Academic Journal of Nawroz University (AJNU), Vol.11, No.3, 2022

https://doi.org/10.25007/ajnu.v11n3a673

OFFLINE OCR ALGORITHM TO DETECT KURDISH/ ARABIC CHARACTERS IN SCANNED DOCUMENT

Sardar Omar Salih [1]

[1] Department of Information Technology, Polytechnic University, Kurdistan Region –Iraq

ABSTRACT

In this paper, Algorithm named (MRWL) Max Rightmost White Line is proposed to detect Kurdish/ Arabic characters' segmentation in scanned document (printed document), it works in preprocess and segmentation stages of OCR processes, these two stages are significant parts of OCR and affect the accuracy of algorithm. The MRWL starts to remove text margins around document to reduce processing time, then, scans to find Top Line (TL) and Bottom Line (BL) for each sentence in paragraph which be used to measure height of characters. Based on TL and BL, the Base Line (BSL) can be detected using horizontally Most Frequency Black Pixel (MFBP) which is useful to find characters' segmentation.

Finding TL, BL and BSL of each sentence help to find characters location in document. Six phases involve in algorithm, each phase has its own functionally. The Algorithm is tested with different input documents and the average accurate rate of detected segmentations is recorded 96.93%.

KEYWORDS: Script, Character, Top Line (TL) and Bottom Line (BL), Base Line (BSL), OCR, Kurdish/ Arabic characters.

1. Introduction

In some situation scanned document (hand written and printed text) are required to encoded to the text form for modifying, coping, searching, sorting and other text manipulating process. The algorithm is designed to find location of characters in scanned document, then, detected characters can be compared with the Kurdish / Arabic characters' using such available comparation algorithm like Fourier analysis, boundary line encoding, polygonal approximation and Chain encoding (Shridhar and Badreldin, 1984). Detecting Kurdish/ Arabic characters' segmentations are the challenge compared to Latin characters like English, reasons , Kurdish/ Arabic script characteristics, such that , characters are cursive, ligatured , overlapped and no such regular patterns between each character, for instance, white line in between which exist in English script, differ in size and diacritics (special mark over or under characters), characters can be written in many glyphs, depend on location in word begin, middle , end and isolate. See Table1 Glyphs of characters in Kurdish/Arabic script. (Althobaiti and Lu, 2017; Shatnawi, 2015)

Table 1: Glyph of characters in Kurdish/Arabic script

No.	Central Kurdish (Soraní - modified Arabic)	No.	Central Kurdish (Soraní - modified Arabic)
.1	ئا ـا ا	.17	ن ـن نـ ـنـ
.2	ب ـب بـ ـبـ	.18	ۆ ـۆ ئۆ
.3	چ ـچ چـ ـچـ	.19	پ ـپ پـ ـپـ
.4	د ـد	.20	ق ـق قـ ـقـ
.5	ە ـە ئە	.21	ر ـر
.6	ێ ـێ ێـ ـێـ ئێ	.22	ڕ ـڕ
.7	ف ـف فـ ـفـ	.23	س ـس سـ ـسـ
.8	گ ـگ گـ ـگـ	.24	ش ـش شـ ـشـ
.9	هـ ـهـ	.25	ت ـت تـ ـتـ
.10	ی ئی یـ ـیـ	.26	و ـو ئو
.11	ج ـج جـ ـجـ	.27	وو ـوو
.12	ژ ـژ	.28	ڤ ـڤ ڤـ ـڤـ
.13	ک ـک کـ ـکـ	.29	و ـو
.14	ل ـل لـ ـلـ	.30	خ ـخ خـ ـخـ
.15	ڵ ـڵ ڵـ ـڵـ	.31	ی ي يـ ـيـ
.16	م ـم مـ ـمـ	.32	ز ـز

2. RELATED WORK

There are many attempts to increase accuracy rate of recognition characters in Kurdish / Arabic scanned documents, in 1988 Al-Badr proposed method named "shape primitives" and obtain accurate rate 94.1% for scanned symbols (Al-Badr & Haralick, 1998).

After that, 2004, Haraty,and Ghaddar use algorithm based on neural networks , the accuracy rate is recorded 73% (Haraty,and Ghaddar 2004).

In 2004, Tomeh, Nadi, et al. Apply linguistically and semantically features to pre-achieved OCR system followed an n-best list re-ranking approach. The result shows by 10.1% to 11.4%-word error rate on both scanned and handwrite document (Tomeh et al., 2013).

