Neural net based complete character recognition scheme for Bangla printed text books

SK Alamgir Hossain
Computer Science and Engineering Discipline,
Khulna University, Khulna, Bangladesh
alamgir@cseku.ac.bd

Tamanna Tabassum
CFS Limited
Khulna, Bangladesh
tamanna@thecodeandfix.com

Abstract
In this paper we propose a neural net based characters recognition scheme for Bangla printed text books. There are a lot of scientific literature, novels, magazines and books etc that are written in Bangla language. More than 400 million people use Bangla language. Most of the library and educational institutions want to keep copy of the books in a digital format. For storing those books in digital text format we need a good character recognition schemes by which we can convert the scanned text book images to editable texts. The key contribution of our research highlights this issue. We propose four main stages namely preprocessing, segmentation, training-recognition and post-processing. In the beginning the input book images preprocessed by rotation, scaling, binarization and noise elimination. The binarized image is then segmented and extracted into individual characters that are trained and recognized by an artificial neural network. Finally, the process ends by reconstructing the text in the post processing stage.

Keywords - Binary Image, Boundary Fill, Character Recognition, Neural Network, OCR

I. INTRODUCTION

The significance of optical character recognition (OCR) [7] increases day by day. Manual conversion of documents is too slow and expensive. Many commercial companies have developed OCR systems for English, Japanese, Chinese and other European languages. OCR in Bangla language is comparatively new and very few research works are available in this era. OCR in Bangla language is more complex than other languages as Bangla has a large number of single characters as well as compound characters. Table I shows the Bangla alphabet characters. Two or more of this alphabet characters may combine to form complex shaped characters called conjunctive characters. Each character requires different level of detection mechanism. Detection of simple characters such as ক খ গ ঘ ঙ চ ছ is easier than the detection of characters such as ট ঠ ই ি ী ু. In this paper we present a method of OCR system which can work efficiently specially for those books which are written in printed text. The main contribution of our method is its rotation invariant capability. Our method also performs well for recognizing conjunctive characters.

TABLE I: BANGLA ALPHABET CHARACTERS

Alphabet consonants	ক খ গ ঘ ঙ চ ছ জ ঝ ঞ ট ঠ ড ঢ ণ ত থ দ ধ ন প ফ ব ভ ম য র ল শ ষ স হ ড় ঢ় ক্ষ য় ৎ ং ঃ
Alphabet vowels	অ আ ই ঈ উ ঊ ঋ এ ঐ ও ঔ
The vowel marks	া ি ী ু ূ ৃ ে ৈ ো ৌ ঁ

In our proposed method, first the scanned grayscale image is converted into black & white image, afterwards we perform rotation and noise removal operations. These preprocessing steps are described in Section IV. The segmentation scheme is presented in Section V. The segmented characters are trained and recognized by artificial neural network (ANN); this process is described in Section VI. Further, in Section VII we describe the methodology by which the recognized characters are combining to rebuild the original text. We performed some experimental study to show the performance of our scheme in section VIII.

II. RELATED WORKS

A.R. Forkan et al. [1] describe Bangla OCR system only of conjunctive characters. In their work they suggested a flexible matching between sample data and training data components applying Multi-layered Feed-Forward ANN. The accuracy level of some special fonts such as RinkiyMJ (22pt) shows result of 87.56% recognition. They do not consider complex regular characters such as ট ঠ ই ি ী ু. A. Dutta et al. [2] describe both isolated handwritten and printed OCR system for Bangla. A detail description of the characteristics of Bangla text is discussed. They use a curvature related feature for characterizing the strokes which constitute the characters. One main limitation of the scheme is with their method; it is not possible to recognize composite characters.

S.M. Shoeb Shatil et al. [3] describes segmentation-free optical character recognition system for Bangla using a Kohonen network. The methodology presented in this paper assumes that the OCR preprocessor has minimally segmented the input words into easily segmentable chunks, and presenting each of these as an image to the classification engine. The main problem of this work is that the recognition process is not rotation invariant. Md. Abul Hasnat et al. [4] presented a complete Bangla OCR for domain specific document images. The performance level for some domain is not satisfactory. M.A Sattar et al. [6] only discussed the segmentation stage. They do not mention any information about how to apply the segmented characters in an actual recognition process.

