

# **Bangla OCR - A Complete Working System**

**A.Q. M. Saiful Islam.**

[niposust@gmail.com](mailto:niposust@gmail.com)

Shah jala University of Science  
& Technology. Sylhet-3114.

([www.sust.edu](http://www.sust.edu))

Bangladesh.

**Md. Mahbubul Haque**

[rabbi.sus@gmail.com](mailto:rabbi.sus@gmail.com)

Shah jala University of Science  
& Technology. Sylhet- 3114.

([www.sust.edu](http://www.sust.edu))

Bangladesh.

**Mohammad Mahadi Hasan.**

[mhasan\\_sust@yahoo.com](mailto:mhasan_sust@yahoo.com)

Shah jala University of Science &  
Technology. Sylhet- 3114.

([www.sust.edu](http://www.sust.edu))

Bangladesh.

## **Abstract**

This paper represents a complete working system on Bangla character recognition from input Bangla Script image. During the recognizing process, pre-processing of input image, segmentation of both lines, words and characters and finally recognize them based on their criteria. Experience with OCR problem teaches that for most subtasks involve in OCR, there is no single technique that gives perfect results for every type of document Image. We have proposed a new algorithm to recognizing a character particularly based on character's characteristics. We have assigned a character code for every possible character in Bangla script. Each character code is divided by three subgroups. First 5 digit come from character criteria and last 6 digit(3-digit horizontal and 3-digit vertical code) is generated from the character during execution of the system. In this approach there is no need to separate a conjunct to their identical characters. The algorithms that we have applied here has the accuracy about to 80% for general style characters in real time environment. There are also some drawback exists when we try to recognize different style fonts like Italic Style, different sized font. For the measurement of the efficiency of the algorithm that we proposed in this paper we think that the given document image is noise free. However, in the case of noisy images, we face problems of a much more serious nature.

## 1. Introduction

OCR stands for Optical Character Recognition. Optical Character Recognition is regarded as one of the most challenging steps in the process of digitization of literature. OCR consists of mapping a collection of arrays of pixels onto a set of characters.

There are several problems encountered in processing a document. A document may be multicolumn, consisting of images etc. The text zone of the document needs to be extracted before the recognition of the text can take place. In printed text, a skew results in overlap between text lines when scanned horizontally makes it difficult to extract text lines. The background noise can further complicate the situation. The ink spread results in character fusions and fading fragments a character.

In OCR, there is a conflicting demand of classifying a large set of natural variants into a single class and at the same time discriminate between closely resembling patterns. It is obvious that a merely statistical classificatory approach will not succeed and a deeper study into the structure of the scripts is required. The last 50 years of research has clearly demonstrated that no single strategy is sufficient for dealing with the complexity of the problem. Moreover, the strategy cannot be same for reading texts of different scripts/languages.

Segmentation of a document into lines and words and of words into individual characters and symbols constitute an important task in the optical reading of texts. Presently, most recognition errors are due to character segmentation errors (1). Very often, adjacent characters are touching, and may exist in an overlapped field (1). Therefore, it is a complex task to segment a given word correctly into its character components. In this paper, we have considered the problem of segmenting printed text in Bangla. Bangla is the official reading, writing and speaking script of Bangladesh and a part of India.

An optical character recognition system usually consists of two main stages, namely segmentation and recognition. There is a possibility of ambiguity at each stage. The line segmentation may be ambiguous due to tilt and overlapping of text lines. Some of the closely written words may not get segmented into individual words. The characters may be fused due to unwanted breaks, resulting in wrong segmentation. The recognition process may not be able to correctly identify a character due to wrong segmentation or inadequate set of features considered for classification. Inadequate or wrong training may also lead to incorrect recognition of a character.

## **2. Related work in Bangla Text Recognition**

Bangla script is alphabetic in nature and the words are two dimensional composition of characters and symbols which makes it deferent from Roman and ideographic scripts. The algorithms which perform well for other scripts can be applied only after extensive pre-processing which makes simple adaptation ineffective. Therefore, the research work has to be done independently for Bangla script.

