

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Development of a recognizer for Bangla text: present status and future challenges

Saima Hossain, Nasreen Akter,
Hasan Sarwar and Chowdhury Mofizur Rahman
*United International University
Bangladesh*

1. Introduction

Optical Character Recognition (OCR) System, by virtue of its usefulness, has emerged as a major research area since 1950. Now it is becoming a more challenging issue all over the world to have efficient and more accurate recognizers. There are many widely spoken languages in the world like Chinese, Arabic, Hindi, English, Spanish, Bangla, Russian, Japanese etc. Bangla is one of the most widely spoken languages, ranking 5th in the world. 21st February is observed as the international mother language day to pay homage to the martyrs fought for the establishment of Bangla as the mother tongue of Bangladesh. With the automation everywhere, it is a burning issue to digitize huge, volume of Bangla documents by using an efficient OCR. However as of today there is no such good recognizer available for Bangla compared to other languages. From 80s, it took huge interest and now becomes as a major research area particularly in Bangladesh and India. Lots of works have been done in different sections of pattern recognition tasks (i.e, pre-processing, segmentation, feature extraction, classification) but there is a lack of synchronization between these works. That is why we put our effort into a comprehensive review of the current status of research to develop an all-inclusive Bangla OCR which will enable one to understand the difficulties and challenges involved, to know how much progress has been done and to estimate what more to be done to come out with a successful Bangla OCR.

2. Overview

This chapter includes introduction to Bangla language which provides a brief description on it, present status on the development of Bangla OCR and some comparative analysis of several proposed methods on Noise Reduction, Skew Detection and Correction, Segmentation, Feature Extraction, Classification published in different articles. Future works in this field explaining the futures challenges are summarised at the end of this chapter.

3. Introduction to Bangla Language

Bangla is an eastern Indo-Aryan language and evolved from Sanskrit (Barbara F. Grimes, 1997). The direction of the writing policy is left to right. Bangla language consists of 50 basic characters including 11 vowels and 39 consonant characters and 10 numerals. In Bangla, the concept of upper case or lower case letter is not present. Bangla basic characters have characteristics that differ from other languages. Bangla character has headline which is called matraline or matra in Bangla. It is a horizontal line and always situated at the upper portion of the character. Among basic characters, there are 8 characters which are with half matra, 10 characters with no matra and rest of them with full matra. All consonants except ঙ এঃ ণ ঙ্ ঙ্ ঙ্ ঙ্ ঙ্ are used as the starting character of a word whereas, vowels are used everywhere. Vowels and consonants have their modified shapes called vowel modifiers and consonant modifiers respectively. Both types of modifiers are used only with consonant characters. There are 10 vowels and 3 consonant modifiers which are used before or after a consonant character, or at the upper or lower portion of a consonant character or on the both sides of a consonant character, likewise, খা খি খু খে খৌ ক্য কঁ ক্র . In Bangla, some special characters are there which are formed by combining two or more consonants and acts as an individual character. These types of characters are known as compound characters. The compound characters may further be classified as touching characters and fused characters. Two characters placed adjacent contact to each other produce a touching character. Touches occur due to horizontal placement of only two characters and/or vertical placement of two or more characters. About 10 touching characters are there in Bangla. Fused characters are formed with more than one basic character. Unlike touching characters, the basic characters lose their original shapes fully or partly. A new shape is used for the fused characters. In sum, there are about 250 special characters in Bangla except basic and modified characters. Table 3.1 illustrates different types of Bangla characters.

Vowels	অ আ ই ঈ উ ঊ ঋ ঌ ঍ ঐ ঔ ঔ
Consonants	ক খ গ ঘ ঙ চ ছ জ ঝ ঞ ট ঠ ড ঢ ণ ত থ দ ধ ন প ফ ব ভ ম য র ল শ ষ স হ ড় ঢ় ঞ্ ঞ্ ঞ্
Vowel Modifiers	া ি ি ু ূ ে ৈ ো ৌ
Vowel Modifiers attached with consonants	খা খি খী খু খূ খ্ খে খৈ খৌ খৌ
Consonant Modifiers	্য ঁ ং
Consonant Modifiers attached with consonants	ক্য কঁ ক্র

Compound Characters: Horizontal Touching Characters	ড + ড = ডড ব + ব = বব হ + ব = হব চ + চ = চচ চ + ছ = চছ খ + ব = খব ট + ন = টন ঠ + ন = ঠন
Compound Characters: Vertical Touching Characters	ক্কা ক্ক ক্ক ক্ক ক্ক ক্ক ক্ক ক্ক ক্ক ক্ক গ্গ গ্গ গ্গ গ্গ গ্গ গ্গ গ্গ গ্গ গ্গ খ্খ খ্খ খ্খ খ্খ খ্খ খ্খ খ্খ খ্খ খ্খ ঙঙ ডড ডড টি টি থু থু থু থু থু থু জ্জ জ্জ জ্জ জ্জ জ্জ জ্জ জ্জ জ্জ জ্জ
Compound Characters: Fused Characters	ট্ট ক্ক ক্ক ক্ক ক্ক ক্ক ক্ক ক্ক ক্ক ক্ক গ্গ গ্গ গ্গ গ্গ গ্গ গ্গ গ্গ গ্গ গ্গ খ্খ খ্খ খ্খ খ্খ খ্খ খ্খ খ্খ খ্খ খ্খ ঙঙ ডড ডড টি টি থু থু থু থু থু থু জ্জ জ্জ জ্জ জ্জ জ্জ জ্জ জ্জ জ্জ জ্জ
Numerals	০ ১ ২ ৩ ৪ ৫ ৬ ৭ ৮ ৯

Table 3.1. Different types of Bangla characters. A subset of 112 compound characters out of about 250 characters (B.B. Chaudhuri, 1998) is shown here.

The occurrence of vowels and consonants are larger compared to special characters in most of the Bangla documents. A statistical analysis, shown in Table 3.2, took 2 sets of data populated with 100,000 words from Bangla books, newspapers and 60,000 words from Bangla dictionary respectively (B.B. Chaudhuri, 1998).

Global characteristics	1 st set of data	2 nd set of data
Vowel characters	38.70%	37.10%
Consonant characters	61.30%	62.90%
Compound characters	4.10%	7.20%
Entropy of the alphabet	4.07 (bits/char)	3.54 (bits/char)
Average word length	5.73 (char)	4.99 (char)
Words with suffix	69.80%	0.00%

Table 3.2. Statistical analysis on the occurrence of different characters

Some of the modifiers are there which are used on top of the character (basic or special character) as well as a few in the bottom of the character. Again some basic characters also

have upper portion which is treated as a part of the character. So it can be said that the construction of Bangla characters require 3 zones named upper zone, middle zone and lower zone. Upper portion from matra line is called upper zone. Middle portion that is situated under the matraline is called middle zone. As some modifiers are used at the bottom of the middle zone, this portion is called lower zone. Most of the characters are situated in the middle zone. Fig. 3.1 shows a simple example explaining the construction of a Bangla word.

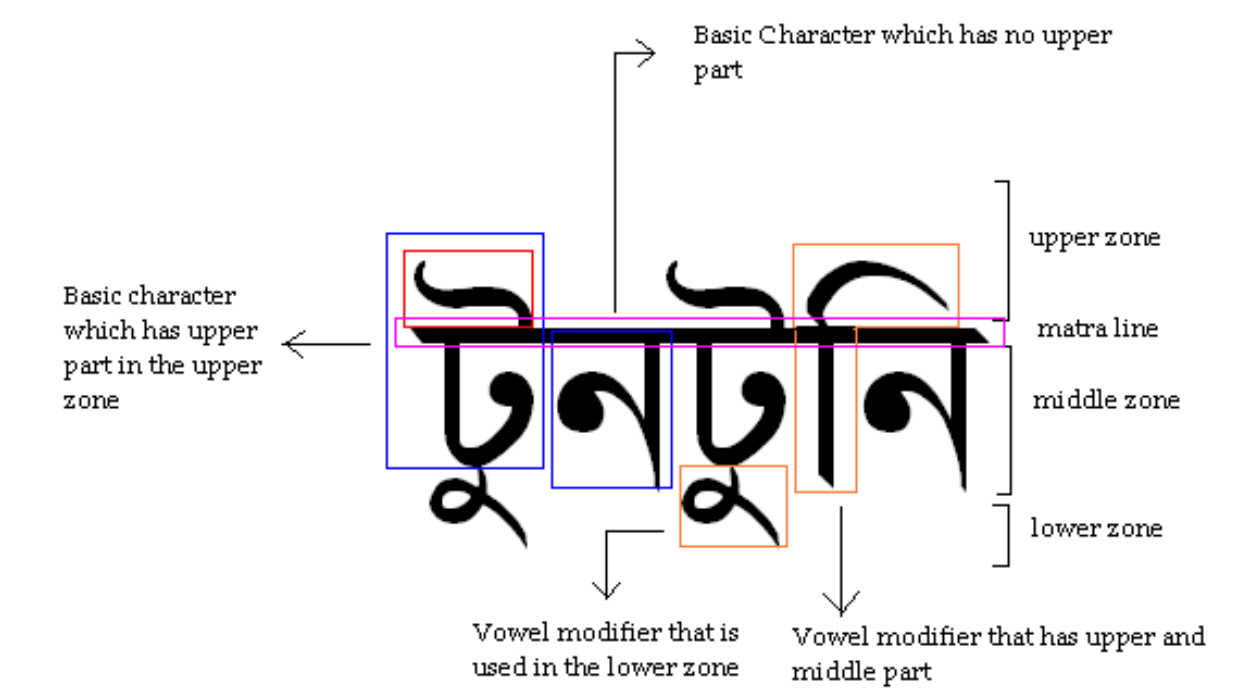


Fig. 3.1. Dissection of a Bangla Word

4. Scanning and Image Digitization

Before going into the ocr process, one must scan the paper through a flat-bed scanner. It is better not to use hand-held scanner, which may create local fluctuation for hand movement (B.B. Chaudhuri, 1998). It is crucial to have good quality printed document scanning. If the quality is poor and the color contrast is too low, it will be hard for the OCR software to read the text and to make correct interpretation. The scanned image is stored, for example, as a jpeg/bmp format file which is converted to a binary image. In order to improve the quality of the image to make the OCR correct interpretation, noise reduction and elimination and skew detection and correction processes are performed.

5. Noise Detection and Removal

Noise is naturally added during scanning process. When documents or papers are scanned, some noises are added automatically into it. There are two different types of noises known as background noise and salt and pepper noise which are given most importance. A histogram-based thresholding approach is used to convert gray tone into two-tone images

(B.B. Chaudhuri, 1998). The histogram shows two prominent peaks corresponding to white and black regions. The threshold value is chosen as the midpoint of the two histogram peaks. The two-tone image is converted into 0-1 labels where 1 and 0 represent object and background, respectively. Authors also claim that their approaches are better than J. N. Kapur, P. K. Sahoo and A. K. C. Wong (J. N. Kapur, 1985), and N. Otsu (N. Otsu 1979). However, salt and pepper noise is not mentioned here. Different authors have suggested for applying noise elimination process at different stages of Preprocessing or Segmentation. (B.B. Chaudhuri, 1998) and (A. Roy, 2002) used it during binarization of the image.

