method. Following that, the points are grouped utilizing the k-means method, finally, ultimate lines are generated.

A recognition-based character segmentation technique is used in Inkeaw *et al.* [28] The segmentation and recognition stages are carried out in this technique at the same time. The technique searches for components in word segments that fit specified classes in its alphabet and divides them into their letters without breaking them into smaller units. This method also used in Elnagar and Bentrçia [29] at the first, the pre-processing is done for cleaning and extracting the features for the image, then the pre-processed text image was segmented into text lines, and the segmented text lines were segmented into words. For each segmented word, the thinning is applied, and then the main linked block was obtained in all thinned words, three kinds of regions features were extracted from these main linked blocks, seven factors are used to

determine the start and end of the cutting points, finally, the features were extracted from segmented regions and feed it artificial neural network to classify it as stroke or a character.

3. PROPOSED APPROACH

In this work, we proposed an approach for segmentation of Arabic text document images into lines, words, and characters. The input of system is text document images (printed or handwritten). The proposed approach involves four main steps: preprocessing, line segmentation, word segmentation, and character segmentation. The preprocessing step aims at cleaning up the text document images for the next segmentation steps. Segmentation of lines step aims to isolate the text lines of the text document image. The segmentation of words step allows us to segment the segmented text lines into word images. We compute thresholding (T) from each extracted text line. The T is used to determine the distances between connected components and isolate these connected components with between-word space. The last step is to divide those words into individual characters. In order to recognize ligatures and characters, the geometric characteristics of the characters are utilized.

3.1. Preprocessing

The preprocessing stage involves preparing the text document images before segmenting them to simplify the segmentation steps processing, and the output of this stage is used to improve the segmentation procedure performance. Figure 1 depicts the results of Arabic text image pre-processing step. Preprocessing step includes binarization, slant correction, and cropping process:

- Binarization aims to separate the text image background whether the original image is in color or grayscale, resulting in a two-tone image with black background and white in the text or the reverse [30]. Figure 1(a) shows the original text image and Figure 1(b) depicts the result of the binarization step.
- Slant correction: Slant correction, also known as a skew correction, seeks to adjust the inclination angle of a text image [31]. Figure 1(c) depict result of Slant correction step.
- Cropping aims to eliminate the excess space surrounding the rectangular region carrying the picture of the noise-free text document image. Figure 1(d) depict result of cropping step.



Figure 1. Arabic text image pre-processing: (a) original text image, (b) text image after binarization, (c) text image after slant correction, and (d) text image after cropping

3.2. Text line segmentation

In order to segment of text line, we used the method presented in Lamsaf *et al.* [32] method used horizontal projection approach. The method first calculates the histogram of the horizontal projection of the text document image, and then impacts the value of 0 for lines with sums less than or equal to the threshold $l=12$. Second, the method recalculates the histogram of horizontal projection and computes the difference between each two successive vector components of the result. Third, it calculates the local maximum and minimum of the previous step's result, and then impacts local maximum and minimum neighbourhood lines with the value 1, finally separating the text lines. The text line segmentation results are shown in Figure 2.

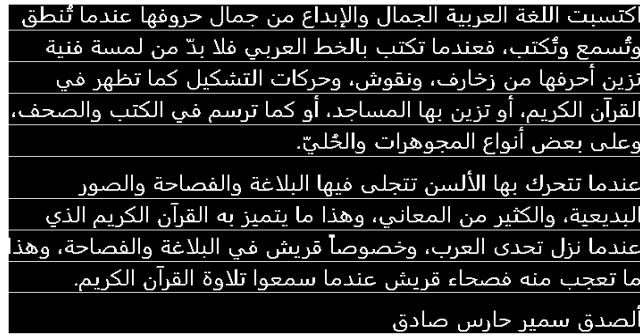


Figure 2. Result of text line segmentation

3.3. Word segmentation

The spaces in Arabic writing are two types: The spacing among sub-words of a single word is known as within-word space, and the spacing among two words is known as between-word space, detecting between-word spaces is the first step in word segmentation. Let $CSW = (CSW_1, CSW_2, \dots, CSW_n)$ be the spaces in a text line, for the space threshold T , we determine one space CSW_i as a word space when the sum of the number of consecutive foreground pixels with a value of zero is greater than space threshold T , The space threshold is the average of all gaps among linked components in the text line, All word space records as $SW = (SW_1, SW_2, \dots, SW_l)$. The word segmentation process as shown in Algorithm 1 and the word segmentation results are shown Figure 3.

