Line, Word, and Character Segmentation from Bangla Handwritten Text—A Precursor Toward Bangla HOCR

Payel Rakshit, Chayan Halder, Subhankar Ghosh and Kaushik Roy

Abstract The basic functionalities of optical character recognition (OCR) are to recognize and extract text to digitally editable text from document images. Apart from this, an OCR has other potentials in document image processing such as in automatic document sorter, writer identification/verification. In current situation, various commercially available OCR systems can be found mostly for Roman script. Development of an unconstrained offline handwritten character recognition system is one of the most challenging tasks for the research community. Things get more complicated when we consider Indic scripts like Bangla which contains more than 280 modified and compound characters along with isolated characters. For recognition of handwritten document, the most convenient way is to segment the text into characters or character parts. So line, word and character level segmentation plays a vital role in the development of such a system. In this paper, a scheme for tri-level segmentation (line, word, and character) is presented. Encouraging segmentation results are achieved on a set of 50 handwritten text documents.

Keywords OCR · Bangla handwritten character recognition · Line segmentation Word segmentation · Character segmentation

P. Rakshit (✉) · C. Halder · S. Ghosh · K. Roy
Department of Computer Science, West Bengal State University,
Barasat, Kolkata 700126, West Bengal, India
e-mail: prmylife20@gmail.com

C. Halder
e-mail: chayan.halderz@gmail.com

S. Ghosh
e-mail: sgcs2005@gmail.com

K. Roy
e-mail: kaushik.mrg@gmail.com

R. Chaki et al. (eds.), *Advanced Computing and Systems for Security*,
Advances in Intelligent Systems and Computing 666,
https://doi.org/10.1007/978-981-10-8180-4_7

1 Introduction

In recent era of digital evaluation, computer-aided document processing (reading/writing) is getting more importance in our day-to-day life. Here, optical character recognizer (OCR) will be utilitarian, if developed properly. Character recognition of printed text document has achieved a great success rate following huge interest by researchers. Commercially available OCR systems like fine reader by ABBYY is one of the prime examples [1]. Various offline handwritten optical character recognition (HOCR) strategies for non-Indic scripts such as English [2], Chinese [3], Japanese [4] are already proposed by different authors but only a few stray works are done on offline Bangla Handwritten character recognition [5–12]. The impediments behind are inconstant variations of human writing style and similarities of distinct character shapes, overlapping and touching of neighboring characters, spatial variation of characters when combined with other characters (modified complex and compound characters), etc. [6]. Hence, an efficient Bangla OCR system needs to be developed for recognition of handwritten text.

In Indian subcontinent, Bangla is the second most popular script after Devanagari [13]. Many research works are already investigated for handwritten character recognition of different Indic scripts. Sahlol et al. proposed an OCR algorithm for recognition of handwritten Arabic characters [14], a deep learning-based large-scale handwritten Devanagari characters recognition scheme was proposed by Acharya et al. [15], Kamble and Hegadi [16] came up with an idea for handwritten Marathi character recognition, whereas Varghese et al. [17] developed a novel recognition tri-stage scheme for recognition of handwritten Malayalam characters. Some works on Bangla isolated alphabets and numerals are also proposed by Rahman et al. [12], Sarkhel et al. [9], and Wen et al. [8]. Halder and Roy [10] suggested a method for Bangla Handwritten character segmentation from words. Das et al. [7] presented a system for handwritten Bangla basic and compound character recognition. A structural composition based Bangla compound characters recognition strategy was provided by Bag et al. [6].

In this proposed work, an attempt is made for development of a tri-level (line, word, and character) segmentation scheme without any normalization. The rest of the paper is organized as follows: Sect. 2 describes the overview of the study, methodology is presented in Sect. 3, whereas results are shown in Sect. 4 and finally, we concluded in Sect. 5.

