A Complete OCR System for Continuous Bengali Characters

Jalal Uddin Mahmud, Mohammed Feroz Raihan and Chowdhury Mofizur Rahman
Department of Computer Science and Engineering
Bangladesh University of Engineering & Technology
880-2-9665612(7109)
mrony@accesstel.net,fraihan2002@yahoo.com, cmrbuet@yahoo.com

Abstract ---This paper is concerned with a complete optical character recognition (OCR) system for Bengali character. Recognition is done for both isolated and continuous printed multi font Bengali characters. Preprocessing steps includes segmentation in various levels, noise removal and scaling. Free man chain code has been calculated from scaled character which is further processed to obtain a discriminating set of feature vectors for the recognizer. The unknown samples are classified using feed forward neural network based recognition scheme. It has been found from experimental results that success rate is approximately 98% for isolated characters and 96% for continuous character.

Index Terms: Chain code, preprocessing, segmentation, feature extraction, neural network, Back propagation, Connected component.

1. INTRODUCTION

There has been particular interest over the last decade in recognition of printed characters. Recognition of printed character is itself a challenging problem since there is a variation of the same character due to change of fonts or introduction of noise. Recognition of Bengali character is a subject of special interest for us and many works have been done in this area. Various strategies have been proposed by different authors. Multi font character recognition scheme suggested by Kahan and Pavlidis[1]. Roy and Chatterjee[2] presented a nearest neighbor classifier for Bengali characters employing features extracted by a string connectivity criterion. Abhijit Datta and Santanu Chaudhuri [3] suggested a curvature based feature extraction strategy for both printed and handwritten Bengali characters. B.B . Chaudhuri and U.Pal [4] combined primitive analysis with template matching to detect compound Bengali characters . Most of the works on Bengali character are recognition of isolated characters. Very few deal with a complete OCR for printed document in Bengali. From that standpoint, this paper will effectively promote the researchers who are interested to develop a complete OCR system. In this work, continuous characters are segmented using some traditional methodology as well as some new methodology in various levels and other preprocessing is performed to find a stream of isolated character. We have adopted the chain code method of image representation [5], which allows a compact representation and reduction of data and hence processing time. Chain code is a linear structure that results from quantization of the trajectory traced by the centers of adjacent boundary elements in an image array. Feature extraction from chain code representation can be effectively used for recognition of the character image. As chain code representation gives the boundary of the character image, thickness of the character is useless in such case. Thinning of the character image is needless when chain code representation is used. Trajectory traced by the chain code implies the shape of the character that can be further processed for feature extraction. Slope distribution of chain code implies the curvature properties of the character that has been used as local feature. Further, use of the back propagation based learning scheme in the recognition strategy enables the system to learn from examples. The generalizing capability of this learning scheme has been harnessed to achieve font invariant recognition of the characters. In section 2 we present the methodology of the system. In section 3 we discuss the techniques used for pre-processing. Section 4 is concerned with the feature extraction strategies. In section 5 the training and recognition scheme has been discussed. Empirical results have been presented in section 6 and conclusion is drawn in section 7.

2. METHODOLOGY

The basic objective of the present scheme is to develop a complete OCR(optical character recognition) system for different fonts and sizes of Bengali characters. Difference in font and sizes makes recognition task difficult if preprocessing, feature extraction and recognition are not robust. There may be noise pixels that are introduced due to scanning of the image. Besides, same font and size can have bold face character and normal one. So width of the stroke is also a factor that affects recognition. So a good character recognition approach must eliminate the noise after reading binary image data, smooth the image for better recognition, extract features efficiently, train the system and classify patterns. In the next figure a typical Bengali script is shown.

0-7803-7651-X/03/$17.00 © 2003 IEEE

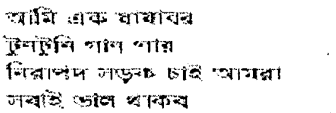


Fig 1: A Typical Bengali Script

Before recognizing isolated character, preprocessing steps segments the paragraphs into lines, lines into words and words into isolated characters. Noise removal and smoothing constitutes the next step. Then connected components within each character have been detected using depth first approach. Each connected component has been divided into four regions depending on the center of mass of each component. Distribution of directional slopes of Freeman chain code in each region makes the feature set for that region of the individual connected component.

3. PREPROCESSING

In the present system character images have been obtained by optical scanning of the character images on the plain paper. The input data obtained by scanning of printed text is almost contaminated with noise and contains redundant information. Preprocessing includes segmentation, scaling, noise removal and elimination of redundant information as far as possible.

3.1 Segmentation

Segmentation of the input binary image data in different level is performed. The segmentation is done in the following steps:

3.1.1 Text Line Detection----Text line detection has been performed by scanning the input page image horizontally. Frequency of black pixels in each row is counted in order to construct the row histogram .The position between two consecutive lines, where the number of pixels in a row is zero denotes a boundary between the lines. Here it is assumed that the text block contains only single column of text. Fig.2 shows the line segmentation process.

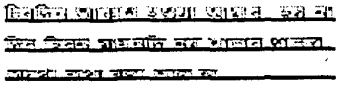


Fig.2: Text line segmentation

3.1.2 Word Segmentation---After a line has been detected, it is scanned vertically. In order to find the column histogram, number of black pixels in each column is calculated. 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 taken experimentally. Fig 3 shows the word segmentation process.

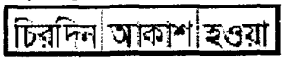


Fig.3: Word Segmentation

3.1.3 Character Segmentation---To segment the individual character in a word we first find the head line of the word which is called 'Matra' in Bengali. From the word, row histogram is constructed by counting frequency of each row in the word. The row with the highest frequency value indicates that head line which has been denoted as 'Matra' . Some times there are consecutive two or more rows with almost same frequency value. In that case , 'Matra' row is not a single row. Rather all the rows that are consecutive to the highest frequency row and have frequency very close to that row constitutes the 'Matra' which is now a thick headline. To find the demarcation line between characters a vertical scan is initiated from the row that is just beneath the 'Matra' row. If the scan can reach the bottom of the word then it has successfully found the demarcation line between characters. But only linear vertical scan fails to find the demarcation line for some words. In this case linear searching to find the demarcation line between characters fails for the presence of a portion of the next adjacent character in the column associated for the current character. To overcome this situation, we have used an approach where the vertical scan is not linear only. It is piecewise linear in the sense that the scan takes turns whenever it sees an obstacle (i.e. black pixel) and tries to reach the bottom of the word. Fig 4 shows the approach.

Fig 4.a Linear scan for character segmentation

Fig 4.b Ordinary linear search fails to segment

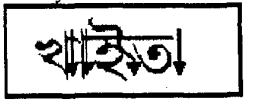


Fig 4.c Piecewise linear scan for segmentation

To find the portion of any character above the Matra we then check if we can move upward from the Matra row from a point- just adjacent to the Matra row and between the two demarcation lines. If it is, then a Greedy search is initiated from that point and the whole character is found.

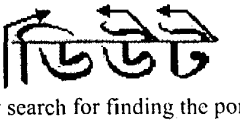


Fig 5: Greedy search for finding the portion of the character above Matra row

Besides of these difficulties , there are characters which are positioned below another character. To segment the characters below another character base line of the word image has been calculated. Each word can be considered to have an imaginary line that crosses at the middle of the word. If a greedy search is initiated from the word image that searches for presence of black pixels below the imaginary line then the search will result some connected

components below that imaginary line. All those components contain a lowest point which are called 'Base Point' . Base line is the highest frequency row of those points. After determining the base line, a depth Base line is the highest frequency row of those points. After determining the base line, a depth first search easily extracts the characters below base line.

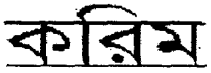


Fig 6. Base Line of the word image

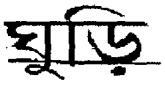


Fig 7. Extracting characters below base line

3.2 Scaling

Scaling of the isolated character has been performed so that size invariant recognition can be possible. Though the recognition is size invariant, better result is obtained when the characters are assumed to be in a specific range. For our system , the range of characters has been taken as 10 pt from 24 pt. The characters that are scaled using an efficient scaling algorithm [6] converted to standard size which is 64 x 64 for our system.

3.3 Noise Removal

After scanning, because of unwanted noise, arbitrary extrusions and intrusions may be found at the boundary of the character images. Noisy cavities in the character images are also common. These distortions detrimentally affect the shape of the characters. Noise removal includes removal of single pixel component and removal of stair case effect after scaling. Stair case effect occurs when the scaled character have junctions so thin that inner and outer contour required for chain code representation cannot be found. Each pixel has been replaced by a filtering function to avoid such effect.

4. FEATURE EXTRACTION

Feature extraction is the most challenging part for character recognition and choice of good features significantly improves the recognition rate and minimizes the error in case of noise. The steps of feature extraction has been discussed below:

4.1 Connected Components Extraction

In Bengali language a character can have more than one connected component. In these characters recognition of two components actually yields the desired result. There fore all the connected components are detected from the isolated character. Detection of connected component is done using depth first search approach.

4.2 Center of Mass for Each Component

Center of mass has been calculated for each connected component. Center of mass for i th connected component is (Xi,Yi). where

$$Xi = \sum_{j=1}^{Ni} Pij / Ni \quad \text{.....(1)}$$

$$Yi = \sum_{j=1}^{Ni} Qij / Ni \quad \text{.....(2)}$$

Here,
Ni = Number of Black pixels in connected component i.
Pij = x Coordinate of the jth Black pixel in ith connected component.
Qij = y Coordinate of the jth Black pixel in ith connected component.

4.3 Bounded Rectangle Calculation

If the grid is searched in a row wise manner, and connected component is found using depth first search strategy, top left and bottom right co ordinate of a component can be found. Besides its minimum and maximum span in x direction as well as in y direction can be found which results a bounded rectangle of the component.

4.4 Division of the Components into Regions

Each connected component has been divided into four regions indicating four quadrants in two-dimensional geometric system. The origin of the two dimensional geometric system is the center of mass of that connected component. With the origin and the bounded rectangle of the connected component, four regions can be established.

Fig 8: Four Regions for a connected component.

4.5 Chain Code Generation:

After the character has been divided into connected components and boundary of the connected components are established, chain code is to be calculated .There are several chain code convention used for image representation, but the most popular one is Freeman chain code. Freeman Chain code is based on the observation that each pixel has eight neighborhood pixels.

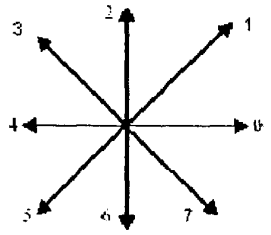


Fig 9: Slope Convention for Freeman Chain code

The 8 transitional positions defined by freeman chain code are then divided into 4 transitional zones in order to facilitate the searching and to keep the correct order of searching.

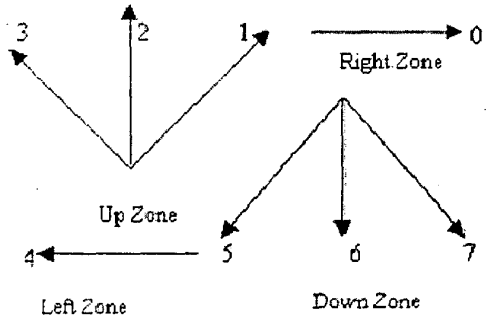


Fig 10: 4 direction zones for searching

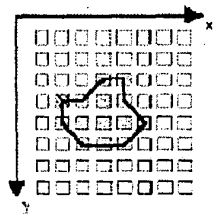


Fig 11: Determining the chain code of an image

4.6 Slope Distribution Generation :

When searching for a closed contour continues, there is a variation of slope in each region. The frequency of each directional slope at each region is recorded and updated during the traversal. There are eight directional slopes in a region, therefore total 32 directional slope for the whole component. The frequency of jth directional slope at ith region is local feature S i,j ,where j = 0,1, ... ,7 and i = 0,1,2,3.

4.7 Normalized Slope Calculation:

In order to obtain fractional value, feature values must be normalized to (0-1) scale. The rule for normalization is: If a1, a2, a3, ... , an are n feature vectors in n dimensional feature space, then their normalized values are $\bar{a}_1$, $\bar{a}_2$, ..., $\bar{a}_n$.

Here $\bar{a}_1 = a_1/N$, $\bar{a}_2 = a_2/N$, ... $\bar{a}_n = a_n/N$.
$N = \sqrt{(a1^2 + a2^2 + ... + an^2)}$.

4.8 Conversion to Character Slope Distribution:

If there is more than one connected componént in the character, then 32 normalized slope for each connected component will be found after the previous step . But recognition step recognizes the whole character, not its individual connected component. Therefore normalized features for each connected components are averaged to get the total features for the character.

5. CLASSIFICATION AND RECOGNITION

Neural network approach has been used for classification and recognition. Training and recognition phase of the neural network has been performed using conventional back propagation algorithm.

5.1 Training

Neural network has been trained by normalized feature vector obtained for each character in the training set. Four-layer neural network has been used with two hidden layers for improving the classification capability of the neural network with minimum error tolerance rate. For 32 dimensional feature vector and 4 layer, number of neuron used in the first hidden layer is 90 and that in the second hidden layer is 75.

5.2 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. The confidence value obtained from the output layer of the neural network, which closes to 1 implies the presence of that character class. Except the confidence value of the recognized character, other confidence values are closes to 0.

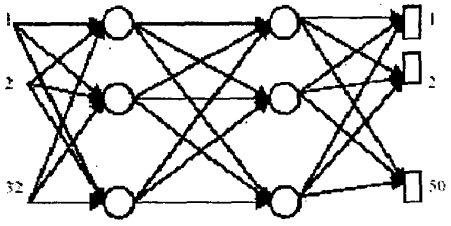
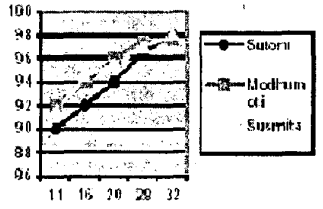


Fig 12: A Neural Network with 4 layers, 90 neurons and 75 neurons in hidden layers.

6. EMPIRICAL RESULT

The system has been tested extensively for both isolated and continuous characters. Many test files have been generated for this purpose. Different complexity levels of test have been used. The degree of accuracy in recognition rate is presented chart 1. Training of the system was performed

with three types of fonts : Sulekha, Susri and Sunetra. Test files were generated with sample from training fonts as well as from some unknown fonts for the system. Chart2 presents success rates when test files contained multiple fonts with varying sizes. Recognition rate is superior for isolated characters than for continuous characters. The discrepancy in performance in case of continuous characters may have happened due to segmentation procedure.



Comparison of Recognition Rate for various font sizes

Chart 1: Success Rates in Recognizing Continuous Characters for unknown fonts for the system

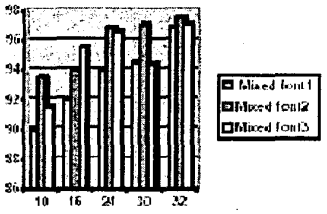


Chart 2: Success Rates in Recognizing Continuous Characters for mixed fonts with various sizes.

7. CONCLUSION

A fast chain code based optical character recognition (OCR) system for Bengali alphabet has been presented and implemented. Performance analysis has also been given. Development of efficient methods for preprocessing, segmentation and feature extraction resulted an increase of speed. Contour representation is suitable for representing characters both in printed and handwritten. Though we are concerned only with printed characters in this paper, the same methodology can be used for recognition of handwritten characters. Obviously in that case, preprocessing and segmentation algorithms have to consider more practical problems than printed character recognition.

REFERENCES

[1] S. Kahan and T.Pavlidis, "Recognition of printed characters of any font and size", *IEEE Trans. Pattern Anal. And Mach.Intell.9,274-288*,1987.

[2]. A.K.Roy and B.Chatterjee, "Design of nearest neighbor classifier for Bengali character recognition", *J.IEEE 30*,1984.

[3]. Abhijit Dutta and Santanu Chaudhury, "Bengali Alpha-Numeric Character Recognition Using Curvature Features", *Pattern Recognition Vol-26, 1707-1720* ,1993.

[4]. B.B.Chaudhuri and U.Pal , "A Complete Printed Bangla OCR System", *Pattern Recognition Vol-31, 531-549* ,1997.

[5]. H. Freeman , "Computer Processing of Line Drawing Images",*ComputingSurveys, Vol –6, no –1,pp –57-97*,1974.

[6]. Suman Kumar Nath and Muhammad Mashroor Ali, "An Efficient Object Scaling Algorithm for raster device", *Graphics and Image Processing, NCCIS* ,1997.