Algorithm 1. Word segmentation

```

Input: TLI as a text line image.
ListOfCSW←ListOfCountOfForegroundColorInEachClumons where Count=0.
CSW←MergeConsecutiveListOfCSW.
T←ComputeSpaceThresholding.
while each segment in CSW.
    if CSW >T.
        SWI←Segment (CSW) .
    End while.
Output: segmented word images.

```

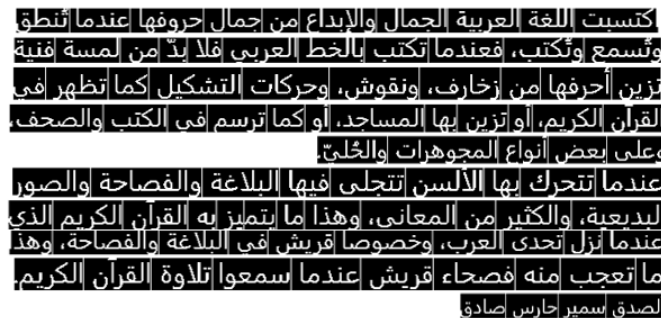


Figure 3. Result of segmented text line into word

3.4. Character segmentation

Arabic characters have three types. First category the characters containing closed loop such as HAA “هـ”, TAH “ط”, SAD “ص”, DTAD “ض”, QAF “ق”, the second category the characters containing semi-loop such as LAM “ل”, TAA “ت”, BAA “ب”, NOON “ن”, AIN “غ”, and the third category characters is similar to ligatures such as RAA “ر”, and ALIF MAQSURA “ى”.

In open characters, distinguishing between ligatures and character segments is difficult. Ligature is a term used to describe a connection between two or more consecutive letters. In written Arabic language words, Successive TAA “ت”, NOON “ن”, and THAA “ث” may look like SHEEN “ش” or SEEN “س” and vice versa. Successive FAA “ف” and HMZA “هـ” may appear as DHAD “ض”. The proposed algorithm (Algorithm 2) analyses geometric characteristics of the characters to detect ligatures and characters.

Algorithm 2. Character segmentation

```

Input: SWI as segmented word images.
OIMG←CopyFromImage(SWI) .
WSL←SkeletonOfWord(SWI) .
ODAD←OmitDotsAndDiacritics(WSL) .
ListOfCSC←ListOfCountOfForegroundColorInEachClumons where Count=0 or 1.
CSC←MergeConsecutiveListOfCSC.
LEP←LastOfEndPointFromLeft.
[[Set CSC] _n=LEP where [[SetLEP-CSC] _n<d, d=4, n is last CSC.
ListOfsegmentedCharacter←CSC(Segment,OIMG) .
For Each Segment in ListOfsegmentedCharacter.
    if structural similarity of merged TwoConsecutiveSegments S_i, s_(i+1) ≈ structural
    similarity of "ص" or "م" or "ف" or "د" then merge S_i, s_(i+1)
    if structural similarity of merged ThreeConsecutiveSegments S_i, s_(i+1) and s_(i+3)
    ≈ structural similarity of "س" or "د" or "ش" or "ث" then merge S_i, s_(i+1) and
    s_(i+3)
Output: S as segmented character images.
    
```

3.5. Description of the proposed algorithm

Before starting the analysis of geometric characteristics of the characters, we applied thinning algorithm on segmented word images to make their stroke width 1 pixel as shown in Figure 4. The thinning technique creates the image's skeleton. A skeleton has a width of one pixel and is created by outlining the word's centerline. The existence of dots and diacritics above or down of some opened characters causes an over-segmentation error. We used the connected component algorithm to omit the dots and diacritics from the image. First, we detect all connected components and then deleted all small, connected components. Figure 5 shows the output after omitted the dots and diacritics from word image.

