A New Segmentation Technique of Handwritten Offline Arabic Text

Asst. Prof. Dr. Ban N. Dhannoon, Dr. Imad J. Mohammed, Muna k.Dhamad

Abstract—Arabic handwriting recognition is the hardest applications in the optical character recognition (OCR) because of the characteristics of Arabic characters and the variety of written styles which differ from writer to another, even for the same writer at a different time. To get good recognition, there must be a correct segmentation. This paper proposes a new segmentation technique for Arabic text. Three stages are needed to reach to highest ratio of segmentation. Line segmentation which depends on the row density to detect spaces among lines and baseline for each line. The second stage is used to extracts dots information from each line and erases them. Final stage is shape segmentation which is proposed to segment word/sub word into characters. The segmentation model is designed based on the nature of Arabic writing. Results are tested on the IfN/ENIT database of Tunisian city names which indicate the effectiveness of the proposed system.

Index Terms—Arabic handwriting, Offline Arabic Text, Optical character recognition, Segmentation.

1 INTRODUCTION

Optical Character Recognition (OCR) is a branch of computer vision whose objective is to convert text in digital image, to editable text [1]. It is deal with machine-print or handwriting but is more commonly machine-print.

The handwritten automatic recognition has been classified into two types of categories depends on the input data “offline" and “online" [2].

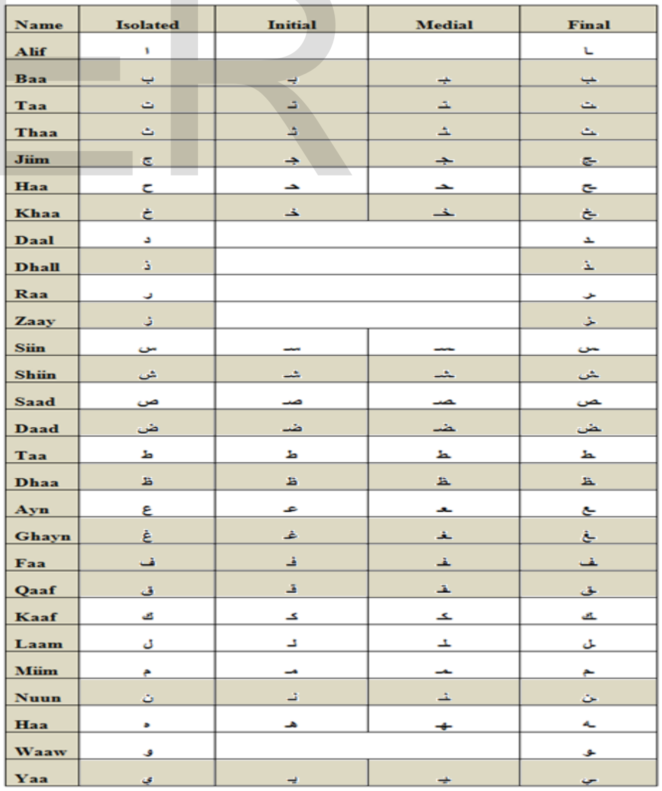
Online handwriting recognition is considered less difficult than offline since the temporal information in the script is available. Also pen speed and even pressure information may be available [3,4]. Furthermore, it is also clear that the offline is the case that simulates the conventional reading task performed by humans.

Handwriting Arabic character recognition is one of the most challenging tasks of research in handwriting recognition, due to, the complexities of Arabic script like cursiveness, multiple shapes of one character depending on its relative position in the word this expands the number of character classes, Arabic writing contains many fonts and writing styles and characters of the same font have different sizes [5, 6]. Table1 explains shapes of Arabic characters

Our contribution in this research proposes a new algorithm that is able to segment a variety of Arabic handwritten words depending on the special structural characteristics of the Arabic characters, it can separate each character according to their position (beginning, middle, end, and isolated characters) that’s led to improve the recognition process, it can extract dots information for each character according to their position and their number (below single dot, above single dot, below double dots, above double dots, above triple dots) that’s led to improve the recognition process.