Another related work, in 2012, conducted to recognize Kurdish/ Arabic text by Aljarrah, Inad, et al. they use "Lookup Dictionary" to correctly classify characters. This method improves accuracy rate ranging from 93.5% and 96.1% (AL jarrah et al., 2012).

Moreover, in 2018, Rasty and Hossein proposed the algorithm to enhance characters recognition in Kurdish characters, they come up with accuracy result 90.82%.

During this journey, proposed MRWL algorithm in this paper come up with a result of accuracy 96.93% of scanned documents according to the cases exclude non-hand write documents.

3. PHASES OF DESIGNING MRWL

MRWL has six main phases in order to locate characters in document, the followings are the explanation of each of them.

3.1 PHASE ONE: CONVERTING SCANNED DOCUMENT TO THE MATRIX OF BINARY DATA RESPRESENTS WHITE AND BLACK PIXELES

In order to detect characters, scanned image is converted to binary with bunch of '0's and '1's and save in the array of two dimensions (width by height). '0' indicates to black pixel (characters) and '1' white pixel (document background). Figure 1 shows how characters are represented in matrix. The algorithm checks scanned image , in case, if it is a RGB image (colored), then, it converts to grayscale image using formula 0.30*R + 0.59*G + 0.11*B or this can be done using MATLAB (8.3.0.532 (R2014a)) build in functions , then, ,gray image can be converted to binary using threshold is a constant, which it is value calculated using threshold methods (Sezgin and Sankur, 2004) .

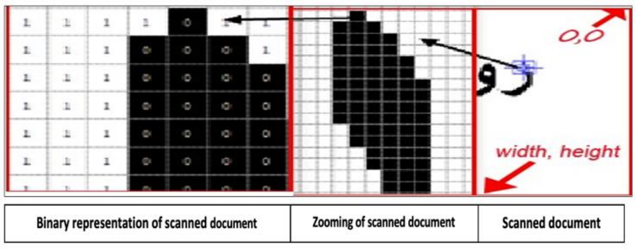
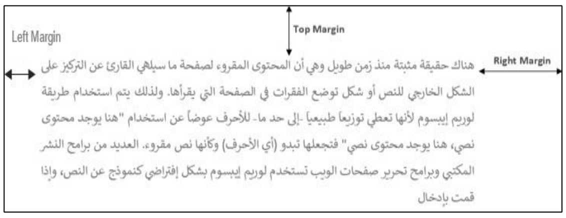

0,0
width, height
Binary representation of scanned document
Zooming of scanned document
Scanned document

Fig. 1: characters represent as bunch of 0s and 1s

3.2 PHASE TWO: CROPPING TOP, RIGHT AND LEFT MARGINS OF SCANNED DOCUMENT

Cropping margins (white space from edge of the document to the text) help to reduce time of scanning. The figure bellow shows margins of scanned document.


Left Margin
Top Margin
Right Margin

هناك حقيقة مثبتة منذ زمن طويل وهي أن المحتوى المقروء لصفحة ما سيلهي القارئ عن التركيز على الشكل الخارجي للنص أو شكل توضع الفقرات في الصفحة التي يقرأها. ولذلك يتم استخدام طريقة لوريم إيبسوم لأنها تعطي توزيعاَ طبيعياَ -إلى حد ما- للأحرف عوضاً عن استخدام "هنا يوجد محتوى نصي، هنا يوجد محتوى نصي" فتجعلها تبدو (أي الأحرف) وكأنها نص مقروء. العديد من برامح النشر المكتبي وبرامح تحرير صفحات الويب تستخدم لوريم إيبسوم بشكل إفتراضي كنموذج عن النص، وإذا قمت بإدخال

Fig. 2: Top, left and right margins of scanned document

To find, Right Margin (RM), it scans from right to left and looks at black pixel (0), if it is detected, then, first right black pixel location (distance of pixels from right edge of document) is stored in temporary array , then, it goes to next row and same process is applied till to reach end of rows (height of array), and minimum number in temporary array is considered to be (RM) . the following Figure 3 pseudocode of RM.

```
function RightMargin():
 temArray[height]
 K=1;
 for i=1 to height do
   for j=1 to width do
     if image[i,j]== 0 then
       temArray[k++] = j       /* j is a distance from right
e.          dge of document to rightmost black pixel */
       break;
     end
   end
 end
 return Min(temArray)
```

```
end
```

