B. Sc Engineering Thesis

Handwritten Writer Independent Bangla Character Recognition

by

Mir Toornaw Islam
Student ID: 0805024
Session: 2011-12

Khandokar Md. Nayem
Student ID: 0805052
Session: 2011-12

Supervised by
Dr. Md. Monirul Islam
Associate Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

A thesis submitted to the
Department of Computer Science and Engineering, BUET
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
Dhaka-1205, Bangladesh

July, 2014

Declaration

This is to certify that the work presented in this thesis entitled "**Handwritten Writer Independent Bangla Character Recognition**" is the outcome of the investigation carried out by us under the supervision of Dr. Md. Monirul Islam, Associate Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka. It is also declared that neither this thesis nor any part of this thesis has been submitted or is being currently submitted anywhere else for the award of any degree or diploma.

(Author)	(Author)
Mir Toornaw Islam	Khandokar Md. Nayem
Student No: 0805024	Student No: 0805052

Abstract

Handwritten Bangla Character Recognition has been an active area of research in the field of Pattern Recognition and Image Processing. Some obstacles and challenges have to be faced during the recognition process. Cursive nature of Bangla characters is one of those challenges. Another challenge is similar shape of Bangla characters. During handwriting, connected characters are also found which is another major obstacle. Besides, multiple gray level of handwritten characters, overlapped characters and lines during handwriting and non-uniform background of handwritten script are also some other major challenges for Bangla character recognition.

Feature extraction is a major phase for any character recognition system. In our work, we use gradient of pixels of the preprocessed image and distance between connecting points of a single character as features. To classify the handwritten characters, we use Feed Forward Artificial Neural Network (ANN) with 50 neurons in input layer, two hidden layers with 50 neurons each and 50 neurons in output layer.

Applying on a writer independent handwriter Bangla character sample database, we get about 62% accuracy rate in correctly recognising characters. Here, horizontal histogram technique is applied for character segmentation which gives us 65% accuracy rate in segmenting individual characters. So, applying this character recognition technique on better optimized character segmentation technique, this character recognition can show significant improvement in recognizing handwritten Bangla character.

Keywords: Character Recognition, Handwritten Bangla OCR, Bangla Text Recognition

Acknowledgements

First of all we would like to thank our supervisor, Dr. Md. Monirul Islam, for introducing us to the fascinating field of Pattern Recognition and guiding us on how to perform our research work. Without his continuous supervision, guidance and valuable advice, it would have been impossible to complete the thesis. We are especially grateful to him for allowing us to chose our course of actions with freedom, for enlightening us during the moments of frustration, for his patience throughout the entire research work and for his moral support to the extra-curricular activities that we accomplished aside from the thesis.

We would also like to thank our friend, Aunik Ahmed, who provided us great support. We are also grateful to all other friends for their continuous encouragement and for helping us in thesis writing.

We would like to express our gratitude to all our teachers. Their motivation and encouragement in addition to the education they provided meant a lot to us.

Last but not least, we are grateful to our parents and to our families for their patience, interest, and support during our studies.

Contents

Chapter 1

Introduction

1.1 Background of the study

Optical Character Recognition (OCR) is a popular research topic in pattern recognition, aimed to enable computers to recognize optical characters without human intervention. The process of OCR involves several steps including segmentation, feature extraction,and classification. Each of these steps has multiple field to research. In the field of recognition of handwritten script, variation of writing style and the skewness and overlapping of free style handwriting complicate the task of segmenting and recognizing characters. Although many methods on handwritten character segmentation and recognition have been published in the literature for Roman, Chinese, Japanese and Arabic scripts, but only few works are reported on the recognition and segmentation of Bangla handwritten scripts.

1.2 Motivation

Recognition of handwritten scripts has a great potential in data and word processing for instance, automated postal address and ZIP code reading, data acquisition in bank checks, processing of archived institutional records etc. Combined with a speech synthesizer, this process has proper utility for the people who are suffering from physical disability. Digitization in Bangladesh has started in very near past. Even before almost a decade or so, all records whether of official or personal kind are kept in handwritten form, if not with a typewriter.

Segmentation of handwritten script is a critical task for the Optical Character Recognition System. In Bangla, touching or overlapping occurs frequently because of modified characters of upper-zone and lower-zone. Segmentation of these touching characters is the main bottleneck in handwritten recognition system.

Introduction of computer in personal work has lessened, but to convert the previous works into digital format, scanned image is all we have and it needs human help to retrieve the information out of a scanned image. So, recognition of handwritten scripts is the prime topic in our field of research.

1.3 Objective of study

Bangla, like other Indic languages has a large set of character consisting of 50 characters and 13 character modifiers. Compound Characters add to the complexity of Bangla alphabet. In the segmentation and recognition of handwritten script, we can combine the knowledge of the difference of the basic Bangla characters from other language. So, the main objectives of our study are as follows:

- Getting familiar with the alphabet and the compound characters of Bangla language.
- Studying the challenges and potential obstacles that may happen during the recognition of a handwritten Bangla script.
- Studying the existing approach for recognition of handwritten script.
- Propose a novel approach for the recognition of Bangla handwritten script.
- Testing the proposed approach with a large number of experimental data.

1.4 Contribution

Choice of a good feature can greatly enhance the recognition rate of a typical OCR system. Many features have been devised by other researchers for Bangla character recognition. In our work, we have used gradient of pixels and distance between connecting points of a character as features.

1.5 Thesis Organization

Chapter 1 discusses the motivation behind our work. Different features and their extraction techniques are described in Chapter 2. In Chapter 3, preprocessing phase for the scanned image, segmentation strategy for line, word and character segmentation, our proposed feature and its extraction technique and the recognition process of the characters using a classification tool are discussed. The experimental results and remarks on the experiments are discussed in Chapter 4. Chapter 5 finally mentions our findings, summarizes our work and discusses the future works regarding the recognition approach.

Chapter 2

Related Works

2.1 Introduction

The prime objective of an Optical Character Recognition (OCR) system is to convert a handwritten or printed document into machine readable text. A typical OCR system first preprocesses the scanned image of the document known as pre processing stage. Pre processing stage includes gray scale image conversion, text binarization, noise removal and skew detection and correction. At the start of pre processing stage, the scanned document image is converted into a gray scale image. This gray scale image is then converted into binary image known as text binarization. Binarization separates the foreground (text) and background information of the image. The most common method for binarization is to select a proper threshold for the intensity of the image and then convert all the intensity values above the threshold to one intensity value, that is white, and all other intensity values below the threshold to the other intensity, that is black. Scanned documents often contain noise that arises due to printer, scanner, print quality, age of the document etc. That's why, it is necessary to remove this noise before we process the image. Now, text lines of the document image may be skewed. This skewness must be corrected in order to make the text lines horizontal. Skew angle is the angle that the text lines of the document image makes with the horizontal direction. Skew correction can be achieved in two steps. First, we estimate the skew angle θ_t and second, we will rotate the image by θ_t, in the opposite direction.

After the completion of preprocessing stage, segmentation is done on the binary image. Segmentation phase includes line segmentation, word segmentation and character segmentation. Different segmentation techniques are used by different researchers. Our segmentation strategy is mentioned in Chapter 3. After the segmentation phase, features are extracted from characters. A good choice of feature can greatly improve the recognition accuracy of an OCR system. After extracting features from characters, classification and recognition of the characters are done using various classifiers like SVM, MLP, ANN, HMM etc. We have introduced our feature extraction process and character classification tool in Chapter 3.

In this chapter, some popular features and their extraction process are described. Besides, the fundamental characteristics of basic Bangla characters are also introduced here.

2.2 Overview of Bangla characters

Bangla is one of the most popular languages in the world. The set of basic characters of Bangla consists of 11 vowels and 39 constants. So, there are 50 different shapes in the Bangla basic character set. The concept of upper and lower case is absent in this script. There are two types of character modifiers in Bangla. These are vowel character modifier and consonant character modifier. A vowel taking a modified shape is called a vowel modifier. Similarly consonants taking modified shape are called consonant modifiers. A consonant or a vowel following a consonant sometimes takes a compound orthographic shape, which is called as compound character. The following figures Fig: 2.1, Fig: 2.2, Fig: 2.3 and Fig: 2.4 show Bangla basic characters, vowel modifier, consonant modifier and compound character respectively.

অ	ঊ	ঔ	ঙ	ঞ	ণ	ন	ম	ষ	য়
আ	ঋ	ক	চ	ট	ত	প	য	স	◌ং
ই	এ	খ	ছ	ঠ	থ	ফ	র	হ	◌ঃ
ঈ	ঐ	গ	জ	ড	দ	ব	ল	ড়	◌ঁ
উ	ও	ঘ	ঝ	ঢ	ধ	ভ	শ	ঢ়	ৎ

Figure 2.1: Bangla basic characters.

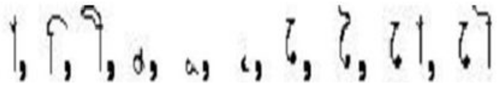


Figure 2.2: Vowel character modifiers in Bangla.

ব্য	ব্র	র্ব

Figure 2.3: Consonant character modifiers in Bangla.

ল্ক(ল + ক)	ক্ল(ক + ল)	ক্ট(ক +ট)	ক্ত (ক+ত)	ঙ্ক(ন+ ক)

Figure 2.4: Compound characters in Bangla with orthographic shape.

