A New Character Segmentation Approach for Printed Arabic Text

El Mamoun Mamouni [1], Yamina Ouled Jaafri [1] and Kaddour Sadouni [2]

✉ mamouni.elm@gmail.com

[1] LDDI Laboratory, Mathematics and Computer Science Department, Faculty of Sciences and Technology University of Adrar, Algeria

[2] SIMPA Laboratory, Computer Science Department University of Science and Technology Med Boudiaf of Oran Algeria

Received: 18 November 2020
Accepted: 02 December 2020
Published online: 30 December 2020

Abstract

Optical character recognition has been the subject of considerable research activity for many years, segmentation is a very important step in the process of character recognition, and has a major impact on the quality of the recognition system. An error produced in this step can be very serious, since it affects the performance of the recognition systems carrying the character to be either misrecognized or in most cases rejected completely. In this paper we propose a segmentation approach for printed Arabic text for different segmentation stages; line segmentation, subword segmentation and character segmentation based on projection profile technique and pen thickness estimation, interesting results have been obtained.

Keywords Segmentation, Printed Arabic text, Projection profile, Pen thickness.

1. Introduction

The human being can easily segment the Arabic word into characters; however, it is not easy to segment it directly into perfect characters by the computer, the ligature increases the difficulty of segmentation, which does not allow applying the algorithms developed for other scripts to the Arabic script.

Some of the reasons may be taken to the lack of research comparing to Latin, Chinese and other languages, but, at the same time, the nature of the Arabic script even in its printed form poses real challenges to researchers [1] [2].

A text segmentation system must go through three main steps which are the segmentation of text into lines, then lines into words, and finely words into characters, each of these steps alone represents a challenge for researchers. For example, we cite the works in [3] [4] to line segmentation, [5] [6] to words segmentation and in [7][8] to character segmentation.

The development of new segmentation approach is one of our objectives for contribution in the field of recognition of printed Arabic script. The proposed approach is based on the projection profile and pen thickness to extract segmentation points.

This paper is organized as follows. In Section 2, we discuss the characteristics of Arabic text. Section 3 describes our proposed approach while section 4 briefly presents our database. In section 5 we discuss the results obtained. Finally, Section 6 concludes the paper including some future work.

2. Characteristics of the Arabic Writing

Arabic writing is the basis of many other languages such as Urdu, Persian (Farsi), Sorani and Luri dialects of Kurdish, Jawi, Pashto and Sindhi [9]. The Arabic handwritten or printed has different characteristics that differentiate it from other languages, which makes the recognition process of Arabic languages very difficult. In this section, we present some of the characteristics of the Arabic writing [10]:

- Arabic language consists of 28 characters.
- Cursive: Unlike the Latin language, Arabic language script is cursive within the same word; this connection can be interrupted in the middle of the word at few certain characters.
- The Arabic characters are written cursively from right to left.
- Dots: Several characters may have the same body but a number and / or a position of different dot. Dotting is a significant source of confusion in AOCR (Arabic Optical Character Recognition) systems, especially noisy scanned documents. There are 15 characters in the language have dots; 10 with one dot, 3 with two dots and 2 with three dots see Table 1.

Table 1 Number of dots

Number of dots	Character name
1	ن, ف, غ, ظ, ض,ز , ذ,خ, ج ,ب
2	ي, ق,ت
3	ش,ث

- Multiple grapheme cases: As mentioned before, Arabic language script is cursive (connected). This characteristic causes the characters to be context sensitive to changing its form and have

multiple variants according to its position (Start, Middle, End, Isolated). All Arabic characters and their shapes variation are shown in Table 2.

- Size variation: Arabic characters are not like English characters in its structured size. In English, a single characters has fixed width and height characters, while in Arabic single characters the width and height are variables.

