An Efficient Line Segmentation Approach for Handwritten Bangla Document Image

K. Mullick and S. Banerjee
Department of Computer Science and Engineering
Heritage Institute of Technology
Kolkata, India
Email: koustav.mullick@yahoo.com, sudiz.7052@gmail.com

U. Bhattacharya
Computer Vision & Pattern Recognition Unit
Indian Statistical Institute
Kolkata - 108, India
Email: ujjwal@isical.ac.in

***Abstract*—Text line segmentation plays a vital role in the overall performance of a document recognition system. In the literature, similar segmentation works for offline handwritten Bangla documents are rarely found. On the other hand, certain peculiarities of handwritten Bangla script such as widespread occurrences of ascenders and descenders or some of its characters appearing only as an ascender or descender often cause unique difficulties to this segmentation task. Existence of connected components over a number of successive text lines is a common phenomenon in unconstrained handwritten Bangla documents. In this article, we propose a novel and efficient approach for text line segmentation where initially, we smudge the input document image, to blur-out white spaces between words, while preserving gaps between consecutive lines. Next, we obtain an initial segmentation scheme by shredding the image based on the white most pixels in between consecutive smudged lines. Multi-line connected components have been taken care of by thinning, and then finding the most probable point of separation in the component. Combining it with the initial segmentation, we obtain the final output. The proposed approach has been evaluated on ICDAR 2013 Handwriting Segmentation Contest dataset of Bangla. The segmentation results show the efficiency of the proposed approach.**

Index Terms – Line segmentation, Handwritten document segmentation, Handwritten Bangla document image

I. Introduction

Text line segmentation and extraction is one of the most important steps and plays a vital role towards the overall performance of any handwritten document recognition system. The main aim of handwriting segmentation is to segment a given document into distinct text lines to reveal the one dimensional natural reading sequence. And also since it is in the beginning of a pipeline of processing, it is very important to keep the errors as low as possible in this step.

While this can very easily be done in case of modern printed documents, it is not so for handwritten texts. Some of the challenges that need to be overcome in order to achieve good accuracy are as follows: skewed lines, curvilinear lines, fluctuating lines, touching and overlapping components, irregular geometrical properties in words or letters between consecutive lines, leftmost starting position, distance between consecutive lines and line height and width.

Rest of this article is organized as follows: Section II briefly describes the existing approaches towards the problem, Section III describes the proposed approach, Section IV presents the experimental results obtained using our approach, Section V concludes the present report and discusses the scope for future works.

II. Previous Works

Unlike printed documents, handwritten documents comes with, much more challenges due to its' complex local structures. Segmentation of such documents is still an open field of research even though there are several methods for it. Roughly they can be categorized into i) horizontal projection based, ii) Hough transform based, iii) bottom-up approach, iv) thinning based and v) smearing based.

Global horizontal projection analysis of black pixels has been utilized in [1], [2]. Piece-wise horizontal projection analysis of black pixels instead of the global projection technique was also employed in several works [3], [4]. Horizontal projections can't deal with skewed, curved and fluctuating lines. Hough transform is employed in the field of document analysis for many purposes such as skew detection, line detection, slant detection and text-line segmentation [5]. A block-based Hough transform as a modification of the conventional Hough transform is used in [6], [7]. A bottom-up approach in which pixels are connected to their closed ones based on geometrical criteria to form text lines is another method [8]. A thinning algorithm followed by some post-processing operations is employed for the entire background region of an input text image to detect the separating borderlines and segment Japanese text documents in [9]. Smearing is another widely used technique for text line segmentation [10], [11] in which white gaps between words in each text line is filled with black pixels to obtain large bodies of black pixels. They are bad at handling connected or overlapping components. Smearing technique combined with a generalized adaptive local connectivity map (ALCM) using steerable directional filter is used in [12].