2.3 Different feature extraction schemes

2.3.1 Global feature extraction based on angular information

We know relative position of a pixel with respect to its neighbouring pixels of a character will not change if the character is rotated in arbitrary directions. Hence, relative angular information of the pixel with respect to its neighbouring pixels will also not change. Since, relative angular information does not change because of rotation, this angular information is used as one of the features [18]. Internal and external contour pixels of a character image are used to determine the angular information feature [Fig: 2.5]. These angular information are computed in such a way that they do not depend on the size and rotation of the characters.

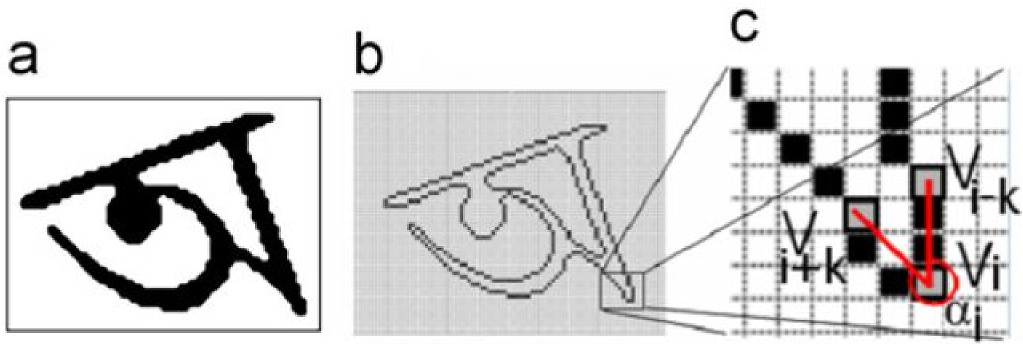


Figure 2.5: Illustration of angle computation from contour pixels. Here, (a) is the original image, (b) contour pixels of the image shown in (a), and (c) is an enlarged version of (b) where α_i is the angle at the contour pixel V_i when $k = 3$.

To get higher accuracy, a character is divided into zones to have zonal information. Zone-wise angular information is computed here and hence higher dimensional features are found. Circular and convex hull rings have been used to get zone-wise features in this scheme [Fig: 2.6].

These circular and convex hull features are then combined. The purpose for combining these features is to improve the recognition and classification results. These features are then fed to a support vector machines(SVM)for recognising the characters.

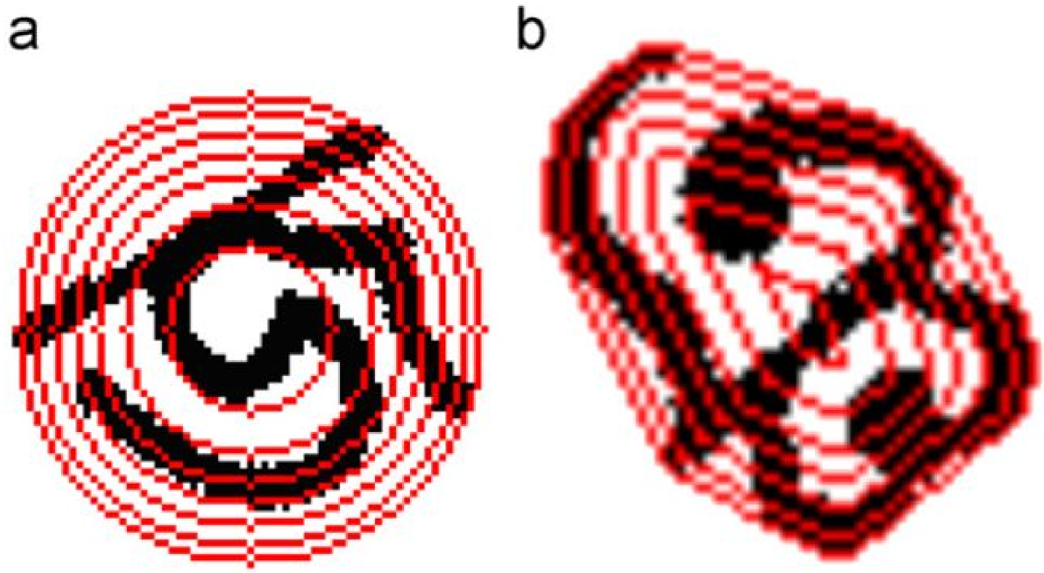


Figure 2.6: (*a*) Seven circular rings and (*b*) seven convex hull rings are shown on Bangla characters.

2.3.2 Chain code based feature extraction

Here, connected components of a character are used as features [4]. In Bangla language, a character can have more than one connected component. Depth First Search (DFS) approach is used for the detection of connected component. Then, we need to determine the boundary rectangle of a connected component to divide it into regions. After determining the boundary rectangle of the component, this component is divided into four regions [Fig: 2.7] indicating four quadrants in 2-D geometric system. The origin is the center of mass of that component. With the origin and the bounded rectangle of the connected component, four regions can be established.

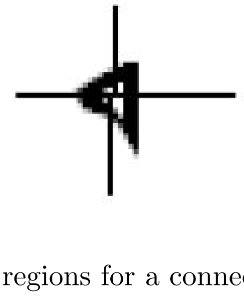


Figure 2.7: Four regions for a connected component.

After the character has been divided into connected components and boundary of the connected components is established, a chain code is calculated for each boundary. A chain code has two components: the coordinates of the starting point and a chain of codes representing the relative position of the starting pixel and its followers. The chain code is generated by using the changing direction of the connected pixels contained in a boundary. The representation is based on 4-connectivity or 8-connectivity of the segments. Here, for the generation of chain code, it is assumed that each pixel has eight neighbourhood pixels.

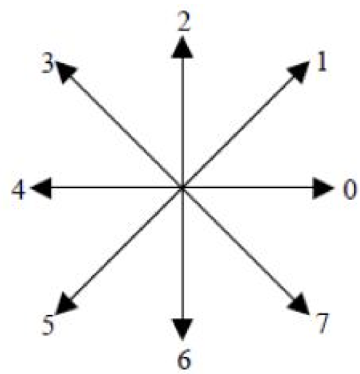


Figure 2.8: Slope convention for chain code.

As chain code for a closed boundary depends on the starting point of the boundary, to make it invariant to the starting point, the chain code can be normalized so that the resulting sequence of numbers forms a minimum integer. Chain code formation is shown in Fig: 2.9.

When searching for a closed boundary continues,there is a variation of slope in each region. The frequency of each directional slope at each region is recorded and updated during traversal. There are 8 directional slopes in each region [Fig: 2.8], therefore total $8X4$=32 directional slopes for each connected component feature. These 32 slope values or feature values are then normalized. Now, recognition step recognizes the whole character, not its individual connected component, therefore normalized features for each connected component are averaged to get the total feature for the character. Then, a feed forward neural network is used for the classification and recognition of the character.

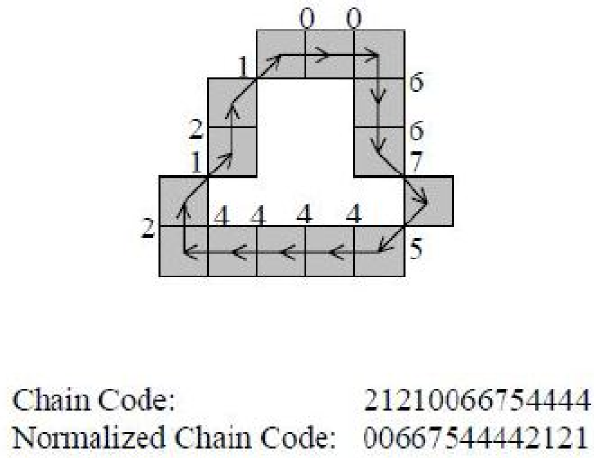


Figure 2.9: 8-directional chain code generation.

2.3.3 Shadow feature, centroid feature and longest-run feature extraction

In this case, feature set consists of 76 features, which include 24 shadow features, 16 centroid features and 36 longest-run features [6]. These features are computed from $64X64$ pixel size binary images of characters.

For computing shadow features, each character image is enclosed within a minimal square, divided into eight octants, as shown in Fig: 2.10$(a - d)$. Lengths of projections of character images on three sides of each octant are then computed. Finally lengths of all such projections on each of the 24 sides of all octants are summed up to produce 24 shadow features of the character image under consideration. For taking the projection of an image segment on one side of an octant, existence of a fictitious light source in the opposite side is assumed. Directions of light rays for each side of all octants in a minimal square is shown in Fig: 2.10$(a - d)$. Shadow features of a sample character image are illustrated also in Fig: 2.11.

Coordinates of centroids of black pixels in all the 8 octants of a digital image are considered to add 16 features in all to the feature set. Fig: 2.12(a) and (b) show approximate locations of all such centroids on two different character images.

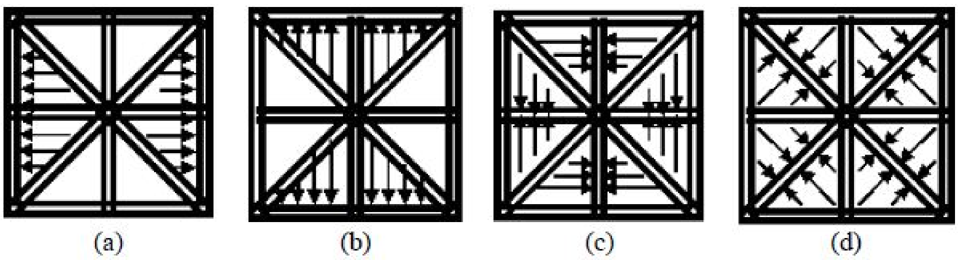


Figure 2.10: An illustration of shadow features.Here,directions of fictitious light rays are shown for taking the projection of an image segment on each side of all octants.