Table 2 Arabic language primitives

Character name		Arabic Language Primitives			
		Isolated	Connected		
			End	Middle	Start
Alif	ألف	أ	ـا	-	-
Baa	باء	ب	ـب	ـبـ	بـ
Taa	تاء	ت	ـت	ـتـ	تـ
Thaa	ثاء	ث	ـث	ـثـ	ثـ
Jeem	جيم	ج	ـج	ـجـ	جـ
Haa	حاء	ح	ـح	ـحـ	حـ
Khaa	خاء	خ	ـخ	ـخـ	خـ
Daal	دال	د	ـد	-	-
Thaal	ذال	ذ	ـذ	-	-
Raa	راء	ر	ـر	-	-
Zaay	زاي	ز	ـز	-	-
Seen	سين	س	ـس	ـسـ	سـ
Sheen	شين	ش	ـش	ـشـ	شـ
Saad	صاد	ص	ـص	ـصـ	صـ
Dhaad	ضاد	ض	ـض	ـضـ	ضـ
Ttaa	طاء	ط	ـط	ـطـ	طـ
Dthaa	ظاء	ظ	ـظ	ـظـ	ظـ
Ain	عين	ع	ـع	ـعـ	عـ
Ghen	غين	غ	ـغ	ـغـ	غـ
Faa	فاء	ف	ـف	ـفـ	فـ
Qaf	قاف	ق	ـق	ـقـ	قـ
Kaf	كاف	ك	ـك	ـكـ	كـ
Lam	لام	لا	ـل	ـلـ	لـ
Mem	ميم	م	ـم	ـمـ	مـ
Noon	نون	ن	ـن	ـنـ	نـ
Haa	هاء	ه	ـه	ـهـ	هـ
Wow	واو	و	ـو	-	-
Yaa	ياء	ي	ـي	ـيـ	يـ

- Subword(s): An Arabic word is made by connecting some characters together. However, six characters (د , أ ,و , ز , ر , ذ) cannot be attached to their successor. Therefore, if one or more of these characters exist in a word, the word is divided into two or more subwords, as shown in Fig. 1.

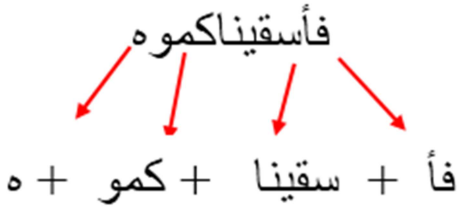


Fig. 1 Subwords example

- Overlapping: The characters in the same word may overlap vertically without touching, as shown in Fig. 2.

Fig. 2 An example of overlapped Arabic words

3. Proposed Approach

Our proposed system of segmentation is based on the projection profile, and it consists of three levels: lines segmentation, subwords segmentation and characters segmentation. The horizontal projection is used for line segmentation and vertical projection is used for subword segmentation and characters segmentation.

The aim of the projection method is to simplify drastically our system of segmentation by reducing two-dimensional information into one dimension; this method is based on the fact that the connection stroke is always of less thickness than other parts of the words.

The horizontal projection is defined as:

$$h(i) = \sum_{i} P(i,j) \tag{1}$$

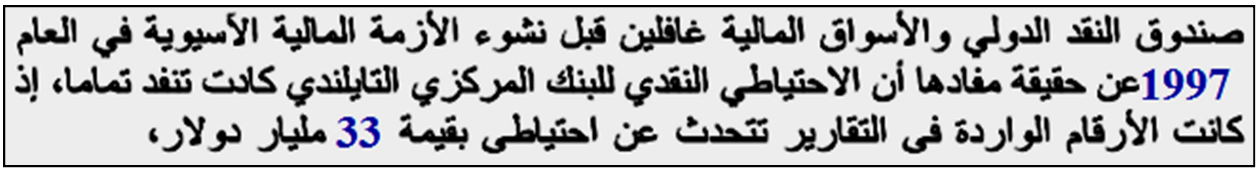
And vertical projection is defined as:

$$h(j) = \sum_{j} P(i,j) \tag{2}$$

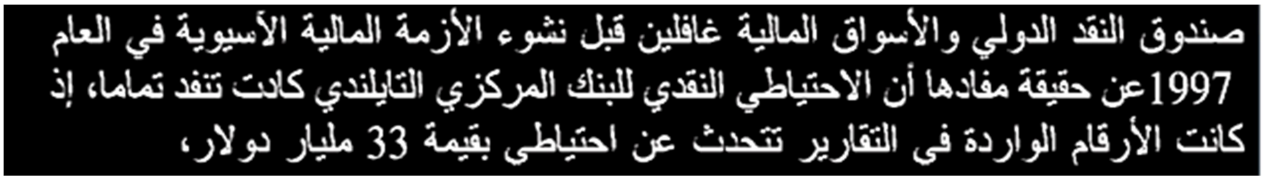
Where $P(i,j)$ is the pixel value which is either zero (white or background) or one (black), i,j refer to rows and columns respectively.