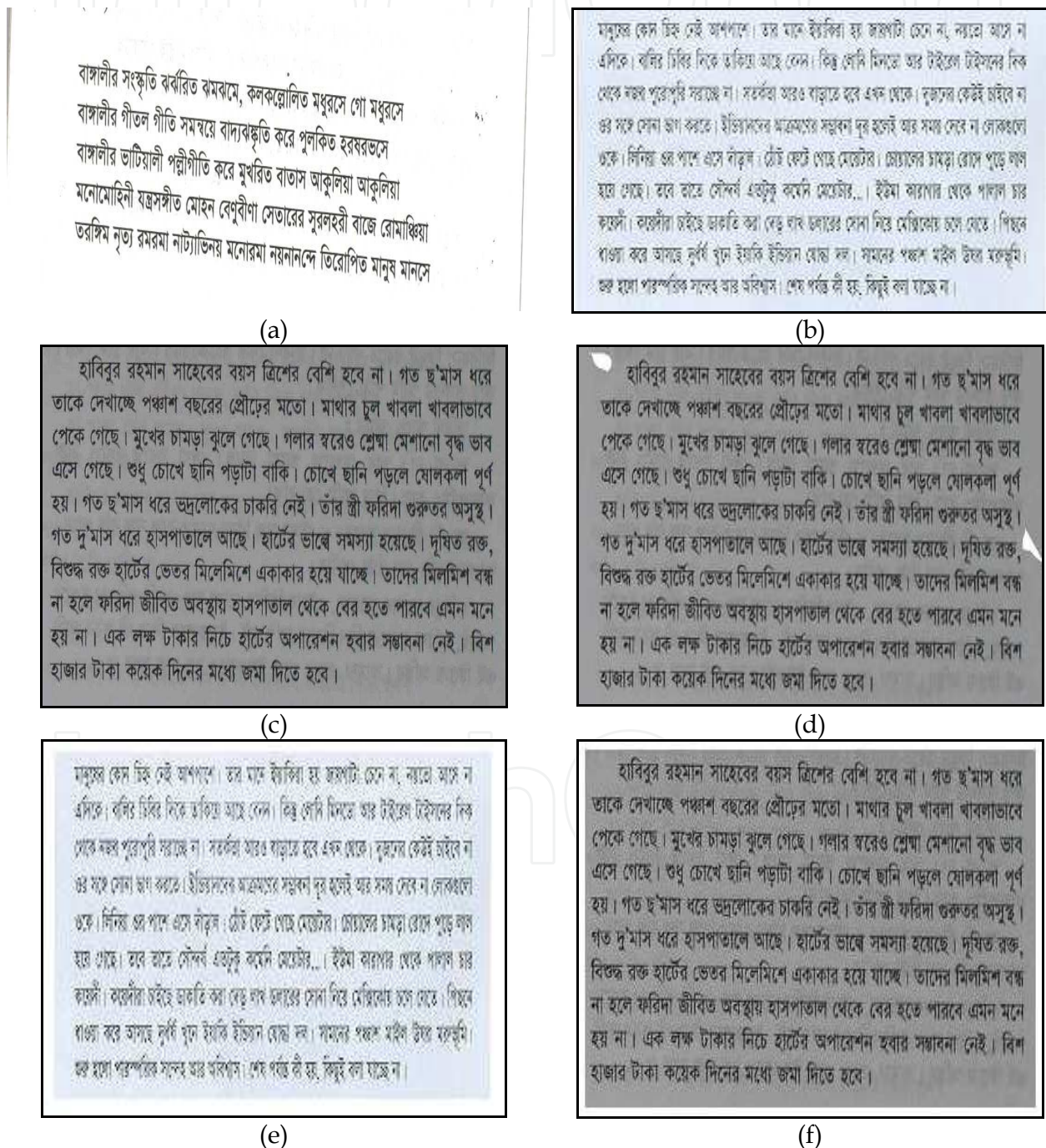


Fig. 5.1. Images (a) – (f) with different types of noises

In (A. Roy, 2002), gray-level images are median filtered and then Otsu’s thresholding algorithm is used to binarize the images of word. The binary images are then filtered to obtain smooth images. In (Jalal Uddin Mahmud, 2003), noise is removed from character images. Noise removal includes removal of single pixel component and removal of stair case effect after scaling. Stair case effect occurs when the scaled characters 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. However, this does not consider background noise and salt and pepper noise. In (Md. Abul Hasnat, 2007), the authors used connected component information and eliminated the noise using statistical analysis for background noise removal. For other type of noise removal and smoothing they used wiener and median filters (Tinku Acharya, 2005). Connected component information is found using boundary finding method (such as edge detection technique). Pixels are sampled only where the boundary probability is high. This method requires elaboration in the case where the characteristics change along the boundary. A comparative performance study in Table 5.1 is given below for some of the images shown in fig. 5.1.

Papers	Fig. 5.1a	Fig. 5.1b	Fig. 5.1c	Fig. 5.1d	Fig. 5.1e	Fig. 5.1f
B.B. Chaudhuri, 1998	yes	yes	x	x	yes	x
A.Roy, 2002	yes	yes	yes	yes	yes	x
Jalal Uddin Mahmud, 2003	yes	yes	Yes	yes	yes	x
Md. Abul Hasnat, 2007	yes	yes	yes	yes	yes	x

Table 5.1. Applicability of different techniques on noisy images shown in fig. 5.1

6. Skew Detection and Correction

Skew is basically an angle that is created due to an angular placement of document in the scanner. (B.B. Chaudhuri, 1998) says that it can be corrected in two steps, i) estimation of skew angle θ_s and ii) rotation of image by θ_s in the opposite direction. Many skew detection and correction algorithms are available. B.B. Chaudhuri (1998) suggests an approach suitable for Bangla scripts. Basically, it tries to detect the head line of document words. Head line is a straight line given on the upper side of a character when used in words. It has been found that average length of Bangla words is six characters (R. M. Bozinovic, 1989), 30-35% of characters are vowel modifiers having very little contribution to head word, 5% is compound character. In Bangla, 41 characters can appear in the first position, of them 30 characters have headlines. Probabilistic analysis reveals that, in 99.39% cases, there will be at least one character with head line in a Bangla word. In this process, firstly a bounding box (an upright rectangle containing a word/component) is defined. The mean box width is b_m and the standard deviation is b_s . Components having boundary box equal to b_m and less than $b_m + 3b_s$ are retained, others are discarded, which fall into categories like dots, punctuation marks, isolated characters and characters without headlines. Next, upper envelop of selected component, G, is found. From each pixel of the uppermost row of the bounding box, a vertical scan is performed until a pixel labelled G is encountered; it is converted into U label, known as the upper envelope. Hough transform technique may be

applied on the upper envelopes for skew estimation. (B.B. Chaudhuri, 1998) has suggested a new idea which is faster, robust and accurate compared to Hough transform. The idea is based on Digital Straight Line (DSL). The upper envelope may contain non-linear parts which require deletion, for which chain code representation has been used. Conditions for straightness of chain code digital arc are given in (J. N. Kapur, 1985).

A subset of DSL is known as SDSL. SDSL consists of runs of pixels in at most two directions which differ by 45° . For runs of two directions, the run lengths in one of the directions are always one. The run length in the other direction can have at most two values differing by unity. An example is shown in fig-6.1. Here, the angle between two directions $d1$ and $d2$ is 45° and run lengths in $d1$ direction are two (n) or three ($n+1$) occurring alternately.

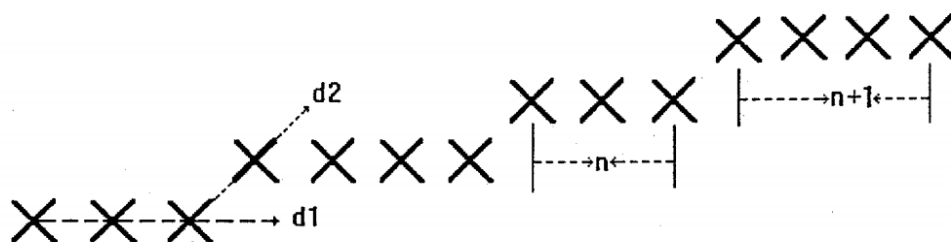


Fig. 6.1. Example of a digital straight line (DSL). Here, "X" denotes DSL pixel

A reasonable number of SDSLs can be used to find the average estimate of skew angle, which is formed between the first and last pixel of every SDSL with horizontal direction. To get a better estimate, clusters of SDSLs are found. Each line of text contains a number of SDSLs, which are grouped into a cluster. The SDSLs that have equal normal distances from a reference point are considered to be members of same cluster. The leftmost and the rightmost pixels of each cluster is taken and considered as the leftmost and rightmost coordinate of a line represented by that cluster. The Skew estimation algorithm is:

- STEP 1: Find connected components in the binary document image and find the mean b_m and standard deviation b_s of their bounding box widths.
- STEP 2: Choose the set S of connected components having bounding box width greater than or equal to b_m and less than $b_m + 3b_s$.
- STEP 3: For each component in S find the upper envelope described above. From each envelope component, find the SDSLs. If more than one SDSL is found choose only the longest one and form the subset R_1 . Let the longest SDSL in R_1 be C_L .
- STEP 4: From the line C_L or its continuation, find the normal distances to the leftmost pixel of other SDSLs of R_1 .
- STEP 5: Cluster the SDSLs of R_1 corresponding to individual text lines and find the leftmost and rightmost pixel of each cluster, as described above.
- STEP 6: For each cluster find the angle of line joining the leftmost and rightmost pixels (e.g., A and B in Fig. 6.2d) with horizontal direction.
- STEP 7: Average of such angles over all clusters (e.g., text lines) gives an accurate estimate of the skew angle.