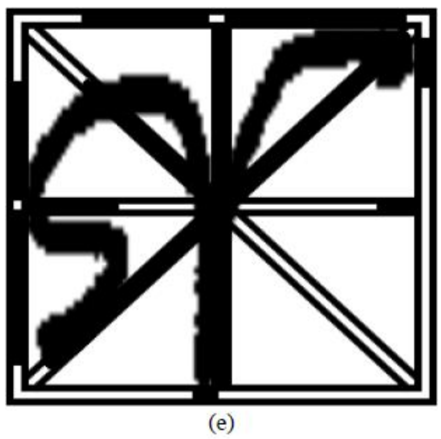


Figure 2.11: Projection of a sample image.

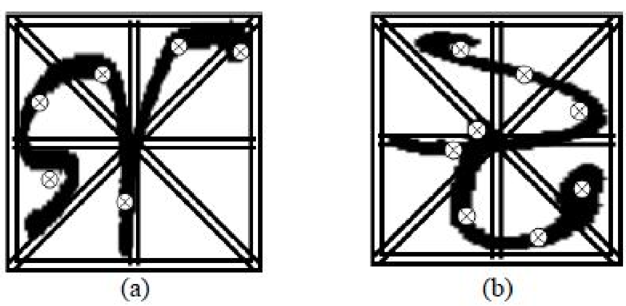


Figure 2.12: Centroid features of two different character images.

For computing longest-run features from a character image, the minimal square enclosing the image is divided into 9 overlapping rectangular regions. In each such rectangular region, 4 longest-run features are computed row wise, column wise and along two of its major diagonals. The row wise longest-run feature is computed by considering the sum of the lengths of the longest bars that fit consecutive black pixels along each of all the rows of a rectangular region, as illustrated in Fig: 2.14. The three other longest-run features are computed in the same way but along the column wise and two major diagonal wise directions within the rectangle separately. Thus, total $9X4$=36 longest-run features are computed from each character image. A feed forward MLP (Multi Layer Perceptron) is used for the classification of the characters.

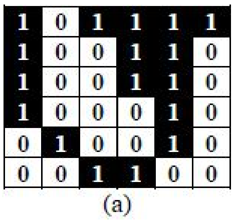


Figure 2.13: An illustration for computation of the row wise longestrun feature. The portion of a binary image enclosed within a rectangular region.

						Length of the Longest Bar
1	0	4	4	4	4	4
1	0	0	2	2	0	2
1	0	0	2	2	0	2
1	0	0	0	1	0	1
0	1	0	0	1	0	1
0	0	2	2	0	0	2
				Sum =		12

(b)

Figure 2.14: Every pixel position in each row of the image is marked with the length of the longest bar that fits consecutive black pixels along the same row.

2.3.4 Topological feature extraction

Topological features can be used for Bangla character recognition. Some examples of topological features are junction points, holes, stroke segments, curvature maxima, curvature minima etc. Here, stroke segments of a character are used as topological feature [5]. At first, thinning is applied on a scanned character image to produce single pixel thick skeleton image [Fig: 2.15(b)]. Then, straight line approximation is applied on the thinned image [Fig: 2.15(c)]. After that, convex shape segments are segmented out from the straight line approximation of the character image [Fig: 2.17]. The skeleton image with strokes approximated as straight line segments can be represented as a graph with the edges corresponding to the approximation points (i.e intersection points and end points of the straight line segments). Using graph traversal, the strokes in the skeleton image are identified.

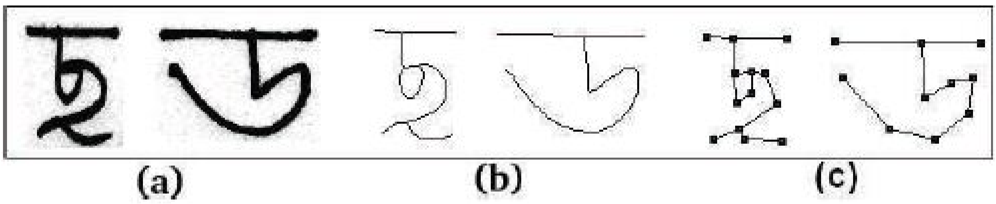


Figure 2.15: Straight line approximation of skeleton images. (a) Input image, (b) Skeleton image, (c) Straight line approximation.

The strokes are then partitioned into convex segments [Fig: 2.17]. Each convex segment in the stroke is approximated by a shape prototype selected from a fixed set of shape primitives [Fig: 2.16]. To identify best matching shape,a set of parameters is computed. So, the feature vector will be a sequence of shape primitives, one for each of the detected convex segments.

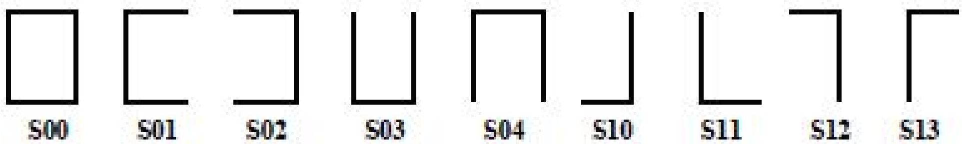


Figure 2.16: Different convex shape primitives.

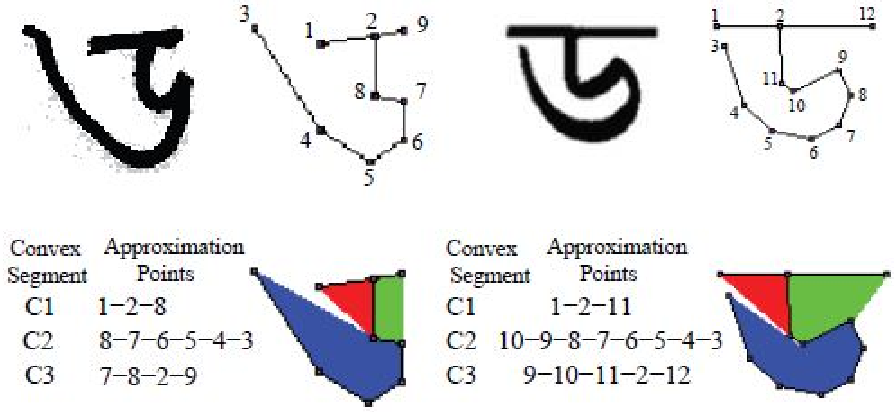


Figure 2.17: The top row shows the straight line approximations of a handwritten and a printed Bangla character. The bottom row shows the identified convex segments and the set of approximation points which belong to these segments.

2.3.5 Gradient based feature extraction

Here, gradient of the gray scale image is used as feature [15]. The gray-scale image of each handwritten character is normalized into $64X64$ size. The Sobel gradient operators are then used to compute the gradient. The Sobel operators are two $3X3$ weighted masks to compute the gradient components in horizontal and vertical directions. The gradient values and directions are then computed.

The directions of gradient are divided into eight equi-length ranges. These eight directional ranges are mapped to eight direction codes, which are integers starting from 0 and ending at 7. Fig: 2.18 shows a basic Bengali character represented by these direction codes obtained by the Sobel gradient operators.

Now, each character image (binary and normalized, as mentioned above) is divided into $8X8$=64 blocks, each block containing $8X8$ pixels. The directions corresponding to each of these 64 blocks are computed. We get 8 direction values for these blocks (as explained above). We further reduce them from 8 to 4 directions in total, by considering d=d mod 4 so that the larger between two opposite directions, namely d and d mod 4, map to the smaller between them. That is, 0 and 4 are mapped to 0 only, 1 and 5 mapped to 1 only, and so on. Thus, finally we get 4 direction codes for each block; and as there are 64 blocks, we get a total of $64X4$=256 features.

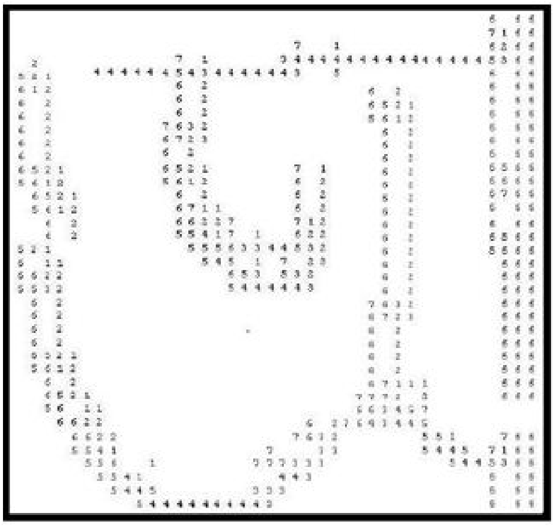


Figure 2.18: A Bengali character represented by eight direction codes $(0, 1...6, 7)$, with the 0 information omitted (white fields inside) for sake of clarity.

2.3.6 Feature extraction based on curvelet transform

Here, a feature extraction technique based on curvelet transform is introduced [13]. Curvelets are good at representing edges in an image. Character recognition using curvelet transform is primarily dependent on the edge information. Recognition accuracy of the characters can be increased by changing the edge information in the spatial domain by morphologically thinned and thickened and taking the curvelet transform of the changed versions of the image. For each character image,there will be five sets of curvelet coefficients. These five sets are used to train five KNN classifiers. The main idea behind using this scheme is that, if a character is failed to be recognized in its original form from its edge information, it may be recognized if the edge positions are slightly varied.