3.1. Binarization

The first step is the image thresholding that converts the greyscale image to binary format, as shown in Fig. 3.



(a)



(b)

Fig. 3 (a) Original image (b) image after Binarization

3.2. Text to line segmentation

In this step, horizontal projection is used to extract lines; the image is scanned row by row to determine the margins with no data (fully black continuous space). The algorithm is presented in Table 3.

Table 3 Lines segmentation algorithm

Step 0	If the current row index i is smaller than the max row's index
	Build up the horizontal projection of this row.
	If its value equals 0
	Go to step 1
	Else
	Go to step 2
	End IF

Step 1	Cut the corresponding row.
Step 2	Go to the next raw. Go to step 0.

The existence of dots in Arabic writing can cause over segmentation problem as shown in the figure 4.

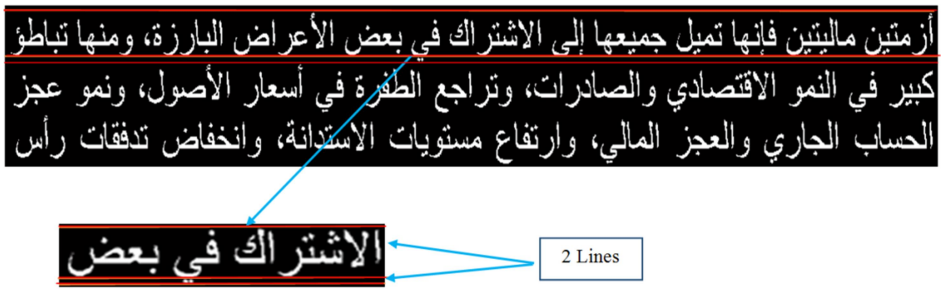


Fig. 4 Over segmentation problem

To handle this issue, first, we compute the pen thickness which is the pin size used for writing, the pen thickness can handle by taking the most frequent value in the vertical projection, figure 5 shows an example.

We notice that the space between the first cut line and the dot (Dot location) is equal to:

$$\text{Dot location} = \text{Pen thickness} / 2 \tag{3}$$

So before cutting, we must verifier space between the first cut and dot location, to ensure that the cut location is not above the dots.

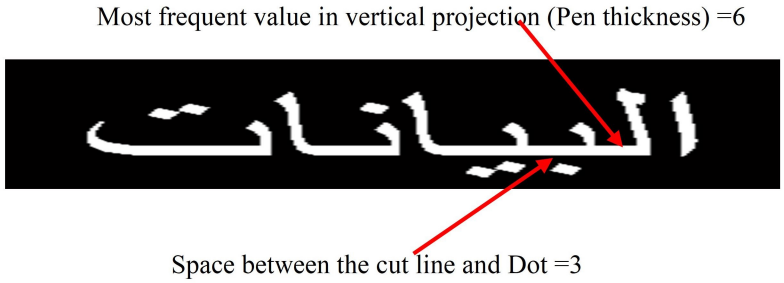


Fig. 5 The relation between the pen thickness and the cut place

3.3. Line to words and subwords segmentation

To facilitate this stage and character segmentation stages we extracted the main body of the line without dotting (Dot elimination) as shown in Fig. 6. And then, the vertical projection profile is performed to each line in order to divide it into subwords. The algorithm is shown in Table 4.

(a) Original line (main bodies + dotting).

(b) Dotting of line.

(c) Main body of line.

Fig. 6 Dot Elimination.

Table 4 Word/Subword segmentation algorithm

Step 0	If the current column index is smaller than the max columns index
	Build up the vertical projection of this column.
	If its value equals 0
	Go to step1
	Else
	Go to step 2
	End IF

Step 1	Cut the corresponding column.
Step 2	Go to the next column. Go to step 0.

The challenge in this step is how to know that the text between two gaps is a word or a subword, and if it's a subword to which word belongs, Moreover, the gap between two consecutive words or subwords is not fixed and depends on the font type and size.

To handle this issue, the pen thickness is compared with the separation space. Thus, if the separation space between two consecutive words/subwords is larger than the mean of the pen thickness of these two consecutive words/subwords, then the separation region performs a separation between two different words else the separation region is between two subwords in the same word, see Fig. 7.

Fig. 7 Space between two words/subwords

3.4. Character segmentation

The proposed algorithm for character segmentation is based on the vertical projection; we summarize it in Table 5.

Table 5 Character segmentation algorithm.

Step0	Read the words/subwords binary image (the input) go to the next step
Step 1	Calculate the vertical projection vector v[i] go to step 2
Step 2	Searching in the vertical projection vector v[i] of values equal to the pen thickness as follow : S=0 For I =1:Length(V) If V[i] == Pen_thickness S=S+1 End End If S = 0 the input subword is one character (ة; ج ; 9 ;0 ; أ ;م...) go to step 6 Else go to the next step
Step 3	Searching all over the vertical projection vector v[i] for sequence of values equal to the given Pen_thickness and this sequence mast contains more than pen thickness over two values (Pen_thickness /2 values). If no sequence is found, the input subword is also detected as one character (ﻯ; ﻯ...). Else go to Step 4.
Step 4	The middle of each sequence is considered as initial segmentation point, these last ones are saved in N[i].
Step 5	For i =2: Number of segmentation points If the length between segmentation point N[i-1] and N[i] is < 12 Ignore the segmentation point N[i]. End End
Step 6 Output	Segmented Characters

4. Text Database

In January 2017, we have created a database that contains different resources of texts, these resources include digital magazines and papers taken from the website of Aljazeera.net, Ar.wikipedia.org and others, the most of these texts speak in the fields of politics and economics. 35 paragraphs of 2 to 6 lines were taken randomly, for a total of 139 lines. These paragraphs have been converted to images with 300

dpi resolution, generated using both Times New Roman and Arial font types and sizes 14-16. Samples from this database are shown in Fig. 8.

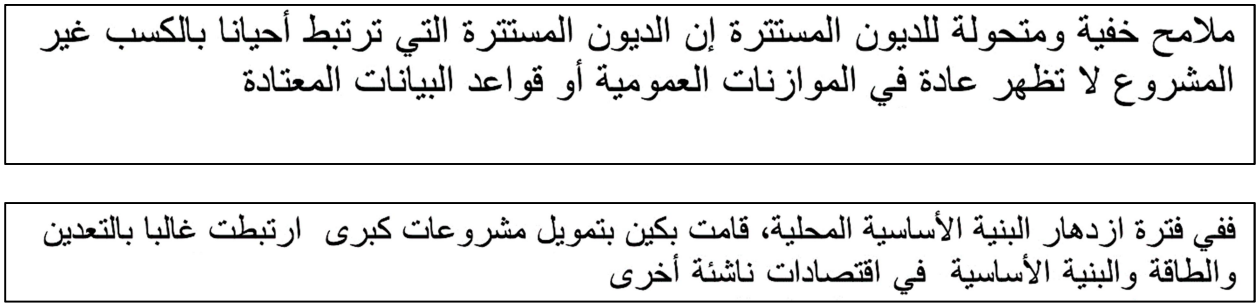


Fig. 8 Samples of text database

5. Experimental Results

5.1. Results of line segmentation

Table 7 Line segmentation results for regular style with different font and sizes.

Font Name	Total number of lines	number of lines correctly segmented	Accuracy
Times New Roman	71	71	100%
Arial	68	68	100%
TOTALS	139	139	100%

As it appears in the Table 7, the results of line segmentation is perfect since we achieved the accuracy 100%, this is due to the elimination of over-segmentation problem, where the second component (dots) was considered as an independent line; which causes two lines, the first one contain a sequence of words some of them contain letters without dots, and the second line contains only the lost dots, instead of one line gathering the both, we have remedied this problem by calculating the distance between the previous cut and the point as explained previously.

5.2. Results of subword segmentation

In this phase, we have chosen to apply the segmentation over subword instead of the word to handle less possible number of connected letters in character segmentation.

We got good results (Table 8) with an accuracy of 97.97%, while the accuracy of 2.03% of subwords is not correctly segmented, and that due to one main reason which is the overlap between two subwords as shown in the Fig. 9 where the value of vertical projection between the

two subwords is different from 0, which violates the condition of subword segmentation. We observed that this issue appear mostly when the letter " أ " is involved and in the middle of the word followed by certain characters such as "ت", " د","صـ".