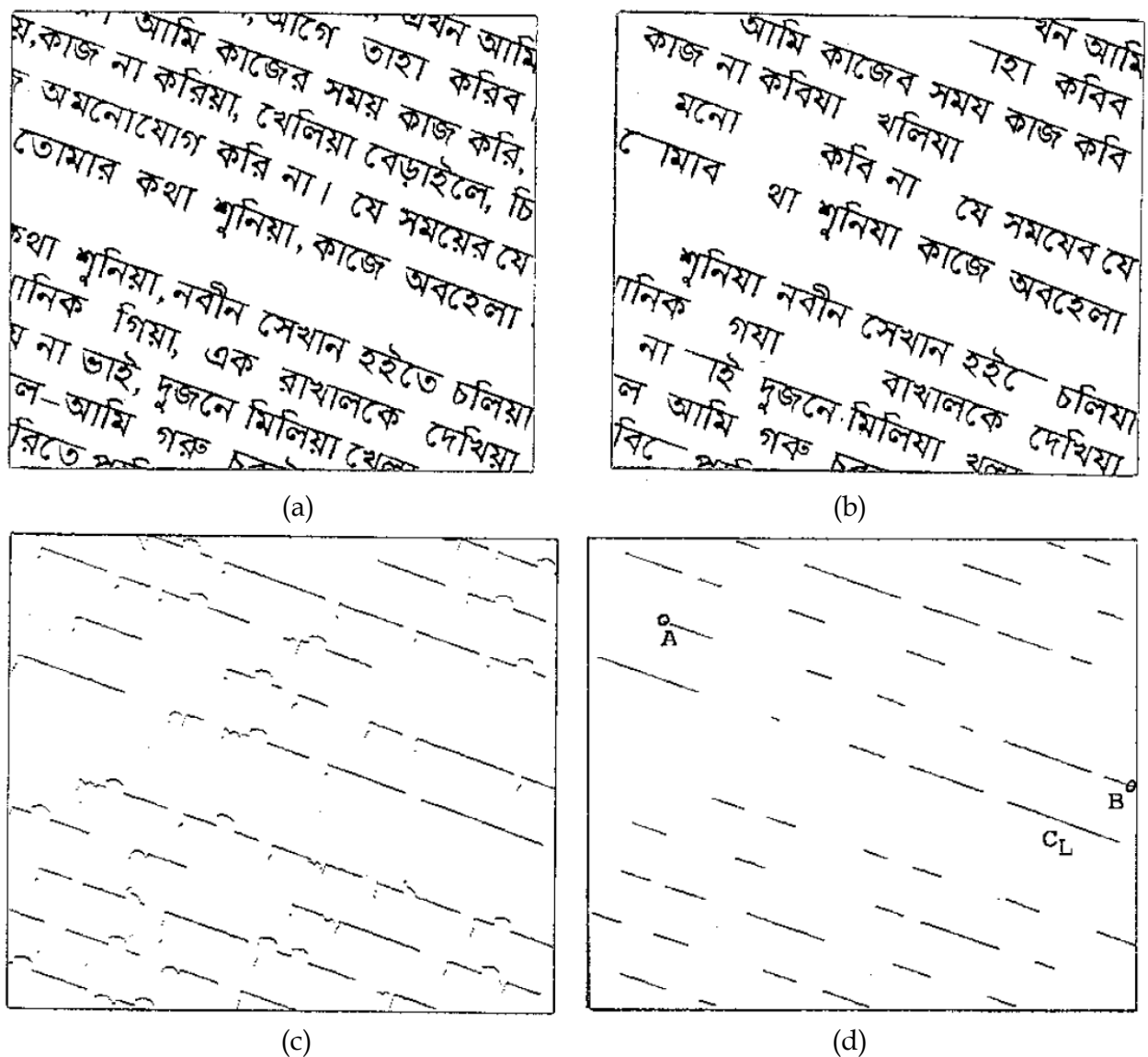


Fig. 6.2. Skew detection approach (Bangla). (a) An example of Bangla skewed text; (b) Selected components from fig. 6.2a; (c) Upper envelope of selected components of fig. 6.2b; (d) SDSL components of fig. 6.2c.

The efficiency of the above described method (A3) in table 6.1 has been measured and compared with Hough transform applied on original document (A1) and on SDSLs (A2), shown in that table. The average execution times for a document of 512X512 pixels on a SUN 3/60 (with Microprocessor MC68020, and SUN O.S. Version 3.0) machine are 620, 312, and 17.80 seconds for methods A1, A2 and A3, respectively shown by (B.B. Chaudhuri, 1997). It is shown that the methods A2 and A3 are statistically equally accurate. However method A3 takes less execution time.

True skew angle (in deg.) (manual)	Mean and SD of estimated skew angles using method					
	A ₁		A ₂		A ₃	
	mean	SD	mean	SD	mean	SD
40	40.396	0.285	40.034	0.256	39.889	0.301
20	20.174	0.439	20.049	0.3162	20.047	0.242
10	10.271	0.393	10.166	0.201	10.112	0.323
5	5.064	0.458	4.962	0.213	5.188	0.233
2	1.986	0.396	2.151	0.234	2.054	0.307

For each true skew angle the statistics is computed over 20 document images.

- A₁: Hough transform over total image.
A₂: Hough transform over SDSLs of upper envelop.
A₃: Proposed quick method.

Table 6.1. Mean and Standard Deviation (SD) of Estimated Skew Angles Obtained by Different Methods shown by (B.B. Chaudhuri, 1997)

7. Segmentation

This is the most vital and important portion for designing an efficient Bangla OCR because feature extraction and recognition process depends on this phase to make the recognition process successful. The output of this phase consists of individual images of basic, modified and compound characters. Segmentation process includes the following steps. They are:

- 1. Line Detection
- 2. Matraline or Headline detection
- 3. Baseline Detection
- 4. Word Segmentation
- 5. Character Segmentation

7.1 Line Detection Process

Generally, a document is written in multiple lines considering one or more than one columns. In this chapter, one columned document is considered. The lines of a text block are detected by finding continuous white pixels between two consecutive matralines. Fig: 7.1.1 shows the result.

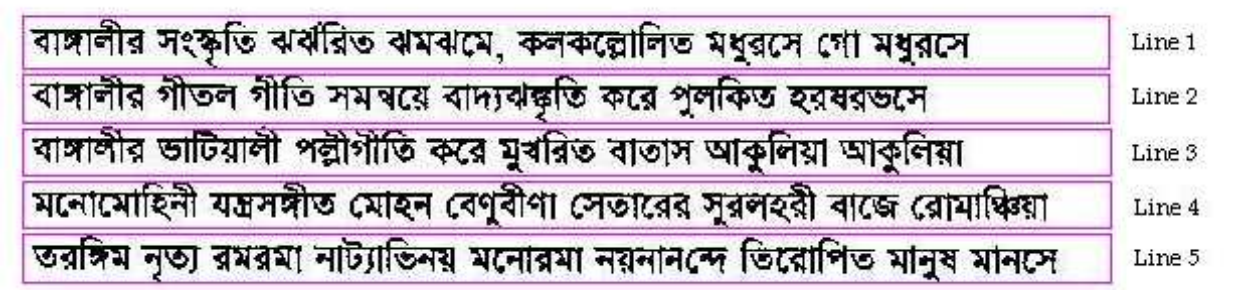


Fig. 7.1.1. Text line detection in a document image

7.2 Matraline or Headline Detection Process

Matraline or headline is an important and distinct feature in bangla. It connects the bangla components together. The matraline consists of highest number of black pixels compared to the upper, middle and lower zone. Under the matraline, the basic shapes of the characters are found. So, if matraline can be detected correctly, it helps to segment the characters in a more flexible way. The row with the highest frequency of black pixels is detected as matraline or headline. It is observed that the height or thickness of the matraline increases in case of larger font size. In those cases we get more rows having similar frequency or nearly close to the row of highest frequency. In order to detect the matraline with its full height, the rows with those frequencies are also treated as matraline (Jalal Uddin Mahmud, 2003). Thus we can say that matraline consists of matra upper line and matra bottom line. Fig. 7.2.1 shows the detection of matarline for different images, (a) and (b).

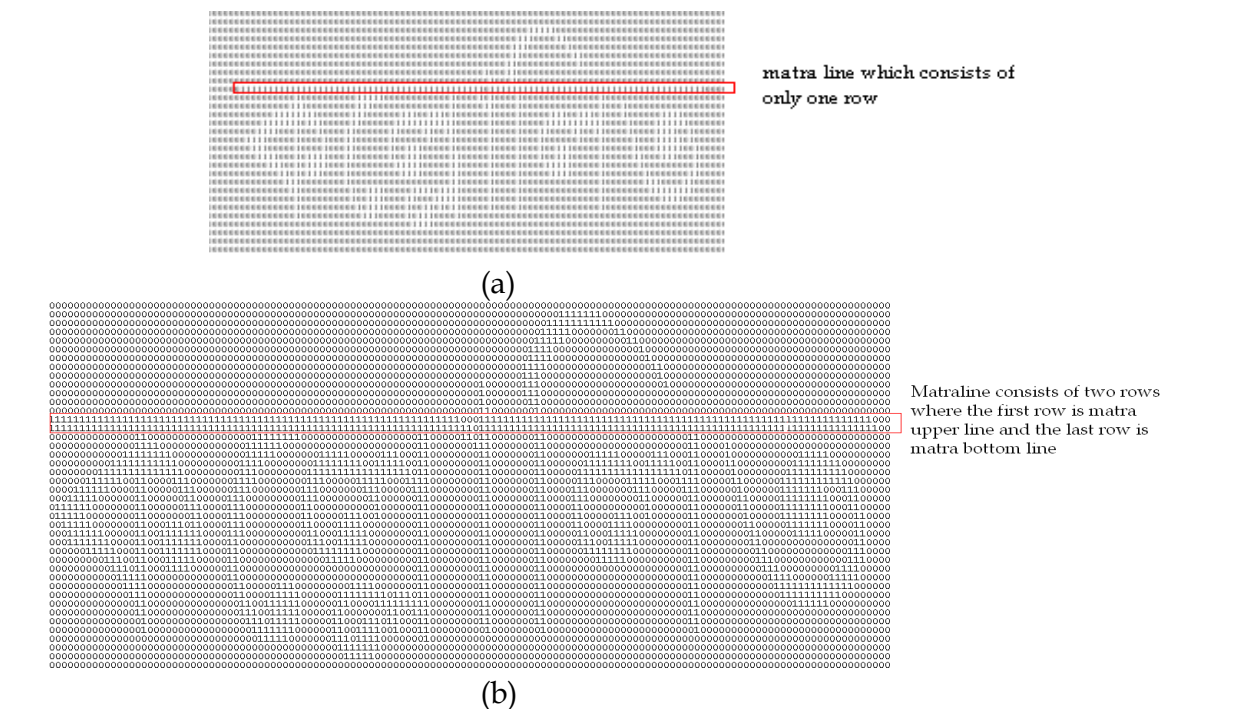


Fig. 7.2.1. Matraline detection process for different mages (a) - (b)

7.3 Baseline Detection Process

A line has a baseline, also named as an imaginary line, which is a row from where the middle zone ends and lower zone starts i.e. a separator between middle and lower zone. Baseline gets equal to end row of the line when the line does not have any lower modifier(s). It is the row where an abrupt change occurs between the previous and next row (S.M. Milky Mahmud, 2004). In a general document, it is observed that about 70% lines hold baseline i.e., 30% lines do not have lower modifiers. So detection of baseline is very important for bangla. Some characters particularly some modifiers are situated here and they are needed to be recognized. Baseline can be detected efficiently searching from lower position of the middle zone to the end row of a line. This is to find out the position from where the black pixels start to increase while they are decreasing in the middle zone. That position is denoted as baseline (Nasreen Akter, 2008). Fig. 7.3.1 shows the baseline.

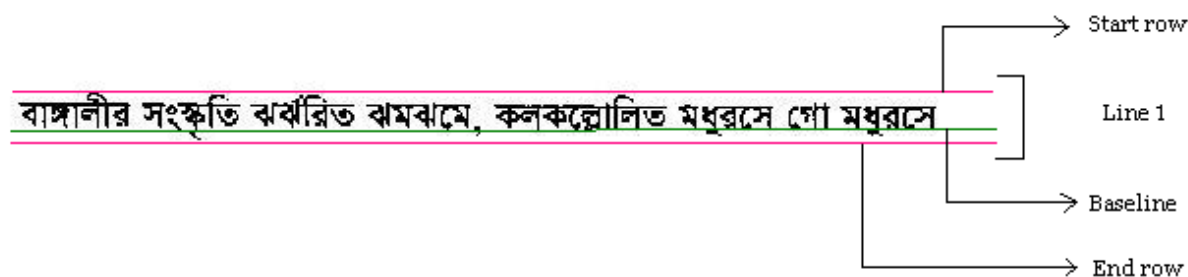


Fig. 7.3.1. Process of baseline detection

7.4 Word Segmentation Process

There are always some white spaces between two words in a text line. Using vertical scan, words are separated by treating the white spaces as a separator. Fig. 7.4.1 shows the word segmentation process.



