Bengali Optical Character Recognition using Self Organizing Map

[1]Muhammad Golam Kibria, [2]Al-Imtiaz
[1 2]Department of CSE
[1 2]University of Information Technology and Sciences, Dhaka, Bangladesh
[1]golam.kibria@uits.edu.bd, [2]imtiaz@itz-one.com

Abstract: Being the 5th Position and sweetest language in the world declared by the UNESCO Bengali is the national language in Bangladesh and one of the major languages in India. Lot of researches has been done to recognize Bengali, English and other major languages using Optical Character Recognition (OCR). To recognize Bengali character from text images and convert into editable text, Self Organizing Map (SOM) – kind of neural network has been used. To collect the character, documents are scanned, which is preprocessed with the Image to Binary Conversion Algorithm. In the binary image, character area is represented by 0 (zero) and rest of the image area is represented with 1 (one). After detecting and correcting the skew and noise, the binary image is processed and grouped, which can be mapped and recognized by SOM. Considering efficiency and fastness, character grouping process has been introduced.

I. INTRODUCTION

Converting hard document, such as newspaper, printed book into editable text to modify or extend is the normal practice nowadays, OCR is the process of converting printed text images into editable text. Optical Character Recognition using optical techniques such as mirrors & lenses and Digital Character Recognition using scanners and computer algorithms were originally considered separate fields. Since few applications survive using true optical techniques, the OCR has been broadened to digital image processing as well.

Bengali OCR involves reading text from paper and translating the images into a form (say ASCII code/Unicode) that the computer can manipulate. OCR system is still in preliminary level in case of language like Bengali, cause of its complexities in character shapes, top bars and end bars. More over it has some modified vowel and compound characters.

II. PROPERTIES OF BENGALI CHARACTER

There are 11 vowels and 39 consonant characters and 10 digits in Bengali language. Most of the characters have a horizontal line at the upper level defined in Fig 1 [1].

Some common properties in Bengali language are given below:

1. Writing style of Bengali is from left to right.
2. No upper and lower case in Bengali language.
3. Vowels takes modified shape called modifiers or allograph [2,4]. as shown in Table 1.
4. There are approximately 253 compound characters composed of 2, 3 or 4 consonants [3] as shown in Table 2.
5. All Bengali characters and symbols have a horizontal line at the upper part called "Matra" that remain connected with another character as in Fig 1.
6. Bengali characters contain three zones which are upper zone, middle zone and lower zone as in Fig 1 [3].
7. Some characters including some modifiers, punctuation etc. have vertical stroke. [2].
8. Most of the character has the property of intersection of two lines in different position, other has one or more corner or sharp angle [7] as in Table 3.

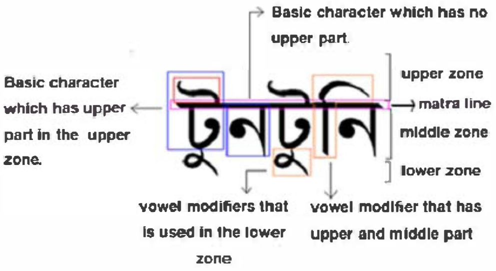

Basic character which has no upper part
Basic character which has upper part in the upper zone.
upper zone
matra line
middle zone
lower zone
vowel modifiers that is used in the lower zone
vowel modifier that has upper and middle part

Fig 1: Dissection of Bengali word [1]

In Bengali script sometimes a vowel takes a modified shape depending on the position in a word. If the first character of the word is a vowel then it retains its basic shape as in Fig 2(a). Generally a vowel followed by a consonant takes a modified shape and placed at the left or right or both or bottom of the consonant shown in Table 1.

Table 1: Example of modified shape of vowel

Vowel	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
Modified shape	া	ি	ী	ু	ূ	ৃ	ে	ৈ	ো	ৌ
ক + vowel	কা	কি	কী	কু	কূ	কৃ	কে	কৈ	কো	কৌ

For two consecutive vowels in a word, the second one remains its basic shape when the first one in modified shape as shown in Fig 2(b).