After omitting the dots and diacritics, the word image scanned from top right to bottom left and compute the number of foreground pixels for each column, the number of foreground pixels with 0 or 1 are termed as a list of candidate segmentation columns (LCSC), and Figure 6 shows the LCSC of the word image. As shown in Figure 7 the consecutive LCSC are merged and termed as candidate segmentation columns (CSC).



Figure 4. Skeleton of word



Figure 5. Word images after omit dots and diacritics



Figure 6. Word image with LCSC



Figure 7. Word image with CSC

Over segmentation occurs in open character when coming at the end of a word like "ص", "س", "ف", to solve this over-segmentation we detect the last end point LEP and subtract the x-axis value of LEP from the x-axis value of CSC_n if the result R satisfied R<D, we replace CSC_n by LEP as shown in Figure 8 and Figure 9, factor D determines the distance between LEP and CSC_n and we have an experienced value: D=4.

Two other over-segmentation Appeared as shows in Figure 10, first one occurs in "ص", "ض" letters as shown in Figure 10(a), second occurs in "س", "ش" letters as shown in Figure 10(b). Step 10 aim to solve these over segmentations, the result of this step shown in Figure 11.



Figure 8. Word after detecting last end point and CSC_n



Figure 9. Segmentation after applied step 8



Figure 10. Over-segmentation of (a) over-segmentation occurs in “ص”, “ض” and (b) over-segmentation occurs in “س”, “ش”

In step 10 to solve the over-segmentation occurs in “ص”, “ض”, we compared structure similarity of two merged consecutive segmented characters (s_i and s_{i+1}) with structure similarity of “ص” or “ض”, if it matches, we merge the S_i and S_{i+1} in one segmentation character as shows Figure 11(a). We used *ssim* function to computes the structural similarity index between two images. Likewise, we compared structure similarity of three merged consecutive segmented character (S_i , S_{i+1} , and S_{i+2}) with structure similarity of or , if it is match, we merge the S_i , S_{i+1} and S_{i+2} in one segmentation character as shown in Figure 11(b). Figure 12 depicted the result of comparison of structural similarity index.



Figure 11. Words images after solving over-segmentation by step 10, (a) result of solving over-segmentation in “ص”, and (b) result of solving over-segmentation in “س”

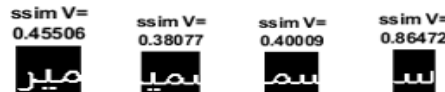


Figure 12. Structure similarity index of merged three consecutive segmented character image with seen image

4. RESULTS AND DISCUSSION

We utilized a collection of 40 text images to test our approach. This collection contains images from KFUPM handwritten Arabic text (KHATT) database, and some images produced by the author. We used a horizontal projection profile method for all text images, segmenting each text block into its text lines, then using the word segmentation method to segment the text line into words, and finally using the character segmentation method to segment all segmented words into individual characters. To assess the efficacy of the suggested method, we tested the algorithm on set of 115 images of Arabic printed and handwritten texts, this set contains images from the KHATT database [33] some images produced by the author.

We calculated the success rate of the obtained result, the segmentation errors were divided into three categories: bad-segmentation, over-segmented, and miss-segmentation. thus, most errors of the bad-segmentation that are common among all fonts occurred in LAMALIF “لا”, “لا” character, and in the “Decotype Thuluth”, “Traditional Arabic” and “Advertising Bold” fonts owing the overlapping letters as “لح”, “لم”, “لح”. On the other hand, most errors of the miss-segmentation appear in certain font types owing to tiny size font. In fact, at tiny font sizes, the segmentation spots may not be seen and therefore cannot be identified since the space among sub-parts is extremely short. It is difficult to compare the findings of the suggested segmentation approach to those of other researchers' segmentation approaches published in the literature since various researchers presented their segmentation results under different limitations and utilized different types of databases. Table 1, Table 2, and Table 3 present the result of text line Segmentation, word segmentation, character segmentation, respectively. Figure 13 shows example of wrong character segmentation, Figure 14 shows sample of segmentation results.

