Bangla Character Recognition System is Developed by using Automatic Feature Extraction and XOR Operation

By Md. Mojahidul Islam, Md. Imran Hossain & Md. Kislu Noman

Islamic University, Bangladesh

Abstract - This paper presents off-line bangle character recognition system using automatic feature extraction and XOR operation. In this system, the Bangla text is accepted as an image file which is first segmented into lines and words and then each word is segmented into characters. The pixels outside the boundary of the character are eliminated. The characters are scaled to a size equal to the database image. A XOR operation is performed between the scaled image and the database image and the error (%) is calculated. Finally, depending on the minimum error, the system recognizes the character to use in the output. The average recognition accuracy rate of the system was about 80%.

Keywords : character recognition; character segmentation; automatic feature extraction; XOR operation.

GJCST-F Classification: I.5.0

Strictly as per the compliance and regulations of:

© 2013. Md. Mojahidul Islam, Md. Imran Hossain & Md. Kislu Noman. This is a research/review paper, distributed under the terms of the Creative Commons Attribution-Noncommercial 3.0 Unported License http://creativecommons.org/licenses/by-nc/3.0/), permitting all non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Abstract - This paper presents off-line Bangla character recognition system using automatic feature extraction and XOR operation. In this system, the Bangla text is accepted as an image file which is first segmented into lines and words and then each word is segmented into characters. The pixels outside the boundary of the character are eliminated. The characters are scaled to a size equal to the database image. A XOR operation is performed between the scaled image and the database image and the error (%) is calculated. Finally, depending on the minimum error, the system recognizes the character to use in the output. The average recognition accuracy rate of the system was about 80%.

Keywords : character recognition; character segmentation; automatic feature extraction; XOR operation.

I. Introduction

The subject of character recognition has been receiving considerable attention in recent years due to the advancement of the automation process. Automatic character recognition improves the interaction between man and machine in many applications like office automation, cheque verification, mail sorting, and a large variety of banking, business and data entry applications. We are concerned here with the recognition of character in Bangla language. Bangla is the mother language of Bangladesh and approximately 10% of the world’s population speaks in Indian, Chinese and other languages trying to develop the complete character recognition system. In our country, research works in this field have achieved a limited success so far as compared to the other foreign languages. Though, the achievement in this fascinating field is not enough to reach the ultimate goal. But the progress of such research with Bangla language is still in an initial level. This research is a simple flourish to implement that dream as the initial step to convert the Bangla text to computer readable form that is development of complete Bangla Character Recognition system. Individual Bangla characters were recognized using various techniques such as geometric shape analysis, black runs and concavity measurement technique.

II. Implementation of Character Recognition System

The character recognition system can be divided as segmentation of text document into character and recognition of the character. The whole process is shown in Fig 1.

![Figure 1: Block diagram of character recognition system](image-url)
a) **Image Acquisition**

The input images are acquired from documents containing text by using scanner as an input device or using Adobe Photoshop or Paint. Acquired images are then stored in Hard Disk in JPG picture format. This image is then passed for preprocessing.

b) **Pre-Processing**

The scanned image is converted into binary image. At first, the RGB image is converted into grayscale image and then binary image i.e. an image with pixel 0 (white) and 1 (black). After converting the image, the unnecessary pixels (0s) from the original image is removed.

c) **RGB to Grayscale and Gray to RGB Conversion**

In practical cases most of the images are generally color (RGB), but it is complex to work with a three-dimensional array. So it needs to convert the RGB image into the grayscale image. The RGB to grayscale conversion is performed by MATLAB command.

\[ I = \text{rgb2gray}(f) \]

For ease of analysis, the grayscale image is converted into binary image by using the following MATLAB command.

\[ \text{BW} = \text{im2bw}(I) \]

III. **Text Segmentation**

Text segmentation is a process where the text is partitioned into its elementary entities i.e. characters [10]. The total performance of the character recognition process depends on the accuracy of the segmentation process of the text into the characters. In the segmentation phase, first the document is segmented into text lines, the text lines are segmented into text words and then the words are segmented into characters.

a) **Line Segmentation**

Text line segmentation is performed by scanning the input image horizontally. Frequency of black pixels in each raw is counted to separate the line. The position between two consecutive lines, where the number of black pixels in a raw is zero denotes a boundary between the lines [13]. The output image is shown in Fig 2.