TABLE 1

THE ARABIC ALPHABET



Name	Isolated	Initial	Medial	Final
Alif	ا			ـا
Baa	ب	بـ	ـبـ	ـب
Taa	ت	تـ	ـتـ	ـت
Thaa	ث	ثـ	ـثـ	ـث
Jiim	ج	جـ	ـجـ	ـج
Haa	ح	حـ	ـحـ	ـح
Khaa	خ	خـ	ـخـ	ـخ
Daal	د			ـد
Dhall	ذ			ـذ
Raa	ر			ـر
Zaay	ز			ـز
Siin	س	سـ	ـسـ	ـس
Shiin	ش	شـ	ـشـ	ـش
Saad	ص	صـ	ـصـ	ـص
Daad	ض	ضـ	ـضـ	ـض
Taa	ط	طـ	ـطـ	ـط
Dhaa	ظ	ظـ	ـظـ	ـظ
Ayn	ع	عـ	ـعـ	ـع
Ghayn	غ	غـ	ـغـ	ـغ
Faa	ف	فـ	ـفـ	ـف
Qaaf	ق	قـ	ـقـ	ـق
Kaaf	ك	كـ	ـكـ	ـك
Laam	ل	لـ	ـلـ	ـل
Miim	م	مـ	ـمـ	ـم
Nuun	ن	نـ	ـنـ	ـن
Haa	ه	هـ	ـهـ	ـه
Waaw	و			ـو
Yaa	ي	يـ	ـيـ	ـي

2. Related Work

Generally, the segmentation method can be grouped into different strategies such as curve analysis, outer contour analysis and vertical projection method … etc.

All the developed algorithms adopting curve analysis strategies segment the thinned word into elementary strokes of its skeleton. These elementary strokes are branching points, intersection points or ending point [7,8].

Algorithms of segmentation by contour analysis extract word contour at first and then find its local minima which filtered to find segmentation point depend on several criteria [9,10].

Algorithms of segmentation by vertical projection method utilize information on the thickness of the word by computed the vertical projection and contour following the words are segmented into pseudo-words which segmented into character with recognition support [11,12].

3. Segmentation Model

The key for reaching a good recognition is by getting a correct segmentation. In this work, a new segmentation algorithm is proposed depending on the structural features of the Arabic characters.

The Segmentation model is divided into: line segmentation, dots extraction, estimation baseline, thinning foreground pixels and character segmentation.

3.1 Line Segmentation

It aims to separate the whole text image into lines by finding the upper and lower bound of each line in the text. At first, the horizontal density histogram for the image is calculated. After that, the rows of the image from top to bottom are scanned to find the upper and lower bound for each line in the text. Upper bound is first row has 0 density and directly followed by row that has a density greater than 0, and any row following upper bound and having 0 density, is considered lower bound for this line of text. This scan is repeated until reaching the end of the text image. Fig .1 and Fig .2 show the result of line segmentation.