Fig. 7.4.1. Process of word segmentation

During the word segmentation process, peculiar situations occur when some matraless character is used in a word (fig. 7.4.2). These situations create some false separators which cause the word to be broken into small pieces. So the process of word segmentation becomes faulty. To avoid this error, widths of each separator in a line is calculated and an average of them is found. Separators which have widths greater than or equal to the half of the average are considered to be true separator (Nasreen Akter, 2008). (B.B. Chaudhuri, 1998) used the midpoint of a run of at least k_1 consecutive 0s (i. e., white pixels) if the run exits in a vertical projection profile of a line.

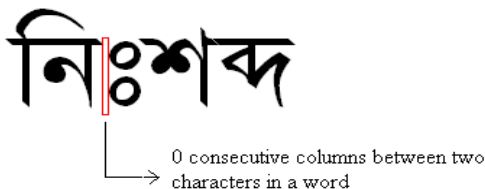


Fig. 7.4.2. Word with 0 consecutive columns between two characters

7.5 Character Segmentation Process

It is the most difficult and challenging part to build printed Bangla OCR. As Bangla is an inflectional language, the ornamentation of the characters in a word causes many peculiarities and makes the segmentation difficult. Many works have been done in order to solve those problems. A word can be constructed either with only taking the basic characters or basic characters and modifiers or basic and compound characters or basic, modified and compound characters. The main parts of all the characters are situated in the middle zone. So the middle zone area is considered as the character segmentation portion. Since matraline connects the characters together to form a word, it is ignored during the character segmentation process to get them topologically disconnected (B.B. Chaudhuri, 1998).

In order to segment টা and ঠা, more than one gaps are found in the immediate upper row of the matra upper line for ঠা which determines there is ঠ. Then টা is segmented into characters by finding the cutting point in the first and second half of the upper zone and ঠা is by the thickness column (Nasreen Akter, 2008). Although ঠা has one gap, it does not get mixed up with টা since no cutting point is found in the first half of the upper zone. The segmentation process of ঠা is similar with the segmentation process of ঠ only differing with the scan direction (Nasreen Akter, 2008). টা is segmented by finding the cutting point (Nasreen Akter, 2008).

More peculiarities are formed if the modifier ঠ or ঠ gets included with those situations. These issues have not been addressed yet by researches. However, modifier, ঠ with basic character, which is used in the upper zone, is detected by considering its distinct feature which is, it always makes a regular angle with matra (Md. Abdus Sattar , 2007). Three other modifiers are there, ঠ, ঠ, <, which are used in the lower zone. They are extracted by doing DFS (Depth first search) below the baseline ((Jalal Uddin Mahmud, 2003) or finding a column of white pixels after the first black pixel is found (Nasreen Akter, 2008). Sometimes a small part of a middle zone character exceeds the baseline. They are distinguished from lower zone modifiers since they don't touch the end row of the line as the characters in the lower zone always touch that row (Nasreen Akter, 2008). Rests of the modifiers are used in the middle zone and both of them are segmented as the regular way. However, sometimes basic character and modifier or two basic characters or two modifiers get into each others' region. If the middle zone modifiers and basic characters get into each others' region, a piecewise linear scanning is applied on them to segment the characters (B.B. Chaudhuri, 1998, Jalal Uddin Mahmud, 2003). Above all, during the character segmentation time, sometimes group of characters do not get segmented as they enter each others' region and seems to act like a character. A piece wise linear scanning, shown in fig. 7.5.1, is done to segment those characters (B.B. Chaudhuri, 1998).

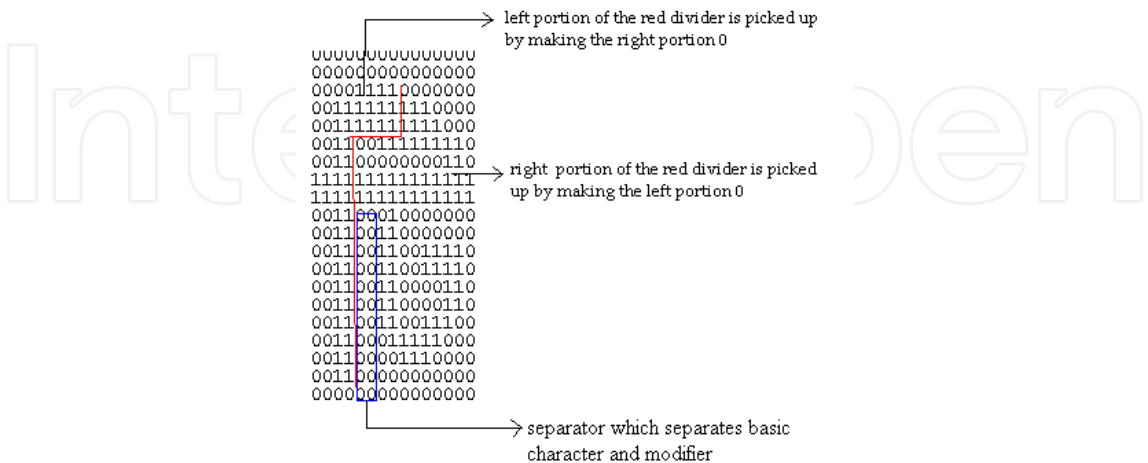


Fig. 7.5.2. Segmentation process of টি

8. Feature Extraction

Feature extraction is an important and challenging part for any character recognition process. Selection of good features leads to improved recognition rate. Many feature extraction algorithms are proposed for both printed and handwritten bangla document. Some of the potential algorithms are described below with comparative analysis.

8.1 Feature extraction algorithm for printed bangla character

- (a) (Jalal Uddin Mahmud, 2003) proposed the following procedure to extract feature from bangla text. In Bangla language, more than one connected components are present. Here, at first, all the connected components of an isolated character are detected using DFS (Depth First Search). Then center of mass has been calculated for each connected component. Center of mass for i th connected component is (X_i, Y_i) . Where

$$X_i = \sum_{j=1}^{N_i} P_{ij} / N_i \quad (1)$$

$$Y_i = \sum_{j=1}^{N_i} Q_{ij} / N_i \quad (2)$$

Here,

N_i = Number of Black pixels in connected component i .

P_{ij} = x Coordinate of the j th Black pixel in i th connected component.

Q_{ij} = y Coordinate of the j th Black pixel in i th connected component.

Then a bounded rectangle of each component is calculated by its minimum and maximum span in x direction and y direction. The freeman chain code is calculated by dividing each component into four regions depending on the center of mass of that component shown in fig. 8.1.1.

Fig. 8.1.1. Four Regions for a connected component

Freeman Chain code is based on the observation that each pixel has eight neighborhood pixels. The 8 transitional positions defined by freeman chain code are then divided into 4 transitional zones in order to keep the correct order of searching. Fig. 8.1.2 describes the freeman chain code.

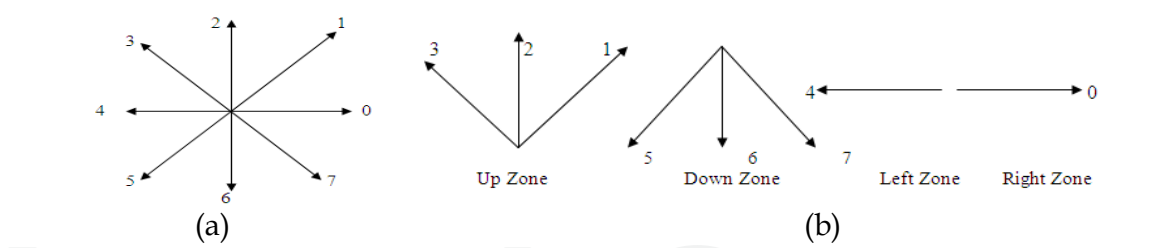


Fig. 8.1.2. (a) Slope Convention for Freeman Chain code, (b) 8 directional slopes divided into 4 direction zones for searching.

Maintaining an anti clock wise order of searching, zonal information is used to modify the chain coded position of the next selected pixel. The algorithm selects the next pixel if it fulfils all of the following criteria:

- The pixel is Black, i.e., it is a part of the character.
- The pixel is within the bounded rectangle of the connected component.
- The pixel is still not visited.
- The pixel is in a zone.

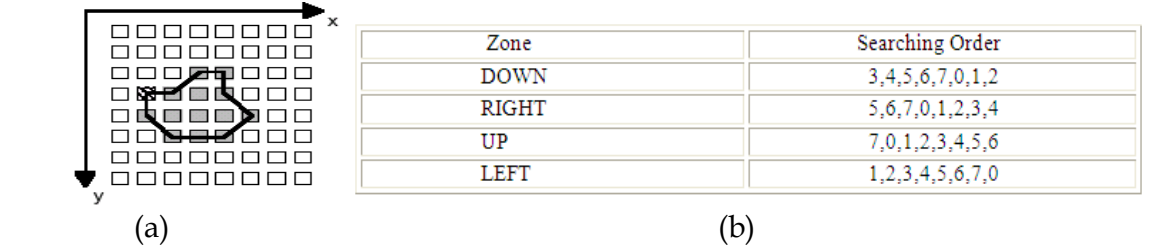


Fig. 8.1.3. (a) Chain code generation for an image, (b) Searching order in the four zones

Fig. 8.1.3 (a) shows the chain code generation of an image marked by gray pixels. When the algorithm starts from the hatched pixel (absolute coordinate, $x=1, y=3$), it marks the current black pixel as visited and initiates its directional zone as DOWN zone. So it searches for an unvisited black pixel in the directional order: 3,4,5,6,7,0,1,2 (Searching order is shown in Fig. 8.1.3 (b) for each zone). In this way the process continues and finally produces the chain code, 06700132454.

The frequency of each directional slope at each region is recorded and updated during the traverse. A total of 32 directional slopes or local features for each component are found. Then they are normalized to 0-1 scale. In Bangla, as there are more than one components in a character, the normalized features for each connected components are then averaged. The calculation of normalized slope distribution is as follows:

If $a_1, a_2, a_3, \dots, a_8$ are 8 directional slopes in region 1, then normalizing constant for region 1 is,

$$N_1 = \sqrt{(a_1^2 + a_2^2 + \dots + a_8^2)} \tag{3}$$

So, normalized slope in region $i = S_{i,j} / N_i$, where $i = 1$ to 4 and $j = 1$ to 8

$S_{i,j}$ = Frequency of j^{th} directional slope in i^{th} region.
 N_i = Normalizing constant in i^{th} region.

- (b) (Abu Sayeed Md. Sohail, 2005) demonstrated a method where a large two dimensional vector is converted into a small one dimensional vector. Here, initially, character images of fixed resolution are separated from original text and digitized into a large two dimensional vector, later converted into a small one dimensional vector. The algorithm they used is:

- i. Find center of the image by $\text{Center}_x = (\text{width})/2$, $\text{Center}_y = (\text{Height} + 1)/2$
- ii. For each black pixel, a radius is calculated as

$$r = \sqrt{(\text{Center}_x - x)^2 + (\text{Center}_y - y)^2}$$
- iii. Sum all pixels within same radius (sum_r_max,....., sum_r_min)
- iv. Calculate the membership for each circle or disk using the membership function, $m_f(bp) = e^{-(|bp - r_{bp}|)/total_pixels}$

This technique is used for the recognition of a single isolated character. It has not been tested for all the characters in a document or at least in a word.

The technique has been used with ANN (Artificial Neural Network). The advantage of this technique, shown by the authors, is that the extracted features are same even if the original character is rotated with rotation angles from 5 to 350 degree having a little bit of distortion. Non rotated characters are recognized with more than 90% of efficiency while rotated characters are recognized over 80- 89% efficiency depending upon the angle of rotation.