2 Overview of the Study

A tri-level segmentation (line, word, and character) scheme which is an essential part of an OCR is proposed here in this work. In Fig. 1, a block level overview of an OCR system is depicted. The most enlightened area of this work is, during segmentation, no normalization is performed, i.e., an attempt is made toward developing a HOCR

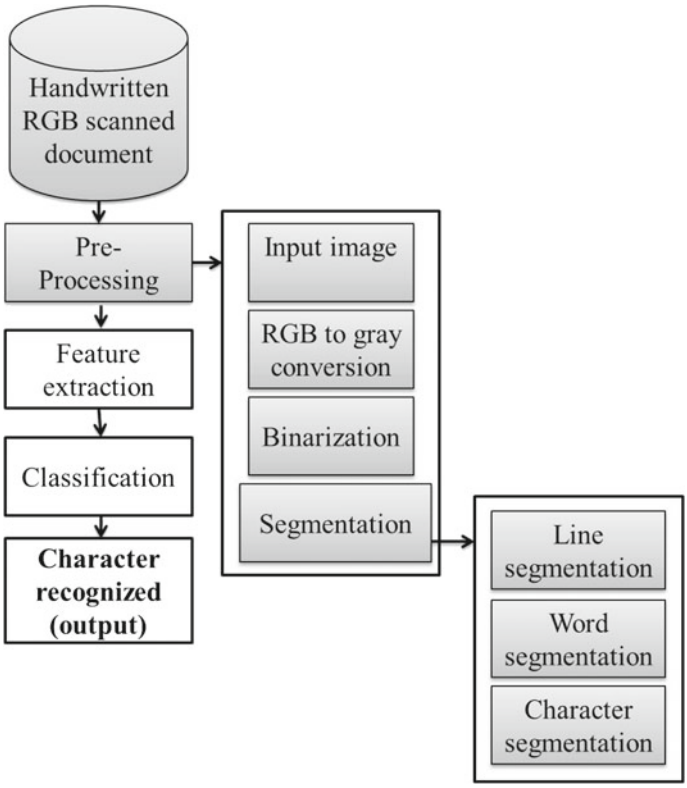

Fig. 1 Block diagram of the proposed HOCR system

without any noise removal and skew/slant correction. In the proposed model, upto preprocessing part is implemented marked as shaded regions in Fig. 1. The remaining stages of feature extraction and classification will be applied on the segmented characters obtained from the current work.

3 Methodology

In this stage, the raw scanned RGB document images are processed to be used for segmentation. Firstly, extraction of only handwritten text is done using a text extraction technique. In this technique, first of all horizontal and vertical boundary lines are detected using histogram localization. After that, these identified boundary lines and all other unnecessary printed texts are deleted to extract only handwritten text part applying minimum bounding box. The extracted grayscale images are then stored in tif format.

3.1 Segmentation

This part consists of a tri-level segmentation as shown in Fig. 1.

3.1.1 Line Segmentation

Line segmentation is the first level of tri-level segmentation. This level itself consists of different sub-levels such as connected component (CC) analysis, Hough transform, filling, and smoothing as shown in Fig. 2. CC analysis is performed to mark word components (WCs) in the binary image. After that, the average word component height (WCh_{avg}) and average WC width and total number of WCs (tot_{wc}) are calculated. Here, we have used a condition ($TH_{min} \leq WC \leq TH_{max}$) to eliminate too large and too small WCs depending on two threshold values TH_{min}, TH_{max}.

Now Hough transform is applied on the binary image to estimate the potential text lines and their start points. To make the segmentation easier, bidirectional horizontal filling and vertical smoothing are performed by segmenting the image into Seg_n (here, $Seg_n = 8$ is considered empirically) vertical segments. Filling operation is used to fill the non-text areas in the image, and smoothing is applied immediately after filling, depending on some certain threshold to fill the small gaps in the filled image. Each transition is marked by two points, namely start transition ($trans_{start}$) and end transition ($trans_{end}$). The total number of transitions in the image is denoted as (t). Average line height (WCh_{avg}) is calculated using Eq. (1). Equation (2) represents calculation of average line gap (LG_{avg}) between two lines.

$$WCh_{avg} = \frac{\sum_{i=0}^{tot_{wc}-1} (WC_{height})_i}{tot_{wc}} \tag{1}$$

where WC_{height} represents height of each word component.