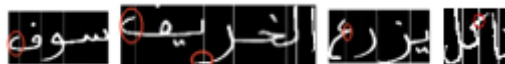


Figure 13. Example of wrong character segmentation

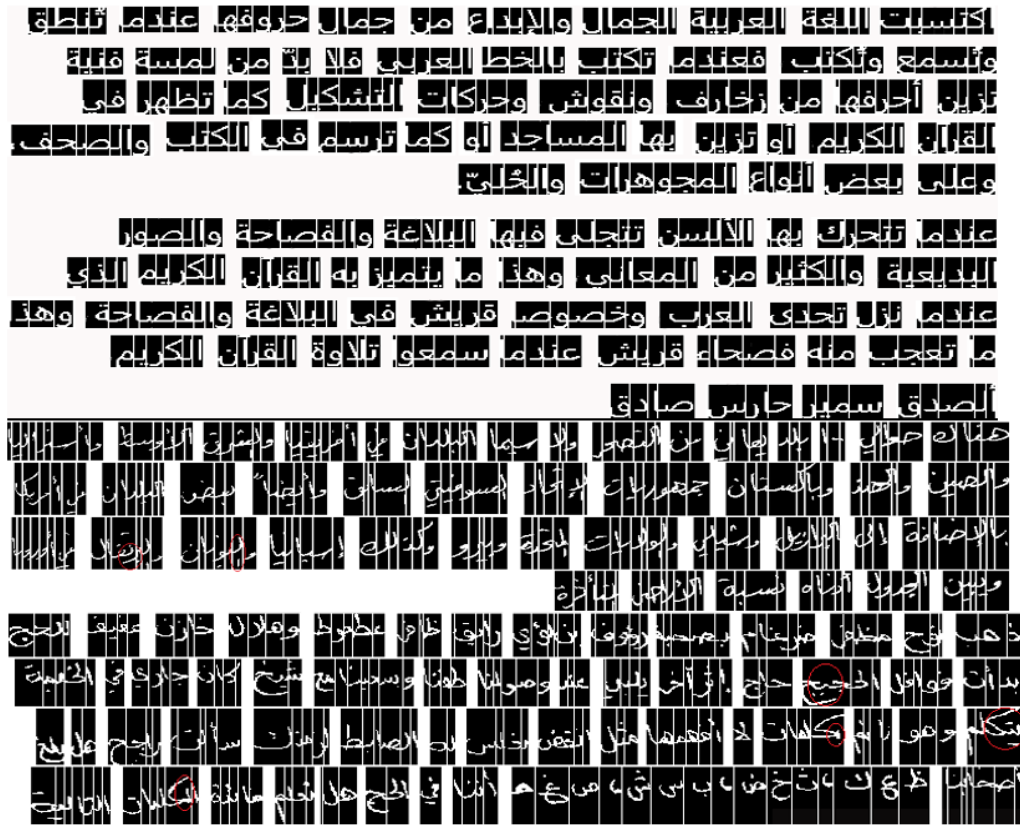


Figure 14. Samples of segmentation results

Table 1. Text line segmentation results

Count of text image used in the experiment	Count text lines in images	Correct segmented text line (percentage)	Incorrect segmented text line (percentage)
115	575	567 (98.6%)	8 (1.4%)

Table 2. Word segmentation results

Count of the words in text lines used	Correct segmented word (percentage)	Incorrect segmented word (percentage)
5175	4968 (96%)	207 (4%)

Table 3. Character segmentation results

Count of character in word images	Correct segmented characters (percentage)	Incorrect segmented characters (percentage)	Incorrect segmented characters (percentage)		
			Over-segmented	Miss-segmented	Bad-segmented
31050	27046 (87.1%)	4004 (12.9%)	1135	2378	491