2.4 Summary

In this chapter, several features of Bangla characters and the extraction techniques of those features are described as we know, choice of a good feature can increase the recognition accuracy of the OCR system profoundly. After studying those features and their extraction techniques, we have chosen gradient of edge pixels and connecting point distance of a character as our proposed features. A thorough description about our proposed features and their extraction technique will be discussed in the coming chapters.

Chapter 3

Proposed Segmentation Method and Features for recognition

3.1 Introduction

Segmentation and Recognition of handwritten Bangla document is one of the most challenging subject in the Optical Character Recognition System because of the skewness of freestyle handwriting and the overlapping of different character components. Variation of writing style and complex character modifiers of Bangla language complicate the segmentation process hardly. And huge character set with complex and modified character increase the recognition complexity.

Even though Bangla is one of the popular language around the world, there are few proposal on Segmentation and Recognition for handwritten Bangla language. There are two acceptable proposal character segmentation based on *water reservoir principle* [17] and *PLM* [20].

In this chapter, we propose a writer independent scheme to segment handwritten Bangla script by following computation based techniques. Using Potential Line Markers(PLMs), document is segmented into lines. Words are segmented on average word size. Compute Junction Point in Matra zone and we segments the characters at those Junction Points. For recognition, Gradient and Connecting Point features are used as feature. Gradient matrix is computed applying *sobel gradient operator*. This angular information of pixel to pixel is processed into 4 buckets which is used as Gradient feature. In extracting Connecting Point feature, merged double gradient response indicates the presence of connecting point in the character. Number of connecting points and their inter-distance are Connecting Point feature. In recognition, 3-layer feedforward Neural Network is fed with in total 50 features to recognise 50 different classes of bangla handwritten character.

3.2 Pre-Processing

The color of document(page color and writing color) and the quality of document image influence the recognition system. The widest contrast between page color and handwriting color shows higher accuracy rate. And better quality document image without shadow effect also shows better accuracy. The whole document of Bangla Handwriting script is prepared for segmentation phase though few techniques.

3.2.1 Grayscale image and Binarization

The color documented is converted as grayscale image. Then using global threshold using *Otsu's method* [16] the document becomes an equivalent binary image where page color is brighter than the handwriting. In our experiment, we observe that, for uniform document page color(background), global threshold method more accurately distinguishes handwriting from background than other methods. And for nonuniform document page color(e.g. watermark, shadow), local threshold method performs better in distinguishes handwriting from background.

3.2.2 Noise Removal

For bad quality document image and others, noise can be infused with the document image. The isolated noise is removed by scanning whole document pixel by pixel to detect black pixel which are unconnected in 8-neighbourhood [9].

3.3 Segmentation

In segmentation phase, document is segmented in to lines, then words and at last in characters. Segmentation process is mostly adapted from A. Rahman et al. [20].

3.3.1 Line Segmentation and Word Segmentation

In Line Segmentation, the whole document is divided in several column segments [20] of average word width. Then Potential Line Marker(PLM) [20] of each column segment is evaluated. Noisy PLMs are removed by using *Noise Removal Algorithm* [20]. Adjacent PLMs are connected applying *Construct-Line Algorithm* and *Add_new_Marker Algorithm* [20]. Thus each PLM's length becomes equal to the whole document's width. Then portion of document between two adjacent PLM is cropped and thus whole document is segmented. In perfectly line segmented document, we get a line of words, then a white gap between lines, so on.

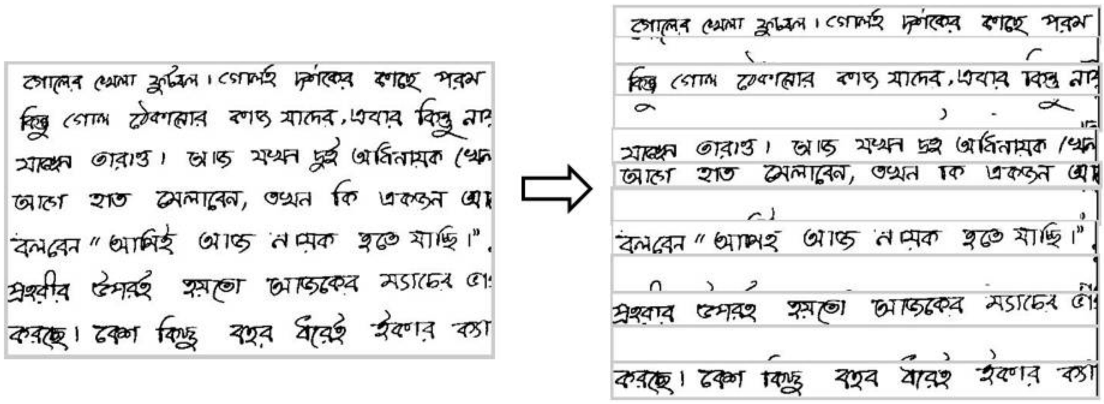


Figure 3.1: Line Segmentation

In each line of words, we search for white pixel column run (across the width of the segmented line). Word resides between two white pixel column runs which are not in consecutive column. Thus words of a line are segmented [20].

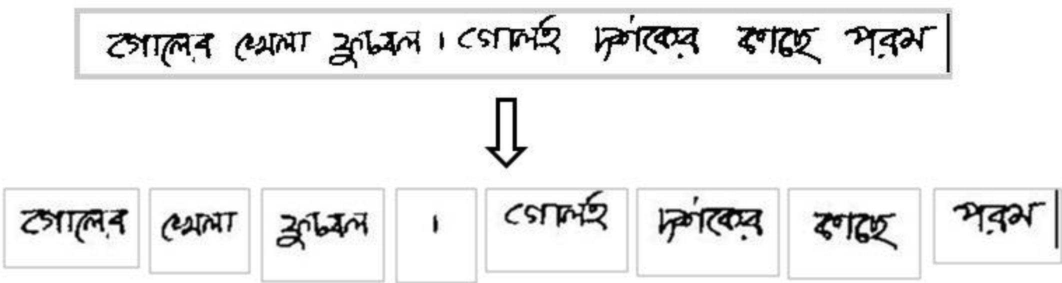


Figure 3.2: Word Segmentation

3.3.2 Character Segmentation

Before starting character segmentation, we apply a few preprocessing with morphological filter [9][7][21] because they give better performance in character segmentation. First, one step *thicken morphological filter* is applied to thicken the handwriting. It connects the possibly disconnected letters of a word. Now we evaluate *Minimum Bounding Box* [2] of each word. Then, inside the *Minimum Bounding Box* of word *skel morphological filter* [3] and *thin morphological filter* [11][19] are applied on the inverted character image. *Skel morphological filter* removes pixels on the boundaries of objects but does not allow objects to break apart; the remaining pixels make up the image skeleton. *Thin morphological filter* thins objects to lines. It removes pixels so that an object without holes shrinks to a minimally connected stroke, and an object with holes shrinks to a connected ring halfway between each hole and the outer boundary.
In a word, bangla letters are connected by matra, so mode of vertical histogram will be or near matra. That is how we evaluate matra of a word. Respect of matra, a word is vertically divided in to two regions, *Matra zone* or *Upper zone* and *Lower zone*. *Matra zone* is defined as from the uppermost point to matra and *Lower zone* is from matra to lowermost point. Then *Matra zone* is searched to detect junction point between two characters. Characters of a word are segmented at these junction point [20].

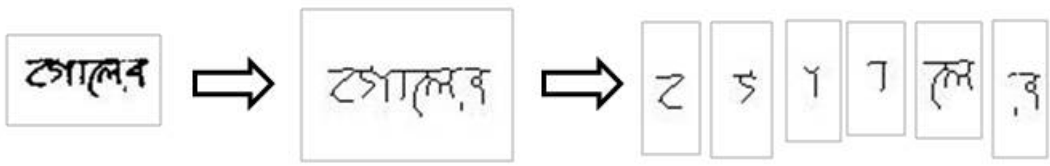


Figure 3.3: Character Segmentation

3.4 Feature Extraction

Feature extraction is one the most important stage for any recognition system. To recognise handwritten bangla character, we consider features which are rotation invariant, translation invariant and scale invariant. Mainly different angular and stationary information of edge pixels of characters are used as features. For different handwriting styles, segmented character size of same character varies over times. Moreover, in bangla, different character has different size. Therefore, we resized each segmented character into (50 × 50).

3.4.1 Gradient of character pixel

Relative position of an edge pixel respect to its neighbouring edge pixels varies from character to character. Human senses a character realizing edge pixel's relative position respect to its neighbouring edge pixel. Hence this angular information of the edge pixels can be used to uniquely recognise individual character.

3.4.1.1 Evaluating Gradient

Detecting change in intensity to identify edge pixels, first-order derivative or gradient can be used. From the segmented character, skeleton image of the character is formed, in which a line is a 1-pixel thick. Then *sobel gradient operator* is applied to get gradient. First compute directional gradients,G_x and G_y, with respect to the *x-axis* and *y-axis*. Here *x-axis* is defined along the columns going right(horizontal) and the *y-axis* is defined along the rows going down(vertical). Then the gradient magnitude,G_{mag} and gradient direction,G_{dir} are then computed from their orthogonal components G_x and G_y[9].