III. Proposed Approach

In this article, we propose a novel approach for text line segmentation from Bangla handwritten documents. A block diagram of the proposed method is shown in Fig. 1. Our approach is based on the idea that segmentation of the individual lines of a handwritten document becomes easier if

978-1-4799-7458-0/15/$31.00

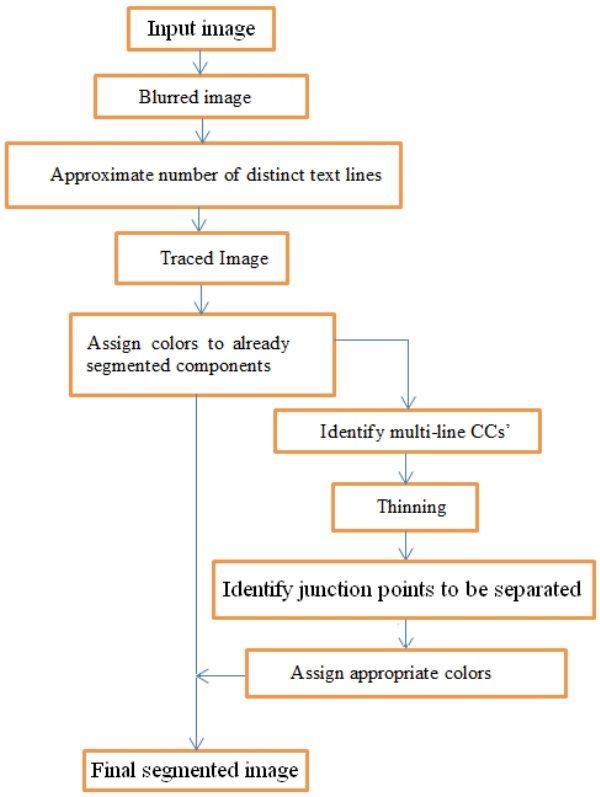
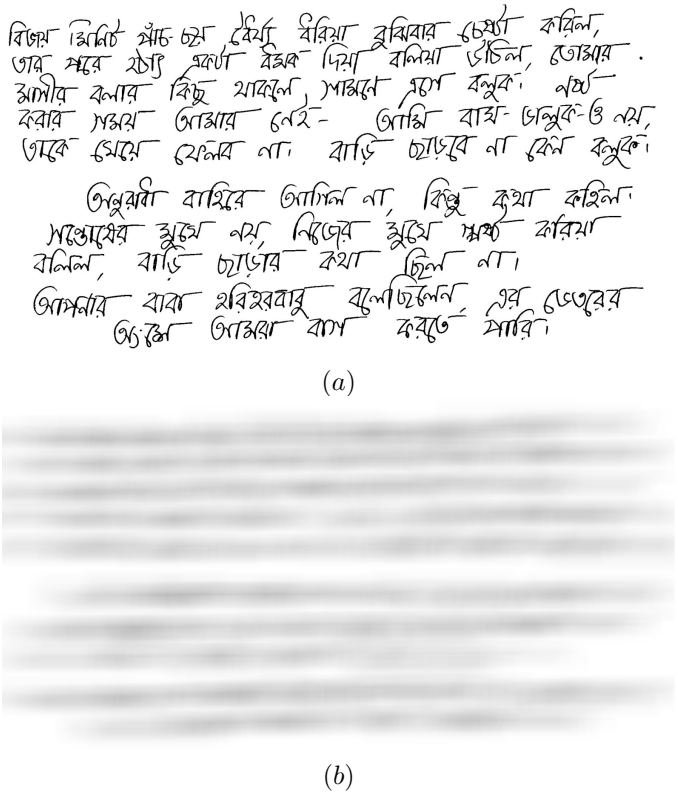


Fig. 1. A block diagram of the proposed method



(a)

(b)

Fig. 2. (a) A sample Bangla handwritten document, (b) the document in (a) is blurred

we can have prior estimation of the number of distinct lines present in the text document. Towards obtaining this estimate, the input image is first blurred and next this blurred image is scanned along vertical columns from the top, at regular intervals to look for continuous change in intensity values in the blurred image which provides useful information of change in text lines. It enables us to obtain a list of points suggesting the point of separation between two consecutive lines. Advantage of this is that unlike [13] we need not draw continuous 'tracers' recursively to find the best separation between two lines. Instead in one run only, using the above identified points, we could find a 'tracer' giving the best separation. Thus we have an initial segmentation but there still exists connected components spanning to multiple lines which need to be separated. For this we employ a very simple approach of thinning the components and finding the best junction point to divide the component. Our results show how effective this simple approach is and we obtain the output where each distinct line is colored using a different color. Further details of the approach are described in the following subsections.