b) **Word Segmentation**

In English text there is a minimum gap between two consecutive characters and two consecutive words. The minimum gap between two consecutive words is greater than two consecutive characters. Although maximum characters in Bangla text line are connected by matra line with each other, the same case occurs if the gap exists between them. For word segmentation from the text line, the vertical scan is performed. If there exists n consecutive scan that find no black pixel, we denote it to be a marker between two words. The value of n is the minimum gap between two consecutive words which is taken experimentally. The output is shown in Fig 3.

c) **Character Segmentation**

For character segmentation from the word, the vertical scan is performed. The starting boundary of a character is the first column where the first black is found. After finding the starting boundary of a character, it continues scanning until a column without any black pixel is found, which is the ending boundary of the character being processed [14]. Fig. 4 shows a single segmented character and its corresponding binary format.
The knowledge base is designed based on the feature matrix of various characters. In order to build the knowledge base, first, the RGB character image is converted into grayscale image then it is converted into binary image. After getting the binary image, the unnecessary pixels from the character boundary is eliminated.

e) Feature Extraction

Feature extraction is the process of extracting essential information content from the image segment. It plays an important role in the whole recognition process [10].

f) Scaling

Depending on the height and width of the database image the segmented characters are scaled. If the size of the segmented character is higher than the database character then the system will be scaled down all the segmented characters to the size of the database character, otherwise scaled up. If C be the segmented character then the scaled image S is obtained by the following MATLAB command: S = imresize(C, [height, width]). Where, height and width is the dimension of the database character. Fig 5(a) shows a database character whose size is $16 \times 16$. The segmented character of size $20 \times 20$ shown in Fig 5(b) is scaled down to the size of database character $16 \times 16$ shown in Fig 5(c). This is repeated for all database characters and finally for all segmented characters.

![Figure 4: (a) Binary Form of a Segmented Character](image)

![Figure 5: Image scaling (a) Database image of size $16 \times 16$, (b) Segmented image of size $20 \times 20$, and (c) Scaled image of (b) of size $16 \times 16$](image)
Character Recognition

Character recognition performance depends on the scaling. If the segmented character is too higher or too lower than the database image then the character recognition performance is reduced. The character recognition procedure is described in following Algorithm:

BEGIN
1. Calculate total\_pixel = height × width.
2. Take XOR between first database character and scaled character S.
3. Calculate no. of correct pixels (0 is the correct pixel), correct\_pixel.
4. Calculate percentage of error using
   \[
   \text{error} (\%) = \frac{\text{total\_pixel} - \text{correct\_pixel}}{\text{total\_pixel}} \times 100\% 
   \]
   and save error (%).
5. Repeat Step 1 to Step 4 for all database characters.
6. Calculate minimum error (%) (e\_min) obtaining from Step 4 for database characters.
7. Define a error tolerance, error\_tolerance.
8. If e\_min < error\_tolerance
   Compare e\_min for all %error
   If e\_min = error\_tolerance
      Then print the i\_th character
      endif
   else
      Print ‘the character is not recognized’
      endif.
9. Repeat Step 1 to Step 8 for all segmented characters
10. End

**Figure 6**: Character recognition (a) Database image of size 16×16, (b) Scaled image of size 16×16, (c) Image after XOR between (a) and (b)

<table>
<thead>
<tr>
<th>Character</th>
<th>Pixel Value</th>
<th>Pixel Value</th>
<th>Pixel Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000000000000</td>
<td>0111110000000000</td>
<td>0111110000000000</td>
<td></td>
</tr>
<tr>
<td>0111111000001111</td>
<td>0111111000001111</td>
<td>0111111000001111</td>
<td></td>
</tr>
<tr>
<td>1011100000001100</td>
<td>1011100000001100</td>
<td>1011100000001100</td>
<td></td>
</tr>
<tr>
<td>1011110010011000</td>
<td>1011110010011000</td>
<td>1011110010011000</td>
<td></td>
</tr>
<tr>
<td>1100111111011000</td>
<td>1100111111011000</td>
<td>1100111111011000</td>
<td></td>
</tr>
<tr>
<td>1101111111101100</td>
<td>1101111111101100</td>
<td>1101111111101100</td>
<td></td>
</tr>
<tr>
<td>1110111111001100</td>
<td>1110111111001100</td>
<td>1110111111001100</td>
<td></td>
</tr>
<tr>
<td>1111100000010000</td>
<td>1111100000010000</td>
<td>1111100000010000</td>
<td></td>
</tr>
<tr>
<td>1111110110111110</td>
<td>1111110110111110</td>
<td>1111110110111110</td>
<td></td>
</tr>
<tr>
<td>1111111111111110</td>
<td>1111111111111110</td>
<td>1111111111111110</td>
<td></td>
</tr>
<tr>
<td>1111111111111111</td>
<td>1111111111111111</td>
<td>1111111111111111</td>
<td></td>
</tr>
<tr>
<td>1111111111111111</td>
<td>1111111111111111</td>
<td>1111111111111111</td>
<td></td>
</tr>
<tr>
<td>1111111111111111</td>
<td>1111111111111111</td>
<td>1111111111111111</td>
<td></td>
</tr>
</tbody>
</table>

