A New Approach to Bangla Text Extraction and Recognition From Textual Image

Md. Al Mehedi Hasan, Md. Abdul Alim, Md. Wahedul Islam
Dept. of Computer Science and Engineering, University of Rajshahi, Rajshahi, Bangladesh.
mehedi_cst@yahoo.com, alim_eva@yahoo.com, wahidcstru@yahoo.com

Abstract
This paper presents a new approach to segment and recognize Printed Bangla Text using Characteristic functions and Hamming network. The main difficulties in printed Bangla text recognition are the separation of lines, words and individual characters. In this paper, a new algorithm has been proposed to detect and separate text lines, words and characters from printed Bangla text. The algorithm uses a set of characteristic functions for segmenting upper portion of some characters and characters that come under the Base line. It also uses a combination of Flood-fill and Boundary-fill algorithm for segmenting some characters that cannot be segmented using traditional approach. Hamming network is used for recognition scheme. Recognition is done for both isolated and continuous size independent printed characters.

Keywords: Optical character recognition, Text segmentation, Hamming Network, Hamming Associative Memory, Characteristic function.

I. INTRODUCTION
Optical Character Recognition [1] is one of the challenging research fields in image processing [2]. Recognition of printed Bangla text is a subject of special interest for us and also very important to Bangla processing. However, there are no precise methods of separating lines, words and characters from printed Bangla text. Although some limited research works have been done in this field, they are not sufficient to segment printed Bangla text. In order to overcome these limitations, we have proposed a new approach to segment printed Bangla text. The algorithm uses a set of characteristic functions for segmenting upper portion of some characters and characters that come under the Base line. It also uses a combination of Flood-fill and Boundary-fill [3] algorithm for segmenting some characters that cannot be segmented using traditional approach. The proposed system uses Hamming Network to recognize characters after being segmented. A Hamming network performs template matching using neural principles and is suitable for classification [4]- [6].

II. BACKGROUND STUDY
In recently published papers, no suitable steps have been taken to segment the upper portion of a character such as ০০, and characters that come under the base line such as ১০০০০০. In Ref. paper [7], segmentation of upper portion of a character and characters that come under the base line are totally ignored which is very important to recognize Bangla characters. On the other hand in Ref. paper [1], for detecting characters that come under the base line, repeatedly decrease the height of the current window and pass the matrix to the network until network can recognize the window as a learned character and then recognize the below window separately. Similar operation is taken for the upper portions such as ০০, ০০, ০০, called modifier in that paper. Repeatedly pass the matrix to the neural network is very time consuming matter. Sometimes these methods fail to detect actual character such as পাঁচ. In this paper, we have used characteristic functions to overcome these problems accurately.

III. PROPOSED SYSTEM ARCHITECTURE
In Bangla Text Extraction and Recognition System, multiple preprocessing and processing stages have been employed. The proposed system can be described in two modules. One is segmentation module and the other is recognition module. The segmentation module includes: (a) Data Acquisition, (b) Preprocessing, (c) Scaling and (d) Text Segmentation. The recognition module includes: (a) Feature Extraction, (b) Hamming Associative Memory, and (c) Character Classification. The simplified block diagram of the proposed system is shown in Fig.1.

Fig.1 Bangla Text Extraction and Recognition System.

A. Data Acquisition
The input images are acquired from documents containing Bangla text by using scanner as an input device. Scanned images are then stored in some picture file
such as BMP, JPG etc. This image subsequently passes through preprocessing, scaling, and segmentation steps.

**B. Preprocessing**

Preprocessing includes noise removal and the elimination of redundant information as far as possible. In a scanned image, it is common to have some pixels of lower concentration in the position of white portion of the image due to various reasons such as fault of scanner [1]. High pass filtering is used to remove this noise from the input image [2].

**C. Scaling**

In order to segment and recognize size independent printed Bangla text, scaling techniques [3] have been used. The proposed system uses 20 pt font to segment the printed Bangla text. If a document containing characters of size more than 20 pt font then the system will be scaled down all of the characters to 20 pt font and if less than 20 pt font then the system will be scaled up all of the characters to 20 pt font.

**D. Text Segmentation**

Segmentation of printed Bangla text is a process by which the text is partitioned into its coherent parts [2][7]. The proposed system uses the following steps to segment printed Bangla text:

- Text Line Detection
- Word Segmentation
- Character Segmentation

**D.1 Text Line Detection**

Text line detection has been performed by scanning the input image horizontally and by keeping record of the number of black pixels in each row, shown in Fig.2(a). Upper boundary of a line is the first row where the first black pixel is found. After finding the upper boundary, it continues scanning until a row whose next two consecutive rows have no black pixels, which is the lower boundary of the text line. It is noted that there exist more than two blank rows between two lines. Fig.2(b) shows the line detection process. The Matra line and the base line are also detected in this process. Matra line is that row where the number of black pixels is the maximum [1]. In a 20 pt font, the number of rows between the Matra line and the base line is constant. When the number of rows between the Matra line and the lower boundary of a text line exceeds this constant then the base line is the row, which is below that constant number of rows from the Matra line. Otherwise the base line and the lower boundary of a text line are the same. 

Fig.2 (a) Explanation Of Horizontal And Vertical Scan 
(b) Text line detection and Word segmentation

**D. 2 Word Segmentation**

After detecting a line, the system scans the image vertically from the Matra line to the base line, shown in Fig.2(a). The number of black pixels in each column is counted. Starting boundary of a word is the first column where the first black pixel is found. After finding the starting boundary, it continues scanning until a column whose next two consecutive columns have no black pixels, which is the ending boundary of the word being processed. It is noted that there exist more than two blank columns between two words. Fig.2(b) shows the word segmentation process.

**D. 3 Character Segmentation**

To detect the character boundary, a vertical scan is initiated from the row that is just under the Matra line to the base line of a word. The starting boundary of a character is the first column where the first black pixel is found. After finding the starting boundary of a character, it continues scanning until a column without any black pixels is found, which is the ending boundary of the character being processed, as shown in Fig 3(a). In some cases, linear vertical scan may fail to detect the ending boundary of the segment, as shown in Fig 3(b). In these situations there will be two characters between the starting and the ending boundary of the segment. To separate these two characters a combination of Flood-fill [3] and Boundary-fill [3] algorithm is used. The algorithm begins with a starting pixel inside a character and checks to see if this pixel is inside the Matra line, base line, starting and ending boundary of this segment and has the black color.

Fig. 3 (a) Linear vertical scan for character detection, 
(b) Linear vertical scan fails to detect character.

To find the portion of a character above the Matra line such as $\text{া}$, $\text{ি}$, $\text{ু}$, the first black pixel is selected which is just adjacent to the Matra line and between the starting and the ending boundary of the character in horizontal scan. To extract the correct upper portion of the character being processed, send this pixel to a set of
characteristic functions, which are devised from the common and adjacent to the common pixels of the upper portion of some standard fonts (SutonnyMJ, Sulekha etc). The characteristic function, which returns the maximum number of black pixels, also returns the upper portion for the character being processed as shown in Fig.4 (a).

![Fig.4 Use characteristic functions for finding upper portion of a character and lower character.](image)

To find the characters situated below the base line such as $y, \sim, \bar{x}, ^{\circ}$, the first black pixel is selected which is just under the base line and between the starting and the ending boundary of a character in horizontal scan. To extract the correct lower character, send this pixel to a set of characteristic functions, which are devised from the common and adjacent to the common pixels of the lower character below the base line of some standard fonts (SutonnyMJ, Sulekha etc). The characteristic function, which returns the maximum number of black pixels, also returns the current lower character as shown in Fig.4 (b).

E. Feature Extraction

Feature extraction provides an important job in character recognition systems. This effectively reduces the number of computations and hence faster the recognition process. In this section, each of the segmented Bangla character is converted into a real valued vector form (here 22x15 matrix) containing 1s and -1s, where 1 represent the presence of character component and -1 represents the absence the character component. To obtain the real valued vector, an N1xN2, N1 is the height and N2 is the width in pixels, segmented character is reduced into 22x15 matrix by dividing N1 and N2 by the height and width reduction factors R1 and R2 respectively. The algorithm for this conversion is presented below.

Algorithm-1: Feature Extraction Algorithm
1. Set $R1 = N1/22$
2. Set $R2 = N2/15$
3. Set $Avg = 0$
4. For $i = 1$ to $22$
5. For $j = 1$ to $15$
6. Set $sum = 0$
7. For $p = (i-1)*R1$ to $i*R1$
8. For $q = (j-1)*R2$ to $j*R2$
9. $sum = sum + N(p,q)$
10. endfor
11. endfor
12. $Avg = sum/(R1*R2)$
13. if (Avg is a certain percentage) then
14. $CharacterMatrix(i,j)=1$
15. endif
16. else
17. $CharacterMatrix(i,j)=-1$
18. endelse
19. endfor
20. endfor

F. Hamming Associative Memory And Hamming Network

In the associative memory problem, one is given a memory set of M prototype patterns or input-output pairs of the form \{xᵢ, yᵢ\}:i=1,2,...,M [5]. The goal is to design a system, which can store each of the \{xᵢ, yᵢ\}, associations in memory so that when pattern xᵢ is presented as input (the memory key), the system reliably retrieves the pattern yᵢ. In addition, the system should retrieve yᵢ even when xᵢ is corrupted with various types of noise. But when the memory key is not sufficiently close to any of the stored prototypes, the system should reject the input as being noise.

The Hamming associative memory (or nearest neighbor classifier) is the simplest type of memory, whereby one stores in memory all of the fundamental input-output pairs \{xᵢ, yᵢ\} [5]. Then for a given input key x, the system simply determines the closest matching stored input pattern xᵢ and then outputs the corresponding output pattern yᵢ. The Hamming network can be modified to provide for a reject or no-decision state when the input pattern is not sufficiently close to any of the stored prototypes as well as modified to allow for local and parallel distance computations [6].

The first layer called the quantifier network – is composed of a number of neurons, which performs the Hamming distance computation. The second layer called the discriminator layer – is traditionally composed of a feed-forward network, which performs the winner take all (WTA) operation, i.e. selects the first-layer neuron of smallest Hamming Distance as the win-
ner [11] and classify the input pattern into the closest group, shown in Fig.5.

![Fig.5 A Hamming Network Example.](image)

The Hamming Net Algorithm [12] is given below:

**Algorithm-2: Hamming Net Algorithm**

1. Assign connection weights and offsets in lower sub-net and in the upper sub-net.
2. Initialize with unknown input character pattern.
3. Iterate until the output of only one unit remains positive.
4. Go to step 2.

**G. Character Classification**

The character matrix obtained from the feature extraction module has been used as input to the Hamming network. Hamming Network is used as a recognizer and its associative memory has been used to store all the fundamental input output pair to make it an efficient recognizer for Bangla alphabet. After inputting the matrix to the network, if the output of the network is very close to one of the characters then the system shows it as a recognized character. If the output is far apart from all the possible outputs, then the system cannot identify the character and wait for the character from user to replace. This process continues until the end of the input document. The following algorithm illustrates this recognition system.

**Algorithm-3: Algorithm for Bangla Text Recognition**

1. Input the Document which will be recognized
2. Set Count=1
3. Repeat step 3 to 10 until whole document is recognized
4. Select the (Count)th segmented character.
5. Determine the character matrix using algorithm-1
6. Input the matrix to HN
7. Calculate the winner
8. If (Winner >0)
   Print “The character is recognized and added to the output list”
9. Else
   Print “The character is Unrecognized”
10. Count=Count+1
11. End

**IV. RESULTS AND DISCUSSIONS**

The system has been tested with several chunks of printed Bangla texts or textual images (.bmp/.jpg file) and observed the system outputs. To study the recognition and segmentation capability of this system, some distorted patterns of text or distorted textual image has been used. Hamming associative memory has been learned with the most familiar Bangla font named “SutonnyMJ”. For accuracy measures, various experimental images of different Bangla fonts have been used including “SutonnyMJ” and got the following statistics. The accuracy of this system is calculated as:

\[ \text{Accuracy} = \frac{\text{Total no. of recognized characters in the document}}{\text{Total no. of characters in the document}} \times 100 \%
\]

<table>
<thead>
<tr>
<th>Total No characters</th>
<th>No. Of Recognized Characters</th>
<th>Font and size</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>820</td>
<td>787</td>
<td>Sulekha, 10</td>
<td>96%</td>
</tr>
<tr>
<td>753</td>
<td>738</td>
<td>SutonnyMJ, 24</td>
<td>98%</td>
</tr>
<tr>
<td>669</td>
<td>649</td>
<td>Moina, 20</td>
<td>97%</td>
</tr>
<tr>
<td>833</td>
<td>833</td>
<td>SutonnyMJ, 20</td>
<td>100%</td>
</tr>
</tbody>
</table>

**V. CONCLUSION**

This paper has presented a new and elegant method of segmenting printed Bangla text. The system is capable of separating size independent Bangla letters, digits and special symbols from printed documents. It can also recognize the segmented characters using Artificial Neural Network such as Hamming Network. By extending the system using spell checker and dictionary, it is possible to make a real time printed Bangla Character Recognition System.

**REFERENCES**


http://www.atis.org/tg2k/hamming_distance.html


[12] Neural Network.  