A. *Preprocessing*

The input image I is a binary image and we consider its pixel values as follows:

$$f(x,y) = \begin{cases} 1, & \text{if } (x,y) \text{ is a foreground pixel} \\ 0, & \text{if } (x,y) \text{ is a background pixel} \end{cases}$$

We first obtain all the connected components in the input image I and collect their individual heights. The simple arithmetic average of these heights is called the letter height and is denoted by LH. A mask similar to the one used in [13] is also considered by us to blur the input image I and obtain the blurred image B as follows:

$$B(x,y) = \sum_{i=-\frac{BW}{2}}^{i=+\frac{BW}{2}} \sum_{j=-\frac{BH}{2}}^{j=+\frac{BH}{2}} I(x+i, y+j),$$

where $BW = 4 \times LH$ and $BH = 0.4 \times LH$ are respectively height and width of the blurring mask. The intent of setting such a value to BW is to blur-out white spaces between two consecutive words in a line and similarly BH is set to blur-out letters in a line preserving white gaps between consecutive lines. In Fig. 2 (b), the grayscale smudged image $B(x,y)$ corresponding to the input binary image $I(x,y)$ of Fig. 2 (a) is shown. Next, we proceed to the following section to obtain an estimate of the number of lines in the document.

B. *Determination of the Number of Distinct Text Lines*

Usually, individual lines of a handwritten document do not either start or end at fixed column positions. The amount of indentations at the starting of a paragraph, title, author's name or the ending column of a paragraph vary widely as shown in Fig. 3. The estimation of distinct lines in a handwritten text document should take care of the above fact. This is, in fact, efficiently tackled with the help of the smudged image (shown in Fig. 2 (b)) as it was obtained in Section III-A. The blurred image $B(x,y)$ has a smooth vertical transition from lighter to darker pixels whenever a new line is encountered

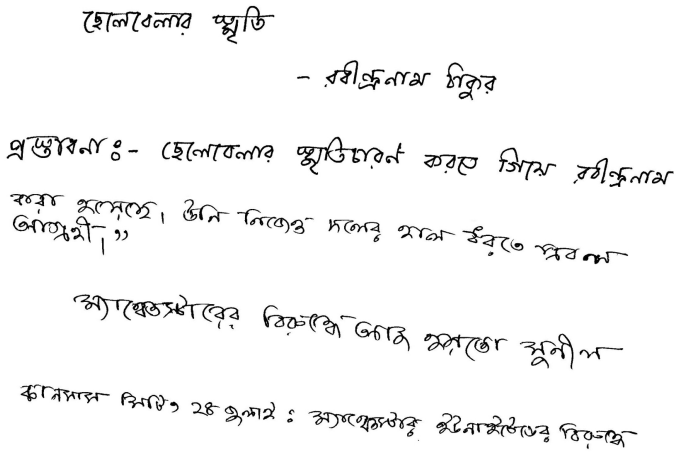


Fig. 3. Examples of variation in the length of text lines

and a similar vertical transition from darker to lighter pixels as the lower rows of the line are approached. We take advantage of this pattern of change in intensity value along the vertical direction to estimate an approximate number of distinct lines present in a handwritten document.

We draw vertical lines at regular intervals starting from the topmost pixel of $B(x, y)$, leaving an offset at the left (since, it occurs only rarely that someone starts writing from the leftmost column of the paper). For each vertical line we check if there is a steady decrease in the consecutive pixel values and then again a gradual increase. We can say that we have encountered a new line after reaching a certain threshold which is calculated from the average height of the components of the image. The algorithm for obtaining the above is described below.