5. CONCLUSION

Analyzing text images into text lines, words, and characters is still a hot topic of study, particularly for cursive writing. Segmentation of characters is considered as one of the most important phases in OCR systems development owing to font differences (e.g., size, type, and style), the presence of complicated types of fonts, and character overlapping. The presented segmentation approach reduced the issue of over-segmentation, which was evident in open character segmentation, particularly in the characters SEEN, SHEEN, SAD, and DTAAD. The proposed approach has shown outstanding results when it comes to segmenting text document images into lines and words. This method, on the other hand, has shown outstanding results in the segmentation of ligatures that occur between successive closed and opened letters, with assured accurate segmentation in the case of printed text document pictures without touching characters. Miss-segmentation problems occurred in certain font types, such as "Decotype Thuluth," "Traditional Arabic," and "Advertising Bold," due to overlapping characters, in which one letter appears over another, making it impossible for the proposed approach to

identifying the ligature between the letters. In future work, there is a need to enhance some of the pre-processing methods, e.g., removing noise and slant correction. In addition, enhancing the character segmentation stage, as well as replacing the ssim function that was used in step 10 with machine learning techniques.




REFERENCES

- [1] F. Fitrianiingsih, S. Madenda, E. Ernastuti, S. Widodo, and R. Rodiah, "Cursive handwriting segmentation using ideal distance approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 5, pp. 2863-2872, Oct. 2017, doi: 10.11591/ijece.v7i5.pp2863-2872.
- [2] M. A. Abuzaraida, M. Elmehrek, and E. Elsomadi, "Online handwriting Arabic recognition system using k-nearest neighbors classifier and DCT features," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 4, pp. 3584-3592, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3584-3592.
- [3] K. Thammarak, P. Kongkla, Y. Sirisathitkul, and S. Intakosum, "Comparative analysis of Tesseract and Google Cloud Vision for Thai vehicle registration certificate," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, p. 1849, Apr. 2022, doi: 10.11591/ijece.v12i2.pp1849-1858.
- [4] R. Arief, A. Benny Mutiara, T. Maulana Kusuma, and H. Hustinawaty, "Automated hierarchical classification of scanned documents using convolutional neural network and regular expression," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 1018-1029, Feb. 2022, doi: 10.11591/ijece.v12i1.pp1018-1029.
- [5] C. Neche, A. Belaïd, and A. Kacem-Echi, "Arabic handwritten documents segmentation into text-lines and words using deep learning," in *HAL Open Science*, pp. 1-7, 2020.
- [6] A. Al-Dmour and A. Al-Dmour, "Segmenting Arabic handwritten documents into text lines and words," *International Journal of Advancements in Computing Technology*, vol. 6, no. 3, pp. 109-119, 2014.
- [7] J. Kumar, W. Abd-Almageed, L. Kang, and D. Doermann, "Handwritten Arabic text line segmentation using affinity propagation," in *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems - DAS '10*, 2010, pp. 135-142, doi: 10.1145/1815330.1815348.
- [8] Z. Shi, S. Setlur, and V. Govindaraju, "A steerable directional local profile technique for extraction of handwritten Arabic text lines," in *2009 10th International Conference on Document Analysis and Recognition*, 2009, pp. 176-180, doi: 10.1109/ICDAR.2009.79.
- [9] B. K. Barakat *et al.*, "Unsupervised deep learning for text line segmentation," in *2020 25th International Conference on Pattern Recognition (ICPR)*, Jan. 2021, pp. 2304-2311, doi: 10.1109/ICPR48806.2021.9413308.