- (c) (B.B. Chaudhuri, 1998) showed that compound characters occupy only 4-6% of the text corpus. In order to introduce fast, accurate and robust technique, basic, modifier and compound characters are distinguished and identified by the following 3 features listed below.

- i. Feature f_1 - Bounding box width.
- ii. Feature f_2 - Number of border pixels per unit width, which is computed by dividing the total number of character border pixels by the width of the character.
- iii. Feature f_3 - Accumulated curvature per unit width.

They have used a feature-based approach for basic and modifier character recognition and a combination of feature-based and template-matching approach for the compound character recognition.

The authors considered a few stroke features for initial classification of the basic characters by a tree classifier. Apart from them, some other features are also used at some nodes of the tree classifier. In fig. 8.1.4, the principal set of chosen stroke features is shown.

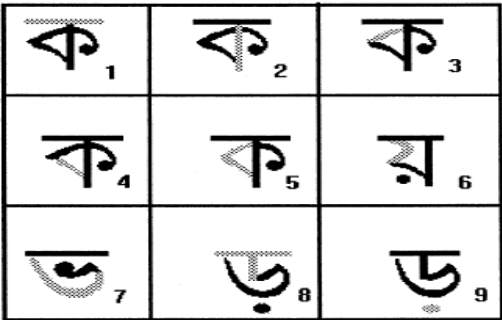


Fig. 8.1.4 Stroke features used for character recognition. (Shaded portions in the character represent the features)

It is mentioned that most of the cases, the strokes, 1, 2, 3, 4, 5 and 8 are correctly detected.

8.2 Feature extraction algorithm for Bangla handwritten characters

- (a) (Subhadip Basu, 2005) has used a set of 76 features which includes 24 shadow features, 16 centriod features and 36 longest- run features, computed taking 64 x 64 pixel size binary images.
- Shadow features are calculated by dividing the image into 8 octants within minimal square. Lengths of all projections on each of the 24 sides of all octants are summed up to produce 24 shadow features, shown in fig. 8.2.1.

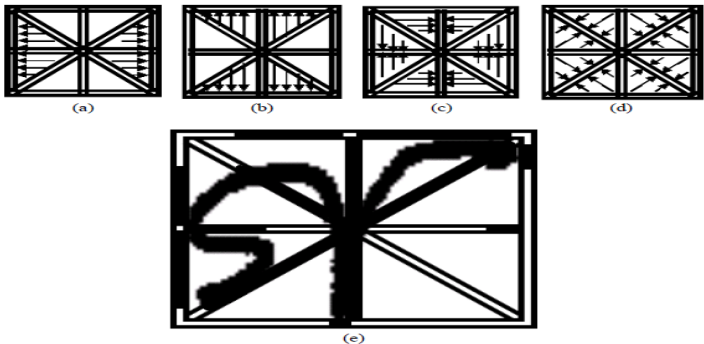


Fig. 8.2.1. An illustration for shadow features. (a-d) Direction of fictitious light rays as assumed for taking the projection of an image segment on each side of all octants. (e) Projection of a sample image

Coordinates of centriods of black pixels in all 8 octants of a digitized image, shown in fig. 8.2.2, are considered to add 16 centriod features.

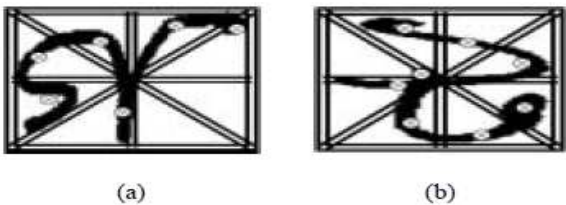


Fig. 8.2.2. Centriod features of two different characters (a)-(b)

Longest- run features are computed dividing the square into 9 overlapping regions and for each, 4 longest-run features are calculated respectively by row wise, column wise along 2 of its diagonal. Thus 36 features are produced (fig. 8.2.3).

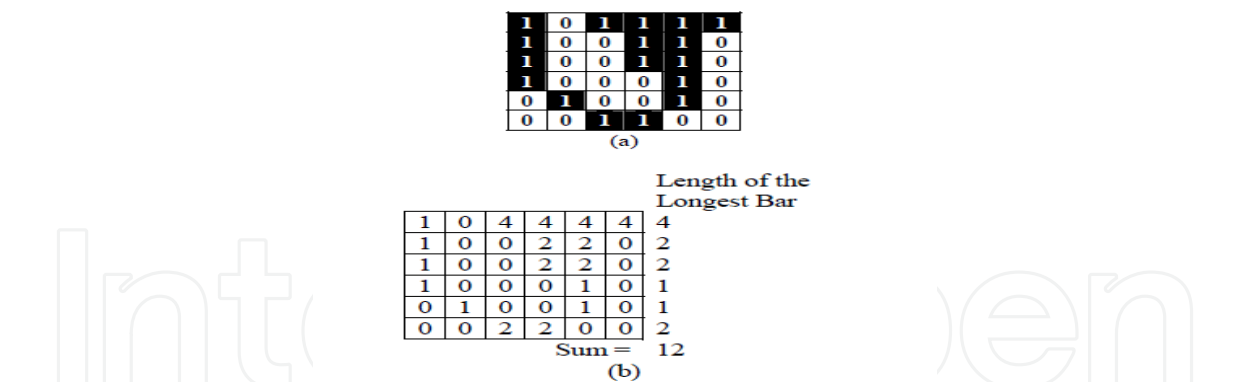


Fig. 8.2.3. An illustration for computation of the row wise longest-run feature.(a) The portion of a binary image enclosed within a rectangular region.(b) every pixel position in each row of the image is marked with the lenght of the longest bar that fits consecutive black pixels along the same row.

(b) (Mohammad Badiul Islam, 2005) suggested various stroke feature generation for handwritten bangla characters (fig. 8.2.4). Stroke identification is performed using curve fitting algorithm and curvature analysis. The equation of curve fitting is, $x=a+by+cy^2$. The point on the stroke that has maximum curvature, k , is calculated with the formula, $k= x_2/(1+x_1^2)^{3/2}$ where $x_1 = dx/dy$ and $x_2 = d^2x/dy^2$. If $k \geq 0.9$ then the stroke is a curve otherwise a straight line. Finally, the code is given using the slope equation, $x = a + by$.

Stroke	Characteristics	Numeric Code
	Horizontal line	1
	Positive sloping line	2
	Vertical Line	3
	Negative sloping line	4
	Vertical concave curve	5
	Vertical convex curve	6
	Rejected stroke	0

Fig. 8.2.4. Strokes used for Bangla character Recognition.

(c) (T.K. Bhowmik, 2003) constructed a feature vector of lengths $((2L + 1) + 3)$ as:
 $f_L = (D_L, P_i, H_v, H_{Run})$, generated for those points whose lower bound and upper bound cardinalities are greater than or equal to L excluding that point.
Where, $D_L = \{d_{i-1}, d_{i-1+1}, \dots, d_{i-1}\} \cup \{d_i\} \cup \{d_{i+1}, d_{i+2}, \dots, d_{i+1}\}$
 $P_i = (\text{height of } x_i \text{ with respect to middle zone}) / \text{height of the middle zone}$
 $H_v = (\text{value of vertical histogram of } x_i) / \text{height of the middle zone}$
 $H_{Run} = \text{value of black pixel run of } x_i$
Then the feature vector, f_L , is normalized to $f_L^N = (D_L/8, P_i, H_v, H_{Run}/\delta)$ where, δ is a constant.

8.3 Comparative analysis on different types of features

In section 8.1 and 8.2, a total of six features are discussed. The technique given in section 8.1b is used for the recognition of a single isolated character. It has not been tested for all the characters in a document or at least in a word. Using stroke feature technique (section 8.1c), a set of characters are found in most of the leaves of the decision tree (fig. 9.1) which needs further care to extract feature for identification of individual character. Chain code feature generation, discussed in section 8.1a, helps to produce a distinct feature set of 32 slopes for each component of the segmented characters. Shadow, centriod and longest-run feature set illustrated in section 8.2a, provides a total of 76 feature values for each characters. Although these two techniques produce distinct feature sets for each character, significant difference is observed between these two techniques among the samples of different sized characters.

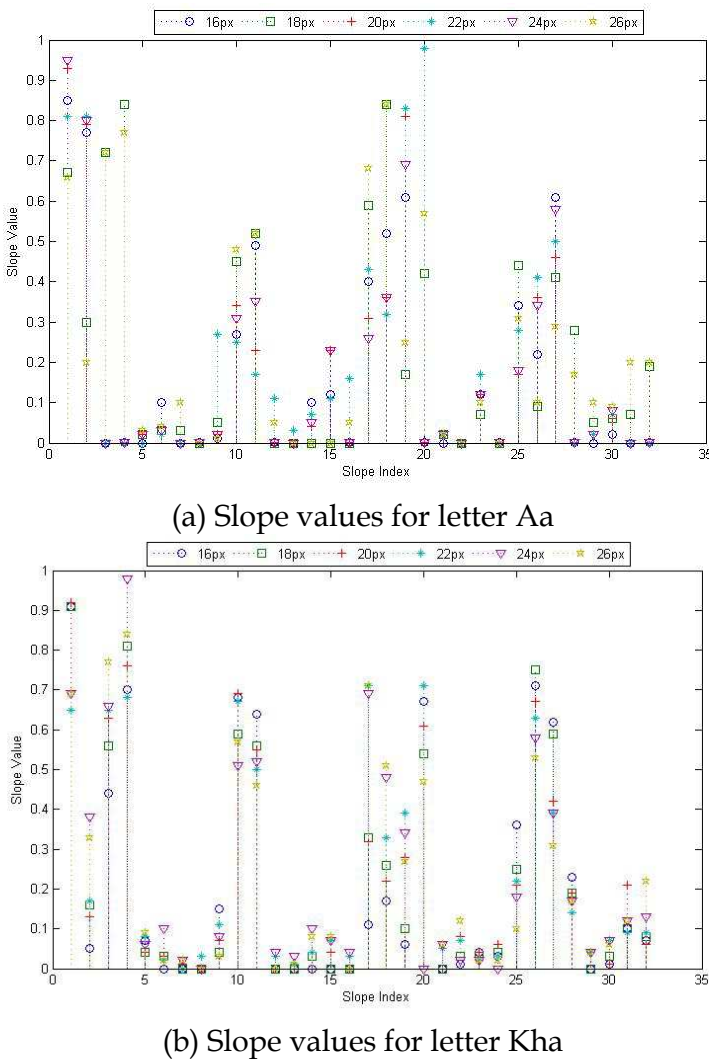


Fig. 8.3.1. Slope produced by chain code (a)-(b)

Fig. 8.3.1a and fig. 8.3.1b display 32 slope values for character Aa (অ) and Kha (খ) respectively. Six different sizes of fonts, namely 16, 18, 20, 22, 24 and 26 pixel font sizes have been considered for both cases. All these fonts have been resized into 64 x 64 pixels. In some cases, a slope value for each of the font size is same, for example, slope values at index 7 for

Aa (অ) while slope values at index 26 differ by a significant amount. Same type of result is found in fig. 8.3.2. On the other hand, similar analysis is performed with shadow, centriod and longest run features. We considered 68 feature values from them used in (Subhadip Basu, 2005). Values of these 68 features for letter ‘Aa’ and ‘Kha’ are presented in fig. 8.3.2. We have considered 6 font sizes for both these characters. It is seen in the figure that each value for all sizes deviates among themselves in a lesser amount than that of chain code representation.