আ+ম=আম (a)   খ+আ+ও = খাও (b)

Fig 2: Non-modified vowel in a word

A consonant or vowel followed by a consonant sometimes takes a compound shape known as compound character as shown in Table 2.

Compounding three consonants is possible, where if the order of two consonants is changed then the compound character is also changed [1].

Table 2: Example of compound character

ক + ক	ক্ক	ব + ব	ব্ব
ক + ত	ক্ত	ব + দ	ব্দ
ক + ন	ক্ন	চ + ছ	চ্ছ
ক + ম	ক্ম	চ + ছ + ব	চ্ছ্ব
ক + ষ	ক্ষ	জ + জ	জ্জ
ক + ষ + ন	ক্ষ্ন	জ + জ + ব	জ্জ্ব
ক + ষ + ম	ক্ষ্ম	জ + ঞ	জ্ঞ
ক + ল	ক্ল	ঞ + জ	ঞ্জ
ল + ক	ল্ক	ঙ + ক	ঙ্ক

Important statistics on Bengali character are given below:

1. Average length of Bengali words is about six characters.
2. About 30% - 35% of characters are vowel modifiers which, being small in size, contribute very little to the head line of the word.
3. Most of the basic characters are consonants, as vowels in basic form can appear at the beginning of the word or when two vowels appear side by side as shown in Fig 2 (b).
4. Compound characters are very infrequent, occurring in about 5% of the cases only.
5. In Bengali 41 characters can appear in the first position of a word. Out of these 41 characters 30 of them have headlines. Probability of getting a character with head line in the first position of a word is: p1= 30/41 and getting a character without head line in the first position is: p1=1-p1=11/41.
6. In other positions of a word, there are mostly consonants and 28 out of 39 Bengali consonants have headlines (matraline). Probability of getting a consonant with head line for other positions in a word is: p2=28/39 and probability of getting a character without head line in other positions is: p2=1-p2=11/39.

III. BENGALI CHARACTER RECOGNITION

The procedural steps to recognize the Bengali character are defined in the flow chart as shown in Fig 3.

A. Data Collection

The scanned image is taken as raw input from where the character level segmentation will be executed. PDF file also can be taken as raw input.

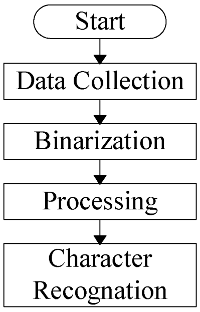


Fig 3: Flow chart to recognize Bengali character

To extract characters from scanned images it is necessary to convert the image into a proper digital image. This process is called text digitization. The process of text digitization can be performed either by a Flat-bed scanner or a hand-held scanner. Hand held scanner typically has a low resolution range. Appropriate resolution level typically 300-1000 dots per inch for better accuracy of text extraction [8].

B. Binarization

Binarization is the technique by which the images are converted in to binary images based on the pixel value. The pixel that make up the letter only require one bit of data each. Based on the pixel value black or white image will be replaced by either with 0 or 1 respectively. Binarization process which is generated from the algorithm that extracts data from the scanned image file and convert the data defined in the flowchart in Fig 4.

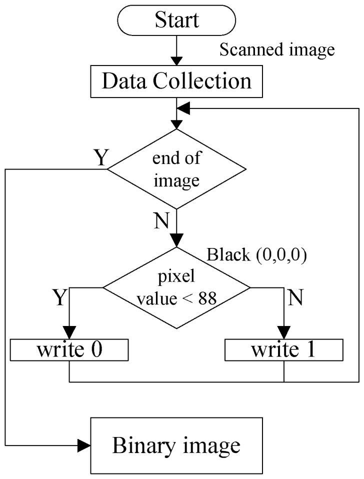


Fig 4. Binarization process

Some binarization methods such as, Global Fixed Threshold, Otsu Global Algorithm, Niblack's Algorithm and Adaptive Niblack's Algorithm are discussed in [5, 6].

A digital text image containing Bengali character is generally an RGB image. Scanned image containing digital Bengali character called "soreo" is shown in Fig 5.