- [10] A. Alghamdi, D. Alluhaybi, D. Almeahmadi, K. Alameer, S. Bin Siddeq, and T. Alsubait, "Text segmentation of historical Arabic handwritten manuscripts using projection profile," in *2021 National Computing Colleges Conference (NCCC)*, Mar. 2021, pp. 1-6, doi: 10.1109/NCCC49330.2021.9428836.
- [11] N. Arvanitopoulos and S. Susstrunk, "Seam carving for text line extraction on color and grayscale historical manuscripts," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, Sep. 2014, pp. 726-731, doi: 10.1109/ICFHR.2014.127.
- [12] N. Ouwayed and A. Belaïd, "Separation of overlapping and touching lines within handwritten Arabic documents," in *Computer Analysis of Images and Patterns*, 2009, pp. 237-244.
- [13] M. J. Mohammed, S. M. Tariq, and H. Ayad, "Isolated Arabic handwritten words recognition using EHD and HOG methods," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 2, p. 801, May 2021, doi: 10.11591/ijeecs.v22.i2.pp801-808.
- [14] A. Al-Dmour and R. Abu Zitar, "Word extraction from arabic handwritten documents based on statistical measures," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 5, pp. 436-444, May 2016, doi: 10.15866/irecos.v11i5.9384.
- [15] A. M. Zeki, M. S. Zakaria, and C.-Y. Liang, "Segmentation of Arabic characters," *International Journal of Technology Diffusion*, vol. 2, no. 4, pp. 48-82, Oct. 2011, doi: 10.4018/jtd.2011100104.
- [16] K. Anwar, Adiwijaya, and H. Nugroho, "A segmentation scheme of Arabic words with harakat," in *2015 IEEE International Conference on Communication, Networks and Satellite (COMNESTAT)*, Dec. 2015, pp. 111-114, doi: 10.1109/COMNESTAT.2015.7434299.
- [17] S. A. Mahmoud, I. AbuHaiba, and R. J. Green, "Skeletonization of Arabic characters using clustering based skeletonization algorithm (CBSA)," *Pattern Recognition*, vol. 24, no. 5, pp. 453-464, Jan. 1991, doi: 10.1016/0031-3203(91)90058-D.
- [18] M. M. Altuwaijri and M. A. Bayoumi, "Arabic text recognition using neural networks," in *Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*, vol. 6, 1994, pp. 415-418, doi: 10.1109/ISCAS.1994.409614.
- [19] J. Cowell and F. Hussain, "Thinning Arabic characters for feature extraction," in *Proceedings Fifth International Conference on Information Visualisation*, 2001, pp. 181-185, doi: 10.1109/IV.2001.942056.
- [20] Y. Osman, "Segmentation algorithm for Arabic handwritten text based on contour analysis," in *2013 international conference on computing, electrical and electronic engineering (ICCEEE)*, Aug. 2013, pp. 447-452, doi: 10.1109/ICCEEE.2013.6633980.
- [21] S. Wshah, Z. Shi, and V. Govindaraju, "Segmentation of Arabic handwriting based on both contour and skeleton segmentation," in *2009 10th International Conference on Document Analysis and Recognition*, 2009, pp. 793-797, doi: 10.1109/ICDAR.2009.152.
- [22] K. Mohammad, A. Qaroush, M. Ayesh, M. Washha, A. Alsadeh, and S. Agaian, "Contour-based character segmentation for printed Arabic text with diacritics," *Journal of Electronic Imaging*, vol. 28, no. 4, p. 1, Aug. 2019, doi: 10.1117/1.JEI.28.4.043030.
- [23] M. Omidyeganeh, K. Nayebi, R. Azmi, and A. Javadtalab, "A new segmentation technique for multi font Farsi/Arabic texts," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2005, pp. 757-760, doi: 10.1109/ICASSP.2005.1415515.
- [24] M. A. Radwan, M. I. Khalil, and H. M. Abbas, "Predictive segmentation using multichannel neural networks in Arabic OCR system," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9896, 2016, pp. 233-245, doi: 10.1007/978-3-319-46182-3_20.
- [25] A. M. Elgammal and M. A. Ismail, "A graph-based segmentation and feature extraction framework for Arabic text recognition," in *Proceedings of Sixth International Conference on Document Analysis and Recognition*, 2001, pp. 622-626, doi: 10.1109/ICDAR.2001.953864.