Fig. 3: Finding right margin

Same process is applied to find Left Margin (LM) but, scanning is opposite to RM, it scans from left to right and stores leftmost black pixel, then maximum number is considered LM. See the following Figure 4 pseudocode of LM.

```
function LeftMargin():
 temArray[height];
 K=1;
 for i=1 to height do
   for j=width to RM do
     if image[i,j]== 0 then
       temArray[k++] = j
/* j is a distance from left edge of document to leftmo
st black pixel */
        break;
     end
   end
 end
 return Max(temArray)
end
```

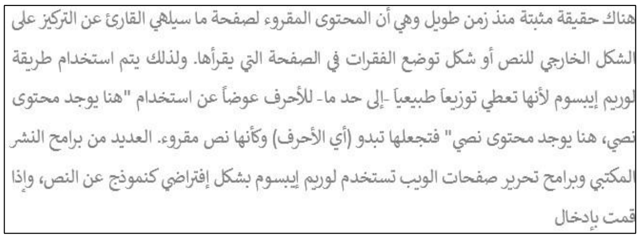
Fig. 4: Finding left margin

For Top Margin (TM), it scans from up to down until find black pixel and store in temporary array, then it goes to next columns and same process is applied until reach LM, then, minimum number in temporary array is TM. See the following Figure 5 pseudocode of TM.

```
function TopMargin():
 temArray[height];
 K=1;
 for i=1 to height do
   for j=RM to LM do
     if image[j,i]== 0 then
       temArray[k++] = j
/* j is a distance from top edge of document to topmo
st black pixel */
       break;
     end
   end
 end
 return Min (temArray)
end
```

Fig. 5: Finding top margin

Last sentence bottom's line is considered a Bottom Margin (BM), this can be finding with bottom line (BL) function will be explained later. The following Figure 6 is margins cropped of document in Figure 2 above, after applying above functions.



هناك حقيقة مثبتة منذ زمن طويل وهي أن المحتوى المقروء لصفحة ما سيلهي القارئ عن التركيز على
الشكل الخارجي للنص أو شكل توضع الفقرات في الصفحة التي يقرأها. ولذلك يتم استخدام طريقة
لوريم إيبسوم لأنها تعطي توزيعاَ طبيعياَ -إلى حد ما- للأحرف عوضاً عن استخدام "هنا يوجد محتوى
نصي، هنا يوجد محتوى نصي" فتجعلها تبدو (أي الأحرف) وكأنها نص مقروء. العديد من برامح النشر
المكتبي وبرامح تحرير صفحات الويب تستخدم لوريم إيبسوم بشكل إفتراضي كنموذج عن النص، وإذا
قمت بإدخال

Fig. 6: Figure two after margins cropped

3.3 PHASE THREE: TOP, BOTTOM, MIDDLE AND BASE LINES OF EACH SENTENCE IN SCANNED DOCUMENT

TL, BL, ML and BSL in each sentence are the keys to detect characters in scanned document for this algorithm. See figure 7 TL, BL, ML and BSL in sentence. To find BSL (is a horizontally line across sentence that has highest intensive black pixel, see Figure 7) in each sentence, the TL and BL first need to be located. if BSL is assumed as ML between TL and BL (TL – BL / 2), in most cases, this ML is not crossed in BSL, there for, it calls Inaccurate Base Line (IBL), see Figure 7 ML is not crossed in BSL. Next phase explains how to find accurate BSL?

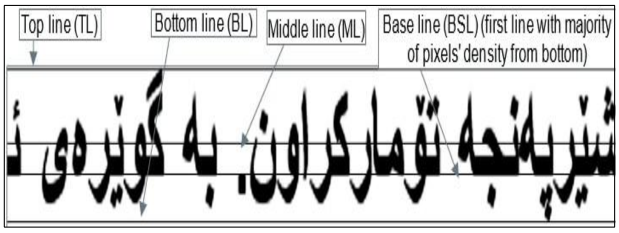

Top line (TL)
Bottom line (BL)
Middle line (ML)
Base line (BSL) (first line with majority of pixels' density from bottom)