Table 8 Subword segmentation results for regular style with different font and sizes

Font Name	Total number of subword	number of subword correctly segmented	Accuracy
Times New Roman	2301	2269	98.61%
Arial	2192	2133	97.31%
TOTALS	4493	4402	97.97%

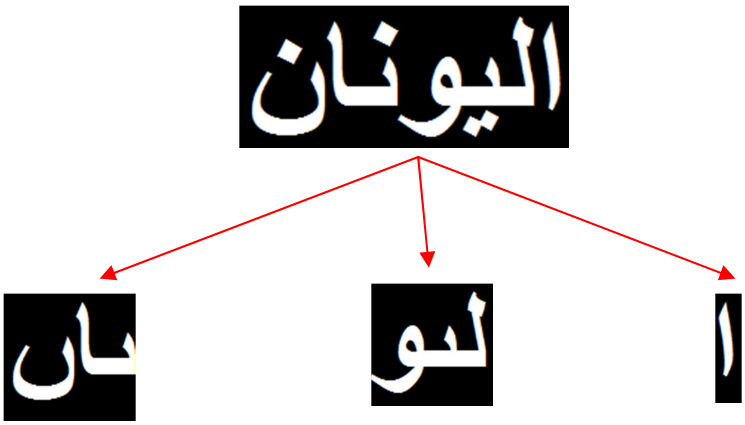


Fig. 9 Example of under Segmentation problem in subword Segmentation

5.3. Results of character segmentation

This level is considered as the most difficult level in the entire segmentation process due to the cusivity of Arabic writing.

Table 9 Character segmentation results for regular style with different font and sizes

Font Name	Total number of Character	Number of Character correctly segmented	Under segmentation	Over Segmentation	Accuracy
Times New Roman	3927	3653	4.7%	2.28%	93.02%
Arial	3651	3393	4.46%	2.61%	92.93%
TOTALS	7623	7046	4.58%	2.44 %	92.97%

The study shows that the results of the proposed algorithm of segmentation characters (Table 9) are good, where we achieve an accuracy of 92.97% for both Times New Roman and Arial font. On the other hand, the segmentation errors appear with a precision of 7.02%, the problem is divided in two main problems, under and over segmentation:

The under segmentation appears with an accuracy equal to 4.58%, the main reason of this issue back to overlap problem, where the shape of letters interfere with each other vertically, which makes the separate point over the junction line hard to defined (see Fig. 10).

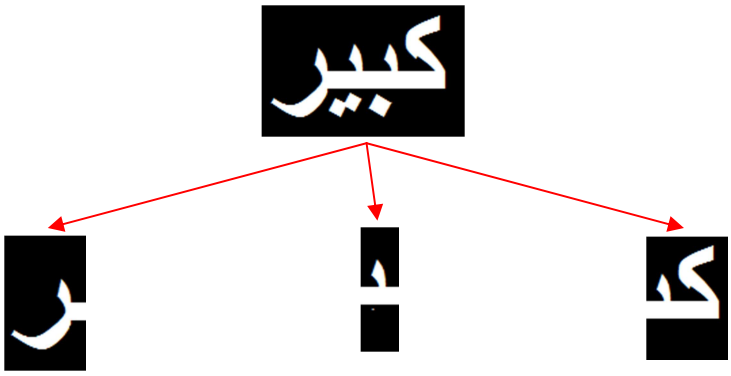


Fig. 10 Example of under segmentation problem in character segmentation

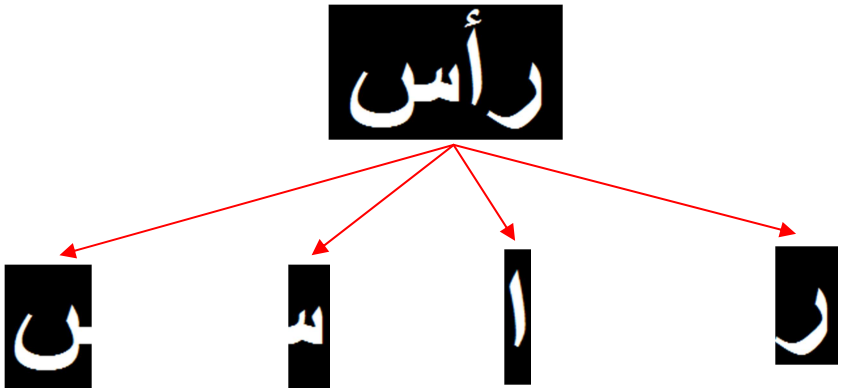


Fig. 11 Example of Over Segmentation problem in Character Segmentation

The under segmentation problem appears the most when the character "ـكـ " is involved in both start and the middle followed by certain characters such as" ـتـ ", " ـسـ "," ـبـ ".Furthermore, the other reason of the under segmentation issue back to the miss segmented subword in the previous level.

The over segmentation problem appears with an accuracy equal to 2.44 %, the main reason of this issue back to the shape of characters itself, where the character shape appears in a certain way that looks like it contains a junction line that prompts us to say that there are two parts in the same letter, as presented in figure 11 for the letter "س".

6. Conclusion and Future Work

In this paper, a new segmentation approach based on projection profile is proposed. This approach consists of three levels: lines segmentation, subwords segmentation and characters segmentation.

For line segmentation stage the over segmentation problems are addressed and solved, the subword segmentation is performed by applying vertical projection, the proposed method also determines if the subword are related to the same word or to different words regardless to the font type or size by estimating the pen thickness for each line. However the character segmentation algorithm has introduced some problems, since it does not differentiate between the junctions that occur between letters and some other junctions that occur in the main body of the letter.

Results using the proposed segmentation method show good results, the recognition rate achieve 100%, 97.97%, 96.23% for line, subwords and character segmentation respectively.

As a future work, we suggest concentrating on the phase of character segmentation only and solving the problem of overlapping and under segmentation between Characters.

References

[1] Yamina, Ouled Jaafri, Mamouni El Mamoun, and Sadouni Kaddour. "Printed Arabic optical character recognition using support vector machine." In 2017 International Conference on Mathematics and Information Technology (ICMIT), pp. 134-140. IEEE, 2017. https://doi.org/10.1109/mathit.2017.8259707

[2] Saabni, Raid, "Efficient recognition of machine printed Arabic text using partial segmentation and Hausdorff distance", International Conference of Soft Computing and Pattern Recognition (2014): 284-289. https://doi.org/10.1109/socpar.2014.7008020

[3] Mohammad, Khader, Aziz Qaroush, Mahdi Washha, Sos Agaian, and Iyad Tumar. "An adaptive text-line extraction algorithm for printed Arabic documents with diacritics." Multimedia Tools and Applications (2020): 1-28. https://doi.org/10.1007/s11042-020-09737-1

[4] Ayesh, Muna, Khader Mohammad, Aziz Qaroush, Sos Agaian, and Mahdi Washha. "A Robust Line Segmentation Algorithm for Arabic Printed Text with Diacritics." Electronic Imaging 2017, no. 13 (2017): 42-47. https://doi.org/10.2352/issn.2470-1173.2017.13.ipas-204

[5] Anwar, Khaerul, and Hertog Nugroho. "A segmentation scheme of arabic words with harakat." In 2015 IEEE International Conference on Communication, Networks and Satellite (COMNESTAT), pp. 111-114. IEEE, 2015. https://doi.org/10.1109/comnetsat.2015.7434299

[6] Qaroush, Aziz, Bassam Jaber, Khader Mohammad, Mahdi Washaha, Eman Maali, and Nibal Nayef. "An efficient, font independent word and character segmentation algorithm for printed Arabic text." Journal of King Saud University-Computer and Information Sciences (2019). https://doi.org/10.1016/j.jksuci.2019.08.013

[7] Mohammad, Khader, Aziz Qaroush, Muna Ayesh, Mahdi Washha, Ahmad Alsadeh, and Sos Agaian. "Contour-based character segmentation for printed Arabic text with diacritics." Journal of Electronic Imaging 28, no. 4 (2019): 043030. https://doi.org/10.1117/1.jei.28.4.043030

[8] Firdaus, Fakhry Ikhsan, Achmad Khumaini, and Fitri Utaminingrum. "Arabic letter segmentation using modified connected component labeling." In 2017 International Conference on Sustainable Information Engineering and Technology (SIET), pp. 392-397. IEEE, 2017. https://doi.org/10.1109/siet.2017.8304170

[9] Naz, Saeeda, Khizar Hayat, Muhammad Imran Razzak, Muhammad Waqas Anwar, Sajjad A. Madani, and Samee U. Khan. "The optical character recognition of Urdu-like cursive scripts." Pattern Recognition 47, no. 3 (2014): 1229-1248. https://doi.org/10.1016/j.patcog.2013.09.037

[10] Odeh, Ammar, Khaled Elleithy, and Miad Faezipour. "Steganography in Arabic text using Kashida variation algorithm (KVA)." In 2013 IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1-6. IEEE, 2013.