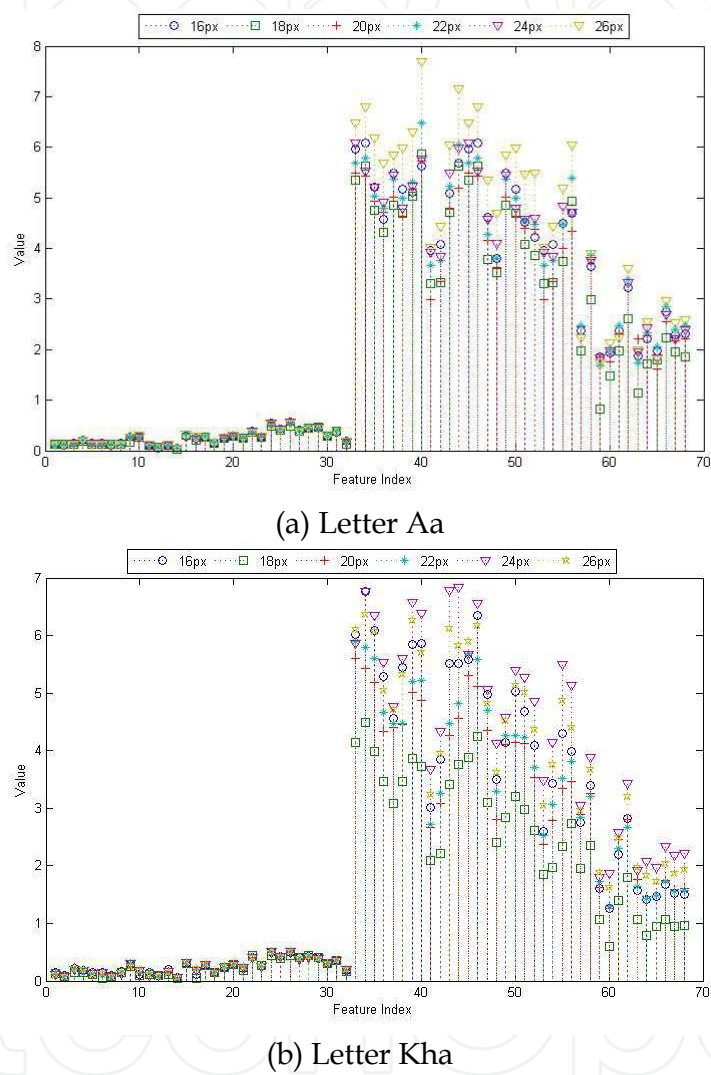


Fig. 8.3.2. Shadow, centriod and longest run feature values (a)-(b)

9. Classification

This is the last phase of the whole recognition process. Several approaches have been used to identify a character based on the features extracted using algorithms described in previous section. However, there is no benchmark databases of character sets to test the performance of any algorithm developed for Bengali character recognition (Angshul Majumdar, 2009). Each paper has taken their own samples for training and testing their proposals. In choosing classification algorithms, use of Artificial Neural Network (ANN) is a

popular practice because it works better when input data is affected with noise. Since a detailed definition of different Neural Network is outside the scope of this chapter, the readers are suggested to see established literature for detailed description of these methods. However a brief overview is provided for Decision tree and MLP classifier with some other neural networks.

9.1 Decision tree

(B.B. Chaudhuri, 1998) used a binary tree classification. Only one feature is tested at each non terminal node. The decision rules are mostly binary e.g. presence or absence of the feature. Here, the features are positional strokes. As one goes down the tree, the number of features to choose from gets reduced. Most basic characters can be recognized by the principle features alone. In some cases, more than one characters share the same non-terminal node of the tree. To separate them, some additional features are used (likewise, क and ख - vertical line touches headline or not, न and न, a vertical line at center position cross न 3 times while न is 4 times).

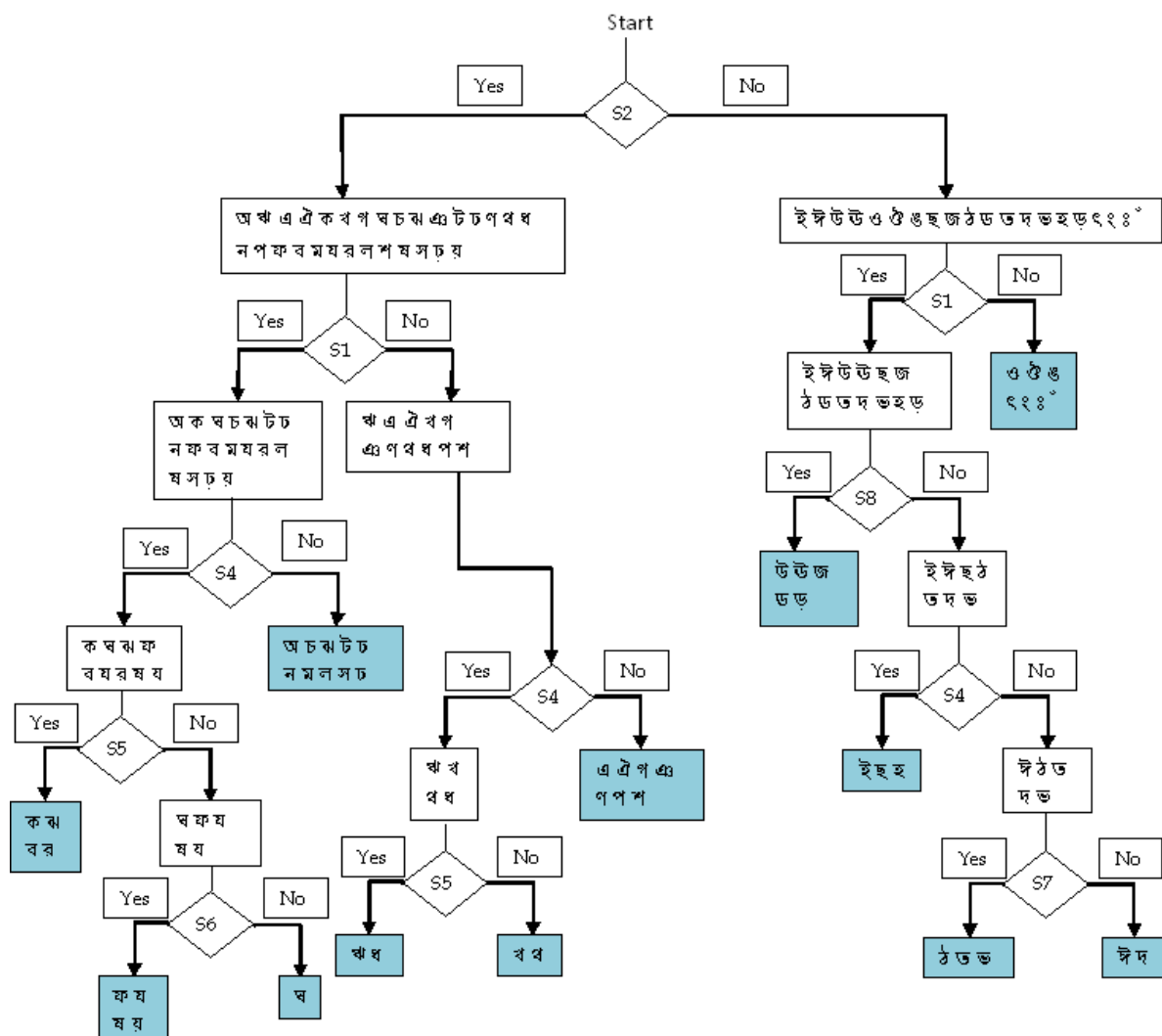


Fig. 9.1. A flow chart representation of a certain portion of tree classifier for basic character recognition

Compound character recognition is done in 2 stages. In the first stage, the characters are grouped into small subsets by a feature-based tree classifier. At the second stage, characters in each group are recognized by a sophisticated run-based template matching approach. The features used in the tree classifier are headline, vertical line, left slant (i.e., features 1, 2, 3 of fig. 8.1.4), boundary box width, presence of signature in upper zone etc. A terminal node of this tree corresponds to a subset of about 20 characters. These character templates are ranked in terms of their bounding box width and stored during the training phase of the classifier. When a character reaches the terminal node in search phase, firstly bounding box width is matched, then a matching score is completed by superimposing the candidate on the template. Different algorithms have been prescribed to compute matching score. In this process, a reasonable amount of character size variation can be accommodated by rescaling. If templates for 12 point character size are stored, it was found that characters from size ranging from 8 to 16 points can be matched by rescaling the candidate without appreciable error. Fig. 9.1 depicts an elaborated decision tree for basic character recognition.

9.2 MLP

(Subhadip Basu, 2005) has used Multi-Layer Perceptron (MLP) classifier to classify handwritten alphabetic characters. It is a special kind of ANN, a feed-forward neural network with artificial neurons. An MLP consists of one input layer, one output layer and a number of hidden layers. The output of each neuron is connected to each neuron of the immediate next layer as input. Neurons in the input layer are used to simply pass the information to the next layer. Supervised training is applied. Back Propagation has been used here which minimizes the sum of square error for the training samples by conducting a gradient descent search in weight space. It is found that the recognition performance is increased as the number of neurons in the hidden layer is increased (Table 9.2.1). A training set of 8000 samples and a test set of 2000 samples of optically scanned handwritten characters of 50 alphabetic symbols have been used to train and test the network. All these samples were scaled to 64x64 pixel images first and then converted to binary images through thresholding. The Back Propagation algorithm with learning rate 0.8 and momentum term 0.7 were executed. (fig. 9.2.1)

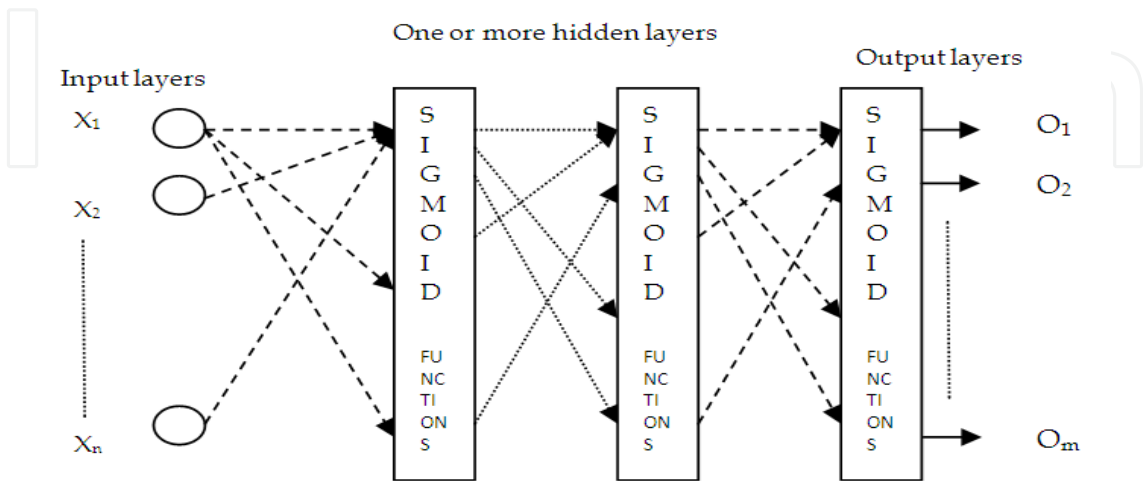


Fig. 9.2.1. A block diagram of an MLP shown as a feed forward neural network

No of Hidden Neurons	35	40	45	50	55	60	65	70	75
Percentage recognition rate on the training samples	80.35	80.93	82.8	84.7	85.70	86.46	87.36	86.69	85.65
Percentage Recognition rate on test samples	70.35	70.70	71.6	73.15	74.2	75.05	73.65	74.7	72.00