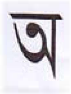
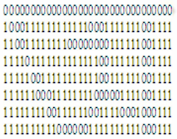


Fig 5: Scanned image of "soreo"

The scanned image is converted into binary image.First, the scanned image is saved to bitmap image, and then the algorithm is applied to convert the bitmap image into binary image. Bitmap is one of the types of file formats to store images in a computerized form. It carries the extension .BMP. Computers use bits of 1 and 0 to store the data.

A scanned image converted into binary image after binarization process looks like the Fig 6.



```
00000000000000000000000000000000
10001111111111110001111111001111
11001111111100000000111111001111
11110111111111111001111111001111
11111001111111111100111111001111
11111100011111111100000111001111
11111111001111110011110001000111
11111111110000001111111111000111
```

Fig 6: Binary image of Bengali character "soreo"

In the next section processing of the algorithm is discussed.

IV. PROCESSING

If the scanned image contains any skew and noise that will be remain in the binary image. During scanning, noise can be added, such as dot that is not part of the text, any unwanted line and so on. Character recognition becomes easier if the character is in normal form, but within the text it is common case for the character to be in Italic or in other style. That's why before mapping the character, skew and noise detection and correction is processed.

The steps to process the scanned image are defined in flowchart shown in Fig 7.

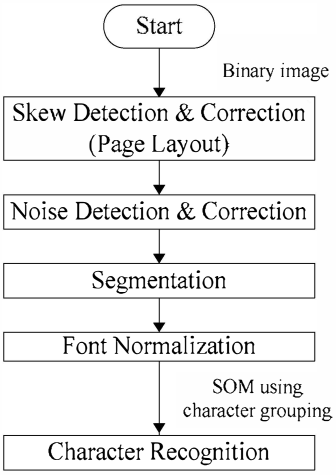

Start
Binary image
Skew Detection & Correction (Page Layout)
Noise Detection & Correction
Segmentation
Font Normalization
SOM using character grouping
Character Recognition

Fig 7. Flowchart to Process from Binary image

A. Skew detection and correction

Sometimes digitized image may be skewed and in such case skew correction is necessary to make text lines horizontal. Skew correction can be achieved in two steps. First, estimate the skew angle θt and second, rotate the image by θt, in the opposite direction [8]. Here detecting the skew angle is using Matra. Two types of methods discussed in [12] & [13] for this purpose. Hough transform technique may be applied on the upper envelopes for skew estimation, but this is slow process. So in [12] an approach is discussed which is fast, accurate and robust. The idea is based on the detection of DSL segments from the upper envelope. In [13], they applied Radon transform to the upper envelope to get the skew angle. Radon transform and the Hough transform are related but not the same. Then applied generic rotation algorithm for skew correction and then applied bi-cubic interpolation.

B. Noise detection and correction

During the scanning the quality of the image might be decreased, hence some noise might be added. Two types of noises are generated, such as background noise and paper noise. Complex script like Bengali, wide pixels from the upper or lower portion of a character cannot be eliminated, because it might be a part of a character, such as উ or উঁ into same character. Lot of methods to detect noise is available as given below:

1. Dots existing in a character like য় may be treated as noise. In [9], the author proposed a new method for removing noises similar to dots from printed documents. In this method, first estimate the size of the dots in each region of the text. Then the minimum size of dots in each region is estimated based on the estimated size of dot in that region.

2. In [20], the authors used connected component information and eliminated the noise using statistical analysis for background noise removal [10].

3. In [11], noise is removed from character images. Noise removal includes removal of single pixel component and removal of stair case effect after scaling.

Upper zone and lower zone are defined in Fig 1. It is very important to detect the blank space in between two lines as noises might be added during scanning or for other reason and eliminated those noises. Elimination can be achieved by removing any unwanted elements in between the lower zone of above line and the upper zone of below line as in Fig 8.