III. BANGLA OCR STEPS

In our proposed characters recognition approach (Fig.1.) we divided the whole system into a number of steps as the following:

a. Printed book images are taken directly from the scanner in grayscale mode. The gray scale image is then converted into binarized image.
b. The scanned image may rotate so the next step is to rotate the binarized image into its correct position.
c. If the image is not in the proper size, it will be resized through scaling. The next step is to make the image noise free in the noise elimination stage.
d. A next important step is the segmentation step which is divided into two parts: we segment the lines into words and the words into characters.
e. After segmenting the image into characters, we pass those gained to the training and recognition stage. In the training stage, we train the neural network using randomized data. After the training process is complete, actual character recognition process is carried out. The neural net will update its internal weights in case errors are found in the gained output.
f. In the final steps the recognized characters are reconstructed to the actual text format.

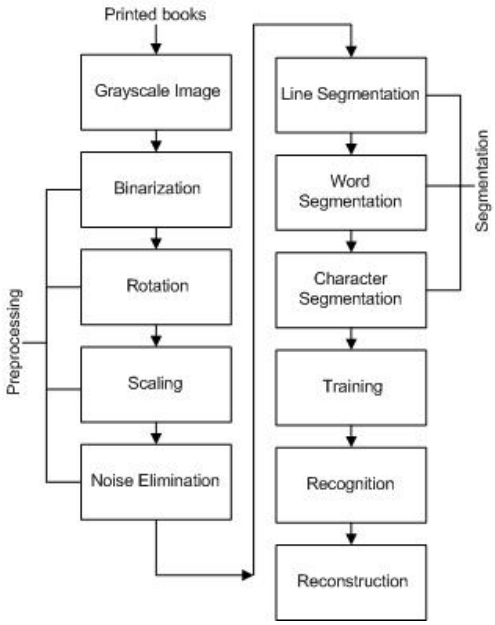

Fig. 1. Proposed Bangla OCR steps

IV. PRE-PROCESSING

In the first step a Bangla printed book's text image is obtained from a flat bed scanner in grayscale mode with 300dpi resolution. The scanned images may have some noise or be scanned in different sizes. So we need to preprocess the scanned images before sending them to the segmentation stage. Preprocessing stage is divided into four functions namely binarization, rotation, scaling and noise elimination.

A. Binarization

The scanned gray scale image is converted to a black & white image. In the grayscale image the pixel values are in the range from 0 to 255. In the black & white image each pixel has only one value either 0 or 1. A black pixel has value 0 and a white pixel has value 1. In Fig.2 the histogram of a gray scale image and its corresponding black & white image is shown.

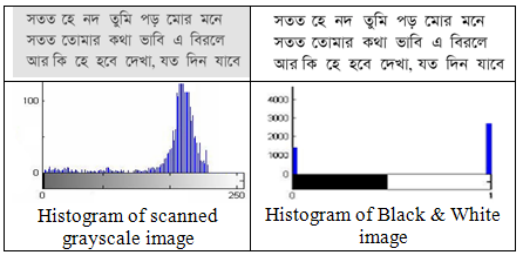

Fig. 2. Histogram of grayscale image (values from 0 to 255) converted Black/White image (two values: 0 and 1)

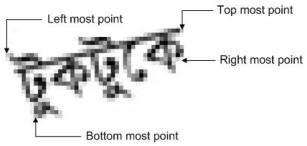

Fig. 3. Rotated image and its different points.

B. Rotation

Sometimes the scanned images may rotate due to the position of the book on the scanner. Most of the Bangla OCR fail to recognize the characters if the input image is rotated. In our proposed method before segmenting the images, we rotate it to its correct position. The rotation point of each word is depicted in Fig.3. The main challenge in this operation is to measure the exact degree of information by which the image needs to be rotated to its actual position. We have devised the following algorithm in order to perform proper rotation of the image.

Algorithm ROTATE_IMAGE (*img: IMAGE*)
*/*gets the input image and returns the number of degree that needs to be rotated so that the image comes to its correct position. leftPoint, rightPoint, topPoint, botPoint are local variables which represents the left, right, top and bottom points. LD, RD, dx, dy are local variables (See Fig. 3)*/*

1. **for** every column (*col)* **from** leftPoint **to** rightPoint
2. **if** image position (topPoint, col) is 0 **then**
3. Set *LD* to (col – leftPoint)
4. Set *RD* to (rightPoint – col)
5. **end if**
6. **endfor**
7. **for every** *row* **from** topPoint **to** botPoint
8. **if** LD greater than RD **then**
9. Set *dx* to LD
10. **if** image position (row, leftPoint) is 0 **then**
11. Set dy to (row – topPoint)
12. go to line 22
13. **end if**
14. **else**
15. Set dx to RD
16. **if** image position (row, rightPoint) is 0 **then**
17. Set *dy* to (row – topPoint)
18. go to line 22
19. **endif**
20. **endif**
21. **endfor**
22. **output** atan(*dy*/*dx*)*180/pi

C. Scaling

The rotated image may not be in the proper size, therefore before segmenting the characters we need to scale the image to an appropriate size. In our method we scale the images to

a size with an aspect ratio of 1 to 0.8. This is the ratio of standard letter size pages.

D. Noise Elimination

The scanning and or transformation steps of the image may add some noise. Before taking further steps we eliminate this noise. In our proposed method we have used averaging and linear smoothing filter for noise elimination.

V. SEGMENTATION

After preprocessing the image the text blocks are first segmented into lines, lines are then transformed into words and finally words transformed into individual characters.

A. Line Segmentation

Line segmentation is performed by scanning the input image horizontally from top to bottom. The first black pixel of a horizontal line denotes a starting boundary of a line. We continue to scan until we find a horizontal line which has no black pixel. This indicates the ending boundary of the line. Fig. 4 demonstrating the starting and ending boundary of different lines.

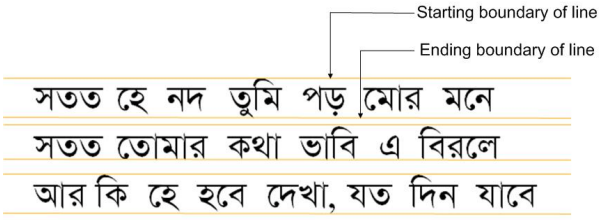

Fig. 4. Line segmentation.

B. Word Segmentation

After text line segmentation is performed, the system segments each line into words. For word segmentation each line is vertically scanned from left to right in order to detect a vertical line with at least one black pixel. This precondition denotes the point of a word. For detecting the ending point of a line we continue to scan vertically until finding a vertical line which has no black pixel. Fig. 5 depicts a text line and its different parts. In Bangla several characters are joined by a horizontal line called "*Mattra*" line. Mattra line is that row or set of rows where the number of black pixel is the maximum [11]. The upper zone denotes the portion above the mattra line, the middle zone covers the portion of basic and compound characters below the mattra line and the lower zone may contain where some vowel and consonant modifiers can reside. The imaginary line separating the middle and lower zone may be called the base line. Upper mattra line is the upper boundary of a line and lower base line is the lower boundary of a line.

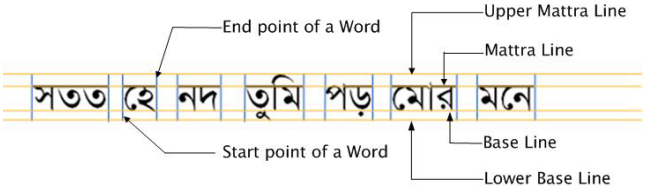

Fig. 5. Different parts of word segmentation.

C. Character Segmentation

The performance of an OCR mainly depends on the accuracy of the character segmentation stage. Character segmentation is performed in some sub stages: (a) removing the mattra line (b) segment those characters which is within the mattra line and base-line (c) segment the parts which are above the mattra line and under the base line. For removing the mattra line, we scan the image horizontally and count the number of black pixel in each row. All the horizontal pixel lines containing this maximum number of pixels or at least 90% of this maximum value is considered as mattra lines. For segmenting the character within the mattra line and the base line we scan vertically. The starting point of a character is the first column where the first black pixel is found. After finding the starting point of a character, we continue scanning until a column without any black pixels is found, which we consider as the ending point of the character being processed (see Fig. 6).

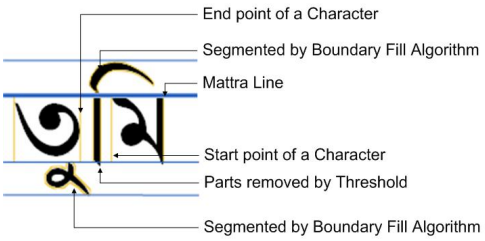

Fig. 6. Character segmentation.

To segment the upper portion and lower portion of the mattra line and base line we use the following BOUNDARY_FILL algorithm.

Algorithm BOUNDARY_FILL (*x, y, img:* IMAGE)
*/*the algorithm fill a region in image img from black to white color. Filling starts at point (x, y). It also copy the filled region into another image img2*/*

1. **if** pixel (*x, y*) in image *img* is black **then**
2. Copy the current pixel value to image *img2*
3. Reset the current pixel location with a white pixel
4. Call BOUNDARY_FILL with values (*x+1, y*)
5. Call BOUNDARY_FILL with values (*x-1, y*)
6. Call BOUNDARY_FILL with values (*x, y+1*)
7. Call BOUNDARY_FILL with values (*x, y-1*)
8. Call BOUNDARY_FILL with values (*x-1, y-1*)
9. Call BOUNDARY_FILL with values (*x-1, y+1*)
10. Call BOUNDARY_FILL with values (*x+1, y-1*)
11. Call BOUNDARY_FILL with values (*x+1, y+1*)
12. **end if**

VI. TRAINING AND RECOGNITION

For training and recognition, each character images is first converted to a 16x16 pixel image; the only feature extracted from the images is a 256-bit long vector, which is then trained or classified using an artificial neural network.

A. Training

Neural network has been trained by normalized feature vector obtained for each character in the training set. In our system the normalized vector is 256-bit long. Four layer neural networks have been used with two hidden layers for improving the classification capability of the neural network with minimum error tolerance rate. At first we will train our neural net with some randomized data. This data will update (hidden weights) based on the error found in the recognition stage.

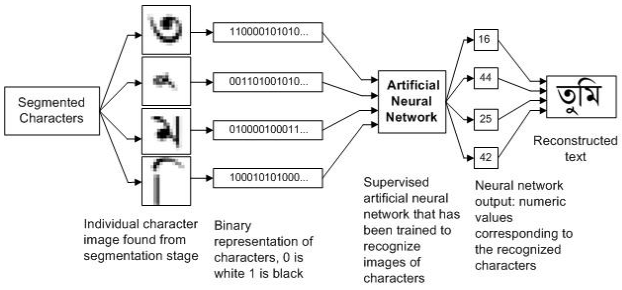

Fig. 7. Segmentation and Regeneration process.

B. Recognition

In the recognition phase of the network, a single iteration is enough to give the confidence value for each class of the character set. After applying the 256-bit long vector (Fig.7.) to the network, if the output of the network is very close to one of the characters with a certain acceptable tolerance then our system will output a numerical value representing the recognized number which then again joins to form the original texts. If the output is far apart from the threshold or the acceptable range from all the possible outputs, then our system cannot identify the character and return a numerical value 0, indicating an unrecognized character.

7. RECONSTRUCTION

After recognition of each character with the neural network we need to regenerate the original text. For reconstruction of the original texts we follow the Bangla word formation rules [9]. According to Bangla word formation rules if a vowel is immediate before the consonant then it will be a vowel marks like Fig. 8.

Fig. 8. Reconstruction the text.

VIII. EXPERIMENTAL RESULTS

We tested our OCR system to measure the performance, for this we scanned 32 pages from popular Bangla novel *Pather Panchali* [8]. There are about 920 lines and 10,400 words in our test case. We separately measure the line, word and character segmentation. We tested our code written in MATLAB. According to our test result our line segmentation success rate is about 98.8% for 920 lines. Our word segmentation success rate is 96.2 % for 10,400 words and the character segmentation success rate is 81% for 10,400 words. In our experiment we used the very popular true type Unicode Bangla font name *AdorshoLipi*. We also tested in some other popular Bangla true type fonts, the results are depicted in Fig. 8. From the result in Fig. 9. our method shows a good performance for most of the fonts except some fonts like *Suleka, Siyam Rupali*.

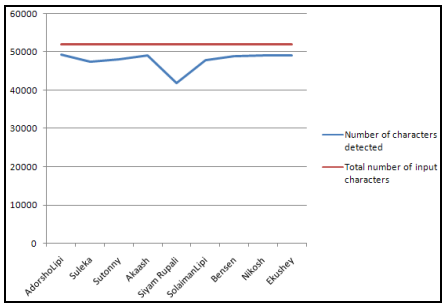

Fig. 9. Performance curve in different Bangla fonts.

We have compared our method with some other popular methods [3, 5], the results are shown in Fig.10.

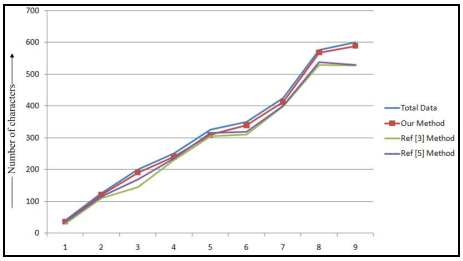

Fig. 10. Comparison graph with other methods.

We have also tested our method in different positive and negative angles and the percentage of success to calculate the exact angle is shown in Fig. 11.

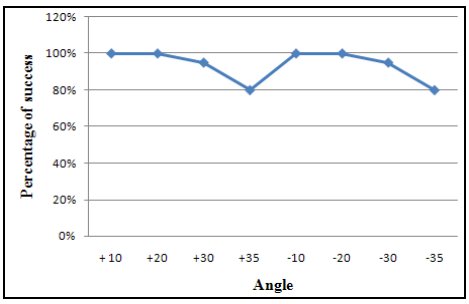

Fig. 11. Different angle and percentage of success.

IX. CONCLUSION

The main objective of this research is to develop a complete character recognition scheme for Bangla printed text books which will be rotation invariant with an acceptable performance rate. Based on our experimental results we can comment that our line and word segmentation method have satisfactory performance. Our character segmentation shows some error because there are some characters in Bangla for which recognition is very difficult. Moreover our method shows a good performance in different types of true type Bangla fonts. Based on the result study our research shows the potential to be used in Bangla text book digitization. We believe that the proposed method of character recognition scheme for Bangla printed text books will bring new direction in future research in this research area.

REFERENCES

[1] A.R. Forkan, S. Saha, M.M. Rahman, M.A. Sattar, "Recognition of Conjunctive Bangla Characters by Artificial Neural Network", *ICICT 2007*, 7-9 March 2007.

[2] A. Dutta, S. Chaudhury, "Bengali Alpha-Numeric Character Recognition Using Curvature Features", *Pattern Recognition*, Vol. 26, No. 12, pp. 1757-1770, 1993.

[3] S. M. Shoeb Shatil and Mumit Khan, "Minimally Segmenting High Performance Bangla OCR using Kohonen Network" , *ICCIT 2006, Proc. of 9th International Conference on Computer and Information Technology*, 2006.

[4] Md. Abul Hasnat, M.H Mumit Khan, "A High Performance Domain Specific OCR For Bangla Script", *Center for Research on Bangla Language Processing*.

[5] B.B. Chaudhuri and U. Pal, "A Complete Printed Bangla OCR System", *Pattern Recognition*, Vol. 31, No. 5, pp. 531-549, 1998.

[6] M.A Sattar, K. Mahmud, H. Arafat and A.F.M Noor Uz Zaman,"Segmenting Bangla Text for Optical Recognition", *ICCIT 2007, 10th international conference on Computer and information technology*, 27-29 Dec. 2007.

[7] R.C. Gonzalez and R.E. Woods, "Digital Image processing", second edition, *Pearson Education*, 2003.

[8] Bibhutibhushan Bandopadhyay, "Pather Panchali", *Ranjan Prakashalay*, 1929.

[9] Palit, Rajesh and Sattar, Md Abdus, "Representation of Bangla Characters in the Computer Systems". *Bangladesh Journal of Computer and Information Technology*, Vol. 7, No. 1, December 1999.