- [26] A. M. Gouda and M. A. Rashwan, "Segmentation of connected arabic characters using hidden markov models," in *2004 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2004. CIMSA., 2004*, pp. 115–119, doi: 10.1109/CIMSA.2004.1397244.
- [27] S. Alirezaee, M. Ahmadi, H. Aghaeinia, and K. Faez, "An efficient restoration algorithm for the historic middle-age Persian (Pahlavi) manuscripts," in *2005 IEEE International Conference on Systems, Man and Cybernetics, 2005*, vol. 3, pp. 2114–2120, doi: 10.1109/ICSMC.2005.1571461.
- [28] P. Inkeaw, J. Bootkrajang, P. Charoenkwan, S. Marukatat, S.-Y. Ho, and J. Chaijaruwanich, "Recognition-based character segmentation for multi-level writing style," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 21, no. 1–2, pp. 21–39, Jun. 2018, doi: 10.1007/s10032-018-0302-5.
- [29] A. Elnagar and R. Bentrchia, "A recognition-based approach to segmenting Arabic handwritten text," *Journal of Intelligent Learning Systems and Applications*, vol. 07, no. 04, pp. 93–103, 2015, doi: 10.4236/jilsa.2015.74009.
- [30] K. Saddami, K. Munadi, Y. Away, and F. Arnia, "Improvement of binarization performance using local otsu thresholding," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, p. 264, Feb. 2019, doi: 10.11591/ijece.v9i1.pp264-272.
- [31] S. S. Bafjaish, M. S. Azmi, M. N. Al-Mhiqani, and A. A. Sheikh, "Skew correction for mushaf Al-Quran: a review," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 1, p. 516, Jan. 2020, doi: 10.11591/ijeecs.v17.i1.pp516-523.
- [32] A. Lamsaf, M. Aitkerroum, S. Boulaknadel, and Y. Fakhri, "Text line and Word extraction of Arabic handwritten documents," in *Innovations in Smart Cities Applications Edition*, 2019, pp. 492–503.
- [33] S. A. Mahmoud *et al.*, "KHATT: An open Arabic offline handwritten text database," *Pattern Recognition*, vol. 47, no. 3, pp. 1096–1112, Mar. 2014, doi: 10.1016/j.patcog.2013.08.009.

BIOGRAPHIES OF AUTHORS






Hakim A. Abdo    is a Ph.D. Scholar in Computer Science, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India. He received M.Sc. Engineering degree in Computer Software and Theory from Yunnan University of Finance and Economics, China. He received the B.ED. Degree from Hodeidah University, Yemen. He has worked as an Assistant professor in the Department of Computer Science, Hodeidah University. His main research interests are pattern recognition, machine learning, Image Processing, and Computer Vision. He can be contacted at email: hakim.abdulghaffar@gmail.com.






Ahmed Abdu    is a Ph.D. Scholar in software engineering, Northwestern Polytechnical University, Xi'an, China. He received his Bachelor of Engineering from the Software Engineering Department, Faculty of Engineering and Information Technology, Taiz University, Taiz, Yemen. He received his M.Sc. information engineering from China University of Geoscience, Wuhan, China. His research Interests software quality, software defect prediction, code analysis, Deep learning models. He can be contacted at email: ahmedabd39@hotmail.com.



Prof. Dr. Ramesh R. Manza    is Professor at Dr. Babasaheb Ambedkar Marathwada University. He received his Ph.D. (Image Data Compression) from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, in 2006. He received his M.Sc. (Computer Science) from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, in 1998, with First Division, 3rd Rank. He received his B.Sc. from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, in 1996, with First Division, and His Specialization Areas are Bio Medical Image Processing, Computer Vision, Remote Sensing, Nano Robotics, MEMS, Microprocessor, Embedded, and Networking. He can be contacted at email: manzaramesh@gmail.com.



Dr. Shobha Bawiskar    is Assistant Professor at Government Institute of Forensic Science, Aurangabad 431001. She received Ph.D. from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India, in 2016. She received M.Phil. (Information Technology) from Yashwantrao Chavan Maharashtra Open University, Nasik, India, in 2008. She received M.Sc. Computer Science from Vivekanand College of Arts & Science, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, in 2007. She received her B.C.S. from College of Information Technology and Management, Aurangabad Affiliated to Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India. Winner of "international researcher award 20-21" which is registered with Ministry of MSME Govt of India. She can be contacted at email: shobha_bawiskar@yahoo.co.in.