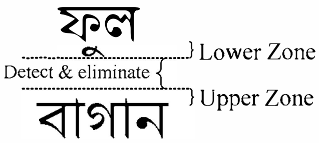


Fig 8. Noise detection and elimination

C. Segmentation

Mapping to recognize the character is done individually. Hence, they should be segmented from the scanned image to line, then from line to word and finally from word to character. To segment the line from scanned image it is mandatory to detect the Matra.

Line Segmentation: The lines of a text block are detected by scanning the input image horizontally. Frequency of binary values (0 for black) in each row is counted in order to construct the row histogram. To segment the individual line from the segmented image, it is required to find out the headline (Matra). The row with the highest frequency of '0' (zero) is detected as matraline or headline. For the larger font size, it is observed that the height or thickness of the matraline increases. In order to detect the matraline with its full height, the rows with those frequencies are also treated as matraline [11].

Word Segmentation: When a line has been detected, then each line is scanned vertically for word segmentation. Number of binary value-'0' in each column is calculated to construct column histogram. When all binary values-'1' are found in vertical line that is considered as the space between two words. Fig 9 shows how the word is segmented.

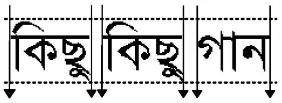


Fig 9. Word Segmentation

During this process, a typical situation may occur when matraless character is found in a word [14]. In such case 20% of blank space can be considered to segment words if a character is considered as 100%.

Character Segmentation: In [1], the construction of word is defined. To segment the word both the Matra and base line of the word needs to be detected. The process to detect the base line is described in [14].

The main parts of all the characters are placed in the middle zone. So the middle zone area is considered as the character segmentation portion. Since Matra line connects the characters together to form a word, it is ignored during the character segmentation process to get them topologically disconnected [15]. A word constructed with basic characters is segmented into characters in a way by scanning vertically [14].

A word in Bengali language is formed using characters and modifiers. To segment the character, these modifiers need to be detected. There are four kinds of modifiers based on their uses. One kind of modifiers known as Middle Zone Modifiers, used only in the middle zone, for example and the Lower Zone Modifiers are used only in the lower zone, such as . Modifier called Upper and Middle Zone Modifiers consists of . The last kind of modifier is called the Upper Zone Modifier, such as [1]. Lower zone modifiers should be below the base line and must be connected to a character. In case of upper and lower zone modifiers, modifiers are joined with Matra at a single point and they take less space than that of for the character vertically as well as these modifiers are not connected with the character at the middle zone. Upper zone modifiers should be at the top of the Matra line and not be connected.

Most of the characters are connected to Matra. Hence, three types of characters can be identified, one who is connected to Matra on top of the character, second who does not have any Matra and third who is connected to Matra on its top right side. Character connected to Matra on its top right side is shown in Fig 10.

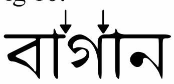


Fig 10. Character segmentation for Matraless

The character segmentation process including modifiers has been described in details in [11, 14, 15 & 16].

D. Font Normalization

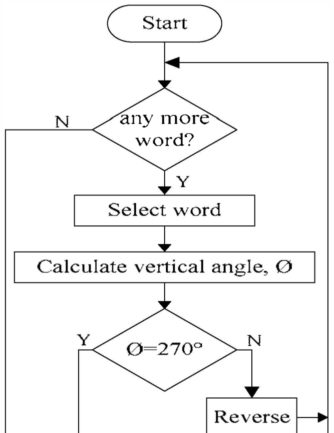


Fig 11. Font normalization

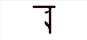
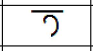
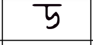
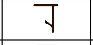
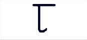
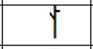
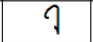
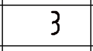
Words or character in the text might be in Italic or other style, but not all of them. It is necessary to transform the Italic into normal font for character grouping and recognition.

So, after the skew detection and correction for the page layout and word segmentation, the process needs to detect the skew for the words or characters in Italic form and normalize them. The process to detect the skewed word or character in Italic and normalize them is defined in flowchart in Fig 11.