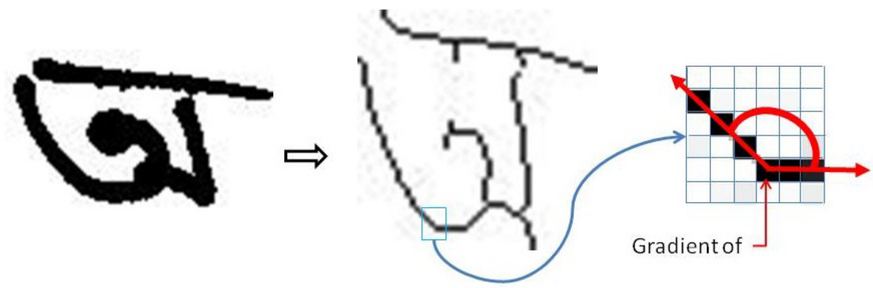


Figure 3.4: Gradient of a pixel

3.4.1.2 Histogram of Gradient

Gradient Operator gives response when it encounters a pixel or edge. By the nature of gradient operator for an edge, we get double responses(positive response before the edge, zero positive response on the edge and negative response after the edge; here scan is done top to bottom, left to right). So only the response pixels from G_{dir} bear the relative angular pixel to pixel information.

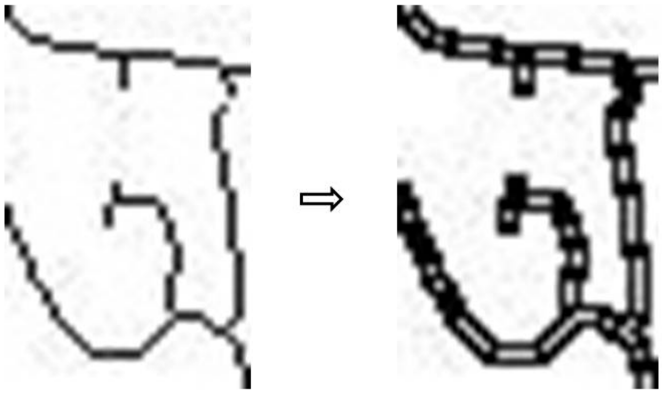


Figure 3.5: Double gradient response at contour pixels

G_{dir} contains angles in degrees within the range $[-180, 180]$ measured counterclockwise from the positive *x-axis* (horizontal). As G_{dir} is implicitly *tangent* operation and in cartesian domain, prime period of periodic function *tangent* is π; the range minimizes into $[0, 180]$.
Based on the value of an angle, it is grouped into one of the 4 bucket corresponding to 4 angular intervals of 45° (0°-45° as bucket no. 1, 45°-90° as bucket no. 2, 90°-135° as bucket no. 3, 135°-180° as bucket no. 4). Total number of angles in each bucket is the frequency of an angle of a bucket. This normalized gradient histogram gives 4 features(4 buckets).

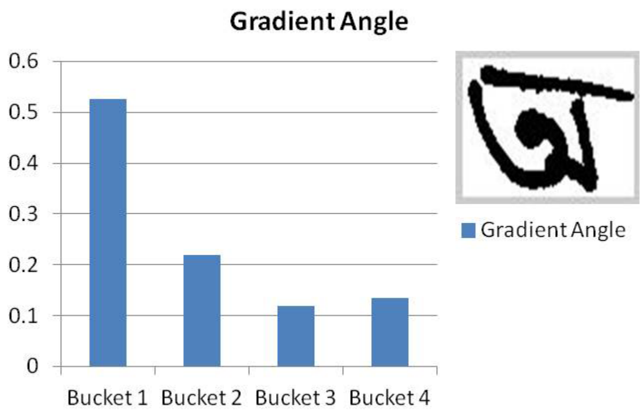


Figure 3.6: Histogram of Gradient

Rotation invariance of Gradient feature As rotation invariant of feature is expected, this gradient feature has to maintain that. In a character, relative position of a pixel with respect to its neighbouring pixels will not change if the character is rotated in any arbitrary directions. Hence relative angular information(gradient direction) of the pixel with respect to its neighbouring pixels will also not change. Moreover, rotation is a member of *Rigid-Body Transformation*. So it will preserve pixel to pixel angle. Since relative angular information does not change because of rotation, Gradient feature is rotation invariant.

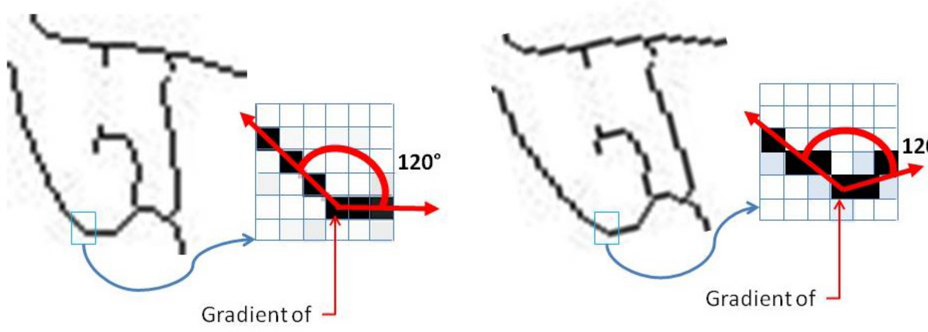


Figure 3.7: Rotation invariance of Gradient

Translation invariance of Gradient feature As the relative angular information is considered here, the actual coordinate of pixels does not influence that. Whether pixels are near the origin(topmost and leftmost corner) or near the end(bottommost and rightmost corner), the relative angular information will still be same. And translation is also a member of *Rigid-Body Transformation*. So it will definitely preserve pixel to pixel angle. Since relative angular information does not change because of translation, Gradient feature is translation invariant.

Scale invariance of Gradient feature Isotropic Scale is a member of *Similitude Transformation*, it will preserve pixel to pixel angle. But for non-isotropic scale, it will not preserve. Literally, Scale is a member of *Linear Transformation* which does not ensure about preserving angle. So different size of same character will exhibit different pixel to pixel relative angular information. By default, Gradient is not scale invariant. However, as we have fixed the size of segmented character, different size of same character will be no more different. All segmented character is now (50×50). Now gradient feature is indifferent of scaling.

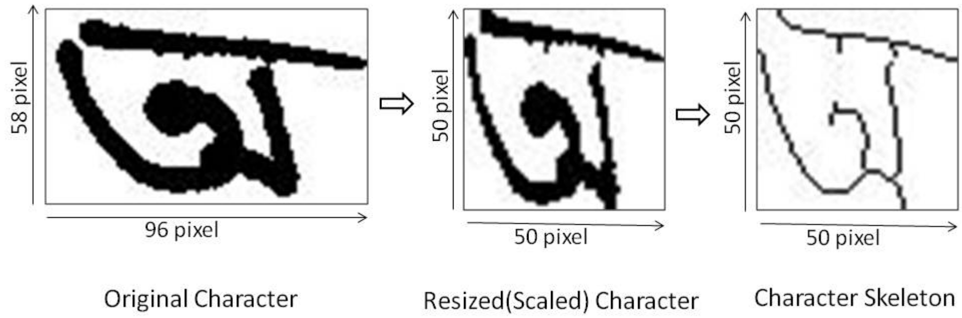


Figure 3.8: Scale invariance of Gradient

3.4.2 Number and inter-distance of Connecting Points

In primitives, each character is composition of several lines. Lines can be straight or curvy. That is why, in *syntactic method* [10] various primitives are used for recognition and these primitives are composed by single or several lines. However, these lines(primitives) are connecting with each other to form a character. And these connecting points of these lines(primitives) and relative position of these connecting points from one another are also district for each character. Here we consider the number of connecting points of a character and inter-distance among these connecting points as feature.

3.4.2.1 Connecting Point detection

Using *sobel gradient operator*, connecting point can be detect. We know that each line gives double response for gradient operator and actual character pixel has zero response. A typical column-wise run of gradient image is $\ldots 45\ 45\ 0\ {-90}\ {-90} \ldots$; where 0 is the actual character pixel and we get positive response(gradient value) above it and negative response(gradient value) below it. This can be expressed in regular expression as,

$$\ldots \left(p^{+}z^{*}n^{+}\right) z^{+} \left(p^{+}z^{*}n^{+}\right) z^{+} \left(p^{+}z^{*}n^{+}\right) \ldots$$

where p=positive gradient value, z=zero gradient value, n=negative gradient value. And each $(p^{+}z^{*}n^{+})$ sequence is well separated by at-least one or more consecutive zero gradient value.

When two lines are connected at a point, two double responses are merged and this merged response is found before encountering the actual connecting point. A typical column-wise *merged response* is look like $\ldots 135\ 161.56\ {-180}\ {-180}$

135 0 -45 $-45\ldots$; which is like in regular expression as,

$$\ldots\left(p^{+}z^{*}n^{+}\right)z^{+}\underbrace{\left(p^{+}z^{*}n^{+}\right)\left(p^{+}z^{*}n^{+}\right)}z^{+}\left(p^{+}z^{*}n^{+}\right)\ldots$$

where p=positive gradient value, z=zero gradient value, n=negative gradient value. And in the *merged response* no zero gradient value is found in between two $(p^{+}z^{*}n^{+})$ sequences. This *merged response* is the indication of a connecting point.

First, we get the gradient response of the character(gradient direction,G_{dir}). Then scan each column applying DFS [12] looking for *merged response*. If a *merged response* is found, we consider the next column coordinate as connecting point. Let, (x_i, y_j) is the actual coordinate of *merged response* where *ith* number of row and *jth* number of column. So, the connecting point is at (x_i, y_{j+1}). In our experiment, we observe that the average number of connecting point in bangla handwriting is below 10. We restrict maximum considered connecting point at number 10. This number of connecting point is considered as a feature.