Total number of pixels, total\_pixel = 16 × 16 = 256
Total number of correct pixels (0\_i), correct\_pixel = 221

\[
\text{error} (\%) = \frac{\text{total\_pixel} - \text{correct\_pixel}}{\text{total\_pixel}} \times 100\% 
\]

\[
= \frac{256 - 221}{256} 
\]

\[
= 13.6719\% 
\]

In this way, for all database character the error (%) calculation is repeated. If the database character exactly or approximately matches with the segmented character then the error (%) will minimum. So base on the minimum error, the system gives the corresponding output character.

IV. RESULT AND PERFORMANCE ANALYSIS

The system is divided in two main phases: segmentation and character recognition. So the overall performance of the system directly depends on the performance of the two individual phases. The accuracy of this system is measured as the success rate for the recognition of characters. It is measured using Eq. (1):

\[
\text{Accuracy} (\%) = \frac{\text{Number of Success}}{\text{Number of Test}} \times 100\% 
\]
a)  **Segmentation Performance**

The segmentation performance of this system is shown in Table 1.

**Table 1 : Text Document Segmentation Result**

<table>
<thead>
<tr>
<th>No. of Lines in a Text Document</th>
<th>Line Segmentation Accuracy (%)</th>
<th>Word Segmentation Accuracy (%)</th>
<th>Character Segmentation Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>100</td>
<td>97</td>
<td>89.05</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>97.5</td>
<td>92.50</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>94</td>
<td>90.71</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>96.67</td>
<td>92.69</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>94</td>
<td>90.32</td>
</tr>
</tbody>
</table>

b)  **Segmented Character Recognition Performance**

For character recognition, this system uses XOR operation which is a very simple matching technique. The character recognition performance of this system is shown in Table 2 for Shoroborno and Table 3 for Numerical Character.

**Table 2 : Bangla Character (Shoroborno) Recognition Result**

<table>
<thead>
<tr>
<th>No. of Test Sample</th>
<th>Total No. of Characters</th>
<th>Total No. of Success</th>
<th>Success Rate (%)</th>
<th>Average Success Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120</td>
<td>90</td>
<td>75</td>
<td>76.96368</td>
</tr>
<tr>
<td>2</td>
<td>150</td>
<td>116</td>
<td>77.33333</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>125</td>
<td>94</td>
<td>75.2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>130</td>
<td>102</td>
<td>78.46154</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>170</td>
<td>134</td>
<td>78.82353</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3 : Bangla Numerical Character Recognition Result**

<table>
<thead>
<tr>
<th>No. of Test Sample</th>
<th>Total No. of Characters</th>
<th>Total No. of Success</th>
<th>Success Rate (%)</th>
<th>Average Success Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>42</td>
<td>84</td>
<td>83.27363</td>
</tr>
<tr>
<td>2</td>
<td>70</td>
<td>53</td>
<td>75.71429</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>65</td>
<td>56</td>
<td>86.15385</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>33</td>
<td>82.5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>44</td>
<td>88</td>
<td></td>
</tr>
</tbody>
</table>

V.  **Discussion and Conclusion**

The aim of this system is to recognize Bangla characters. This system can recognize these characters with slight limitations. The limitations are discussed in the following section.

a)  **Limitation**

The performance of this system depends on the segmentation and recognition. If the characters of text are in very close or overlap to each other, then the system fails to segment the characters. For Bangla characters, different font size is possible in practical. It is not possible to store all the front size in database. So it needs to scale the character which causes distortion in character shape. It should create a problem but the system should not fail always.

b)  **Further Scope**

Due to the limitations described in previous section the system is not suitable for on-line applications. The overlapping character can be segmented by using Flood fill and Boundary fill algorithm. It is further target to perform this work.

c)  **Conclusion**

In this paper the off line Bangla character recognition system is developed by using automatic feature extraction and XOR operation. The efficiency of this system is not so high. In future, MLP and SVM classifier can be used for character recognition.

**References Références Referencias**