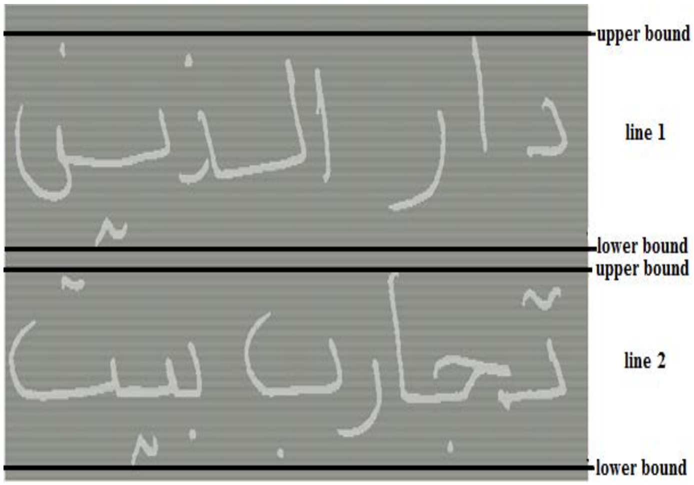


Fig .1. Line segmentation method

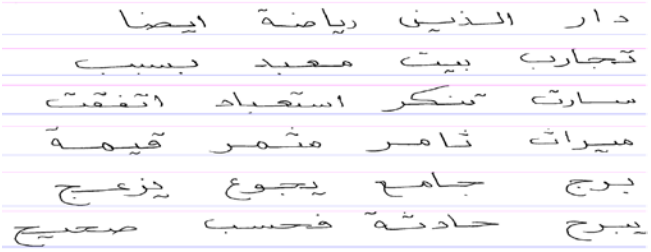
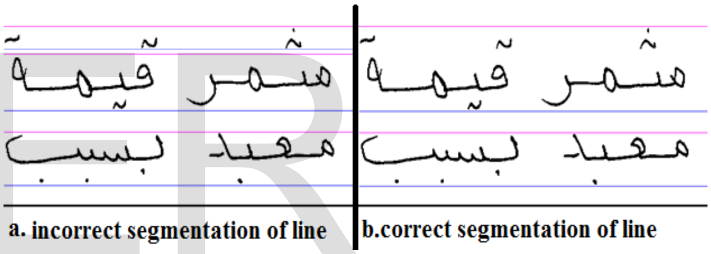


Fig .2. Line segmentation for paragraph

A problem appears in this process, where some dots are determined in a separate line segment. To solve this problem, the following steps are performed after extracting all lines in the text image:

- Find the maximum height of line in the image.
- Any height of line in the image smaller than half of maximum height of line, then it is considered as a sub-line.
- If the sub-line is found, then merge it with the closest line to it.

Fig .3 illustrates this problem and its solving.



a. incorrect segmentation of line

b.correct segmentation of line

Fig .3. Correction of line segmentation: (a) Before correction , b. After correction steps

3.2 Estimate Baseline

Arabic script (hand written or printed) is cursive and letters connected to each other with an imaginary line called a baseline. Dots in Arabic characters can be positioned under, above or in the middle of them. An example of a word with these dots is shown in Fig.4.

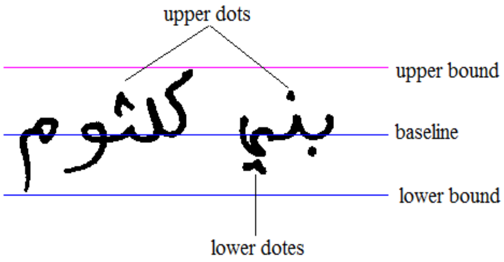


Fig .4. Baseline and dots

The baseline detection is important to determine the position of the dots in Arabic characters with respect to their baseline. The horizontal histogram is used to detect baseline. The baseline corresponds to the horizontal line with the highest density of foreground pixels through each line. Fig .5 explains that.