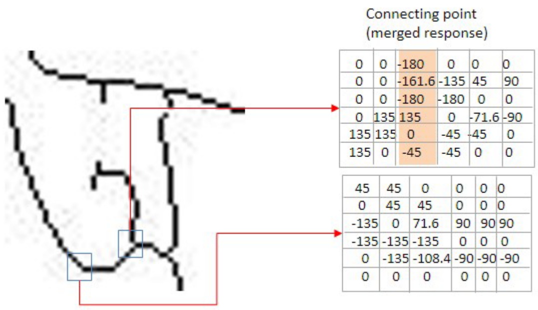


Figure 3.9: Detection of connecting point

3.4.2.2 Inter-distance calculation

After detecting the connecting points, we calculate the inter-distance between connecting points. Using Euclidean distance method, inter-distance between connecting point (x_i, y_j) and (x_p, y_q) is given by,

$$interdistance = \sqrt{(x_i - x_p)^2 + (y_j - y_q)^2}$$

We consider top-most 10 connecting point and measure distance in between each of them. So from inter-distance, we get total $45(^{10}C_2)$ features. Now using the number of connecting points and this 45 features, in total 46 features is extracted by applying Connecting Point feature.

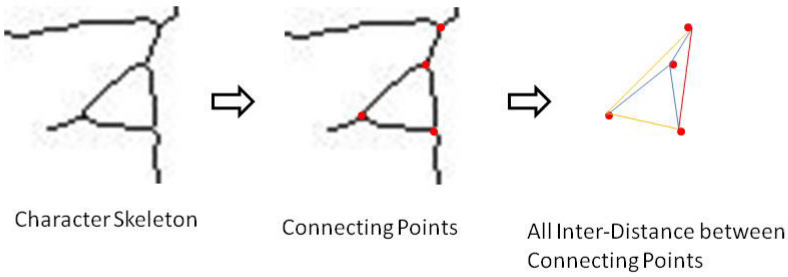


Figure 3.10: Connecting Point Feature

Rotation invariance of Connecting Point feature Rotating the character in any arbitrary direction in any amount does not change the number of connecting point as they are formed by connecting several lines. As rotation is a member of *Rigid-Body Transformation*, it will preserve pixel to pixel distance. Rotation may change the actual coordinate of pixels, but does not change relative distance from pixel to pixel. So Connecting Point feature is rotation invariant.

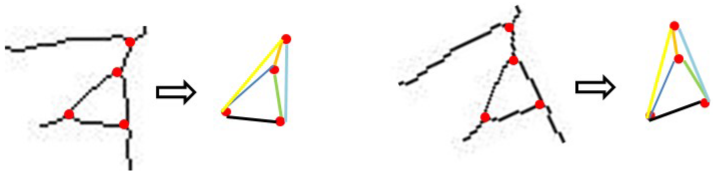


Figure 3.11: Rotation invariance of Connecting Point feature

Translation invariance of Connecting Point feature Translation maps pixels in new coordinate position but still number of connecting points and relative distance from pixel to pixel are unchanged. Moreover, like rotation, translation is a member of *Rigid-Body Transformation*. So it is translation invariant too.

Scale invariance of Connecting Point feature Being a member of *Linear Transformation* class, Scale does not ensure about preserving distance. So different size of same character will exhibit different pixel to pixel distance. The number of connecting point is still unchanged. However, as we have fixed the size of segmented character in (50×50), different size of same character will be no more different. Thus inter-distance among connecting points will not unchange.

3.5 Recognition

For recognition, a Feedforward Neural Network(NN) is fed with the feature vector. As our recognition is multi-classed and NN shows satisfactory accuracy rate in Bangla recognition [22], feedforward NN is chosen for handwritten bangla recognition. Here, we have 50 features and class is considered as 50(total number of bangla characters).

We use 3-layer NN with 2-layer hidden layers. Each of this hidden layer has 50 nodes each and use *tan-sig(tangent sigmoid)* activation function. In output layer, linear activation function is used for 50 classes. For training, *Levenberg-Marquardt* backpropagation [8] algorithm is used; although it uses much memory, it is faster with good accuracy. Training data is collected from DATABASE [1], total 1000 sample bangla handwritten bangla character written by 50 different writer, 20 samples for each 50 class. Samples are randomly chosen for training, validation and testing; where the relative ratios are 70%, 15%, 15%.

3.6 Summary

In this chapter, we proposed a writer independent recognition system for Bangla handwriting. We preprocessed the document for removal of noise and background information. Then we segment the lines using PLM method. Using average word size, words are segmented. Matra zone is detected and the junction points where the two characters touches are also determined to segment the word into characters. For recognition, Gradient and Connecting Point features are used. Applying *sobel gradient operator*, angular information of pixel to pixel is extracted as Gradient feature. For Connecting Point feature, merged double gradient response indicates the presence of connecting point. Number of connecting points and their inter Euclidean distance are used as Connecting Point feature. In recognition phase, 3-layer feedforward Neural Network is used; where 50 features are fed to recognise 50 classes.

Chapter 4

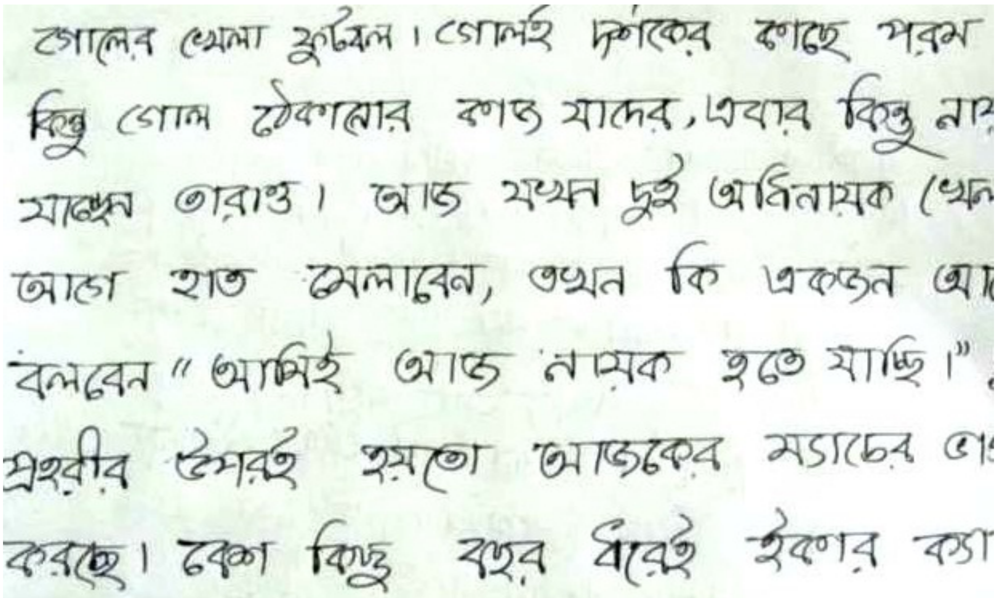
Experimental Results

4.1 Introduction

The proposed scheme of segmentation and recognition is tested mainly on free style Bangla handwritten document image written by different individual. As free style handwriting has skewness and overlapping, these complicated the process to justify the efficiency of this schema. Even though sample handwritten bangla document written by various writer from different background are very difficult to find, various handwritten scripts samples are tested in our scheme to train the recognition system for better accuracy. The proposed algorithm is applied and tested in MATLAB. We find an appreciable accuracy in recognising bangla handwritten character.
In this chapter, experimental results of our proposed algorithm is shown. We also analyze the output for various styles of handwriting.

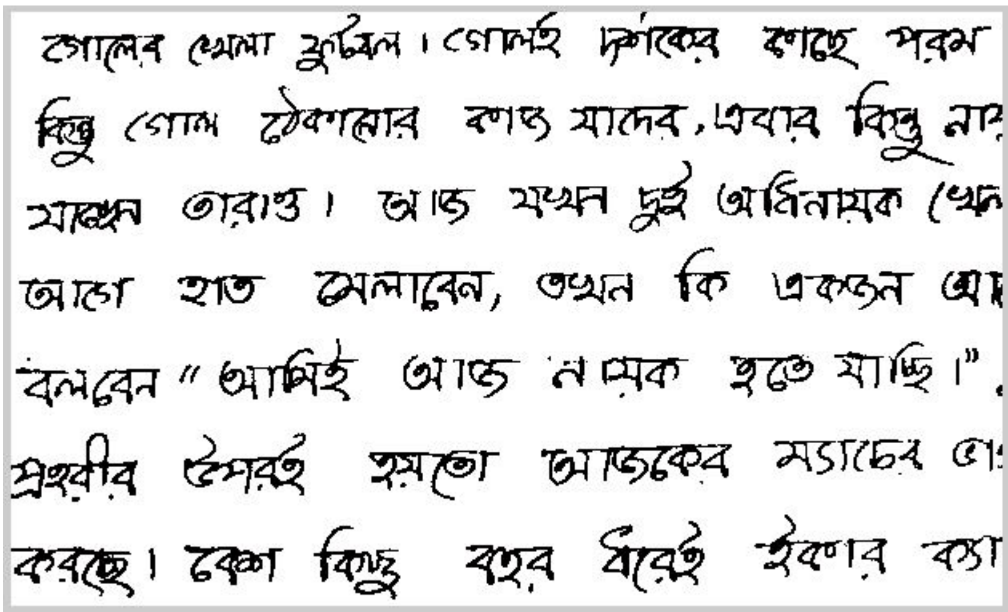
4.2 Result of Binarization

In this stage, a full size bangla handwritten document is preprocessed. The document goes through Grayscale, Binarization and Noise Removal steps. Finally we get a binary(black and white) document of handwritten bangla.



গোলের খেলা ফুটবল। গোলই দর্শকের কাছে পরম
কিন্তু গোল ঠেকানোর কাজ যাদের, এবার কিন্তু নায়
যাচ্ছেন তারাও। আজ যখন দুই অধিনায়ক খেল
আগে হাত মেলাবেন, তখন কি একজন আ
বলবেন "আমিই আজ নায়ক হতে যাচ্ছি।"
প্রহরীর কেউই হয়তো আজকের ম্যাচের গো
করছে। কেননা কিছু বছর ধরেই ইকার কথা

Figure 4.1: Sample Handwritten Bangla Document



গোলের খেলা ফুটবল। গোলই দর্শকের কাছে পরম
কিন্তু গোল ঠেকানোর কাজ যাদের, এবার কিন্তু নায়
যাচ্ছেন তারাও। আজ যখন দুই অধিনায়ক খেল
আগে হাত মেলাবেন, তখন কি একজন আ
বলবেন "আমিই আজ নায়ক হতে যাচ্ছি।"
প্রহরীর কেউই হয়তো আজকের ম্যাচের গো
করছে। কেননা কিছু বছর ধরেই ইকার কথা

Figure 4.2: Handwritten Bangla Document after Preprocessing

4.3 Result of Segmentation

4.3.1 Result of Line Segmentation

The proposed algorithm works well for most of the input images and provides high accuracy output for line segmentation. The output for line segmented images shows that some lines are wrongly segmented that are not actually lines because the gaps between two consecutive lines are bigger than usual gap between two lines or lines are overlapped with each other. We tested a number of sample scanned images of various writing style of different individuals and found that the accuracy was almost 96% for line segmentation.

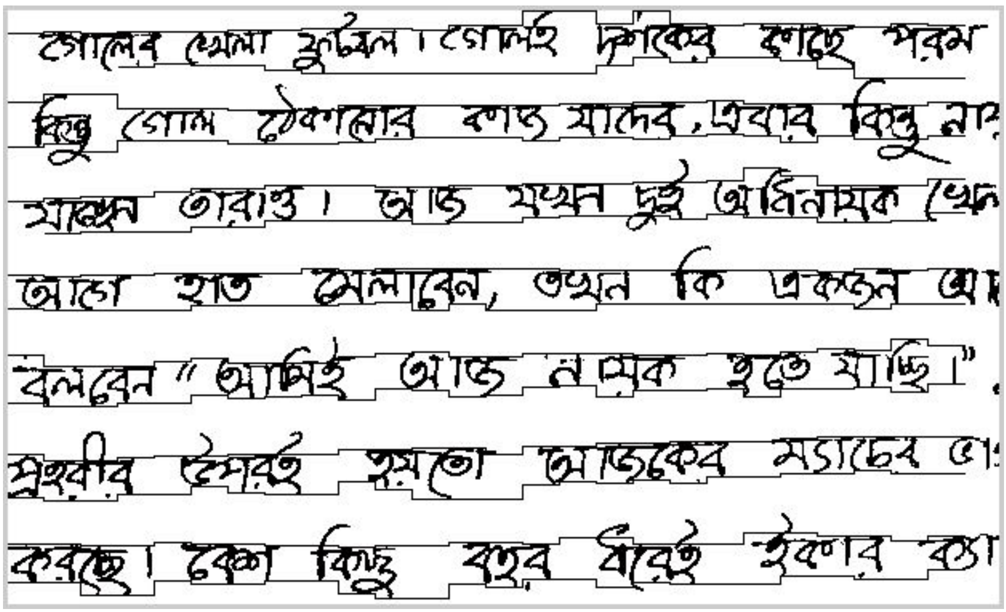


Figure 4.3: Handwritten Bangla Document applying PLM

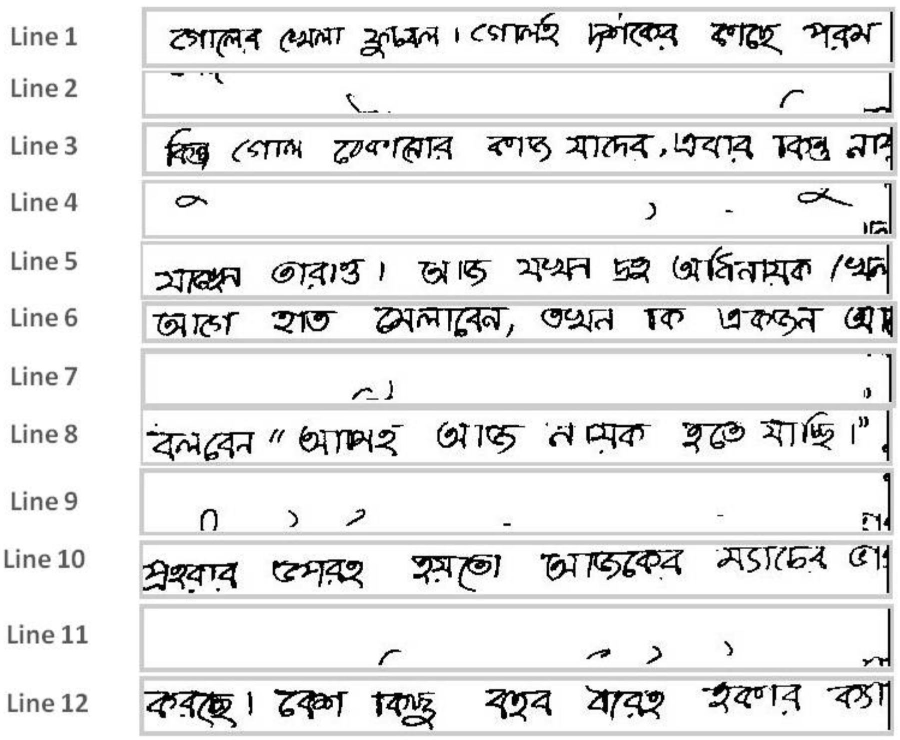


Figure 4.4: Handwritten Bangla Document segmented in Lines

4.3.2 Result of Word Segmentation

The proposed algorithm works also well in the word segmentation. It can compute the word delimiter properly to segment the lines into words. As free-style handwriting samples are taken so the distance between two consecutive black pixel region are not always detect the proper words. Using our proposed algorithm , the accuracy level of ward segmentation was also high and accuracy rate is almost 92%.

Figure 4.5: Handwritten Bangla Document- Line 1,3,5 segmented in Words

4.3.3 Result of Character Segmentation

In the case of character segmentation, our proposed algorithm works well. But due to the skewness and overlapping of Bangla characters in handwritten script, a few character and character modifiers are difficult to segment. Though the accuracy rate is almost 65% to segment the words into characters.

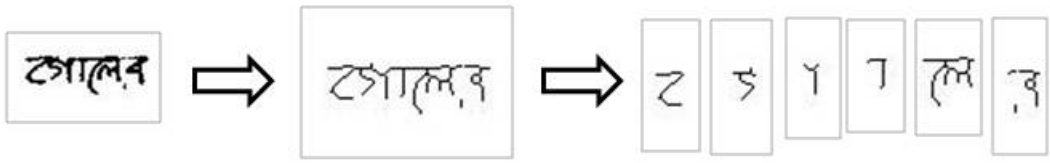


Figure 4.6: Handwritten Bangla Document- Word1 segmented in Characters

4.4 Result Analysis

First, we analyse the result of line segmentation. Applying on various full size bangla documents of various size written by wide range of individuals, we plot *number of actual lines in samples* in $x - axis$ and *number of recognised lines* in $y - axis$. Now, the accuracy rate of line segmentation is calculated by,

$$\text{Accuracy rate of Line Segmentation} = \frac{\text{Number of recognised lines}}{\text{Number of actual lines in samples}} \times 100\%$$

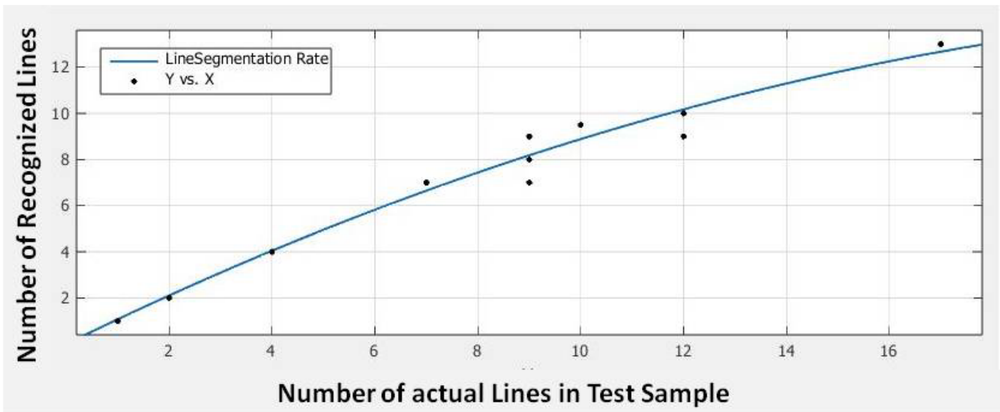


Figure 4.7: Handwritten Bangla Document- Line Segmentation Rate

Here, in our system, accuracy rate of Line Segmentation is 96%.