Table 9.2.1. Recognition Performance of the MLP with different numbers of neurons in the hidden layers

9.3 Kohonen Neural Network (KNN)

KNN (Adnan Mohammad Shoeb Shatil, 2007) differs from the feed forward back propagation neural network architectures. The first difference is that the Kohonen network (fig. 9.3.1) does not contain hidden layers; secondly, training and recognition processes are significantly different, that is, it is trained in unsupervised mode; thirdly, it does not use an activation function, and finally, the Kohonen Network does not use any bias weight in the network. For a particular feature vector of a given pattern, a single neuron will be fired. Input data is first resized to 250x250 pixels, regardless of whether the input image is a single word or character. Skew detection was not taken into consideration. Both computer generated image and scanned image have been used to train the network. A character is converted into a vector of length 625, which also determines the number of input neurons. This work considered 10 Bangla digits, 11 vowels, 36 consonants, in total, 57 characters. Authors have reported the performance of kohonen network given in Table 9.3.1 based on different scopes of character/ words i. e., trained, untrained similar, scanned documents and irregular font. A comparison on accuracy rate between kohonen and neural network is found in Table 9.3.2.

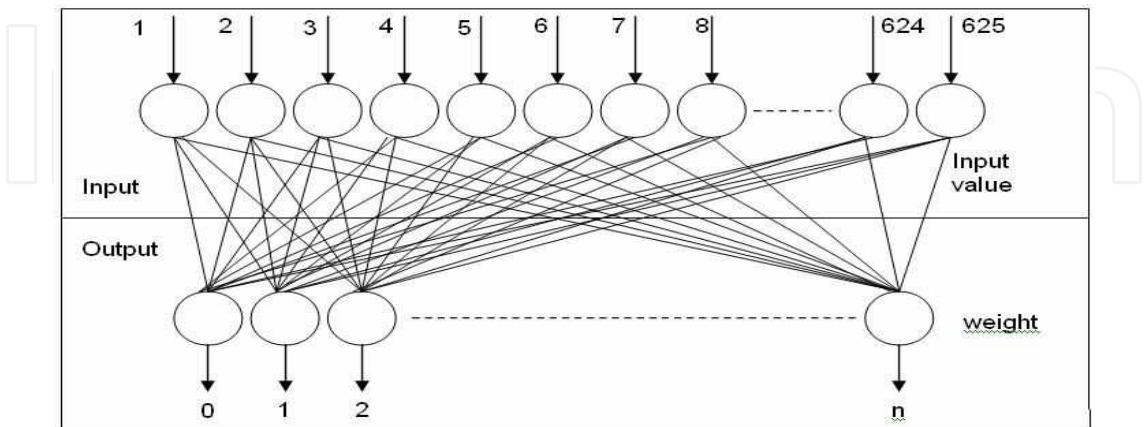


Fig. 9.3.1. Kohonen Neural Network design for Bangla Character Recognition

Character/Words	Rate of accuracy with kohonen network
Trained	100%
Untrained similar	99%
Scanned documents	99%
Irregular font	98%

Table 9.3.1. Accuracy rates corresponding to different sectors

Typeface	Accuracy rate	
	Kohonen network	Neural network
Sutonny	97.6%	94.37%
Sulekha	96.2%	91.25%

Table 9.3.2. Performance comparison between Kohonen Network and Neural Network

9.4 Nearest Subspace Classifier

The concept of Nearest Subspace Classifier has been applied in a different way in (Angshul Majumdar, 2009). The assumption is – samples from each class lie on a subspace specific to that class. As per the assumption, the training samples of a particular class span its subspace. The problem is to determine to which subspace it belongs to. Therefore, any new test sample belonging to a class can be represented as a linear combination of the test samples i.e.

$$v_{k,test} = \sum_{i=1}^{n_k} \alpha_{k,i} \cdot v_{k,i} + \epsilon_k$$

Where $v_{k,test}$ is the test sample for k^{th} class, $v_{k,i}$ is the i^{th} test sample for the k^{th} class and ϵ_k is the approximation error for the k^{th} class
The error term is written as

$$\epsilon_k = ((V_k^T V_k)^{-1} V_k^T - I) v_{k,test}$$

The expression in brackets (the orthoprojector) can be computed beforehand since it does not depend on test sample. The algorithm is as follows:
Training – For each class ‘i’, compute the orthoprojector (the term in bracket of the above equation)
Testing – Calculate the error for each class ‘i’ by hitting the test sample by the orthoprojector. Classify the test sample to the class having the minimum error.
During experimentation, each image is normalized to 16x16 pixels. Two different scenarios, namely pessimistic and optimistic scenarios have been considered. There were 12 different fonts of 5 different sizes of each. For the pessimistic scenario 5 different datasets were created by randomly splitting the font types into groups of 9 for training and 3 for testing (Table 9.4.1). For optimistic scenario, 3 different datasets were created by randomly splitting the font sizes into groups of 3 for training and groups of 2 for testing (Table 9.4.2). The authors have shown that this classifier gives better performance than Nearest Neighbour, Sparse Classifier and Support Vector Machine.

Set No.	KNN	Proposed	SC	SVM
1	0.1529	0.0667	0.1085	0.1200
2	0.3660	0.2157	0.2967	0.3507
3	0.4065	0.2549	0.3098	0.3720
4	0.4157	0.3150	0.3699	0.3853
5	0.2706	0.2065	0.2314	0.2507

Table 9.4.1. Recognition Error for Pessimistic Scenario

Set No.	KNN	Proposed	SC	SVM
1	0.0253	0.0074	0.0114	0.0204
2	0.0376	0.0082	0.0188	0.0302
3	0.0596	0.0245	0.0384	0.0474

Table 9.4.2. Recognition Error for Optimistic Scenario

9.5 Heuristics Neural Network

Heuristics on Neural Network (Golam Sarowar, 2009) showed that variation in implementing Back Propagation (BP) algorithm can significantly affect a Neural Network’s performance. Even better preprocessing and feature extraction may fail to give better accuracy if the recognition process (both training and testing) is incompetent. The rate of convergence of BP depends on the learning rate. Learning rate is reduced when a new train sample increases error after weight adjustment of NN and rate is increased when a new train sample decreases the error in classification. BP with momentum method is such a method where both the weight change at the previous step and the gradient at the current step are used to determine the weight change for the current step. With momentum, a network can slide through a local minimum which otherwise get stuck. In conjugate gradient algorithms, search is performed along conjugate directions, not in the steepest descent direction (negative of the gradient), which generally produces faster convergence. Some conjugate gradient algorithms are Fletcher-Reeves Update, Polak-Ribiere update, Powell-Beale restarts and scaled conjugate gradient. Some heuristics have also been applied to the above variations of BP. Antisymmetric activation or transfer function quickens the learning. Maximum possible information content in the training set can be ensured by introducing a training set that results largest training error. Desired response of a neuron at the output layer is offset by some amount to prevent the free parameters of the network being driven to infinity. Each input variable should have a zero-mean. It is useful to reduce the dimension of input vectors. Use of crossvalidation (B. Yegananarayana, 2008) may be used to prevent overfitting a common problem with neural network training. Table 9.5.1 and 9.5.2 show performance variations in case of different versions of BP algorithm with and without heuristics.

Algorithm	Performance Error	Epoch	Mis-classification	Accuracy %
Backpropagation with momentum	0.021	1251	115	83.57
Backpropagation with adaptive learning rate	0.07	102	180	74.29
Powell-Beale restarts	0.0121	80	86	87.26
Polak-Ribiere update	0.02	74	95	86.43
Scaled Conjugate Gradient	0.021	150	110	84.29

Table 9.5.1. Accuracy without heuristics

Algorithm	Performance Error	Epoch	Mis-classification	Accuracy %
Backpropagation with momentum	0.40114	3163	102	85.43
Backpropagation with adaptive learning rate	0.69554	62	224	68
Powell-Beale restarts	0.405388	65	78	88.86
Polak-Ribiere update	0.4180	43	84	88.00
Scaled Conjugate Gradient	0.399	81	97	86.14

Table 9.5.2. Accuracy with heuristics applied

The authors mentioned that the main issue with recognition of Bengali characters is the scarcity of samples. Typically, parametric classifiers like Artificial Neural Network (ANN) and SVM perform well when there is a lot of training data available; otherwise the parameters used in classification over-fit. In such a case it is logical to use a non-parametric classifier such as the KNN.

10. Future Challenges and Conclusion

Despite so many attempts to solve problems on several aspects of an OCR, a fully fledged solution is still unavailable for Bangle language. Problems that we have identified are summarized below.

Little data are available as sample. So rigorous testing of an implementation is not possible. Each author has used their own set of data. As a result, comparative analysis does not produce a really meaningful result. Some authors addressed noise detection and cleaning phase in their works. However, a comprehensive solution for elimination of all types of noise is not available. The reader has already understood that Bangla has not only basic characters; it is rich with modifiers and compound characters. Placement of modifiers may happen on the upper, lower, left or right side of original characters which generates a lot of complications. Rarely authors could confidently claim that a particular segmentation and classification scheme has dealt with all of them. Again lack of standard or benchmark samples do not allow one to make a comprehensive testing of their application. Investigation of the phases of pattern recognition deserves special attention to be considered. We have done some component level implementation on Intel® Core™2 Duo processor with 4 GB RAM. Fig. 5.1a has been used as a sample image for binarization, segmentation, feature extraction, and classification. The total time length required for all the activities took from 1 minute to 10 minutes depending on the choice of algorithms. This suggests that careful investigation would reveal the best possible combination of algorithms and processes for all the phases of Bangla OCR. It is evident that a full commercial OCR is a demand of the time in this era of digitization.