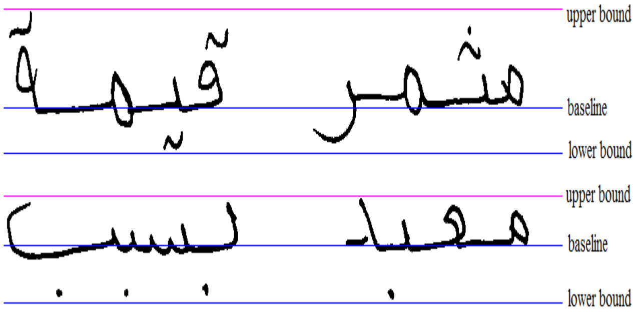


Fig .5. Estimate baseline

3.3 Extraction of Dots

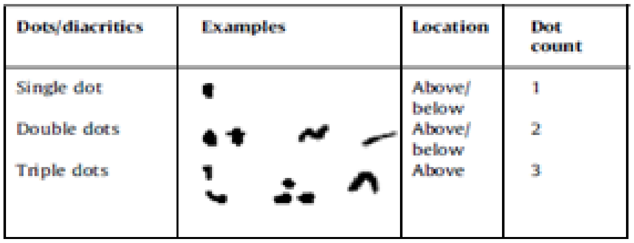
There are fifteen characters from Arabic characters that have dots ('Nuqta'). They are highlighted in table (1). Dots are an integral part of a character; many characters look similar but are distinguished from one another by dots above or below their central part. For example, the Arabic characters "Baa" (ب) and "Thaa" (ث) have the same basic shape, but "ب" has one dot below and "ث" has three dots above.

Dots can provide valuable information about the recognition of Arabic characters. In Arabic handwriting, the dots have different diacritics, as shown in Table 2.

In our research the dots are extracted to identify their number and location. Then dots will be erased to get the correct segmentation point without the effect of dots.

TABLE 2

SOME DOT'S SHAPE USED IN ARABIC HANDWRITING.



Dots/diacritics	Examples	Location	Dot count
Single dot		Above/ below	1
Double dots		Above/ below	2
Triple dots		Above	3

.

At first, the image is scanned with mask from top to bottom and from left to right, this mask is smaller than double and triples dot and any primary components of the character (main body of the character), and it is greater than a single dot. In scanning process, each mask of the image is checked to find if there is a foreground pixel isolated from background pixels, and their number of foreground pixels exceeds a threshold limit of a single dot. If these two conditions are achieved, then this mask is considered containing a single dot. Save x and y coordinate of its centre. Then the dot is erased by converting its pixels to background color.

After extracting all single dots from the text image, the search for double and triple dots will start. The text image is scanned with mask from top to bottom and from left to right, this mask is smaller than other primary components of the character, and is greater than double and triple. In scanning process, each mask of the text image is checked if it contains foreground pixels isolated from background pixels then this mask is considered containing a dot. If the width of the dot is greater than the height of the dot, then the dot is considered a double dot, otherwise it is treated as triple dot. The x and y coordinate of dot's centre are found and save, after that, the dot is erased by converting its pixels to the background color. The steps for dot extraction are illustrated in fig.6.

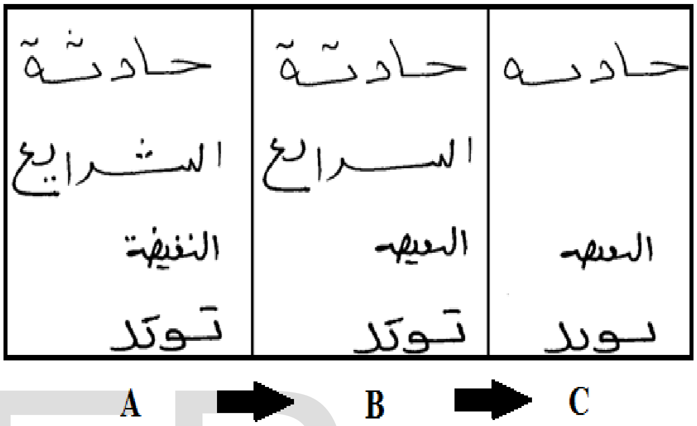


Fig .6. (a) Arabic word, (b) single dots extracted, (c) double and triple dots extracted.

3.4 Thinning

Thinning process simplifies the text shape and reduces the amount of data that need to be handled and obtain the connected character skeleton of the input image. The thinning algorithm Zhang-Suen [13] was implemented here. The thickness of any line in the text after thinning is one pixel wide.

Zhang-Suen thinning algorithm preserves the connectedness of the characters and keeps curves, arcs and isolated points unchanged. The dots are extracted before thinning step because many of dots information is lost during thinning process.