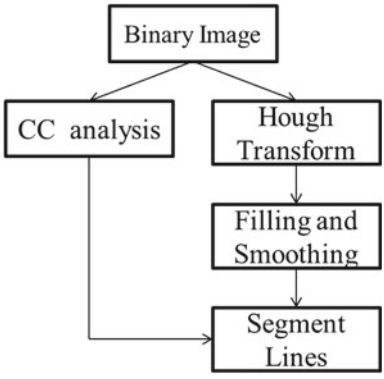

Fig. 2 Steps of line segmentation

$$LG_{avg} = \frac{\sum_{i=0}^{t-1}(trans_{end} - trans_{start})_i}{t} \tag{2}$$

Algorithm 1: Filling of gray image with smoothing

Input : Handwritten gray text image I_gray.
Output: Space filled image I_gray$_{smooth_filled}$.

```
for s ← 0 to Seg_n do
    // Seg_n is the number empirically taken to divide the text
       for filling
    Spoint_seg ← 0 ;
    Epoint_seg ← 0 ;
    for i ← 0 to height do
        // Left to right filling
        diff ← width/Seg_n;
        V ← 0;
        V ← V + (s + diff);
        Spoint_seg ← Epoint_seg ;
        Epoint_seg ← V ;
        Spoint_seg ← for j ← Spoint_seg to Epoint_seg do
            if I_gray(i, j) ≠ Object Pixel then
                I_gray_filled(i,j) ← Fill_color;
            end
        end
        // Right to left filling
        for j ← Epoint_seg to Spoint_seg do
            if I_gray(i, j) ≠ Object Pixel then
                I_gray_filled(i,j) ← Fill_color;
            end
        end
    end
end
// smoothing
for i ← 0 to width do
    for j ← 0 to height do
        for ss ← trans_start to trans_end do
            counter ← counter + 1;
        end
        smt_thresh ← WCh_avg * 20%;
        if count ≥ smt_thresh then
            I_gray_smooth_filled(i,j) ← Fill_color;
        end
    end
end
```

The detected lines are segmented by running a separator through the interline gap between two lines of the smoothed image. Another step is employed to detect and

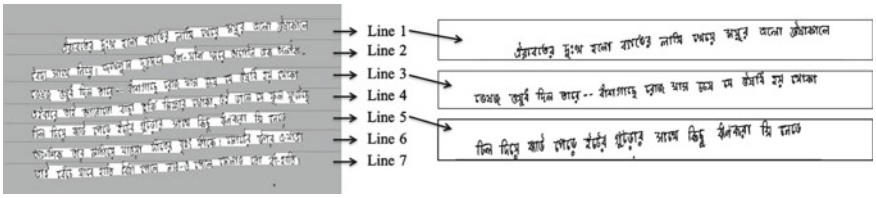

Fig. 3 Lines in a segmented image

separate the text lines that Hough transform failed to detect. The segmentation of the lines is started from their start points, and the separator moves in straight line in forward direction (from left to right) through the filled non-text area between two text lines. At that time, if the separator touches a text pixel (black) or text area (white), a decision making is performed at that point. The separator checks in upward and downward direction through which it can move forward (Fig. 3). If both of the directions are closed, the separator cuts through the text area. After completing this operation, we have a basic set of lines. Now each segmented line height is checked again. If the line height of any segmented line is greater than the threshold LH_{th}, the line is checked again whether there exist multiple lines. Equation (3) is used to calculate LH_{th}. This situation occurs because sometimes Hough transform fails to detect all the lines, and multiple lines are treated as a single line. To segment these lines again, same segmentation procedure is repeated. Finally, we have the segmented set of all lines. A sample segmented image is shown in Fig. 3.

$$LH_{th} = (2 \times (WCh_{avg} + LG_{avg})) \times 0.8 \tag{3}$$

3.1.2 Word Segmentation

In the proposed scheme, word segmentation is performed based on CC analysis. Before applying CC analysis, morphological erosion, and dilation are performed depending on erosion and dilation threshold. In the previous section (line segmentation), CC analysis is already discussed. Here, each WC of a line is treated as a word. As in Bangla script, "matra" or "shirorekha" is mostly used to connect the characters of a word; therefore, words can be recognized easily by identifying the CCs. The CCs of a line are shown using bounding boxes which represent the words in Fig. 4.

3.1.3 Character Segmentation

Zone segmentation is required to segment the characters properly as Bangla words contain character parts (modifiers) in upper and lower zones along with middle zone. In this scheme, busy zone, headline, and baseline information are used to separate