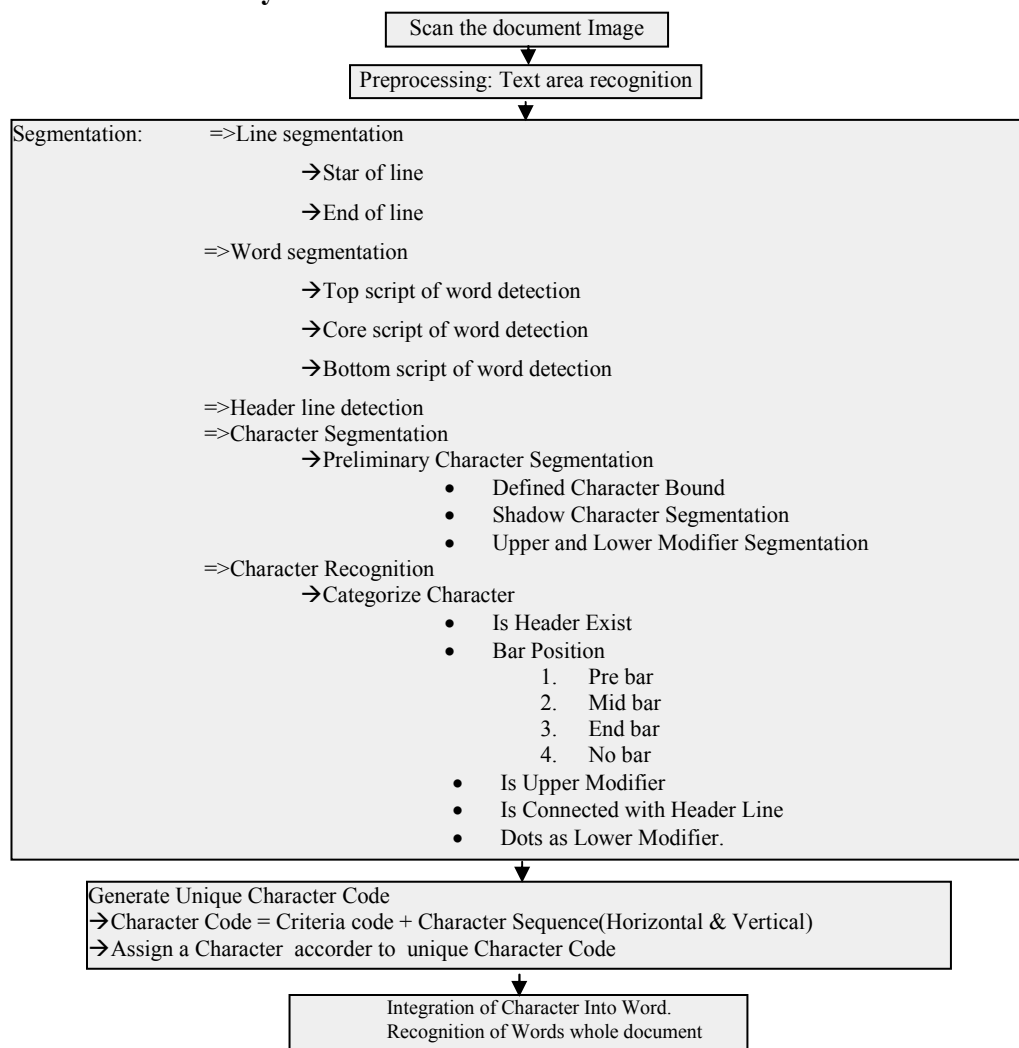
Alamgir [11] have reported various aspect of Bangla script recognition. The post-processing system is based on contextual knowledge which checks the composition syntax. Molla [12] have described Bangla numeral recognition based on the structural approach. The primitives used are horizontal line segment, vertical line segment, right slant and left slant, a decision tree is employed to perform the analysis based on the presence/absence of these primitives and their interconnection. A similar strategy was applied to constrained hand-printed Bangla characters [6]. Neural network approach for isolated characters has also been reported [7]. However, none of these works have considered real-life documents consisting of character fusions and noisy environment. An OCR for Bangla printed script has been described by Chaudhuri and

Pal [8, 9, 10]. The emphasis is on recognizing the vowels and consonants (basic characters) which constitute 94%-96% of the text. They have not presented any sample document and corresponding output for various stages. The quality of their test documents remains hidden. Results are summarized in one statement which does not tell anything about the results at various stages and the contribution of each stage in a comprehensive manner.