E. Character Grouping

Characters in Bengali language are complex in many ways. Each of the character is formed using straight lines or curvature or/and compound among characters. Based on its shape the Bengali character can be grouped. It would be better for the SOM to classify the character if the binary image below the Matra is identified first. Different groups of the Characters have been identified in Table 3.

Table 3: Character Grouping

Group's Name	Identical shape	Member									
ব		ব	র	ক	ধ	ঝ	ঋ				
ত		ত	অ	আ	হ	ই	ঈ				
ড		ড	ড়	উ	ঊ	জ	ঙ				
য		য	য়	ষ	ফ	ঘ	ন	ম	থ	খ	
ঢ		ঢ	ঢ়	ট	চ						
গ		গ	প	ণ	শ	ল	স				
এ		এ	ঐ	ঞ							
ও		ও	ঔ								
Others		ছ	ঠ	দ	ভ	ৎ	ং	ঃ			
Number		০	১	২	৩	৪	৫	৬	৭	৮	৯

ব – Group: one straight vertical lines joined with Matraline and two other lies are connected to this vertical line.

য – Group: one right sided vertical line is connected with another line at its end

ঢ – Group: one left sided vertical line connected to Matra that turns to right when closing to its base line.

Other character groupings have their own characteristics based on the identical shape of the characters.

V. SELF ORGANIZING MAP (SOM)

The main idea is to make it simple and acceptable for SOM which is also known as kohonen neural network [19]. The SOM contains no hidden layer. The SOM differs from the feed forward back propagation neural network in several important ways [7].

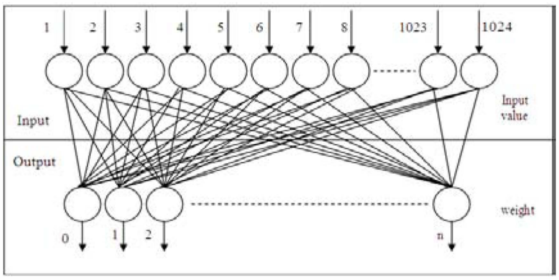


Fig 12: SOM execution process

Number of neuron depends on the number of vector. Vector length for an input layer of 1024 has 1024 neurons. But in the output layer the number of neuron depends on the number of character trained with the network. If 1024 is considered as the input and *n* for the output character, the suitable SOM is shown in Fig 12.

SOM is unsupervised machine learning that learns by self-organizing and competition [17]. It reduces a remarkable amount of time. SOM is clustering the input vector by calculating neuron weight vector according to some measure (e.g. Euclidean distance), thus weight vector that closet to input vector comes out as winning neuron. However, instead of updating only the winning neuron, all neurons within a certain neighborhood of the winning neuron are updated using the Kohonen rule [17].

Suppose the training set has sample vectors X, to choosing the wining neuron these vectors are trained in SOM network, the algorithm is defined in [18].

In Table 3, character grouping is defined. Characters are grouped based on its identical shape. SOM will map the identical shape of the character first to match with the appropriate group which leads SOM to map only the members within that group.

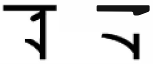


Fig 13. Error in scanned image

Since the characters are grouped based on their identical shape, it will be easier for the SOM to recognize the character even if the character has error as in Fig 13.

VI. RESULTS

Performance of the SOM has been increased due to the character grouping. There are 60 characters including numerical number in Bengali. It would take 60 times to check and map in conventional way. But due to grouping SOM requires maximum 20 times to check and map a character which can save up to 33% of total recognition time. Accuracy has been increased as the font normalization is done during processing the binary image.

VII. CONCLUSION

Character Grouping and font normalization have been introduced in the paper. The Noise detection process has been improved that eleminates unwanted dots or noises in between two lines. Due to introducing the grouping, the total character recognition process becomes faster as it saves time for mapping and classifying the character. Our future work in this regard will be recognizing the compound characters and handwriting. Objectives have been justified by introducing and improving certain processes.