Fig. 7: TL, BL, ML and BSL of scanned document

To find TL, it scans from bottom line of previous sentence (column by column) by constant value 100 pixels and store topmost black pixel in temporary array, the minimum number among temporary array is assigned as TL. 100 pixels constant is chosen in behave of height of document, due to, BL of previous sentence is closed to TL of next sentence. therefore, 100 pixels is enough to find TL, also, if image height is used and sentences length are not equal in paragraphs, this causes to unexpected ML which will be used in BL function. See Figure 8 pseudocode of TL.

```
function TopLine ()
  BL = TM // Bottom Line = Top Margin
  while BL < Height do
   temArray [LM] /*create arrat , size of temp array = Left Margin = document width */
   K=1
   if BL + 100 < Height then
     HE =BL+100 /* HE = Hight Estimated is 100 from BL */
   else
     HE =row
   end
   for j=RM to LM do
     for i=BL+1 to HE do
       if image[i,j]== 0 then
   temArray[k++] = I /* i is a distance from top of sentence to black pixel */
         break
       end
     end
  end
 end
 ML = median(temArray) /* Middle line = Median value, ML will be used to find bottom*/
 return Min(temArray) /* Min number in array which is considered Top Line */
end
```

Fig. 8: Top line function

To detect BL, it scans from assume line ML (ML calculates as Median from Top Line function) row by row until reached to first white line bellow sentence and assigned as BL. See Figure 9 pseudocode of BL.

```
Function BottomLine()
 for i= ML to height do
  findBlack=0
  for j= RM to LM do
   if image[i,j]== 0 then    // Black Pixel = 0
    findBlack = true
    break
   end
  end
  if findBlack == false then    //find white line is bottom of sentence
   BL = i    // i is a bottom line
  end
 end
 if i == heigh then
   BL = heigh // BL = heigh means Bottom Margin ( BM) (Last sentence bottom)
 end
 return BL /* return bottom Line
end
```

Fig. 9: Bottom line function

3.4 PHASE FOUR: ACCURATE BASE LINE IN EACH SENTENCE

As mentioned, the horizontally ML passes between both top and bottom line may not cross sentence's BSL (AlKhateeb et al., 2008). See Figure 10, in case, ML is used as BSL in algorithm, it causes to find inaccurate characters segments. Therefore, BSL is required to locate in sentences, to find it, algorithm scans from BL up to ML and stores detected black pixel position in array, then, Most Frequency Black Pixel (MFBP) is chosen as BSL, reason, BSL is the line has highest intensive black pixel. See Figure 10. but in some cases, this is not best choose due to, in case, characters are isolated and not ligatured, see Figure 6, thus, highest intensive black pixel cannot be best choice, therefore, the distance in sentence from top to bottom is divided into four equal area with four lines, top and bottom lines are included named (L1, L2, L3 and L4) see Figure 10. In case MFBP line is located under L2 and above L4 is a best choose. otherwise L3 is considered best choose not MFBP. In the algorithm, BSL itself is not used for finding character location but line above it is uses for characters segmentation, the next phase explains how to find above BSL line.

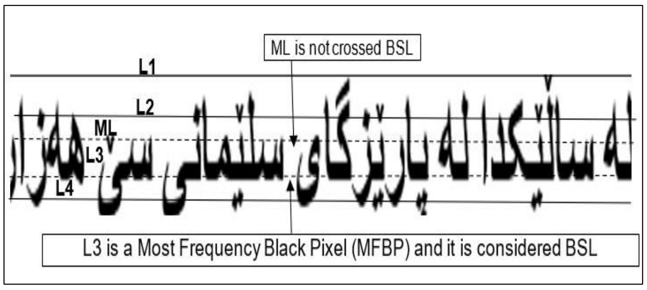

ML is not crossed BSL
L1
L2
ML
L3
L4
L3 is a Most Frequency Black Pixel (MFBP) and it is considered BSL

Fig. 10: Most Frequency Black Pixel (MFBP) with L1, L2, L3 and L4.

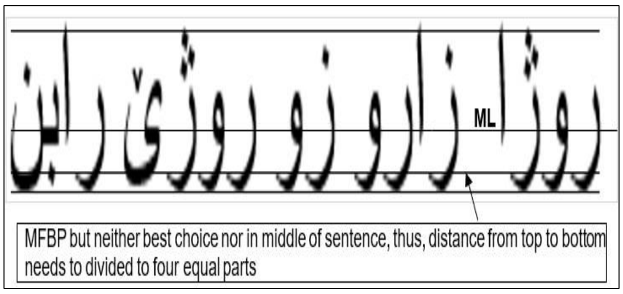

ML
MFBP but neither best choice nor in middle of sentence, thus, distance from top to bottom needs to divided to four equal parts

Fig. 11: MFBP is not cross BSL in isolated glyphs.

3.5 PHASE FIVE CALCULATE ABOVE BSL

To calculate above BSL, the algorithm scans from BSL up to ML (column by column). If white pixel is found, it stops and goes to next column and scans again, if black pixel is found and continuing to go up until reach white pixel, then, topmost black pixel is stored in temporary array. (MFBP) in array is assigned as above BSL. See Figure 12 above BSL.

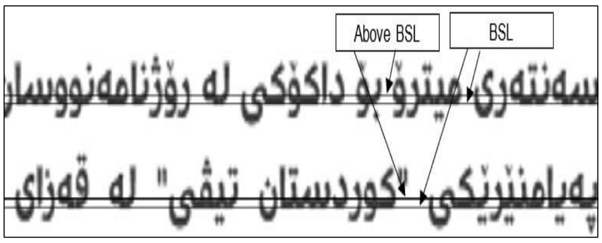


Fig. 12: Above BSL

3.6 PHASE SIX: DETECTING CHARACTER ONE BY ONE BASED ON THE ABOVE FINDINGS (TL, BL AND ABOVE BSL)

After TL, BL and above BSL are detected for each sentence in document, the process of finding character segmentation for each of sentence as follow.

The algorithm starts to detect rightmost white pixels from TL to above BSL minus one quarter (above BSL minus one quarter is used in scanned instead above BSL, to reduce two ligatured character detected as one), see Figure 13, and stores the rightmost white pixels in temporary array, the maximum number in array is assumed start point (SP) to find end of character. Two cases are considerable in this process:

Case one: if character is (isolated/white space after character) (Figure 13, character 1 and 2) : In this case SP is forwarded to first next white line from TL to BL, then, calculate total of white lines until reach to next black line , if they are less than five white lines, the line before black line is end of character, see Figure 13 character (1). Otherwise, start point is end of character if next white lines are more than five line (there is a space after character), see Figure 13 character (2).

Case two: if character is ligatured no white space after character (Figure 13, character 3 and 4):

in this case, SP is forwarded and continue to reach next black line from TL to above BSL minus one quarter and assigned white line before black line as end of character see Figure 13 characters (3) and (4).

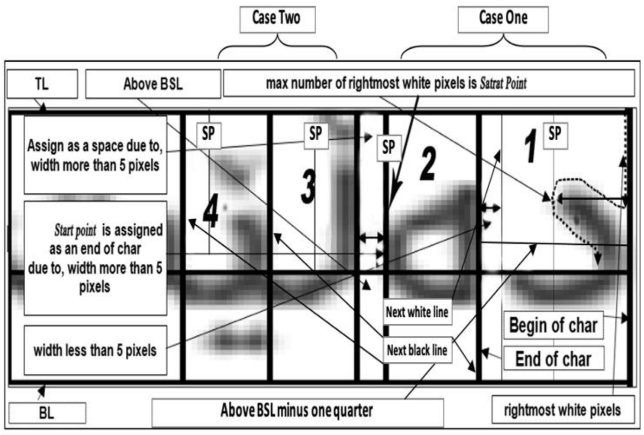
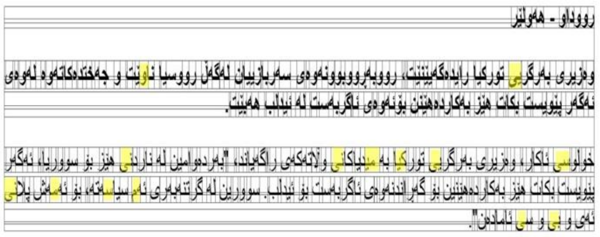

Case Two
Case One
TL
Above BSL
max number of rightmost white pixels is *Satrat Point*
SP
SP
SP
SP
4
3
2
1
Assign as a space due to, width more than 5 pixels
Start point is assigned as an end of char due to, width more than 5 pixels
width less than 5 pixels
Next white line
Next black line
Begin of char
End of char
BL
Above BSL minus one quarter
rightmost white pixels

Fig. 13: Detected characters in sentence.

4. CASES TEST

The following documents (Figure 14 to 18) of no handwrite documents are the output of algorithm and their results shown in bellow Table 2. The highlight locations indicate that the two neighbored characters are detected as one (inaccurate detection) and others are correctly detected. The Figure 19 handwrite document is the output with highlights location indicate that the more than neighbored characters are detected as one and others are correctly detected.

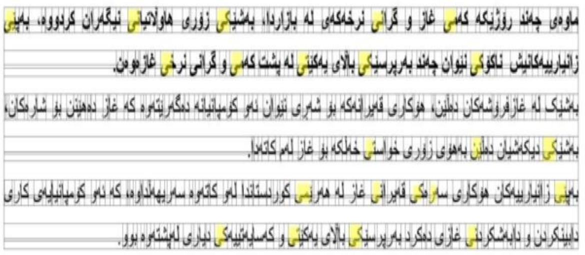


رووداو - هەولێر

وەزیری بەرگریی تورکیا رایدەگەیێنێت، رووبەڕووبوونەوەی سەربازییان لەگەڵ رووسیا ناوێت و جەختدەکاتەوە لەوەی
ئەگەر پێویست بکات هێز بەکاردەهێنن بۆئەوەی ئاگربەست لە ئیدلب هەبێت.

خولوسی ئاکار، وەزیری بەرگریی تورکیا بە میدیاکانی وڵاتەکەی راگەیاند، "بەردەوامین لە ناردنی هێز بۆ سووریا، ئەگەر
پێویست بکات هێز بەکاردەهێنین بۆ گەڕاندنەوەی ئاگربەست بۆ ئیدلب. سوورین لە گرتنەبەری ئەم سیاسەتە، بۆ ئەمەش پلانی
ئەی و بی و سی ئامادەن".

Fig. 14: Random paragraph from RUDAW website

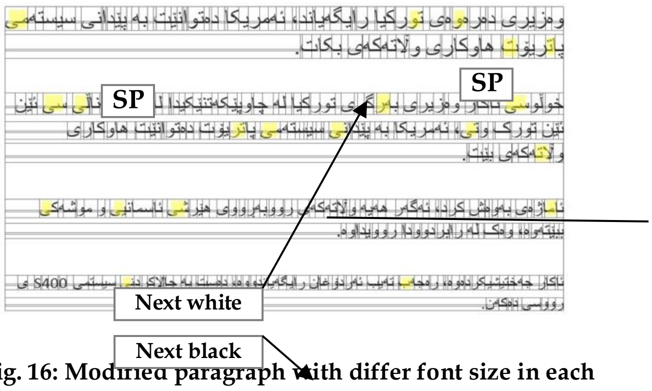


ماوەی چەند رۆژێکە کەمی غاز و گرانی نرخەکەی لە بازاردا، بەشێکی زۆری هاوڵاتیانی نیگەران کردووە، بەپێی
زانیارییەکانیش ناکۆکی نێوان چەند بەرپرسێکی باڵای یەکێتی لە پشت کەمی و گرانی نرخی غازەوەن.

بەشێک لە غازفرۆشەکان دەڵێن، هۆکاری قەیرانەکە بۆ شەڕی نێوان ئەو کۆمپانیانە دەگەڕێتەوە کە غاز دەهێنن بۆ شارەکان،
بەشێکی دیکەشیان دەڵێن بەهۆی زۆری خواستی خەڵکە بۆ غاز لەم کاتەدا.

بەپێی زانیارییەکان هۆکاری سەرەکی قەیرانی غاز لە هەرێمی کوردستاندا لەو کاتەوە سەریهەڵداوە، کە ئەو کۆمپانیایەی کاری
دابینکردن و دابەشکردنی غازی دەکرد بەرپرسێکی باڵای یەکێتی و کەسایەتییەکی دیاری لەپشتەوە بوو.

Fig. 15: Random paragraph from NRT website



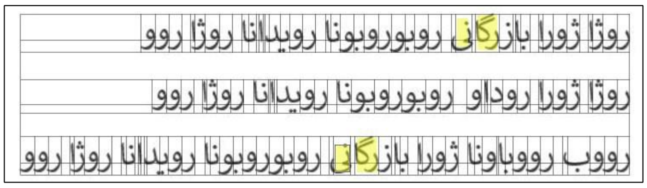
وەزیری دەرەوەی تورکیا رایگەیاند، ئەمریکا دەتوانێت بە پێدانی سیستەمی
پاتریۆت هاوکاری وڵاتەکەی بکات.

خولوسی ئاکار وەزیری بەرگری تورکیا لە چاوپێکەوتنێکیدا لە [illegible]ناڵی سی ئێن
ئێن تورک وتی، ئەمریکا بە پێدانی سیستەمی پاتریۆت دەتوانێت هاوکاری
وڵاتەکەی بێت.

ئاماژەی بەوەش کرد، ئەگەر هەیە وڵاتەکەی رووبەڕووی هێرشی ئاسمانی و موشەکی
ببێتەوە، وەک لە رابردوودا روویداوە.

ئاکار جەختیشیکردەوە، رەجەب تەیب ئەردۆغان رایگەیاندووە، دەستبەجێ بە چالاککردنی سیستەمی S400 ی
رووسی دەکەن.

Fig. 16: Modified paragraph with differ font size in each sentence from KNNC website

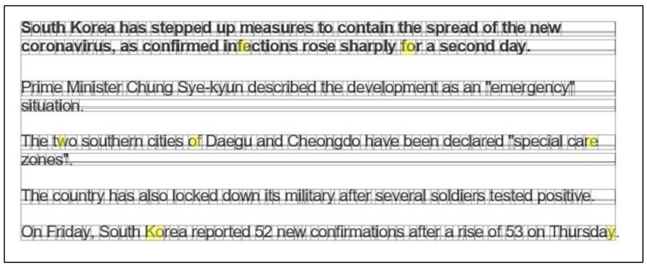


روژا ژورا بازرگانی روبوروبونا رویدانا روژا روو

روژا ژورا روداو روبوروبونا رویدانا روژا روو

رووب رووباونا ژورا بازرگانی روبوروبونا رویدانا روژا روو

Fig. 17: Random paragraph with glyphs is not ligatured sentences



South Korea has stepped up measures to contain the spread of the new coronavirus, as confirmed infections rose sharply for a second day.

Prime Minister Chung Sye-kyun described the development as an "emergency" situation.

The two southern cities of Daegu and Cheongdo have been declared "special care zones".

The country has also locked down its military after several soldiers tested positive.

On Friday, South Korea reported 52 new confirmations after a rise of 53 on Thursday.

Fig. 18: Random paragraph in Latin (English) paragraph from BBC website

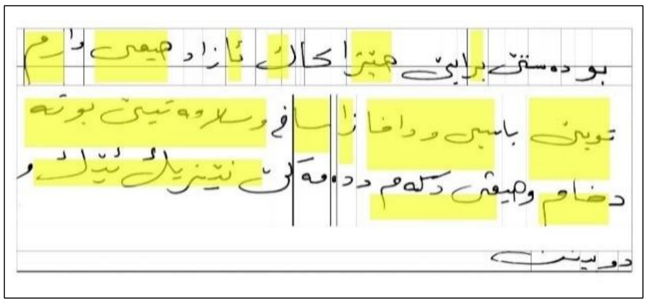


Fig. 19: Handwrite scanned document

The above six random documents (Figure 14-19) with different cases (font characteristics (name, size), language and handwrite) are used as an input of algorithm, the results are concluded in the following Table 2. The following equation is used to calculate rate of accuracy in each document.

The rate of accuracy =

$$\frac{\text{Number of Correct characters detected in document}}{\text{Total of Characters in document}}$$

Table 2: Testing Table

Document Figure	from / cases	Total char	Correct char	Incorrect char	% Of accuracy
Figure 14	from RUDAW	475	461	14	97.05%
Figure 15	from NRT	570	548	22	96.14%
Figure 16	from KNN	461	441	20	95.66%
Figure 17	not ligatured	148	144	4	97.29%
Figure 18	English	475	468	7	98.52%
Figure 19	Handwrite	93	37	56	39.78%
Overall, of accuracy for no hand write document					96.93%

5. DISCUSSION

The MRWL algorithm is tested with output 97.05% of correct detection segmentations in the paragraphs that are taken randomly from RUDAW website, 96.14% and 95.66% correct detection segmentations in paragraphs that are taken from NRT and KNN website in respectively, KNN paragraphs' sentences are modified to different font size for each sentence. The 98.52% correct located from English paragraphs and 97.29% with glyphs are not ligatured. These values indicate that the average of accuracy in algorithm for no handwrite is 96.93%, while, scanned handwrite text is used in the test and 39.78% accurate rate of characters are detected. Therefore, MRWL is not best choose for paragraph written in handwrite.

by looking at (Figure 14 to 18), two common inaccurate patterns are detected, first, two neighbored characters that are in the end of words, like (ـكى) (ـنى) (ـمى), are detected as one, reason, last characters of them (ى) falls under BSL, second, two neighbored characters that are overlapped such that (ـهم)(ـهب)(ـگا) are detected as one, this occur, as a result of, not white line over BSL between two characters. Figure 16, the modified sentences to variant font sizes are not affecting the algorithm accuracy compared to (Rasty and Hossein) proposed algorithm. These two inaccurate patterns, can be developed with more researches to increase accuracy of MRWL up to 99.99%.

The highest inaccurate locations are detected, in the handwrite document (see Figure 19), the reasons, the sentences are not in a straight line, therefore, algorithm cannot detect TL, BL and BSL which are the backbone of algorithm, therefore, MRWL is not working correctly with handwrite scripts. More that, MRWL can work with other than Kurdish/Arabic such as English and 98.52% of accuracy is recorded.

6. CONCLUSION

The MRWL passes six phases, in each of them, there is a function, starts from cutting margin, and to functions of finding TL, BL, BSL and above BSL, These Lines are considered the backbone for detecting characters' segmentation. The algorithm is tested with six random documents with specific cases as input and the output of each of them is measured to find accuracy of MRWL, the average of accuracy of non-handwrite document is recorded as 96.93% is more than the average of accuracy reported by (Rasty, Hossein, et al.) which is 90.82% (Yaseen and Hassani, 2018) and (Omar Al-Jarrah, Al-Kiswany , et al.) is 87% (Al-Jarrah et al., 2006) and 96.1% with lookup dictionary . This rate can be increased, if two inaccuracy patterns mentioned in discussion be addressed. For the hand write documents, MRWL cannot be used, due to, fewer correct characters are detected which is 39.78%.

7. REFERENCES

1. AL-Shatnawi Atallah and Khairuddin Omar. Methods of arabic language baseline detection the state of art. *IJCSNS*, 8(10):137, 2008. Malayappan Shridhar and A. Badreldin. High accuracy character recognition algorithm using fourier and topological descriptors. *Pattern Recognition*, 17(5):515–524, 1984.
2. Hassan Althobaiti and Chao Lu. A survey on arabic optical character recognition and an isolated handwritten arabic character recognition algorithm using encoded freeman chain code. In *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2017.
3. Al-Badr, B., & Haralick, R. M. (1998). A segmentation-free approach to text recognition with application to Arabic text. *International Journal on Document Analysis and Recognition, 1(3), 147–166.*
4. Haraty, R., & Catherine G., (2004). Arabic text recognition.
5. Tomeh, N., Habash, N., Roth, R., Farra, N., Dasigi, P., & Diab, M. (2013). Reranking with linguistic and semantic

features for Arabic optical character recognition. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 549–555.

6. Aljarrah, I., Al-Khaleel, O., Mhaidat, K., Alrefai, M., Alzu'bi, A., & Rabab'ah, M. (2012). Automated system for Arabic optical character recognition. *Proceedings of the 3rd International Conference on Information and Communication Systems*, 1–6.
7. Maad Shatnawi. Off-line handwritten arabic character recognition: a survey. In *Proceedings of the international conference on image processing, computer vision, and pattern recognition (IPCV)*, page 52. The Steering Committee of The World Congress in Computer Science, Computer …, 2015.
8. Optimizing the color-to-gray-scale conversion for imageclassification.
9. Mehmet Sezgin and Bülent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–166, 2004.
10. Jawad H. AlKhateeb, Jinchang Ren, Stan S. Ipson, and Jianmin Jiang. Knowledge-based baseline detection and optimal thresholding for words segmentation in efficient pre-processing of handwritten Arabic text. In *Fifth International Conference on Information Technology: New Generations (itng 2008)*, pages 1158–1159. IEEE, 2008.
11. Rasty Yaseen and Hossein Hassani. Kurdish optical character recognition. *UKH Journal of Science and Engineering*, 2(1):18–27, 2018.
12. Omar Al-Jarrah, Samer Al-Kiswany, Mohammad Fraiwan, and Hani Khasawneh. A new algorithm for arabic optical character recognition. *WSEAS Transactions on Information Science and Applications*, 3, 2006.