### 3. System Hierarchy

A document reading system may be looked at as a hierarchical organization of cooperating modules. At the top level, the document reading problem consists of the document image. And at the bottom level the interpretation of the input image exists in the form of recognized words and sentences. The segmentation and recognition phases lie between these two levels.

#### Different Phases of the System



#### **4. Extraction of Units for Recognition from the Document Image**

A document page may contain images, graphs, tables etc, in addition to the text. Extraction of text-zones from the document has been extensively studied [3,4,5] and still continues to be an active research area. However, in this thesis, we illustrated a pre-processing stage that extracts uniform text zones from the document image. Our system segments each uniform text zone into text lines and text lines into words. Words are further segmented into characters and symbols. The characters and symbols may not be valid Bangla symbols when viewed in isolation. We refer to characters and symbols as ‘units for recognition’ or ‘recognition unit’.

The Extraction of Unit from pre-processed document Image can be done by following steps

- **Line Segmentation.**

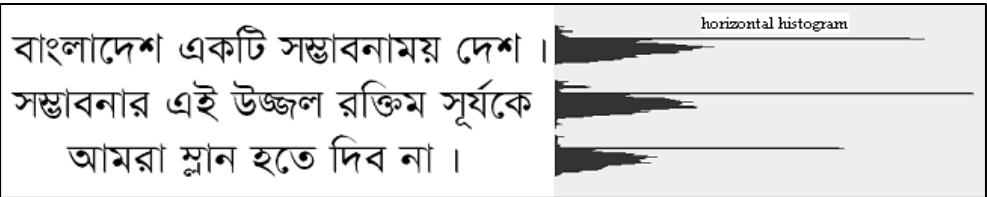
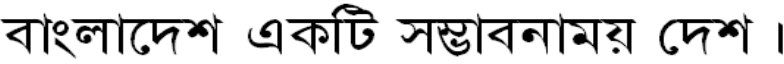

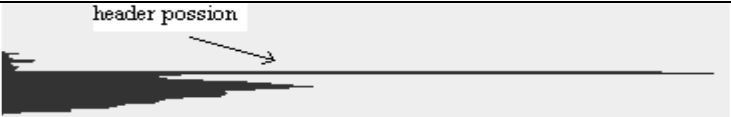


Bangla script is written from left to right, top to bottom. A text line is separated from the previous and following text lines by white space. This segmentation is based on horizontal histograms of the document. A horizontal histogram of the uniform text zone is made. A zero value in the histogram corresponds to a horizontal gap. The horizontal gaps are assumed to be the line boundaries.

- **Identify Header Line Position.**

Header line position is the dominating feature for extracting characters and modifiers correctly. Header line is a long vertical stroke started from left and spans to the right from start to end. As this is the common line, the horizontal projection shows it as a instantaneous stroke at the plot. The stroke becomes very high rapidly and falls down rapidly again having a width of only 1 or 2 pixels. So our OCR system checks if there is a rapid variation in pixel count while checking the rows one after another. If it finds one, it stores the position.

- **Segmentation of a text line into Words.**

In Bangla script all characters and symbols of a word are joined together by a header line. As a result, word boundaries are rarely ambiguous. The gap in header line creates no problem for word boundary identification process as the gap in header line does not create a vertical gap in the word. A vertical histogram of a text line is made and every gap of two or more pixels in the histogram is taken to be the word delimiter. Segmentation of a text line into words is almost self-explanatory.

|        |  |
|--------|--|
| Step 1 |    |
| Step 2 |   |
| Step 3 |  |
| Step 4 |  |
| Step 5 |  |
| Step 6 |  |

- **Segmentation of a Word into Symbols and Characters.**

The header line joins the characters of a word together which makes the segmentation of a word into its constituent characters slightly involved. The region above header line contains upper modifier symbols. The region below the header line contains core characters and lower modifier symbols. Before segmentation can progress any further, header line must be identified. Header

line is easily identified as it is the most dominating horizontal line in a word. After the header line is removed, vertical gap separates the top modifiers from their neighbors. The characters below header line are also separate from their neighbors by vertical gap.

|        |  |
|--------|--|
| Step 1 |  |
| Step 2 |  |
| Step 3 |  |
| Step 4 |  |
| Step 5 |  |

## 5. Recognition of Symbols

### 5.1 Define the general criteria of a character:

We gather information of each character by applying them a criteria matching procedure.