Algorithm

offset =15, $max = 0$, $threshold = \frac{LH}{2}$, $LineBoundary[] = null$

```
for : j = offset to image_width step offset
 flag = 0, W2B = 0, B2W = 0, LineCount = 0, temp[] = null
    for : i = 1 to image_height
        if B(i, j) < B(i − 1, j) and flag = 0
            W2B ++;
        else if B(i, j) > B(i − 1, j) and flag = 0
            if W2B > threshold
                flag = 1;
            W2B = 0;
        else if B(i, j) > B(i − 1, j) and flag = 1
            B2W + +;
        else if B(i, j) < B(i − 1, j) and flag = 1
            if B2W > threshold
                        // new line identified
                temp[LineCount] = i
                LineCount ++;
                flag = 0
            B2W = 0
    end for
    if LineCount > max
        max = LineCount
        LineBoundary = temp
        MaxColumn = j
    end for
return max, LineBoundary and MaxColumn
```

Once the number of lines in the document and separating boundary information of each such line get captured by max and $LineBoundary$ respectively, we can obtain an estimate of the width of respective lines by checking two successive background pixels above and below the line.

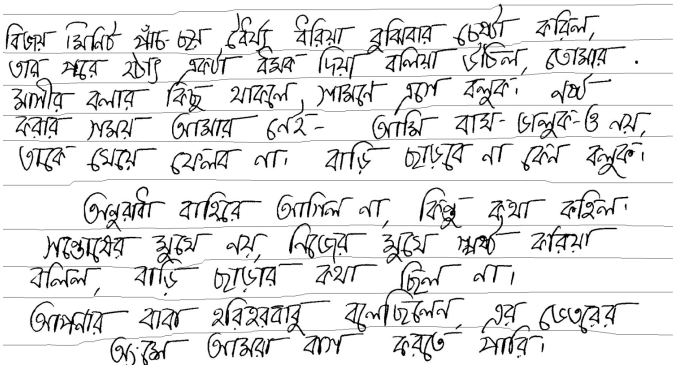


Fig. 4. Distinct lines of the document image in Fig. 2 (a) are identified

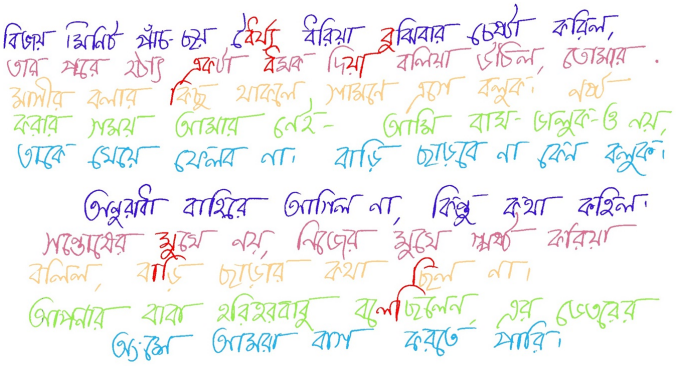


Fig. 5. Characters in distinct lines of the document image in Fig. 2 (a) are shown by different colors. Connected characters (components) across the lines are identified and shown by red color

C. Image Shredding

We now shred the image into individual regions so that each such region represents a single horizontal (possibly curved) strip of text. From the previous step (described in Section III-B) we already have certain information stored in $LineBoundary$ for each line in the document. For each entry in $LineBoundary$ we employ two 'tracers', one to the left and the other one to the right to trace the probable boundaries of each line. The result of identification of individual lines of the document image in Fig. 2 (a) is shown in Fig. 4.

Once we obtain the horizontal strips consisting of individual lines in the document image, next we need to identify the characters / words in each individual line and this is done by color filling each of them by distinct colors for different lines as described in next section.

D. Color Filling

This is in general a simple step, once we have separated the distinct lines with distinct boundaries. But that is not the case because of two reasons. First, certain characters may overlap with the boundary, then how to decide whether it belongs to the line above the boundary or below it. Secondly there may be considerable overlap among characters in consecutive lines. Then it is important to decide where from to separate the component and assign them correctly to their respective lines. We solve the first problem using a very simple heuristic. We calculate the upward and downward extension of the

Fig. 6. A few connected components across multiple lines present in the output image of Fig. 6 (a) are shown

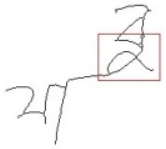


Fig. 7. A rectangular area of the thinned image is shown and the junction points belonging to this rectangular region are verified for prospective segmentation point

overlapping component from the point of overlap with the boundary. If it is more downwards, we assign it to the next line below it, else to the above line. If the difference in extension to either side is not considerable we assign it to the one having darker pixels (in the smudged image) in the nearby region. For the rest of the components which lie well within the margins, we simply assign it to the line it belongs to, since there are no overlaps with the boundary. We have not addressed the problem of components extending to multiple lines and Fig. 5 shows the result obtained. Marked in red are the components which are identified as extending to multiple lines and we try to address this in the next section.

E. Segmentation of Connected Components Across Multiple Lines

The output of the previous stage described in Section III-D usually consists of many components which are connected across multiple consecutive lines. A few examples of such connected components are shown in Fig. 6. Existence of such connected characters is perhaps more frequent in handwritten documents of Indian scripts such as Bangla due to the particular nature of these scripts which is responsible for existence of many characters or their parts occurring above or below the core region. Since we focus only on Bangla documents, the present approach for segmentation of these components is designed based on the properties of these Indian scripts. However, slight modifications of the present scheme should work for other scripts as well. The proposed approach for this segmentation task is based on estimation of the touching point. The identification of the touching point is based on identification of the junction point lying somewhere near the middle of the connected components and separate it accordingly, we could get an acceptable result. The first hurdle is to identify all the junction points. For this we obtain the one pixel thinned versions of these components from which the junction points are identified in a trivial manner.

Once all the junction points in such a component are identified, finding the one lying nearest to the center of the component is trivial. We simply segment the component

Fig. 8. Connected components of Fig. 6 are segmented

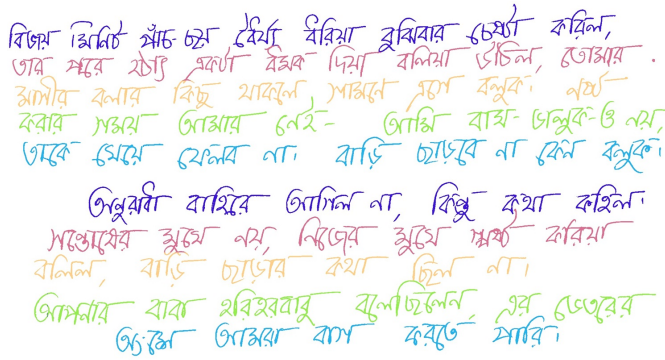


Fig. 9. Final segmentation output of the document of Fig. 2 (a) is shown

at such a junction point and the pixels lying above this segmentation point belong to the upper line and the pixels below it belong to the lower line. If the component belongs to more than two lines, the same approach can efficiently identify the parts of component lying in each individual line. In Fig. 7, the suspect area of the thinned image is shown by marking a rectangle. The junction points inside this rectangle are verified further to decide the segmentation point. In Fig. 8, the connected components of Fig. 6 are shown after their segmentation. In Fig. 9, the final output of the document image of Fig. 2 (a) is shown.

IV. Experimental Results

We used Bangla handwritten document image samples of the ICDAR 2013 handwriting segmentation contest for experimentation of the proposed algorithm. This dataset consists of English, Greek and Bangla handwritten documents from which we used only Bangla document images for the present study. There are 100 such document images of Bangla. For evaluation of the implementation of the proposed approach, we used the 'Viewer and Evaluation Software' provided in the Training Toolkit [13], [14] of the above contest. In the evaluation software, 'MC_thres', the acceptance threshold used to define a one-to-one (O2O) matching, is set to 95% for text line segmentation. It means that if a method detects 95% pixels of a word, it is considered to be detected correctly. This evaluation procedure involves the following quantities: N = count of ground-truth elements (text lines), M = count of text lines that have been detected automatically, RA = Recognition Accuracy, DR = Detection Rate, FM = the performance metric F-Measure which is obtained by using the formula

$$\text{FM} = \frac{2 \times \text{DR} \times \text{RA}}{\text{DR} + \text{RA}}$$

TABLE I. EVALUATION RESULT ON ICDAR 2013 HANDWRITING SEGMENTATION CONTEST DATASET

N	M	DR	RA	FM	O2O
1653	1668	93.16%	92.32%	92.74%	1540

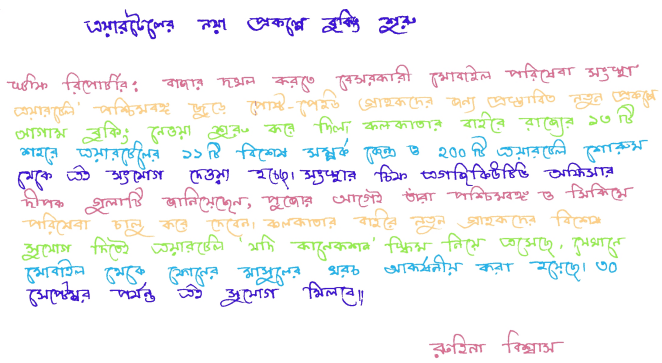


Fig. 10. Outputs of the proposed method on another sample of ICDAR 2013 Handwriting Segmentation Contest dataset are shown

Table 1 summarizes the simulation results provided by the proposed method. It is evident that the performance of the proposed segmentation approach is comparable to any state of the art text line segmentation algorithm.

Segmented document images obtained by the present implementation corresponding to a few samples of ICDAR 2013 Handwriting Segmentation Contest dataset are shown in Figs. 10 and 11.

V. CONCLUSIONS

The proposed line segmentation approach can efficiently handle unconstrained handwritten Bangla document images. It is capable of taking care of situations where touching of characters occur across multiple lines. Also, if the document image has multiple skews or the text lines are separated by widely varying gaps, the proposed approach does not usually suffer due to them. These claims are validated by the segmentation results shown in Figs. 10, 11. However, the proposed line segmentation approach has the following two shortcomings. Firstly if the lines of a document have significant overlap along with improper alignments, then if the gap between two successive words of a line is comparable to the gap between its line and the adjacent lines, then the 'tracer' may pass through the white-most space in such a way that parts of two successive lines may not be segmented as it happened for the sample in Fig. 12 (a). The corresponding 'tracer' line, due to such multiple unfavorable factors failed to follow the desirable horizontal path and could not separate parts of these two lines as it can be seen in Fig. 12 (b). The other drawback is, since our method is dependent on the estimation of the number of lines in the beginning, if estimation of this number is wrong, line segmentation may fail considerably. Also, if there is large difference between the extents of two lines, it may not capture them as separate lines and hence shred the image incorrectly as shown in Fig. 13. In the sample document image, there is a large difference between the lengths of the last line and the line just before this last line. In fact, the last line started near the right margin of the document. The proposed approach often may fail in such an extreme situation. Thus we

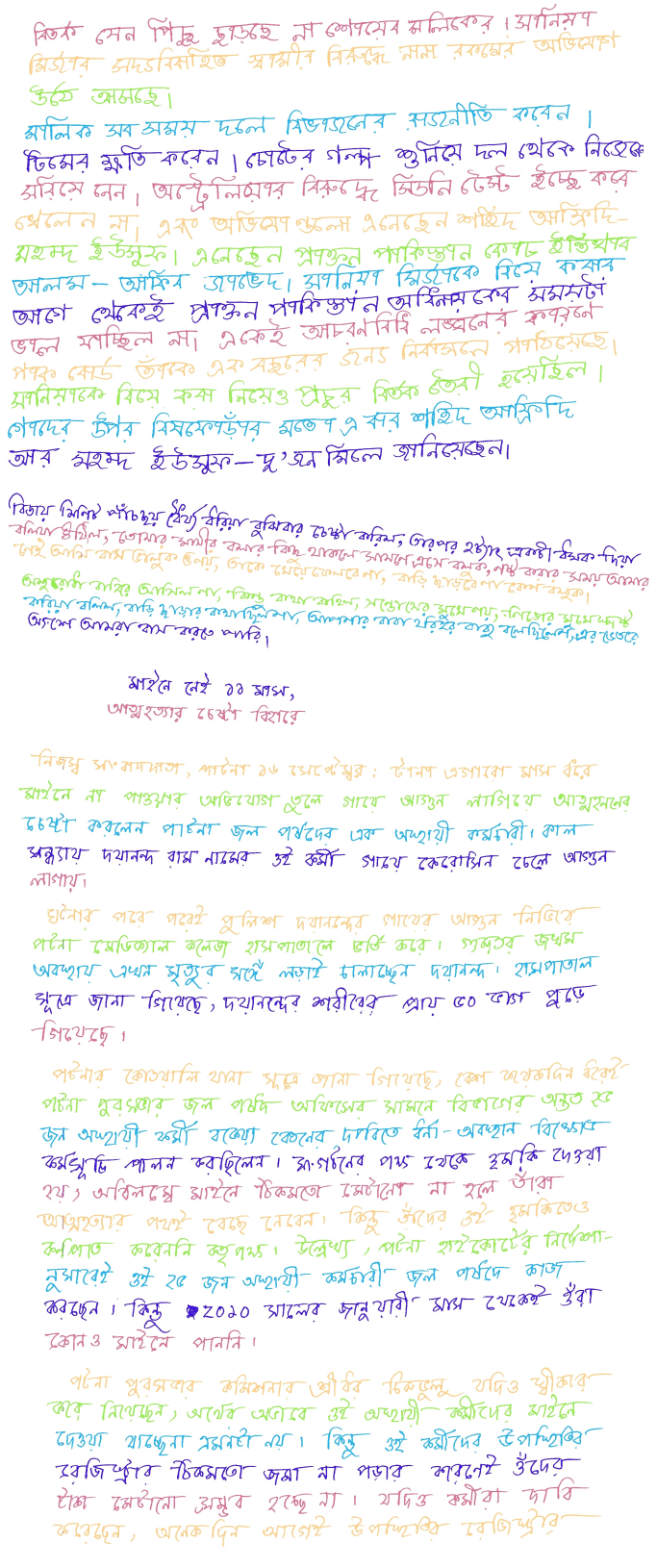
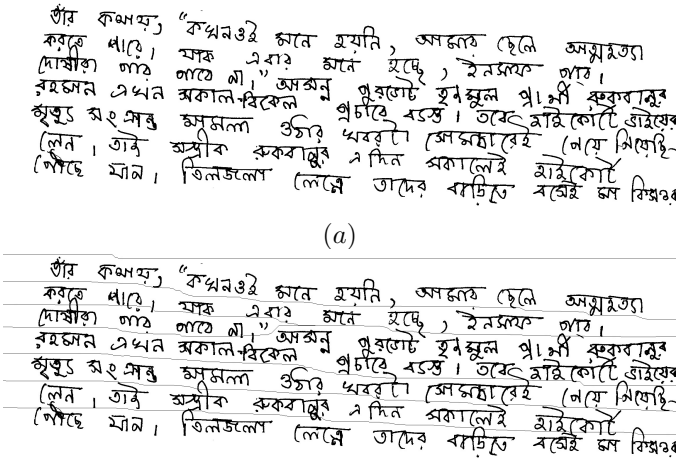


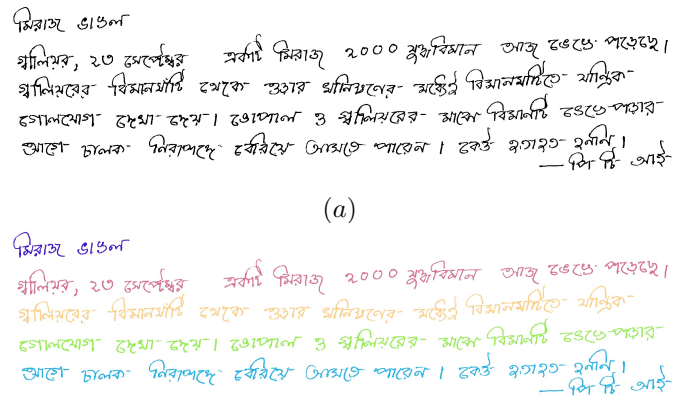
Fig. 11. Outputs of the proposed method on several other samples of ICDAR 2013 Handwriting Segmentation Contest dataset are shown



(a)

(b)

Fig. 12. A sample image in which the proposed method segmented the lines wrongly



(a)

(b)

Fig. 13. A sample image in which the proposed method failed to segment two consecutive lines

obtained inaccurate shredding of lines for the said sample as shown in Fig. 13 (b), where it fails to segment the last line from its immediately preceding line.

In future, we would like to address the above shortcomings of this segmentation approach and extend this work to include segmentation at word level as well. We shall also simulate this approach on handwritten documents of other Indian scripts and study for its necessary adaptations.

ACKNOWLEDGEMENT

We acknowledge the tireless help of Dr. Nikolaos Stamatopoulos of National Center for Scientific Research "Demokritos", Greece for solving all of our problems regarding usage of the evaluation software.

REFERENCES

[1] M. R. Hashemi, O. Fatemi, R. Safavi, Persian Cursive Script Recognition, in: Proceedings of the Third International Conference on Document Analysis and Recognition, 1995, pp. 869-873.

[2] R. Manmatha, J.L. Rothfeder, A scale space approach for automatically segmenting words from historical handwritten documents, IEEE Transactions on Pattern Analysis and Machine Intelligent 27 (8) (2005) 1212-1225.

[3] A.Zahour, L. Likforman-Sulem, W. Boussalaa, B. Taconet, Text-line segmentation of historical Arabic documents, *Proc. of 7th Int. Conf. on Doc. Anal. and Recog.*, 2007, pp. 138-142.

[4] U. Pal, S. Datta, Segmentation of Bangla unconstrained handwritten text, *Proc. of 7th Int. Conf. on Doc. Anal. and Recog.*, 2003, pp. 1128-1132.

[5] L. Likforman-Sulem, A. Hanimyan, C. Faure, "A Hough Based Algorithm for Extracting Text Lines in Handwritten Documents", *Proc. of 7th Int. Conf. on Doc. Anal. and Recog.*, pp. 774-777, 1995.

[6] G. Louloudis, B. Gatos, I. Pratikakis, K. Halatsis, "A block-based Hough transform mapping for text line detection in handwritten documents", *Proc. of 10th Int. Workshop on Frontiers in Handwriting Recognition*, pp. 515-520, 2006.

[7] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, "Text-line detection in hand-written documents", *Pattern Recognition*, vol. 41, pp. 3758-3772, 2008.

[8] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text Line Segmentation of Historical Documents: a Survey", *Int. Journ. Doc. Anal. Recog.* vol 9(2), pp. 123 - 138, Springer, 2007.

[9] S. Tsuruoka, Y. Adachi, T. Yoshikawa, "The segmentation of a text-line for a handwritten unconstrained document using thinning algorithm", *Proc. of 7th Int. Workshop on Frontiers in Handwriting Recognition*, pp. 505-510, 2000.

[10] F. M. Wahl, K.Y. Wong, R.G. Casey, "Block segmentation and text extraction in mixed text/image documents", *Computer Graphics and image Processing*, vol. 20, pp. 375-390, 1982.

[11] Z. Shi, V. Govindaraju, "Line separation for complex document images using fuzzy runlength, *Proc of 1st International Workshop on Document Image Analysis for Libraries*, pp. 306–, 2004.

[12] Z. Shi, S. Setlur, and V. Govindaraju, Text extraction from gray scale historical document images using adaptive local connectivity map, *Proc. of the 8th Int. Conf. on Doc. Anal. and Recog.*, vol. 2, pp. 794-798, 2005.

[13] A. Nicolaou, B. Gatos, "Handwritten Text Line Segmentation by Shredding Text into lines," *Proc. of the 10th Int. Conf. on Document Analysis and Recognition*, pp.626-630, 2009.

[14] N. Stamatopoulos, G. Louloudis, B. Gatos, U. Pal and A. Alaei, ICDAR2013 Handwriting Segmentation Contest, Proc. of the 12th Int, Conf. on Doc. Anal. and Recog., pp. 1402-1406, 2013.

[15] I. Phillips and A. Chhabra, "Empirical Performance Evaluation of Graphics Recognition Systems," *IEEE Transaction of Pattern Analysis and Machine Intelligence*, vol. 21, No. 9, pp. 849-870, 1999.