REFERENCES

[1] Farjana Yeasmin Omee , Shiam Shabbir Himel and Md. Abu Naser Bikas,, *"A Complete Workflow for Development of Bengali OCR"*, International Journal of Computer Applications (0975 – 8887) ,Volume 21– No.9, May 2011

[2] U. Garain and B. B. Chaudhuri, *"Segmentation of Touching Characters in Printed Devnagari and Bengali Scripts using Fuzzy Multifactorial Analysis"*, IEEE Transactions on Systems, Man and Cybernetics, vol.32, pp. 449-459, Nov. 2002.

[3] Minhaz Fahim Zibran, Arif Tanvir, Rajiullah Shammi and Ms. Abdus Sattar, *"Computer Representation of Bengali Characters And Sorting of Bengali Words"*, Proc. ICCIT" 2002 , 27-28 December, East West University, Dhaka, Bengalidesh.

[4] A. B. M. Abdullah and A. Rahman, *"Spell Checking for Bengali Languages: An Implementation Perspective"*, Proc. of 6th ICCIT, 2003, pp. 856-860.

[5] J. He, Q. D. M. Do*, A. C. Downton and J. H. Kim, *"A Comparison of Binarization Methods for Historical Archive Documents"*.

[6] Tushar Patnaik, Shalu Gupta, Deepak Arya, *"Comparison of Binarization Algorithm in Indian Language OCR"*.

[7] Teuvo Kohonen, *"The self-Organizing Map"*, IEEE Invited paper.

[8] Md. MahbubAlam and Dr. M. AbulKashem, *"A Complete Bangla OCR System for Printed Chracters"*, JCIT-100707.pdf

[9] M.HassanShirali-Shahreza, SajadShirali-Shahreza, *"Removing Noises Similar to Dots from Persian Scanned Documents"*, Computing, Communication, Control, and Management, 2008. CCCM '08. ISECS International Colloquium on Issue Date: 3-4 Aug. 2008 On page(s): 313- 317

[10] Tinku Acharya and Ajoy K. Ray (2005). *"Image Processing Principles and Applications"*, John Wiley & Sons, Inc., Hoboken, New Jersey

[11] J. U. Mahmud, M. F. Rahman and C. M. Rahman (2003). *"A Complete OCR System for Continuous Bengali Characters"*, IEEE,PP. 1372-1376

[12] B.B. Chaudhuri and U. Pal, *"Skew Angle Detection Of Digitized Indian Script Documents"*, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 19, pp.182-186, 1997.

[13] S. M. MurtozaHabib, Nawsher Ahmed Noor and Mumit Khan, *"Skew Angle Detection of Bangla script using Radon Transform"*, Proc. of 9th ICCIT, 2006.

[14] Nasreen Akter, Saima Hossain, Md. Tajul Islam & Hasan Sarwar (2008). *"An Algorithm For Segmenting Modifies From Bangla Text"*, ICCIT, IEEE, Khulna,Bangladesh, PP.177-182

[15] B.B. Chaudhuri & U. Pal (1998), *"Complete Printed Bangla OCR System"*, Elsevier Science Ltd. Pattern Recognition, Vol(31): 531-549

[16] Md. Al Mehedi Hasan, Md. Abdul Alim, Md. Wahedul Islam & M. Ganger Ali, *"Bangla Text Extraction and Recognition from Textual Image"*, NCCPB, Bangladesh, PP.171-176, 2005

[17] Kohonen, T. (1990), *"The Self-organizing map*", Proc. IEEE, vol. 78, no. 9, 1464-1480.

[18] R.Indra Gandhi, Dr.K.Iyakutti, *"An Attempt to Recognize Handwritten Tamil Character Using Kohonen SOM"*, Int. J. of Advance d Networking and Applications, Volume: 01 Issue: 03 Pages: 188-192 (2009)

[19] Teuvo Kohonen, "The Self Organizing Map", Proc IEEE, Vol 78, No. 9, September 1990.

[20] Md. AbulHasnat, S M MurtozaHabib and MumitKhan."A high performance domain specific OCR for Bangla script", *Int. Joint Conf. on Computer, Information, and Systems Sciences, and Engineering (CISSE)*, 2007.