This procedure contain the following features

|   |  |
|---|--|
| Existence of Header   |  |
| Position of Vertical Bar<br>Pre Bar, Mid Bar,<br>End Bar, Non Bar |  |
| Upper Modifier  |  |
| Connection to Header Line   |  |
| Dot as Lower Modifier   |  |

## 5.2 Horizontal Zero Crossings

Image of a character is treated as an array of pixels. A black pixel is expressed by a 1 and a white pixel by 0. Tracing the whole array row by row, number of transitions from black pixel to white pixel is recorded for each row. Let  $V_i$  be the number of transitions for  $i^{\text{th}}$  row. The sequence  $V_i$ ,  $0 \leq i \leq n$ , where  $n$  is the pixel height of the character, is referred to as horizontal zero crossing sequence  $S$ .



We have modified the feature to make it font-independent and noise resilient. We have divided the character into  $n$  horizontal segments of equal height. Each segment is represented by number of zero-transitions that is most frequent in the segment. We divide the sequence  $S$  into three subsequences of equal length,  $S_1, S_2$ , and  $S_3$ . For each subsequence the most frequent number is stored in  $S_1\text{-Most}, S_2\text{-Most}$ ,  $S_3\text{-Most}$ . The feature vector consists of  $(S_1\text{-Most}, S_2\text{-Most}, S_3\text{-Most})$ . This feature is used as a filter to reduce the set of probable characters for an unknown character. Fig shows that all 4 font face produces the same output for 'ক' and 'খ' which and (131), (122).

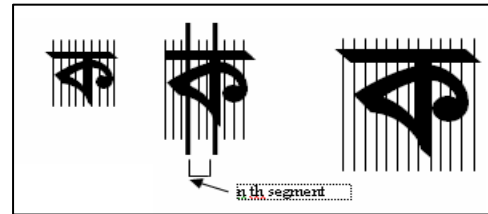
**Sequence S for four samples of character 'ক' and 'খ'**

|                               |  |
|-------------------------------|--|
| <b>Sequence<br/>s for 'ক'</b> | <p>.....111133323321111..... =&gt; 11113 33233 21111 =&gt; 131</p> <p>.....111133343321111..... =&gt; 11113 33333 21111 =&gt; 131</p> <p>.....111133343321111..... =&gt; 11111 33343 32111 =&gt; 131</p> |
| <b>Sequence<br/>s for 'খ'</b> | <p>.....3111 12222 22211..... =&gt; 3111 12222 22211 =&gt; 122</p> <p>.....3111 22212 22211..... =&gt; 3111 22212 22211 =&gt; 122</p> <p>.....3111 22211 22211..... =&gt; 3111 22211 22211 =&gt; 122</p> |

## 5.3 Vertical Zero Crossings



Vertical zero crossing is same as the horizontal zero crossing. In this process the input Image block is divide into vertical segments. Finally apply the segmentation algorithm as for the Horizontal Zero Crossing. By this process we find another 3-digit vertical code. The only difference in between two Zero crossing algorithm is horizontal zero crossing process the block of character that contain bellow the header throw the end of the line and the vertical zero crossing process the block of character that contain the whole line that means upper strip if any, core strip and lower strip if any.



#### 5.4 Character Recognition from the Generated Code sequence:

From the previous portion we have got the Category code of length 5 and the Character sequence code of length 6 which is divide in two sub group one is horizontal zero crossing code and the other is vertical zero crossing code . So we got a total sequence of code of length 11 to be matched to recognize the character.

| Criteria Code (5 Digit)     |  |                                   |   |                                       | Horizontal Code (3 Digit) | Vertical Code (3 Digit) | Unicode |
|-----------------------------|--|-----------------------------------|---|---------------------------------------|---------------------------|-------------------------|---------|
| Header<br>1 = Yes<br>0 = No | Bar<br>1=Pre bar<br>2=Mid bar<br>3=End bar<br>4=No bar | Upper Modifier<br>1= Yes<br>0= No | Connection with header<br>1 = Yes<br>0 = No | Existence of Dot<br>1 = Yes<br>0 = No |                           |                         |         |
| 1                           | 2  | 0                                 | 1   | 0                                     | 133                       | 122                     | ক 004b  |
| 0                           | 3  | 0                                 | 0   | 0                                     | 121                       | 232                     | খ 004c  |

In the above table we display the sample character code that we have built by applying our proposed algorithm. It may vary from font to font with different size and style. But we found good accuracy in result in our experiment when apply a specific font named “ **SutonnyMJ** ” in regular font size(14 px).

## 6. Experiments

In the experiment we have given sample input Image containing different sized font only character set. The charter set is divided in plain set which is combine of vowel and consonants and other set is contains of different sized conjuncts. As for information Bangla script contain more then 600 joint character. Among them 120 conjuncts is frequently used. We have successfully recognize 50+ of them based on their criteria that we have built up. The font size varies from regular size(14) to large size (32).

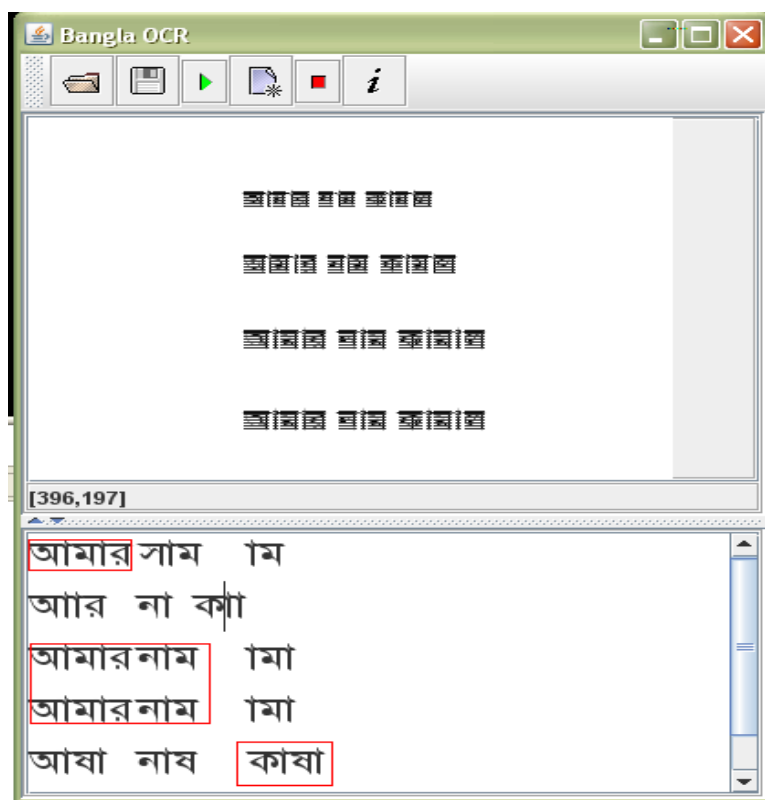


Fig: Developed toolkit for performance evaluation

| Font size | Character [ Plain] | Accuracy of Recognition(%) | Character [ Conjuncts] | Accuracy of Recognition(%) |
|-----------|--------------------|----------------------------|------------------------|----------------------------|
| 14        | 50                 | 99.9                       | 50                     | 99.9                       |
| 16        | 100                | 85                         | 100                    | 90                         |
| 18        | 150                | 87                         | 150                    | 87                         |

|    |     |    |     |    |
|----|-----|----|-----|----|
| 20 | 200 | 90 | 200 | 92 |
| 22 | 250 | 92 | 250 | 93 |
| 24 | 300 | 86 | 300 | 87 |
| 26 | 350 | 82 | 350 | 79 |
| 28 | 400 | 90 | 400 | 92 |
| 30 | 450 | 81 | 450 | 85 |
| 32 | 500 | 88 | 500 | 95 |

*Table: Performance and character recognition*

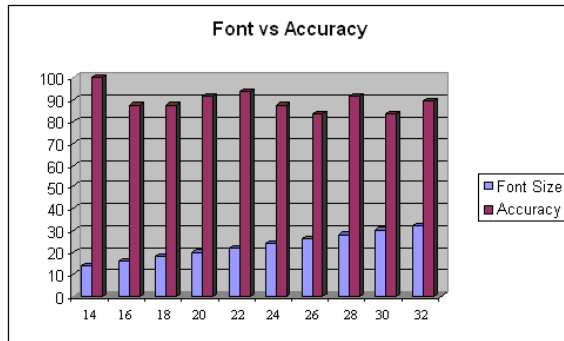


Fig: Result for **simple character**

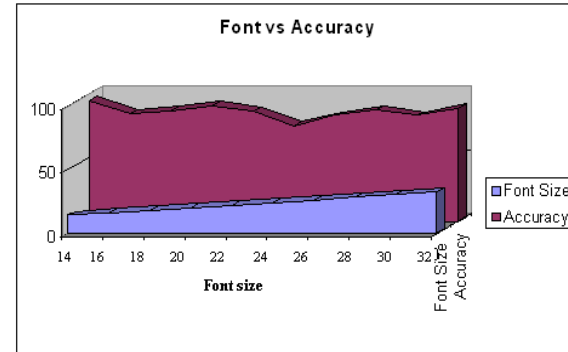


Fig: Result for **conjunct character**

## 7. Conclusions and Future Work

Bangla language in a computer is still lag behind because of it's complexity .Bangla literature has a large number of characters and greater problem of joint letters. Our primary concern is to achieve for recognition of regular sized font with a reasonably good accuracy rate at least for the frequently occurring core characters. Consequently, our future efforts will be directed towards calibrating the recognition of a word that is segmented from a sentence to improve performance. We also need to resolve problems related to conjunct and over segmented characters in the context of noisy images. By integrating the OCR with a well structured dictionary during the post processing phase, we hope to be able to like up the character recognition rate by making a successful use of domain knowledge.

## Acknowledgments

The authors would like to express their gratitude to M. Shahidur Rahman (Associate professor, Dept of CSE, SUST) for being very helpful and his constant support and encouragement. And Muhammed Zafar Iqbal (Professor, Head of Dept of CSE, SUST) because of his continuous interest on Bangla OCR.

## References

- [1] R. C Gonzalez and P. Wintz, Digital Image Processing, 2nd ed. Reading, MA: Addison Wesley, 1987.
- [2] S. Mori et al, Historical Review of OCR Research and Development\_ Proceedings IEEE , vol, 80 no. 7 pp . 1029-1058, July 1992.
- [3] E. Persoon and K. S. Fu, Shape determination using Fourier descriptors, IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-7, 170-179, 1977.
- [4] S.Wendling et al, Use of Harr transform and some of its properties in character recognition, Proceedings International Joint Conference on Pattern Recognition (IJCPR) pp. 844-848,1976
- [5] R. Ott, On feature selection by means of principal axis transform and nonlinear classification, Proceedings International Joint Conference on Pattern Recognition (IJCPR), pp. 220-222, 1974
- [6] R. M. K. Sinha, Computer processing of Indian languages and scripts -potentialities and problems, Journal of Institution of Electronics & Telecommunication Engineers, vol. 30, 133-49,1984.
- [7] J. C. Sant and S. K. Mullick, Handwritten Devanagari script recognition using CTNNSE algorithm, International Conference on Application of Information Technology in South Asian Language, February 1994.
- [8]B. B. Chaudhuri and U. Pal, A Complete Printed Bangla OCR System, Pattern Recognition, vol. 31 no. 5 pp . 531-549,1997.
- [9]B. B. Chaudhuri and U. Pal, A Complete Printed Bangla OCR System, Pattern Recognition, vol. 31no .5, pp . 531-549,1997.
- [10] U.Pal and B. B. Chaudhuri, An improved Document Skew Angle Estimation Technique, Pattern Recognition Letters , 17, pp. 899-904,1996.
- [11] Alamgir, Mohammed, Molla, Md. Khademul Islam and Iqbal Muhammed Zafar, Bangla Character Recognition System, ICCIT'99, SUST, Sylhet, Bangladesh, 3-5 December 1999, P-159-163.
- [12] Md. Khademul Islam Molla and Kamrul Hasan Talukder, ICCIT'2002, East West University, 27-28 December 2002, P-200-2006