11. References

A. Ray Chaudhuri, A Mallick, D. Singh, M. Nasipuri & D.K. Basu (2005). Bangla Document Image Analysis, Recognition and Reconstruction, NCCPB, Bangladesh, pp.276-284

A. Roy, T.K.Bhowmik, S.K.Parui & U.Roy (2005). A New Approach to Skew Detection and Character Segmentation for Handwritten Bangla Words, Proceedings Of The Digital Imafing Computing: Techniques And Applications, IEEE

- Abu Sayeed Md. Sohail, Md. Robiul Islam, Boshir Ahmed & M A Mottalib (2005). Improvement in Existing Offline Bangla Character Recognitions Techniques Introducing Substainability to Rotation and Noise, NCCPB, Bangladesh, pp. 163-170
- Adnan Mohammad Shoeb Shatil and Mumit Khan (2007). Computer Science and Engineering, BRAC University, Dhaka, Bangladesh
- Al-Sakib Khan Pathan, Md. Mahbub Alam, Mostafa Monowar, Forhad Rabbi, Sabbir Ahmed & Tareq Hasan Khan. 12-Segment Display for Bengali Numerical Characters
- Angshul Majumdar & Rabab K. Ward (2009). Nearest Subspace Classifier: Application To Character Recognition
- Angshul Majumdar (1999). Bangla Basic Character Recognition Using Digital Curvelet Transform, *Journal of Pattern Recognition Research*, Vol(1)17-26
- Angshul Majumdar (2007). Bangla Basic Character Recognition Using Digital Curvelete Transform, *Journal Of Pattern Recognition Research*, Vol (1) 17-26
- Angshul Majumdar (2008). Multi Font Bangla Character Recognition Via Multiresolution Transforms, PP.1-14
- Atallah Mahmoud Al-Shatnawi & Khairuddin Omar (2009). Skew Detection And Correction Technique For Arabic Document Images Based On Centre Of Gravity, *Journal Of Computer Science*, Vol(5)(363-368)
- B.B. Chaudhuri & U. Pal (1997). Skew Angel Detection Of Digitized Indian Scripts Documents, *Transactions On Pattern Analysis And Machine Intelligence*, IEEE, PP. 182-186
- B.B. Chaudhuri & U. Pal (1998). Complete Printed Bangla OCR System, Elsevier Science Ltd. *Pattern Recognition*, Vol(31): 531-549
- B. Yegananarayana (2008) "Artificial Neural Network". Prentice-Hall India
- Barbara F. Grimes (1997). *Ethnologue: Languages of the World*, 11th edition.
- Carlos A.B. de Mello & Rafael D.Lins (1991). A Comparative Study on OCR Tools, *Vision Interface'99*, Toris-Riviers, Canada, PP.224-232
- Dipit Deodhare, NNR Ranga Suri & R.Amit (2005). Preprocessing And Image Enhancement Algorithms For A Form-Based Intelligent Character Recognition System, *International Journal Pf Computer Science & Applications*, Vol(2):131-144
- Faruq A. Al-Omari & Omar Al-Jarrah (2004). Handwritten Andian Numerals Recognition System Using Probabilistic Neural Network, *Advanced Engineering Informatics*, Vol(18):9-16
- Golam Sarowar, M.A. Naser, S.M. Nizamuddin, Nafiz I.B. Hamid, Adnan Mahmud (2009). Enhancing Bengali Character Recognition Process Applying Heuristics On Neural Network, *International Journal Of Computer Science And Network Security*, Vol(9):154-158
- J. N. Kapur, P. K. Sahoo and A. K. C. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, *Comput. »ision Graphics Image Process.* 29, 273-285 (1985)
- Jalal Uddin Mahmud, Mohammed Feroz Raihan & Chowdhury Mofizur Rahman (2003). A Complete OCR System For Continuous Bangali Characters, IEEE, PP. 1372-1376
- Kaustubh Bhattacharyya & Kandarpa Kumar Sarma (2009). ANN-Based Innovative Segmentation Method For Handwritten Text In Assamese, *International Journal Of Computer Science Issues*, Vol.(5):9-16

- Laurence Likforman-Sulem, Abderrazak Zahour & Bruno Taconet (2006). Text Line Segmentation Of Historical Documents: a survey, *International Journal On Document Analysis And Recognition*, PP. 1-25
- Li Zhuang & Xiaoyan Zhu (2005). An OCR Post-Processing Approach Based On Multi-Knowledge, R.Khosla et al.(EDs):KES , LNAI 3681,PP. 346-352
- M.Cheriet, N.Kharma, C.L.Liu & C.Y.Suen (2007). Introduction: Character Recognition, Evolution And Development, *Character Recognition Systems: A Guide For Students And Practitioner*, PP.1-4
- M.S. Alam (2009). Research on Bangla Language Processing In Bangladesh: Progress And Challenges, *International Language & Development Conference*, Dhaka, Bangladesh, PP.527-533
- Md. Abdus Sattar, Khaled Mahmud, Humayun Arafat and A F M Noor Uz Zaman. Segmenting Bangla Text For Optical Recognition, *ICCIT 2007*, Dhaka, Bangladesh PP. 27-29
- Md. Abul Hasnat Mumit Khan (2009). Elimination of Splitting Errors In Printed Bangla Scripts
- Md. Abul Hasnat Mumit Khan (2009). Rule Based Segmentation of Lower Modifies In Complex Bangla Script
- Md. Abul Hasnat, Muttakinur Rahman Chowdhury & Mumit khan (2009). Integrating Bangla Script Recognition Support in Tesseract OCR
- Md. Abul Hasnat, S.M. Murtoza Habib, Mumit Khan (2007). A High Performance Domain Specific OCR for Bangla Script
- Md. Al Mehedi Hasan, Md. Abdul Alim & Md. Wahedul Islam , A New Approach To Bangla Text Extraction And Recognition From Textual Image
- Md. Al Mehedi Hasan, Md. Abdul Alim, Md. Wahedul Islam & M. Ganger Ali (2005). Bangla Text Extraction and Recognition from Textual Image, *NCCPB, Bangladesh*, PP.171-176
- Md. Wahedul Islam, MD Al Mehedi Hasan & Ramesh Chandra Debnath (2005). Handwritten Bangla Numerical Recognition Using Back-Propagation Algorithm With and Without Momentum Factor, *NCCPB, Bangladesh*, PP. 177-182
- Md. Waliullah Khan Nomani, Syed Mustafa Khelat Bari, Tanwir Zubayer Islam & Mohammed Nazrul Islam (2007). Invariant Bangla Character Recognition Using A Projection-Slice Synthetic-Discriminant-Function-Based Algorithm, *Journal Of Electrical & Electronics Engineering*, Vol(7): 403-409
- Mohammad Badiul Islam, Mollah Masum Billah Azadi & Dr, M. M. A. Hashem (2005). Bengali Handwritten Character Recognition Using Modified Syntactic Method, *NCCPB, Bangladesh*, PP. 264-275
- Mohammad Masud Al Hossain Khan, Uzzal K. Acharjee & Bipul K. Sen (2005). Detection Of Bengali Text in Digital Images Using Connected Component Analysis, *NCCPB, Bangladesh*, PP.292-298
- Mohammad Monjur Alam & Mohammad Anwer (2005). Feature Subset Selection Using Genetic Algorithm For Bangali Handwritten Digit Recognition, *NCCPB, Bangladesh*, PP. 258-263

- Mohammad Osiur Rahman, Fouzia Ashraf Mousumi, Edgar Scavino, Aini Hussain & Hassan Basri (2009). Real Time Road sign Recognition System Using Artificial Neural Networks For Bengali Textual Information Box, *European Journal Of Scientific Research*, Vol(25): 478-487
- Mohammed Moshiul Hoque & S.M. Faizur Rahman (2007). Fuzzy Features Extraction From Bangla Handwritten Character, *ICCIT*, Dhaka, Bangladesh, PP.72-75
- Munirul Mansur, Naushad Uzzaman and Mumit Khan (2006). Analysis of n-gram Based Text Categorization for Bangla in a Newspaper Corpus, *ICCIT*, Dhaka, Bangladesh, pp.1-24
- N. Otsu (1979). A threshold selection method from gray level histograms, *IEEE Trans. Systems Man Cybernet.* 9,62-66
- Nasreen Akter, Saima Hossain, Md. Tajul Islam & Hasan Sarwar (2008). An Algorithm For Segmenting Bangla Text, *ICCIT*, IEEE, Khulna, Bangladesh, PP.177-182
- Nirmalya Chowdhury & Diganta Shaha (2005). Bengali Text Classification Using Kohonen's Self Organizing Network, *NCCPB*, Bangladesh, PP. 196-200
- Nirmalya Chowdhury & Diganta Shaha (2005). A Neural Network Based Text Classification Method: A Possible Application for Bengali Text Classification, *NCCPB*, Bangladesh, PP. 183-188
- R Sanjeev Kunte & R D Sudhaker Samuel (2007). A Simple And Efficient Optical Character Recognition System For Basic Symbols In Printed kannada Text, *Sadhana*, Vol(32), PP.521-533
- Rajiv Kapoor, Deepak Bagai & T.S.Kamal (2004). A New Algorithm For Skew Detection And Correction, *Pattern Recognition Letters*, Vol(25): 1215-1229
- R. M. Bozinovic and S. N. Shihari (1989). Off line cursive script word recognition, *IEEE trans. Pattern Anal. Mach. Intell.* 11, 68-83.
- S. Abirami, Dr. D. Manjula (2009). A Survey Of Script Identification Techniques For Multi-Script Document Images, *International Journal Of Recent Trends In Engineering*, Vol(1):246-249
- S.K. Parui, K.Guin, U.Bhattacharya & B.B. Chaudhuri (2008). Online Handwritten Bangla Character Recognition Using HMM, *IEEE*
- S.M. Milky Mahmud, Nazib Shahrier, A.S.M Delowar Hossain, Md. Tareque Mohmud Chowdhury & Md. Abdus Sattar (2004). An Efficient Segmentation Scheme For The Recognition Of Printed Bangla Characters, *ICCIT*
- Subhadip Basu, Nibaran Das, Ram Sarkar, Mahantapas Kundu, Mita Nasipuri & Dipak Kumar Basu (2005). Handwritten 'Bangla ' Alphabet Recognition Using an MLP Based Classifier, *NCCPB*, Bangladesh, PP. 285-291
- T.K. Bhowmik, A.Roy and U.Roy (2003). Character Segmentation For Handwritten Bangla Words Using Artificial Neural Network
- Tinku Acharya and Ajoy K. Ray (2005). "Image Processing Principles and Applications", John Wiley & Sons, Inc., Hoboken, New Jersey
- U. Garain & B.B Chaudhuri (2001). Segmentation Of Touching Characters In Printed Devnagari And Bangla Scripts Using Fuzzy Multifactorial Analysis, *IEEE*, PP.805-809
- U.Pal & B.B Chaudhuri (1995). Computer Recognition Of Printed Bangla Script, *International Journal Of Systems Science*, Vol(26): PP. 2107-2123

- U.Pal & B.B. Chaudhuri (2001). Automatic Identification of English, Chinese, Arabic, Devnagari And Bangla Script Line, IEEE, PP.790-794
- W.K. Taylor (1964). Character Recognition, Electronics & Power, PP.6-9
- Wikipedia, http://en.wikipedia.org/wiki/Optical_character_recognition
- Yungang Zhang & Changshui Zhang. A New Algorithm For Character Segmentation Of License Plate
- Zhang Ruilin, Hu yan, Fang Zhijian & Zhang Lei (2009). Skew Detection and Correction Method of Fabric Images Based On Hough Transform, *ICICTA*, IEEE, PP. 340-343



Character Recognition

Edited by Minoru Mori

ISBN 978-953-307-105-3

Hard cover, 188 pages

Publisher Sciyo

Published online 17, August, 2010

Published in print edition August, 2010

Character recognition is one of the pattern recognition technologies that are most widely used in practical applications. This book presents recent advances that are relevant to character recognition, from technical topics such as image processing, feature extraction or classification, to new applications including human-computer interfaces. The goal of this book is to provide a reference source for academic research and for professionals working in the character recognition field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Nasreen Akter, Saima Hossain, Hasan Sarwar and Mofizur Rahman (2010). Development of a Recognizer for Bangla Text: Present Status and Future Challenges, Character Recognition, Minoru Mori (Ed.), ISBN: 978-953-307-105-3, InTech, Available from: <http://www.intechopen.com/books/character-recognition/development-of-a-recognizer-for-bangla-text-present-status-and-future-challenges>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen