RESEARCH ARTICLE

Kurdish Optical Character Recognition

Rasty Yaseen and Hossein Hassani*

Department of Computer Science and Engineering, University of Kurdistan Hewlêr, Erbil, Kurdistan Region – F.R. Iraq

***Corresponding author's email:** hosseinh@ukh.edu.krd

Received: 13 March 2018 **Accepted:** 17 May 2018 **Available online:** 30 June 2018

ABSTRACT

Currently, no offline tool is available for Optical Character Recognition (OCR) in Kurdish. Kurdish is spoken in different dialects and uses several scripts for writing. The Persian/Arabic script is widely used among these dialects. The Persian/Arabic script is written from Right to Left (RTL), it is cursive, and it uses unique diacritics. These features, particularly the last two, affect the segmentation stage in developing a Kurdish OCR. In this article, we introduce an enhanced character segmentation based method which addresses the mentioned characteristics. We applied the method to text-only images and tested the Kurdish OCR using documents of different fonts, font sizes, and image resolutions. The results of the experiments showed that the accuracy rate of character recognition of the proposed method was 90.82% on average.

KEYWORDS: Optical Character Recognition, Character Segmentation, Upper Contour Labeling, Kurdish OCR, Kurdish NLP

1. INTRODUCTION

In this article, we introduce a method to perform the segmentation and recognition process of Kurdish texts written in the Persian/Arabic script. Kurdish is spoken by a population of about 30 million (Hassani, 2017; Hassani & Medjedovic, 2016) in different countries. Kurdish is a multi-dialect language and is written using four different scripts (Hassani, 2017; Hassani & Medjedovic, 2016). Using Persian/Arabic script has a long history and is widely used in different Kurdish dialects (Hassani, 2018; Hassanpour, 1992). To the best of our knowledge, at the time of the writing of this article, there is no offline Optical Character Recognition (OCR) tool for Kurdish. That is, the subject seems to be less explored with a focus on very special cases such as feature extraction (Mohammed, 2013). However, research on Kurdish OCR could benefit from OCR studies in other languages such as Arabic, Urdu, and Farsi (Persian) with similar characteristics and features in their scripts. For example, Arabic OCR systems have a quite long history and their accuracy rates are reported to be over 90% (Cheung, Bennamoun, & Bergmann, 2001), (Zheng, Hassin, & Tang, 2004). Furthermore, work on Farsi showed promising results (Azmi & Kabir, 2001).

The rest of this article is organized as follows. Section 2 provides a review of the related work. Section 3 discusses the suggested method for character segmentation and recognition of Kurdish text written in the Persian/Arabic script and how it is applied. It also addresses the evaluation method for the suggested character recognition approach. Section 4 provides the results of the experiments. Section 5 analyzes the results. Finally, Section 6 concludes the article and provides suggestions for future work.

2. RELATED WORK

The Kurdish Persian/Arabic script has an architecture similar to the architecture used in Arabic, Urdu, Farsi, and Hebrew scripts. It is written Right to Left (RTL) and cursive. Therefore we are interested in the work that has been done in the mentioned languages. However, research on cursive English and Latin scripts are also beneficial with

Access this article online

DOI:10.25079/ukhjse.v2n1y2018.pp18-27 **E-ISSN:** 2520-7792

regard to their methods of segmentation for connected characters.

Amin (1988, 1991) was a pioneer to Arabic OCR. He presented an analytical approach where word segments are analyzed according to their context and whole representation in order to be recognized. The approach requires a large lexicon of the language. He presented a method for vertical projection of sub-words in each word. In this method, first, the text lines are detected with a horizontal projection of the black pixels (assuming the text is black) in the image. A space of zero projection larger than a specified threshold will be the sign for a new line. Then, the sub-words are found by a vertical projection of a line, and a space larger than a pre-defined threshold would mean that there is a sub-word before that space. The vertical projection of sub-words would detect possible separation points in the component by evaluating the projection value of each point and finding the small values less than the average projection value when the point is close to the sub-word baseline. This is expressed by the following formula from (Amin, 1991):

$$AV = \left(\frac{1}{NC}\right)\sum_{j=1}^{NC} X_j \qquad (1)$$

Where AV is the average projection value, NC is the number of columns in the sub-word projection, and Xj is the number of black pixels of the jth column.

Zheng et al. Zheng et al. (2004) presented a holistic approach to the Arabic OCR. In this approach, development of a large lexicon is not necessary for the OCR process. In this approach, words are segmented into characters and each character is recognized individually. This method incorporates the vertical projection suggested in Amin (1991). Also, Zheng et al. (2004) proposed a number of rules for determining whether a component is an isolated character or not. If it is found to be an isolated character, it will not be segmented. Otherwise, a character segmentation algorithm will be applied along with certain rules for determining whether the possible separation points are true segmentation points. The results of their experiment showed an average accuracy rate of 94.8% for two types of fonts with 6 different font sizes.

Cheung et al. (2001) have suggested a modification on the vertical projection for Arabic OCR. They have used the concept of feedback from the output of the classification stage to the segmentation stage in which no extra rules and conditions would be necessary to acquire an accurate character segment, because of the feedback concept. Any fragment not recognized as a character in the classification stage will be sent back to the segmentation stage until a correct character is recognized (Cheung et al., 2001). This work reports 99% accuracy rates, however, through both the provided example results (Cheung et al., 2001) it is difficult to understand how the accuracy of the method was evaluated.

Azmi and Kabir (2001) have worked on Farsi (Persian) script for which they implemented segmentation using the upper contour of sub-words. They have suggested an algorithm that is implemented by finding the global baseline of each text line first. Then, the local sub-word baseline is found using eight-directional Freeman chain coding (Azmi & Kabir, 2001). Next, conditional labeling is applied on the upper contour of sub-words based on a point's distance from the local baseline and the label of the point before it. The fragments will be labeled *u*, *m*, or *d* which stand for *up*, *middle*, and *down* respectively. After that, character segmentation is performed based on certain rules which depend on the fragments' label and the local baseline. Finally, a post-processing algorithm will eliminate over-segmentation based on the rules which depend on the pen-size. The pen-size is the most frequent column size of the text line.

As mentioned in Azmi and Kabir (2001) and Amin (1988), this method suffers from two major inadequacies. First, several Persian/Arabic characters naturally overlap the character adjacent to them, which, in some cases, is used as stylistic calligraphy. Second, "The connection between two characters is often short. Therefore, placing the segmentation points is a difficult task" (Amin, 1988). Even though Zheng et al. (2004) suggest particular rules which intend to resolve the inaccuracies caused by these two problems, these issues continue to have their impacts in the methods based on vertical histogram projection, for example in the work by Cheung et al. (2001). However, in this latter case, the difference is that no extra rules and conditions would be necessary to acquire an accurate character segment, because of the feedback concept. Any fragment not recognized as a character in the classification stage will be sent back to the segmentation stage until a correct character is recognized.

Azmi and Kabir (2001) address the issues of projection-based segmentation by working on the upper contour of the text in an image. They demonstrate their solution by comparing with two other methods, which are profile-based and histogram projection-based segmentation. It turns out that the upper contour based segmentation provides higher accuracy (Azmi & Kabir, 2001).

3. METHODOLOGY

We have based our method upon the approach by Azmi and Kabir (2001). Although this approach is more flexible in designing advanced OCR systems for ancient and/or noisy texts (Fig.1), the upper contour labeling method seems to be competent to be efficiently modified to design regular OCRs, which need to consider noises and to obtain better results through fine-tuning (Azmi & Kabir, 2001). We have proposed new rules for the segmentation process which are more compatible with Kurdish characters in terms of font styles and proportions. We compile our data by extracting texts from Wikipedia in Kurdish (Sorani). We render the extracted data in different fonts and font sizes. The details of the suggested rules are explained below.

3.1. Line Segmentation

We use an optimized horizontal histogram projection (Zheng et al., 2004) to find the text line limits by detecting zero values in the projection. That is, we separate lines of normal text images which do not have slants or overlaps. We have optimized this method for very small overlaps by calculating the median and average text line heights and comparing it to each text line. If a perceived text line has 1.8 times the height of the median text line height, then it is two overlapped text lines. The overlap is removed by splitting them in the middle to get two single text lines.

3.2. Pre-Character Segmentation

We use the contour-based segmentation method (Azmi & Kabir, 2001) for character segmentation. This method requires a number of text line features. These features are:

1. Text line boundaries; which are the limits of each text line including all the signs above and/or below the main character parts. This is already known after line segmentation.
2. Text line pen sizes; which are the most frequent black pixel column sizes of the text line. This feature is found by using an algorithm for identifying the most frequent black pixel vertical run on the text line.
3. Text line baselines; the upper and lower boundaries of the baseline will be at a distance of pen size from one another. So, we find the width of pen size of highest values in the row projection of the text line (Fig. 2).
4. The proportions of pen sizes relative to the text line boundaries. This is necessary to differentiate between font styles of different proportionality.

These features are demonstrated in Fig.3.

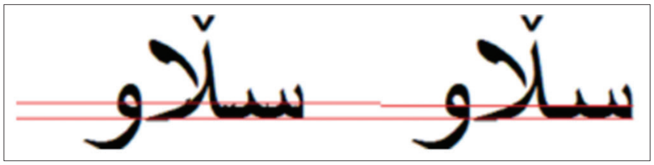


Figure 1. The word on the right side (means Hello in Kurdish) is shown with a normal baseline of pen-size width. The same word on the left is shown as noisy text with a baseline of pen-size width and 3 extra pixels to account for noise

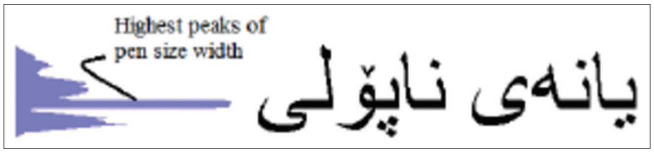


Figure 2. Baseline found with pen-size and row projections. Note that the highest peak is the baseline of a horizontally straight text line

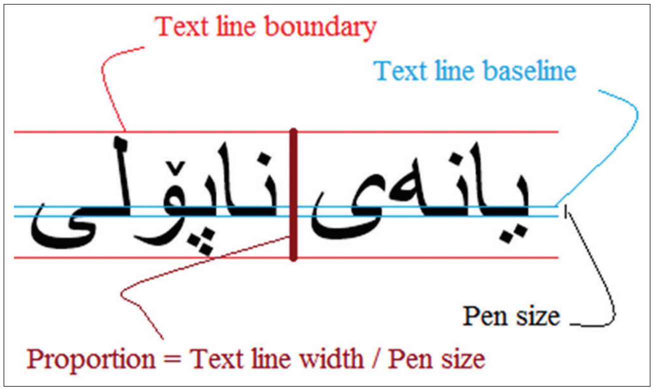


Figure 3. Text line features that are necessary for character segmentation. "Proportion" is the ratio of "Text line width" to "Pen size." "Text line width" is the positive difference between the upper "Text line boundary" and the lower "Text line boundary." "Pen size" is the positive difference between the upper "Text line baseline" and the lower "Text line baseline."

3.3. Character Segmentation

We have optimized the contour-based character segmentation method (Azmi & Kabir, 2001) for character segmentation. The optimization is based on Kurdish fonts and character proportions, in order to increase the overall performance of the method. First, we perform a connected component analysis to separate all contiguous black pixels on each text line. Then, we categorize connected components into three categories as follows:

1. The main body of characters on a text line; these are the main parts of Persian/Arabic based characters that concatenate with other characters. So, the character segmentation takes place on these components that we label as "text line sub-words". We find text line sub-words using the fact that they lie on the text line baselines.

2. Text line upper signs; these are the upper sign components of characters. We find them using the fact that they lie above text line baselines.
3. Text line lower signs; these are the lower sign components of characters. We find them using the fact that they lie below text line baselines.

This categorization makes the system more efficient since only text line sub-words' upper contours are required for segmentation. However, after determining segmentation points on each of these sub-words, we do take upper and lower signs into consideration. This is done so that signs are not misattributed to neighboring characters.

There are a number of font styles in which periods and commas are placed below or above the baselines (Fig 4). These will be detected and corrected by checking if they are above or below any text line sub-words. If they are not, they will be added as text line sub-words.

Furthermore, a number of font styles are designed to have upper or lower signs to be partially on the baseline (Fig. 5). In this case, we check to see whether the characters touch both the upper and lower boundaries of the baseline. If they do not, they will be added to upper or lower signs depending on their position.

After the categorization, the process continues by assessing text line sub-words. For this, we follow the method suggested by Azmi and Kabir (2001). That is, the upper contours of each sub-word are found and compared to the values of the upper and lower text line baseline. We label each pixel on the upper contour based on where they are relative to the baseline. For this, we use the same notation used by Azmi and Kabir (2001), in which labels *u*, *m*, and *d* stand for *up*, *middle*, and *down* respectively. If a pixel is above the upper baseline, it will be labeled *u*. If it is below the lower baseline, it will be labeled *d*, and if it is between or on the baseline boundaries, it will be labeled *m*. Then, sequences of these labels will be generated along with their widths (Fig. 6). Every sequence will start with a *u* sequence to account for some characters which start with an unnecessary (for segmentation) *m* sequence (Fig. 7).

Before starting the rule-based segmentation, we identify characters that should not be over-segmented. For example, the characters س and ش have parts that are similar to a whole characters' main bodies (Fig. 8). These two characters have a "*u-m-u-m-u*" sequence independent of whether they are concatenated with other characters or not. However, many characters in Persian/Arabic scripts have *u-m* sequences when they are connected to other characters (Fig. 9). We filter س and ش characters by detecting whether they have upper and lower signs. If a *u-m-u-m-u* sequence has no signs above or

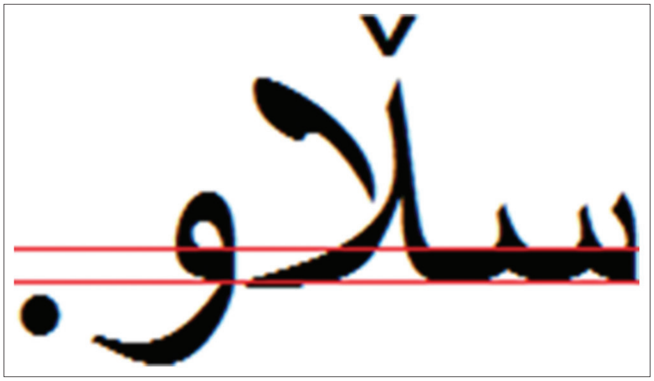


Figure 4. Example of a font style which put periods below the baseline

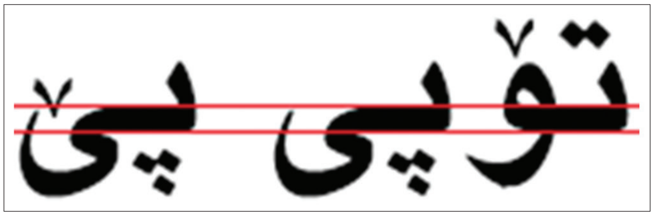


Figure 5. Example of a font style which puts upper signs partially on the baseline

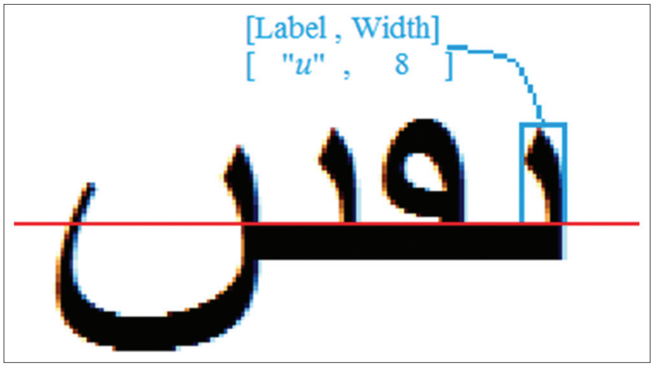


Figure 6. A sample of labeled sequence based on the upper contour

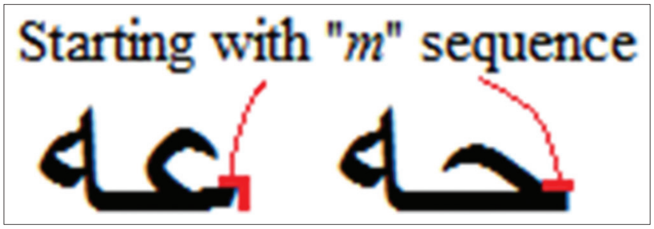


Figure 7. Characters that start with m sequence

below it, then it is س. If a *u-m-u-m-u* sequence has no signs below it, but 3 or fewer signs above it, it can be a ش. The contour properties of the signs will be analyzed to determine if they resemble a sign above ش.

After checking that over-segmentation has not occurred, we perform segmentation based on a series of rules which have been obtained through calculating character proportions in Kurdish fonts. The rules are as following:

1. If there is a sequence *u-m-u*, segment after *u-m*.
2. If there is a sequence *u-m-d*, segment after *u-m*.

Some parts of certain characters have sequences that would over-segment the characters if these rules are applied. We filter these exceptions using the widths and heights of those parts (Fig. 10). For example, a sub-word that ends with a u-m-u sequence is always going to end with ا or ه characters (Fig. 10). The character ا would have a height of at least 4 times the pen size. The character ه would have a width of at least 2 times the pen size. The width is calculated from the highest peak to the left (Fig. 10, right side). These two rules are established after pen size normalization for fonts with a low text line to baseline proportion (Fig. 11).

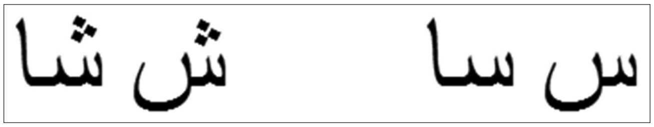


Figure 8. Left-character isolated and connected. Right-character isolated and connected

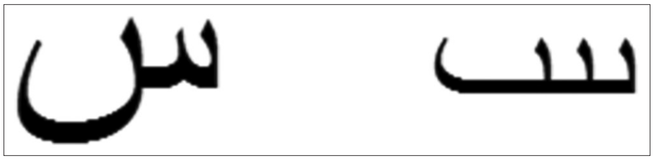


Figure 9. Left - The character. Right - 3 connected characters' main bodies

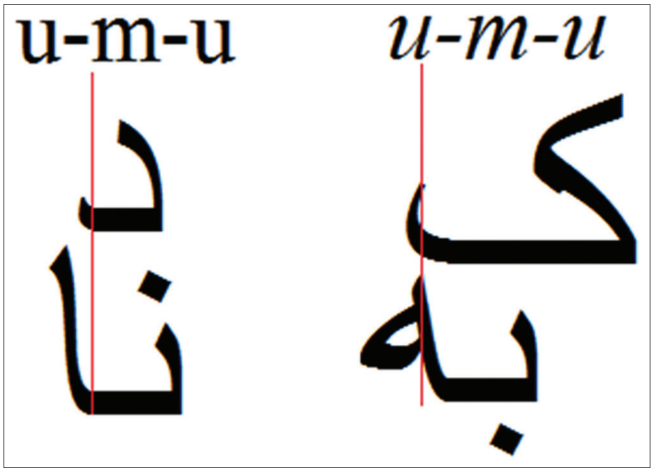


Figure 10. Example of the difference in width and height between actual segmentation points and none-segmentation points of the same label sequence

When the text line sub-words are segmented, the segmented characters are joint together with the signs whose middle points are within the boundaries of each sub-word. Once the segmentation is complete, characters are classified based on their features relative to the data set of characters we have predefined using the Gamera (Droettboom, MacMillan, & Fujinaga, 2003) Framework.

Finally, we perform word segmentation in order to provide a textual output. For this, we apply the vertical (column) histogram projection on each text line and determine word segments based on a specified threshold (Zheng et al., 2004).

The threshold, in our case, is:

$$\text{Threshold}=3.5*\text{MedianSpaceBetweenSubWords} \qquad (2)$$

3.3.1. Evaluation

To test the system, we conduct experiments on different font categories, that is, fonts with different pen size to text line boundary proportions, and different font sizes for each of the fonts. For each, we test different image resolutions in terms of dots per inch (dpi) scanning resolution (Azmi & Kabir, 2001; Kanungo, et al., 1999). The method's accuracy rate is calculated based on character recognition accuracy (CRA) (Rashid, 2014), where:

$$CRA\%=\frac{N\text{-}ED}{N}*100 \qquad (3)$$

In this equation, N is the total number of characters in the original document. ED is the edit distance, which includes insertions, deletions, and substitutions (failure in segmentation and/or recognition). Table 1 demonstrates the selected font parameters.

Although different evaluation approaches such as Word Recognition Accuracy (WRA) exist to assess the accuracy of OCR methods (Agrawal, et al., 2009; Agrawal et al., 2011; Hubert, et al., 2016; Jumari & Ali, 2002), we use CRA because our focus is on character recognition at this stage. However, in the course of developing Kurdish

OCR further (see subsection 6.1), the evaluation should cover other methods as well according to the objectives of the project.

As Table 1 shows, we test two constants against a variable. For example, the first, third, and fifth row show that we experiment with varying fonts against constant font size of 12 and image resolution of 150 dpi.

4. EXPERIMENTS AND RESULTS

Table 1 provides the summary of the experiments. A sample of one of the tested documents and its corresponding result is shown in Figures 12 and 13.

Table 3 shows the average character accuracy rate for each tested font. Table 4 shows the average accuracy rate of each tested font size. Table 5 shows the average accuracy rates for the tested image resolution.

5. DISCUSSION

From the data that we created for the experiments, about 60% was used during the training stage, 20% for developing, and the rest for testing. The reason for using a large amount of data in the development stage came from what we found in the early phases of the development when we found that in several cases the segmentation process would result in incorrect detection depending on the size and type of the font. Therefore, we decided to extend the development data to tune the segmentation algorithm.

The experiment results showed that contour label based segmentation is successful with many different font styles for the Persian/Arabic based Kurdish texts. The average

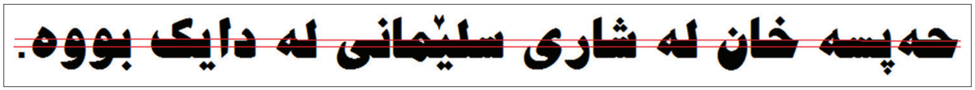
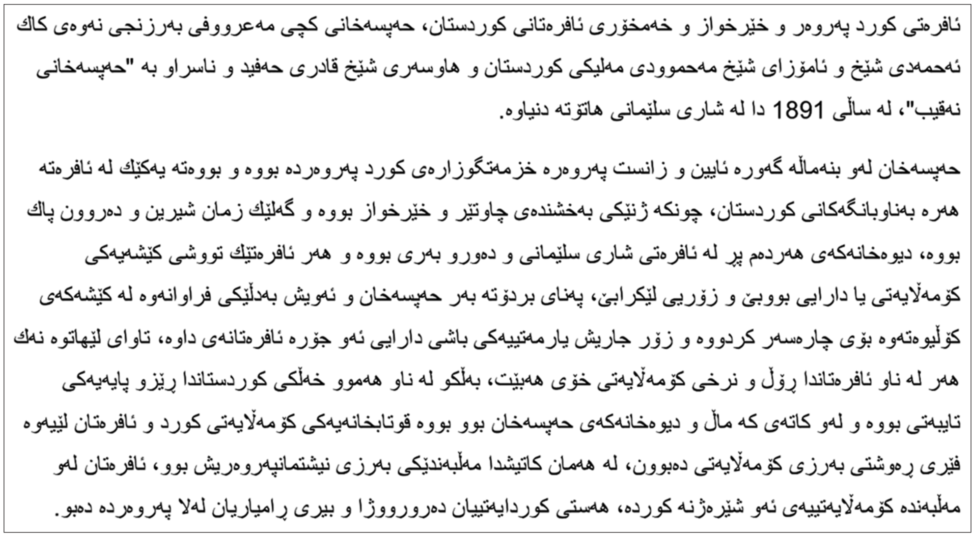


Figure 11. Font with thick original baseline is normalized in pen size



ئافرەتی کورد پەروەر و خێرخواز و خەمخۆری ئافرەتانی کوردستان، حەپسەخانی کچی مەعرووفی بەرزنجی نەوەی کاك
ئەحمەدی شێخ و ئامۆزای شێخ مەحموودی مەلیکی کوردستان و هاوسەری شێخ قادری حەفید و ناسراو بە "حەپسەخانی
نەقیب"، لە ساڵی 1891 دا لە شاری سلێمانی هاتۆتە دنیاوە.

حەپسەخان لەو بنەماڵە گەورە ئایین و زانست پەروەرە خزمەتگوزارەی کورد پەروەردە بووە و بووەتە یەکێك لە ئافرەتە
هەرە بەناوبانگەکانی کوردستان، چونکە ژنێکی بەخشندەی چاوتێر و خێرخواز بووە و گەلێك زمان شیرین و دەروون پاك
بووە، دیوەخانەکەی هەردەم پڕ لە ئافرەتی شاری سلێمانی و دەورو بەری بووە و هەر ئافرەتێك تووشی کێشەیەکی
کۆمەڵایەتی یا دارایی بووبێ و زۆریی لێکرابێ، پەنای بردۆتە بەر حەپسەخان و ئەویش بەدڵێکی فراوانەوە لە کێشەکەی
کۆڵیوەتەوە بۆی چارەسەر کردووە و زۆر جاریش یارمەتییەکی باشی دارایی ئەو جۆرە ئافرەتانەی داوە، تاوای لێهاتوە نەك
هەر لە ناو ئافرەتاندا ڕۆڵ و نرخی کۆمەڵایەتی خۆی هەبێت، بەڵکو لە ناو هەموو خەڵکی کوردستاندا ڕێزو پایەیەکی
تایبەتی بووە و لەو کاتەی کە ماڵ و دیوەخانەکەی حەپسەخان بوو بووە قوتابخانەیەکی کۆمەڵایەتی کورد و ئافرەتان لێیەوە
فێری ڕەوشتی بەرزی کۆمەڵایەتی دەبوون، لە هەمان کاتیشدا مەڵبەندێکی بەرزی نیشتمانپەروەریش بوو، ئافرەتان لەو
مەڵبەندە کۆمەڵایەتییەی ئەو شێرەژنە کوردە، هەستی کوردایەتییان دەرورووژا و بیری ڕامیاریان لەلا پەروەردە دەبو.

Figure 12. Original text image (Arial, 12, 400 dpi)

Table 1. Variable font sizes and image resolutions for experiments

Size*	Dots per Inch (dpi)
12 (normal size text)	150 (low resolution scanner)
12 (normal size text)	400 (medium resolution scanner)
14 (title size text)	150 (low resolution scanner)
14 (title size text)	400 (medium resolution scanner)
32 (large title size text, i.e., chapter title)	150 (low resolution scanner)
32 (large title size text, i.e., chapter title)	400 (medium resolution scanner)

*Tested fonts: Arial, Times New Roman, Chemen, Ezmer, Goran, Hejar, Hemen, Hiwa. Documents with a mixture of these fonts have also been tested

Table 2. Experiment results

Font	Size	dpi	Accuracy (%)
Arial	12	150	89.23
Arial	12	400	98.72
Arial	14	150	94.69
Arial	14	400	98.26
Arial	32	150	96.67
Arial	32	400	100
Times New Roman	12	150	91.81
Times New Roman	12	400	98.48
Times New Roman	14	150	94.88
Times New Roman	14	400	98.50
Times New Roman	32	150	97.10
Times New Roman	32	400	100
Chimen	12	150	97.55
Chimen	12	400	95.90
Chimen	14	150	88.81
Chimen	14	400	92.43
Chimen	32	150	94.68
Chimen	32	400	93.43
Ezmer	12	150	72.01
Ezmer	12	400	93.30
Ezmer	14	150	88.69
Ezmer	14	400	94.61
Ezmer	32	150	94.92
Ezmer	32	400	93.05
Goran	12	150	92.09
Goran	12	400	97.57
Goran	14	150	92.22
Goran	14	400	93.97
Goran	32	150	94.87
Goran	32	400	97.48
Hejar	12	150	83.56
Hejar	12	400	93.84
Hejar	14	150	69.48
Hejar	14	400	88.15
Hejar	32	150	91.44
Hejar	32	400	90.37
Hemen	12	150	77.62
Hemen	12	400	93.35
Hemen	14	150	86.56
Hemen	14	400	92.97
Hemen	32	150	93.57
Hemen	32	400	93.57
Hiwa	12	150	86.32
Hiwa	12	400	91.37
Hiwa	14	150	84.46
Hiwa	14	400	90.04
Hiwa	32	150	88.46
Hiwa	32	400	88.46
Mixed (All of the above)	12	150	79.56
Mixed (All of the above)	12	400	82.70
Mixed (All of the above)	14	150	79.30
Mixed (All of the above)	14	400	82.52
Mixed (All of the above)	32	150	85.83
Mixed (All of the above)	32	400	84.61
Average character accuracy rate			90.82

Table 3. Individual font style accuracy rate

Font	Accuracy (%)
Arial	96.26
Times	96.80
Chimen	93.80
Ezmer	89.43
Goran	94.70
Hejar	86.14
Hemen	89.61
Hiwa	88.19
Mixed	82.42

Table 4. Font size based accuracy rates

Size	Accuracy (%)
12	89.72
14	89.47
32	93.57

Table 5. Font size based accuracy rates

Dpi	Accuracy (%)
150	88.38
400	93.25

character recognition accuracy rate is 90.82%. The font with the least amount of character recognition accuracy is *Hejar*. This font has some character features that are not detected by the rules we have applied for segmentation (Fig 14).

We notice that, in general, the larger the font size for a specific font, the better the accuracy rates we obtain. However, the average accuracy rate for size 14 is lower than size 12 font sizes. This is evident when we look at the accuracy rates of Chimen, Hiwa, and Hejar fonts are lower in size 14 than in size 12. This is because our rules cannot detect the changes that occur between characters in these fonts. (Fig 15 and Fig 16).

The recognition of ش character has also been over-segmented in several cases and showed a lower accuracy rate (Fig 16).

Furthermore, on average, the higher the resolution of the image and the larger the font size, we obtain higher accuracy rates (Fig 17). This occurs due to the fact that low-resolution images suffer from the characters being connected to one another when they should not be connected. The false concatenation of low-resolution images happens because of the presence of the noise (Fig 18).

A certain percentage of the inaccuracies also happens when upper and lower signs are attributed to the wrong characters when they are not directly above or below their corresponding characters main body (Fig 19 and Fig 20).

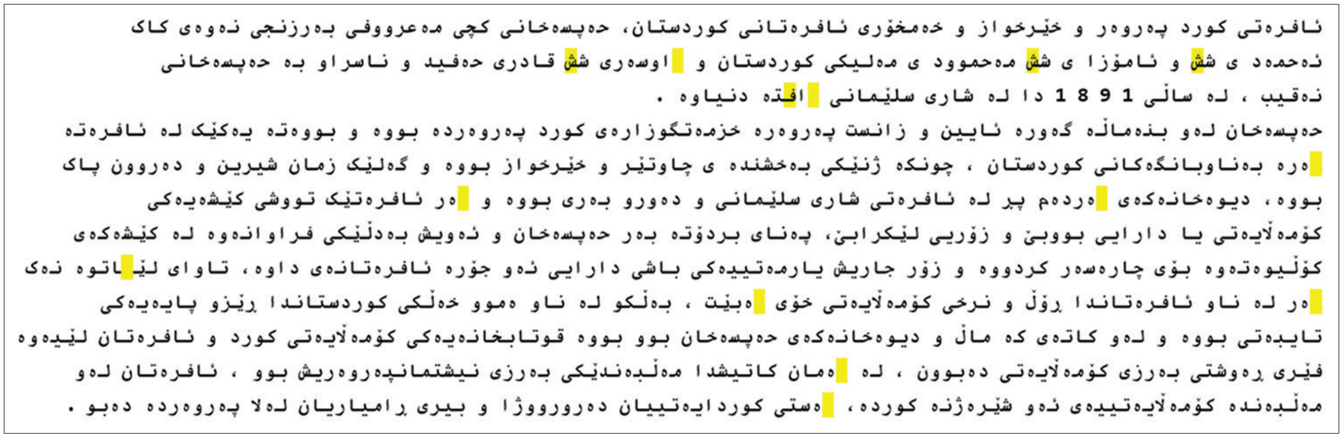

ئافرەتی کورد پەروەر و خێرخواز و خەمخۆری ئافرەتانی کوردستان، حەپسەخانی کچی مەعرووفی بەرزنجی نەوەی کاک
ئەحمەد ی شش و ئامۆزا ی شش مەحموود ی مەلیکی کوردستان و اوسەری شش قادری حەفید و ناسراو بە حەپسەخانی
نەقیب ، لە ساڵی 1 8 9 1 دا لە شاری سلێمانی افتە دنیاوە .
حەپسەخان لەو بنەماڵە گەورە ئایین و زانست پەروەرە خزمەتگوزارەی کورد پەروەردە بووە و بووەتە یەکێک لە ئافرەتە
ەرە بەناوبانگەکانی کوردستان ، چونکە ژنێکی بەخشندە ی چاوتێر و خێرخواز بووە و گەلێک زمان شیرین و دەروون پاک
بووە، دیوەخانەکەی ەردەم پڕ لە ئافرەتی شاری سلێمانی و دەورو بەری بووە و ەر ئافرەتێک تووشی کێشەیەکی
کۆمەڵایەتی یا دارایی بووبێ و زۆریی لێکرابێ، پەنای بردۆتە بەر حەپسەخان و ئەویش بەدڵێکی فراوانەوە لە کێشەکەی
کۆڵیوەتەوە بۆی چارەسەر کردووە و زۆر جاریش یارمەتییەکی باشی دارایی ئەو جۆرە ئافرەتانەی داوە، تاوای لێاتوە نەک
ەر لە ناو ئافرەتاندا ڕۆڵ و نرخی کۆمەڵایەتی خۆی ەبێت ، بەڵکو لە ناو ەموو خەڵکی کوردستاندا ڕێزو پایەیەکی
تایبەتی بووە و لەو کاتەی کە ماڵ و دیوەخانەکەی حەپسەخان بوو بووە قوتابخانەیەکی کۆمەڵایەتی کورد و ئافرەتان لێیەوە
فێری ڕەوشتی بەرزی کۆمەڵایەتی دەبوون ، لە ەمان کاتیشدا مەڵبەندێکی بەرزی نیشتمانپەروەریش بوو ، ئافرەتان لەو
مەڵبەندە کۆمەڵایەتییەی ئەو شێرەژنە کوردە، ەستی کوردایەتییان دەروروژا و بیری ڕامیاریان لەلا پەروەردە دەبو .

Figure 13. Text result output of Arial, size 12, and 400 dpi (character recognition accuracy 98.72%). Yellow highlights are inaccuracies

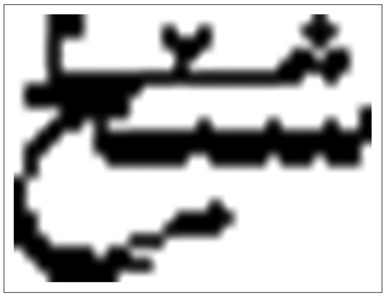


Figure 14. *Hejar* font has a very small change in height from the baseline between connecting characters. The additional connecting lines are artifacts of erroneous segmentation

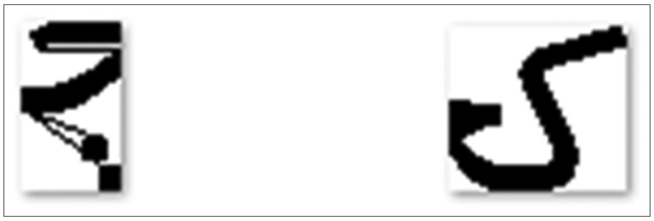


Figure 15. Characters *k* and *ch* are segmented in the wrong place due to the baseline to text line proportions

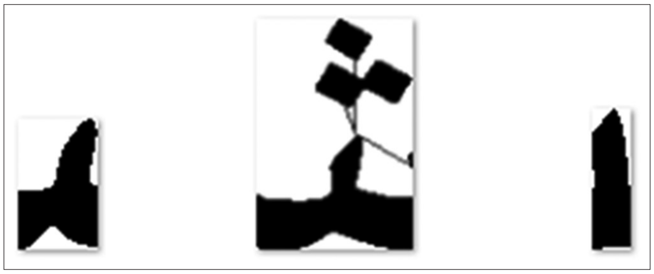


Figure 16. Character *sh* is over-segmented in *Hiwa* font

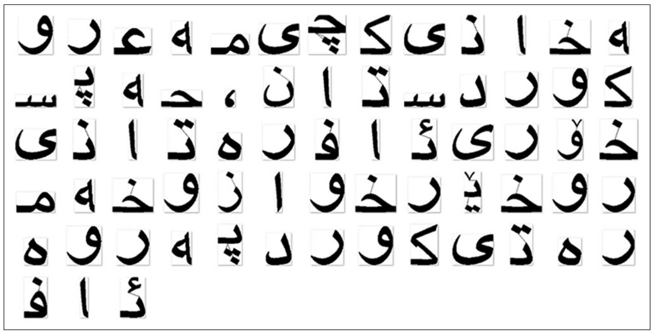


Figure 17. *Goran* font of size 32 and resolution of 400 dpi

In addition, mixing multiple fonts together in one image leads to the least amount of accuracy rates (Table 3), because each font has a specific proportion of heights and widths of character parts. Our segmentation method does not recognize the differences between fonts in one image, which leads to low segmentation accuracy rate (Fig 21).

6. CONCLUSION

This article suggested a method for Optical Character Recognition of Kurdish texts written in the Persian/Arabic script. Our method is a modification and enhancement of the methods that have been suggested for Farsi (Persian) and Arabic, particularly contour labeling based segmentation. We conducted a number of experiments on a variety of fonts, font sizes, and image resolutions. The method showed a recognition of 90.82% on average. However, the accuracy was lower for a few cases, depending on the font style, size, or having mixed-fonts on a document.

6.1. Future Work

To the best of our knowledge, this is the first attempt to develop an offline Kurdish Optical Character Recognition. The work can be improved and expanded in several areas. First, the pre-processing and post-processing algorithms should be implemented in order to leverage the work to a system

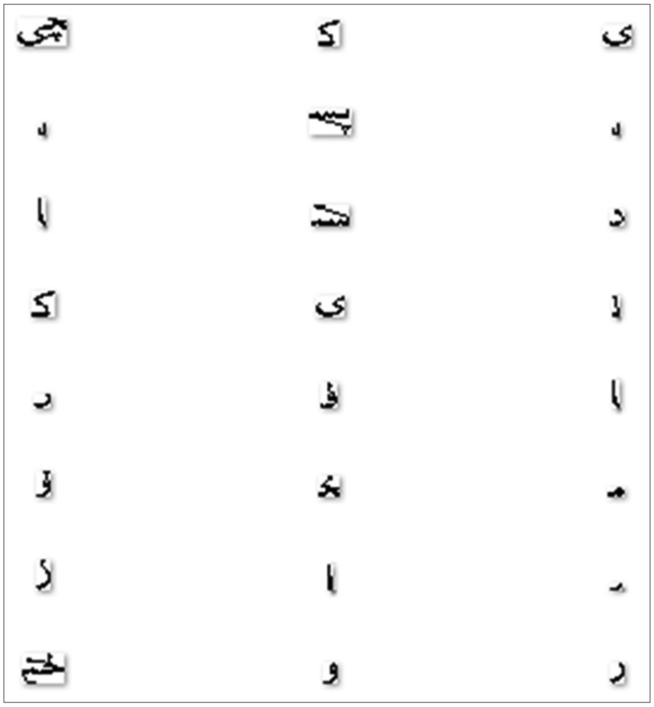


Figure 18. Samples of *Ezmer* font of size 12 and resolution of 150 dpi

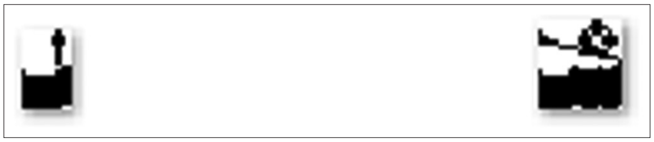


Figure 19. A dot sign above the character on the left is falsely attributed to character on the right

that could be used by ordinary users. Second, pre-processing techniques are needed for old texts and noisy texts as well, which are difficult to digitize at the moment. For example, pre-processing techniques such as smoothing, thresholding, de-skewing, and more advanced document analysis and region categorization concepts could be applied in order to enhance the segmentation process and make it more accurate (Belaïd and Ouwayed 2012, for example). Third, a post-processing stage could also be added, that might use a Kurdish dialect classification (Hassani and Medjedovic (2016) whereby using a language model for the specified dialect, the output could be checked for spelling and grammar accuracy. Finally, applying scale independent feature selection for better recognition (Mohammed (2013) could be another area for further study.

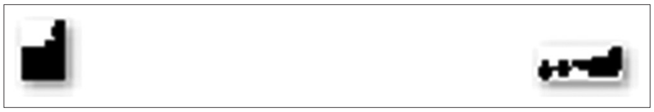


Figure 20. The dot signs below the character on the left are falsely attributed to the character on the right

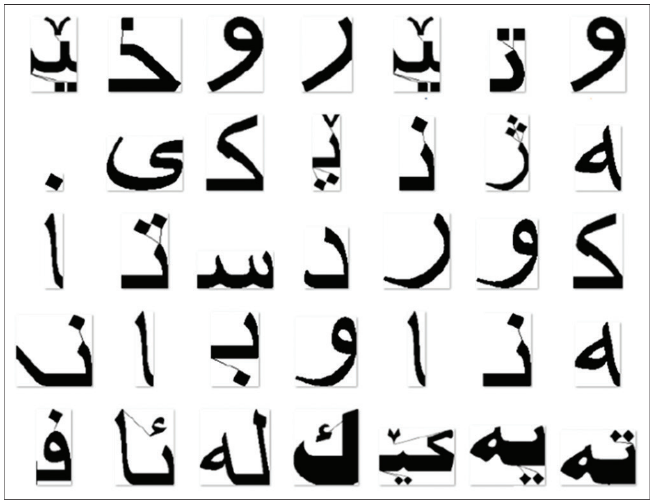


Figure 21. Samples of mixing fonts in one image lead to the least amount of segmentation accuracy rates

REFERENCES

Agrawal, S., Constandache, I., Gaonkar, S. & Choudhury, R.R. (2009). *Phonepoint Pen: Using Mobile Phones to Write in Air.* In: Proceedings of the 1st ACM workshop on Networking, Systems, and Applications for Mobile Hand-Helds. p. 1-6.

Agrawal, S., Constandache, I., Gaonkar, S., Roy, C. R., Caves, K. & DeRuyter, F. (2011). *Using Mobile Phones to Write in Air.* In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. ACM, New York, USA. p. 15-28.

Amin, A. (1988). *OCR of Arabic Texts.* Pattern Recognition, Cambridge, UK. p. 616-625.

Amin, A. (1991). Recognition of Arabic handprinted mathematical formulas. *Arabian Journal for Science and Engineering,* 16(4), 531-542.

Azmi, R. & Kabir, E. (2001). A new segmentation technique for omnifont Farsi text. *Pattern Recognition Letters,* 22(2), 97-104.

Belaïd, A. & Ouwayed, N. (2012). Segmentation of ancient Arabic documents. *In: Guide to OCR for Arabic Scripts.* Springer, Londonp. p. 103-122.

Cheung, A., Bennamoun, M. & Bergmann, N. W. (2001). An Arabic optical character recognition system using recognition-based segmentation. *Pattern Recognition,* 34(2), 215-233.

Droettboom, M., MacMillan, K. & Fujinaga, I. (2003). The Gamera framework for building custom recognition systems. *In: Symposium on Document Image Understanding Technologies.* pp. 275-286. Available from: http://www.gamera.informatik.hsnr.de/. [Last retrieved on 2016 Dec 15].

Hassani, H. (2018). BLARK for multi-dialect languages: Towards the Kurdish BLARK. *Language Resources and Evaluation,* 52(2), 625-644.

Hassani, H. (2017). Kurdish Interdialect Machine Translation. *In: Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial).* Association for Computational Linguistics. p. 63-72).

Hassani, H. & Medjedovic, D. (2016). Automatic Kurdish dialects identification. *Computer Science and Information Technology,* 6(2), 61-78. Available from: http://www.airccj.org/CSCP/vol6/csit65007.pdf. [Last retrieved on 2016 Nov 18].

Hassanpour, A. (1992). Nationalism and Language in Kurdistan, 1918-1985. Mellen Research University Press, San Francisco.

Hubert, I., Arppe, A., Lachler, J. & Santos, E. A. (2016). Training and quality assessment of an optical character recognition model for northern Haida. In: Chair, N. C. C., Choukri, K., Declerck, T., Muud, A., Maegaard, B., Mariani, A., Odijk, A. & Piperidis, J., editors. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (lrec 2016).* European Language Resources Association (ELRA), Paris, France. Available from: http://www.lrec-conf.org/proceedings/lrec2016/pdf/39 Paper.pdf. [Last retrieved on 2018 Apr 27].

Jumari, K. & Ali, M. A. (2002). A survey and comparative evaluation of selected online Arabic handwritten character recognition systems. *Jurnal Technology,* 36, 1-18.

Kanungo, T., Marton, G. A. & Bulbul, O. (1999). Performance evaluation of two Arabic OCR products. In: Proceedings of SPIE-the international society for optical engineering. *SPIE,* 3584, 76-83.

Mohammed, B. O. (2013). Handwritten Kurdish character recognition using geometric discertization feature. *International Journal of Computer Science and Communication,* 4, 51-55.

Rashid, S. F. (2014). *Optical Character Recognition-A Combined ANN/HMM Approach (Unpublished Doctoral Dissertation).* Technical University of Kaiserslautern.

Zheng, L., Hassin, A. H. & Tang, X. (2004). A new algorithm for machine printed Arabic character segmentation. *Pattern Recognition Letters,* 25(15), 1723-1729.