3.5 Character Segmentation

Character segmentation is an operation that seeks to decompose an image of a sequence of characters into an image of individual symbols. A segmentation process is applied to thinned words which pass by segmentation of line and extraction of dots as shown in fig .7.

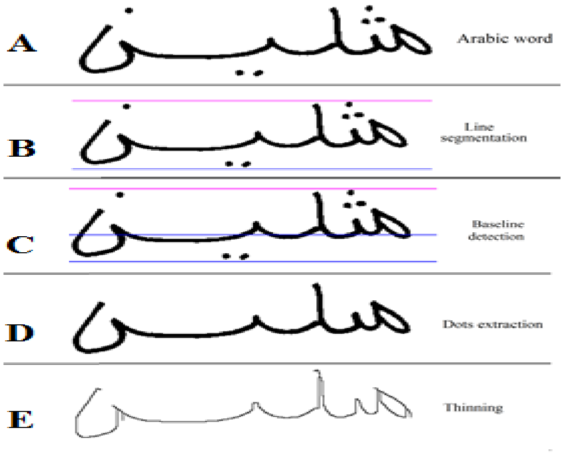


Fig . 7. Stages before the character segmentation.

The idea in character segmentation depends on finding starting and ending words/sub-words (called Parts of Arabic Words—PAWs), as shown in fig .8, and finding segmentation points among characters in these words/sub-words. From these points, each character could be separated.

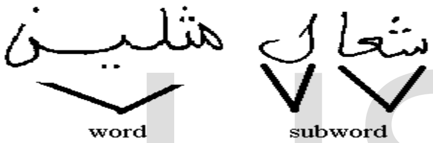


Fig .8. Word/subword

In our research, the vertical projection (also called the vertical histogram) to each line in image is used to find these point (starting, ending and segmentation points), as shown in fig .9.

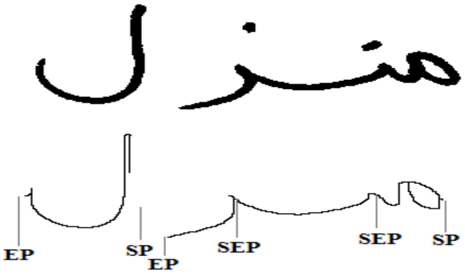


Fig .9. (SP) Starting point, (SEP) segmentation point, (EP) end point.

First, scanning histogram of each line is from left to right to find:

- Starting point (SP) by tracking tipping point from the background pixel area to foreground pixel area.
- End point (EP) by tracking tipping from the foreground pixel area point to background pixel area.
- Probable segmentation point by tracking tipping point from the least density pixel area to more density area pixel. The difference in intensity depends on the parameter let it be called difference of density parameter. These points are shown in fig .10.

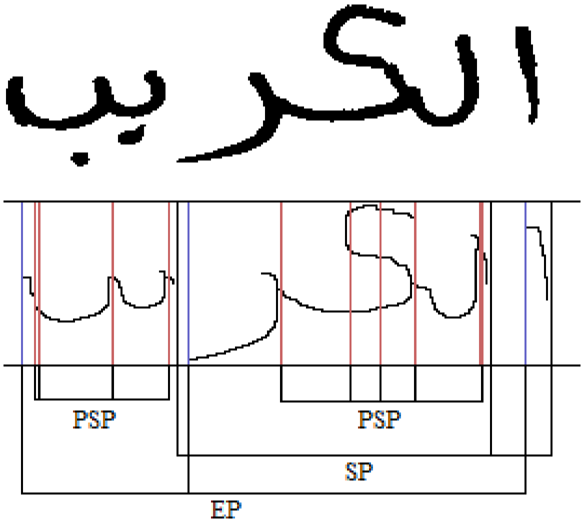


Fig .10. (SP) Starting point, (PSP) probable segmentation point, (EP) end point.

The probable segmentation points (PSP) are filtered to get segmentation point. This filtering process done by scanning the probable segmentation point in line of text from left to right, and the following criteria is checked:

1. The second probable segmentation point from right in every pair of probable segmentation points is cancelled if the pair is close to each other by ten pixels.
2. The segmentation line is the column that cutting the segmentation point. Segmentation line must be cutting only one foreground pixel as shown in fig.11. If that did not materialize, then a number of processing steps should be done on foreground pixels which cut by segmentation line:

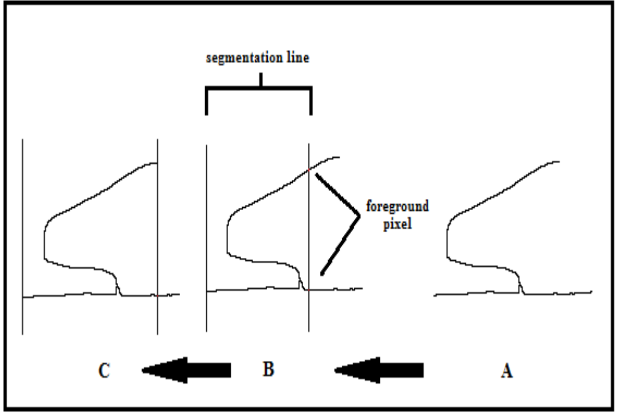


Fig .11. (A): Arabic character “Kafe”, (B): the segmentation line cut two foreground Pixel, (C): the segmentation line cut one foreground pixel.

a) First, track the first foreground pixel closer to the upper bound of the line to test this segment of character to check if it is the one closer to this segmentation line which is greater than the previous segmentation line from left, consider it extra part and erase it. As shown in fig .12(B→C).

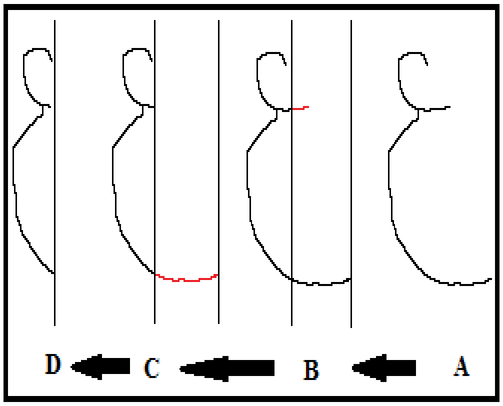


Fig .12. (A): Arabic character “Ayn”, (B): red segment in character is extra part near to upper bound of the line (C): erased the extra part, (D) red segment in character is extra part near to lower bound is erased

b) Second, any segment of character which is not connected with the skeleton of character near the baseline, it is also erased. After all that if the segmentation line is still cutting more than one foreground pixel, that means this segmentation line is cutting a circle or a curve so, the segmentation line is pulled to left until reaching a cutting position at most one foreground pixel as shown in fig .13.

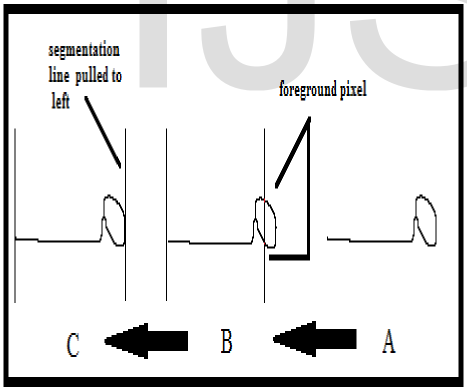


Fig .13. (A): Arabic character” Miim”, (B): the segmentation line is cutting more than one foreground pixel (C): the segmentation line is pulled to left

Finally, the column of the line that cut the end point is end line. The pair of end lines with the left segmentation line is compared with previous pair which lies on left side, if the pair is found smaller than previous pair by 40% then sums these pairs together by cancelling one of segmentation line as shown in fig .14.

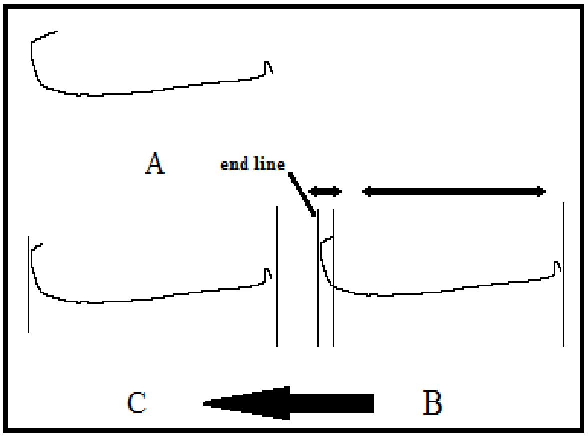


Fig .14. (A): Arabic character Baa, (B): The pair of end lines with the left segmentation line is smaller than previous pair by 40%, (C): the segmentation line is cancelled

More illustration in fig (15) explains this process. After this processing, every segmentation line cutting only one foreground pixel represents a segmentation point in the skeleton of word /sub-word, also, starting and end points are detected.

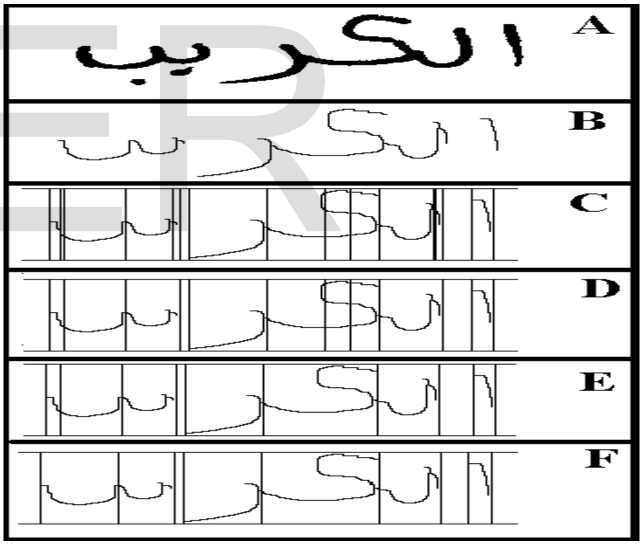


Fig .15. (A) Arabic word الكريب, (B) the skeleton of word, (C, D, E, F) character segmentation stages.

From left to right, each pair of these points can bounded as follows:

- Pair of starting point with end point is bounded as *isolated character.*
- Pair of starting point with segmentation point is bounded as *initial character*.
- Pair of two segmentation points is bounded as *medial character.*
- Pair of segmentation point with end point is bounded as *final character.*

These characters shown in fig.16.

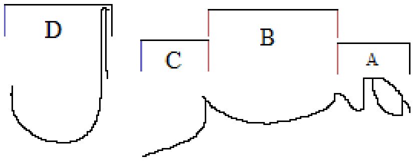


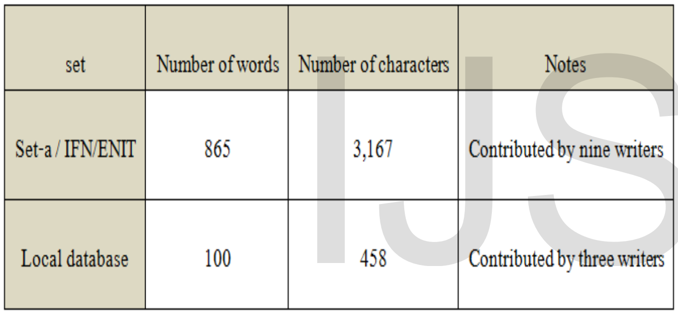
Fig .16.(A) Initial character, (B) medial character, (C) final character,(D) Isolated character.

4. Database and Experimental Result

To test our model, we applied it on two types of database; our local database, and set-a from INFINT database (INFINT database) [14]. Local database contains different texts covering all Arabic characters from three writers without movements and multi-lines. While set-a from IFN/ENIT database contains samples with single line, the characteristic of IFN/ENIT database is containing many bad samples sometimes cannot be read by human. More detail about database illustrated in table 3.

TABLE 3

THE NUMBER OF WORDS AND CHARACTERS IN THE SETS



set	Number of words	Number of characters	Notes
Set-a / IFN/ENIT	865	3,167	Contributed by nine writers
Local database	100	458	Contributed by three writers

Results of segmentation in Set-a of IFN/ENIT database and Local database were 84.3% and 90.8% respectively. Despite the fact that a high percentage of correct segmentation is done in using this new algorithm, there are problems occurring with this segmentation algorithm including misplaced segmentation, over segmentation, or under segmentation. Most errors are back to the shape of the character especially the characters that have a tipping point from the least density of?? more density foreground pixels (س، ش، ص، ض).

5. Conclusions

In this paper, we present our research results on off-line Arabic handwriting text segmentation depending on structural characteristic of Arabic character and introduce several new ideas and techniques for segmentation of Arabic hand- writing text.

An Arabic text lines is segmented to words/sub-words. The dots' number and location are extracted and dots are erased from the lines. At last the words/sub-words are thinned and segmented into characters using new segmentation algorithm which based on the characteristics of Arabic writing.

References

[1] Richard Szeliski, Computer Vision Algorithms and Applications, first edition, springer, 2011.

[2] Liana M. L. and Venu G., Offline Arabic Handwriting Recognition: A Survey, Transaction on Pattern Analysis and Machine Intelligence, IEEE, Vol. 28, No. 5, May 2006.

[3] Abdallah B., Abdellatif E. and Mokhtar S., HMMs with Explicit State Duration Applied to Handwritten Arabic Word Recognition, The 18th International Conference on Pattern Recognition, IEEE, 2006.

[4] Rejean P. and Sargur N., On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey, Transaction on Pattern Analysis and Machine Intelligence, IEEE, Vol. 22, No. 1, January 2000.

[5] Yusuf P., Recurrent Neural Network Method in Arabic Words Recognition System, International Journal of Computer Science and Telecommunications, Vol. 3, Issue 11, November 2012.

[6] Firoj P., The State of the Art Recognize in Arabic Script through Combination of Online and Offline, International Journal of Computer Science and telecommunications, Vol. 4, Issue 3, March 2013.

[7] Alimi, A.M, An evolutionary neuro-fuzzy approach to recognize on-line Arabic handwriting, Fourth International Conference on Document Analysis and Recognition, IEEE, 1997.

[9] Olivier G., Miled H., Romeo K., and Lecourtier Y., Segmentation and coding of Arabic handwritten words, 13th International Conference on Pattern Recognition, IEEE, Vol. 3, 1996.

[8] Safwan Wshah, Zhixin Shi and Venu Govindaraju, Segmentation of Arabic Handwriting based on both Contour and Skeleton Segmentation, ICDAR, IEEE, 2009.

[10] Osman, Y., Segmentation algorithm for Arabic handwritten text based on contour analysis, International Conference on Computing, Electrical and Electronics Engineering (ICCEEE), IEEE, 2013.

[11] Youssef Es Saady, Ali Rachidi, Mostafa El Yassa and Driss Mammass, Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character, International Journal of Advanced Science and Technology, Vol. 33, August, 2011.

[12] Said Elaiwat and Marwan AL-abed Abu-Zanona, A Three Stages Segmentation Model for a Higher Accurate off-line Arabic Handwriting Recognition, WCSIT 2 Vol. 3, 2012.

[13] T. Y. Zhang and C. Y. Suen, A Fast Parallel Algorithm for Thinning Digital Patterns", Image Processing and Computer Vision, Vol. 27, No. 3, March 1984.

[14] IfN/ENIT-database of hand written Arabic words, in: 7th Colloque International Francophone url'Ecritetle Document, CIFED2002, October21–23, 2002.