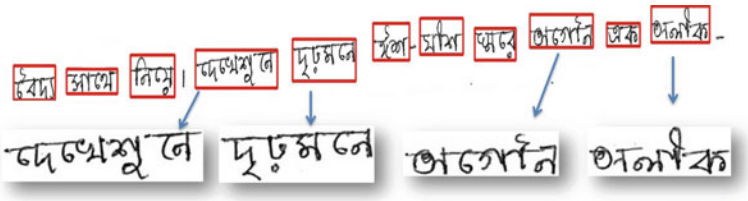

Fig. 4 Segmented words from a line

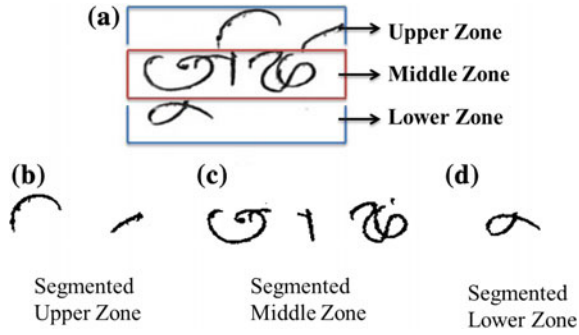

Fig. 5 **a** Original image (**b**) Segmented upper zone (**c**) Segmented middle zone (**d**) Segmented lower zone

three zones. After segmenting the three zones of a word, character segmentation of the base characters of the middle zone is necessary for proper segmentation. Middle zone segmentation is performed by using the interspace between two characters and vertical projection profile method. After identifying the segmentation points, the characters of the middle part are vertically segmented using those segmentation lines. The details about this character segmentation technique can be found in [10] (Fig. 5).

4 Results

The current study of three-level segmentation is employed on a dataset of 50 Bangla unconstrained handwritten text documents from same number of individuals. No restriction is posed on the type of writing instruments used. The dataset contains wide variation of distinct writing style because the writers are of different age, gender, educational qualification. The collected datasets are scanned and stored in 300 dpi RGB mode. Currently, no ground truth data is available, so we have calculated segmentation accuracy manually.

4.1 Performance of Line Segmentation

Table 1 shows line segmentation performance of the current work. Here out of total 482 lines 436 are properly segmented while the others are under segmented. No case of over-segmentation has occurred during the segmentation. An average accuracy of 90.46% was obtained for line segmentation.

4.2 Performance of Word Segmentation

The result achieved for word segmentation is shown in Table 2. Here we have considered total 200 lines out of 436 lines as the manual checking was very time-consuming and difficult to check all the lines manually. Out of 1640 words from 200 lines, 1477 words are properly segmented having accuracy of 90.06%.

4.3 Performance of Zone and Character Segmentation

Here we have considered a set of 500 words for manual checking. The detail result of zone segmentation is shown in Table 3. The accuracies achieved for upper zone, middle zone, and lower zone are 66.56%, 98.80%, and 75.97%, respectively. In Table 4, one can see that out of 3015 characters, 1524 are properly segmented obtaining an average accuracy of 50.55%.

4.4 Error Analysis

In this proposed method, it is found that the line segmentation and word segmentation accuracies are encouraging enough to be used in Bangla OCR. In case of word segmentation, erroneous situations are occurred when word components of a same word

Table 1 Line segmentation result of all the documents

Documents	Total lines	Segmented lines	Accuracy (%)
50	482	436	**90.46**

Table 2 Word segmentation result of segmented lines

Lines	Total words	Segmented words	Accuracy (%)
200	1640	1477	**90.06**

Table 3 Zone segmentation result of segmented words

Words	Total zones of characters			Segmented zones of characters			Accuracy (%)			Average accuracy (%)
	Upper	Middle	Lower	Upper	Middle	Lower	Upper	Middle	Lower	
500	308	500	195	205	494	122	66.56	98.80	62.56	**75.97**

Table 4 Character segmentation result of segmented words

Words	Total characters	Segmented characters	Accuracy (%)
500	3015	1524	**50.55**

Fig. 6 **a**–**b** Some different word examples which causes the errors of character segmentation

are written with significant distance between them which leads to over-segmentation, similarly when two words are touching each other then under segmentation occurs. The character segmentation accuracy is not as high as the existing results due to the fact that segmentation of handwritten text has various challenges that are discussed earlier in Sect. 1. Apart from those, it should also be noted that in this proposed method any kind of correcting measures like skew and slant corrections are not considered. In Fig. 6, erroneous character segmentation of two sample words is presented. In Fig. 6a, the word is multi-oriented creating a multi-level skew with irregularly placed characters at different zones, along with this the characters are also touching which leads to improper character segmentation result. The similar multi-level skew can also be observed in Fig. 6b.

5 Conclusion

In this paper, line, word, and character segmentation of unconstrained handwritten Bangla text, document is presented. The segmentation is one of the most essential and preliminary tasks for many document image processing works. The task is more challenging when it needs to be developed on unconstrained handwriting. The proposed approach of line segmentation is a hybrid approach to improve line segmentation accuracy compared to existing methods. A simple yet effective word segmentation procedure is presented based on connected component analysis. Character segmentation is performed on a relatively large number of words set using an existing zone and character segmentation method. The prime factor of low accuracy for the proposed work is due to existence of skew, slant, touching/overlapping

components, and lack of connections between two consecutive characters of a word. Here, no attempt is made to correct and normalize this natural randomness of writing to make the method simple, so that it becomes easier to be incorporated in most of the OCR systems. The main drawback of this technique lies in word and character segmentation area where it fails to reduce over-segmentation.

In future, to deal with these touching and overlapping writing we will introduce combination of different approaches which will improve the segmentation accuracy. Further, these segmented characters will be used for feature extraction and classification to observe character recognition results in recent future. We have planned to use chain code-based features, histogram gradient-based features, and local binary pattern (LBP) for feature extraction. Different renowned classifiers like MLP, SVM will be tested in future along with combination of classifiers.

Acknowledgements Two of the authors, Ms. Payel Rakshit and Mr. Chayan Halder, are thankful to Department of Science and Technology (DST) for their support as INSPIRE fellowship.

References

1. https://www.abbyy.com/ . Last accessed 07 Dec 2016
2. Rakshit, S., Basu, S.: Recognition of handwritten Roman script using Tesseract open source OCR engine. In: National Conference on (NAQC), pp. 141–145 (2008)
3. Tsukumo, J., Tanaka, H.: Classification of handprinted Chinese characters using nonlinear normalization methods. In: 9th International Conference on Pattern Recognition, pp. 168–171 (1988)
4. Yamada, H., Yamamoto, K., Saito, T.: A non-linear normalization method for handprinted Kanji character recognition line density equalization. Pattern Recognit. **23**, 1023–1029 (1990)
5. Bhunia, A.K., Das, A., Roy, P.P., Pal, U.: A comparative study of features for handwritten Bangla text recognition. In: 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 636–640 (2015)
6. Bag, S., Harit, G., Bhowmick, P.: Recognition of Bangla compound characters using structural decomposition. Pattern Recognit. **47**, 1187–1201 (2013)
7. Das, N., Das, B., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M.: Handwritten Bangla basic and compound character recognition using MLP and SVM classifier. J. Comput. **2**, 109–115 (2010)
8. Wen, Y., Lu, Y., Shi, P.F.: Handwritten Bangla numeral recognition system and its applicaiton to postal automation. Pattern Recognit. **40**, 99–107 (2007)
9. Sarkhel, R., Das, N., Saha, A.K., Nasipuri, M.: A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition. Pattern Recognit. **58**, 172–189 (2016)
10. Halder, C., Roy, K.: Word & character segmentation for Bangla handwriting analysis & recognition. In: 3rd National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, pp. 243–246 (2011)
11. Maitra, D.S., Bhattacharya, U., Parui, S.K.: CNN based common approach to handwritten character recognition of multiple scripts. In: 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 1021–1025 (2015)
12. Rahman, M.M., Akhand, M.A.H., Islam, S., Shill, P.C., Rahman, M.M.H.: Bangla handwritten character recognition using convolutional neural network. Int. J. Image Graph. Signal Process. (IJIGSP) **7**, 42–49 (2015)

13. Halder, C., Obaidullah, S.M., Roy, K.: Effect of writer information on Bangla handwritten character recognition. In: 5th National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp. 1–4 (2015)
14. Sahlol, A.T., Suen, C.Y., Elbasyouni, M.R., Sallam, A.A.: A proposed OCR algorithm for the recognition of handwritten Arabic characters. J. Pattern Recognit. Intell. Syst. **2**, 8–22 (2014)
15. Acharya, S., Pant, A.K., Gyawali, P.K.: Deep learning based large scale handwritten Devanagari character recognition. In: 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 1–6 (2015)
16. Kamble, M., Hegadi, S.: Handwritten Marathi character recognition using R-HOG feature. In: International Conference on Advanced Computing Technologies and Applications (ICACTA), Procedia Computer Science, vol. 45, pp. 266–274 (2015)
17. Varghese, K.S., Jamesa, A., Chandran, S.: A novel tri-stage recognition scheme for handwritten Malayalam character recognition. In: International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST), Pattern Recognition, vol. 24, pp. 1333–1340 (2016)
18. Htike, T., Thein, Y.: Handwritten character recognition using competitive neural trees. Int. J. Eng. Technol. **5**, 352 (2013)