To analyse the result of word segmentation, we feed the segmented lines to character segmentation process. And plot the result as *number of actual words in sample line* in $x - axis$ and *number of recognised words* in $y - axis$. Now, the accuracy rate of word segmentation is calculated by,

$$\text{Accuracy rate of Word Segmentation} = \frac{\text{Number of recognised words}}{\text{Number of actual words in sample line}} \times 100\%$$

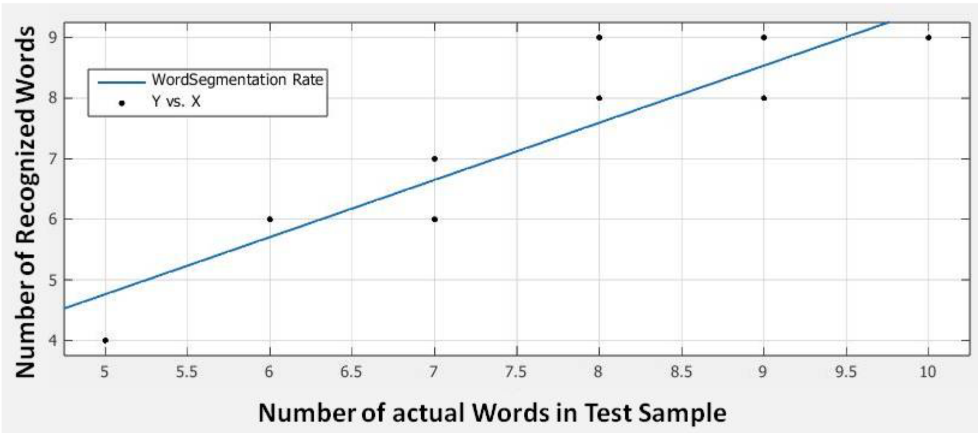


Figure 4.8: Handwritten Bangla Document- Word Segmentation Rate

Here, our system's accuracy rate of Word Segmentation is 92%.

We plot the result of character segmentation as *number of actual characters in sample words* in $x-axis$ and *number of segmented characters* in $y-axis$. Now, the accuracy rate of character segmentation is calculated by,

$$\text{Accuracy rate of Character Segmentation} = \frac{\text{Number of segmented characters}}{\text{Number of actual characters in words}} \times 100\%$$

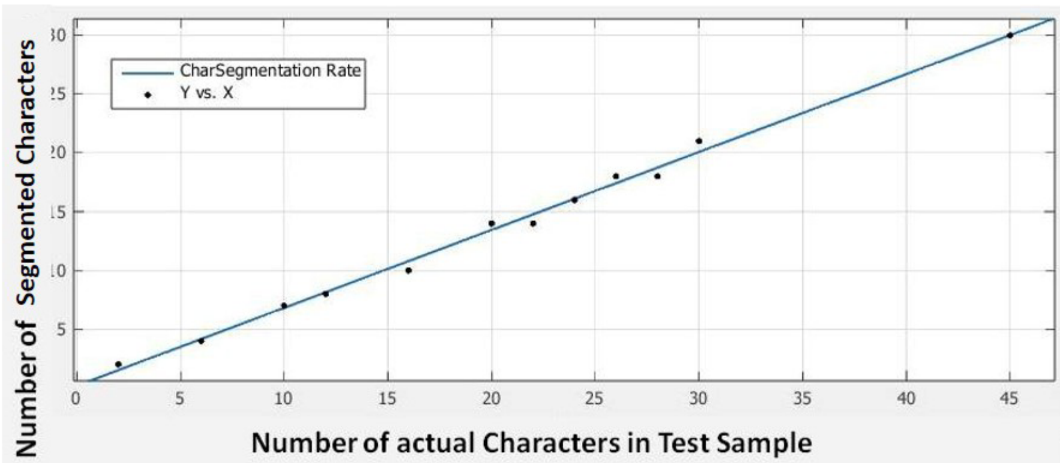


Figure 4.9: Handwritten Bangla Document- Character Segmentation Rate

Accuracy rate of Word Segmentation is 65% in our system. As we feed the output of line segmentation into word segmentation system and the output

of word segmentation into character segmentation system, the unrecognised mistakes in any level propagate along the system. These decrease the accuracy of our recognition system.
The accuracy of character recognition is given by,

$$\text{Accuracy} = \frac{1}{N} \sum_{k=1}^{\text{Number of training sets}} \frac{\text{Number of recognized characters}}{\text{Number of actual characters}} \times 100\%$$

In our recognition system, accuracy rate is about 62%. Due to low character segmentation rate, our character recognition accuracy is also affected.

4.5 Summary

The proposed scheme of segmentation is tested mainly on free style Bangla handwritten script image written by different individual. As free style handwriting has skewness and overlapping that complicate the process to justify the efficiency of that schema. Various handwritten scripts samples are tested in our scheme to gain the proper accuracy. Based on the line word and character segmentation method ,our algorithm has gained a high accuracy of almost 96% for line segmentation, 92% for word segmentation, 65% of character segmentation and 62% of character recognition respectively.

In character segmentation the accuracy level is pretty low because of the limitation of the detection a few characters. These characters are divided into two pieces that cause a bad impact on recognizing the character.

Chapter 5

Conclusion and Future Study

5.1 Findings

In this paper an effective scheme to recognise writer independent handwritten Bangla script is proposed. This proposal is a total statistic free approach where computational based techniques are implemented to segment the Bangla script document into lines,words and characters; and angular and stationary relative relation of character pixels are used to recognise. We use construct line algorithm for the line segmentation and word delimiter to segment the line into words. We also introduced a different approach of the blockage of vertical white pixel flow to determine the Matra zone and ease the character segmentation process. For recognition, pixel gradient information and relative inter-distance between connecting point are introduced. Due to poor character segmentation rate and simple feature types, the accuracy level is not high enough. So in future we consider the constraint to make better accuracy on this method.

5.2 Future Study

A number of researcher worked with Bangla handwriting recognition system but very few researchers became succeeded in properly recognition the Bangla characters along with complex and modified characters. But our new algorithm outputs accurate result even in cursive characters and skewed lines. It cant say about which character modifier should be attached with which character modifiers. Different character size and different orientation are not recognizable in our proposed system. If further research is done then it can be told which character modifiers should be attached with character modifiers, different font size can be correctly identified and oriented character can be recognised.

Our next step will be to improve the Character segmentation technique and features. Then we have planned to develop a Multi-resolution and Multi-Oriented Handwritten Bangla Character Recognition approach to use the result of segmentation process to build a complete OCR system. For contracting multi-resolution and multi-Oriented character recognition system, two kinds of approaches are found:

- Wavelet method to detect multi-resolution [14]
- Circular and Convex hull feature to detect multi-orientation [18]

Our recognition accuracy will improve by following a satisfactory character segmentation method that can properly segment the free style handwritten characters even in the presence of skewness and overlapping of characters along with features that can identify multi-resolution and multi-oriented character even when complex and modified characters are present. Our future study will be on finding satisfactory character segmentation approach and features that will eventually maximize our character recognition rate to a satisfactory level.

List of Figures

References

[1] Handwritten Bangla Character Database. http://www.isical.ac.in/~ujjwal/download/Banglabasiccharacter.html, July 2014.

[2] Minimum Bounding Box. http://en.wikipedia.org/wiki/Minimum_bounding_box, July 2014.

[3] Morphological Filter. http://www.mathworks.com/help/images/ref/bwmorph.html, July 2014.

[4] M. M. Alam and M. A. Kashem. A Complete Bangla OCR System for Printed Chracters. *Journal of Convergence Information Technology(JCIT)*, 2010.

[5] S. Bag, G. Harit, and P. Bhowmick. Topological Features for Recognizing Printed and Handwritten Bangla Characters. September 2011.

[6] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu. Handwritten Bangla Alphabet Recognition Using an MLP Based Classifier. pages 285–291. 2nd National Conference on Computer Processing of Bangla, February 2005.

[7] E. R. Dougherty. *An Introduction to Morphological Image Processing.* 1992.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Second edition.

[9] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Pearson Prentice Hall^{TM}, 3rd edition, 2008.

[10] M. B. Islam, M. M. B. Azadi, M. A. Rahman, and M. M. A. Hashem. Bengali Handwritten Character Recognition Using Modified Syntactic Method.

[11] L. Lam, S. W. Lee, and C. Y. Suen. Thinning Methodologies-A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 879, September 1992.

[12] C. E. Leiserson, T. H. Cormen, C. Stein, and R. Rivest. *Introduction to Algorithms*. The MIT Press, third edition, July 2009.

[13] A. Majumdar. Bangla Basic Character Recognition Using Digital Curvelet Transform. *Journal of Pattern Recognition Research(JPRR)*, January 2007.

[14] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, Burlington, 2008.

[15] S. Mandal, S. Sur, A. Dan, and P. Bhowmick. Handwritten Bangla Character Recognition in Machine-printed Forms Using Gradient Information and Haar Wavelet. International Conference on Image Information Processing(ICIIP), 2011.

[16] N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 62–66, 1979.

[17] U. Pal and S. Datta. Segmentation of Bangla Unconstrained Handwritten Text. *Document Analysis and Recognition*, pages 1128 – 1132, August 2003.

[18] U. Pal, P. P. Roy, N. Tripathy, and J. Llados. Multi-oriented Bangla and Devnagari Text Recognition. *Computer Vision and Pattern Recognition Unit(CVPRU)*, pages 4124–4136, June 2010.

[19] Pratt and K. William. *Digital Image Processing*. J. Wiley & Sons Inc., 1991.

[20] A. Rahman, F. Yasir, H. M. Cyrus, and W. Bari et al. Segmentation of Handwritten Bangla Script. Undergraduate B.Sc Thesis, Bangladesh University of Engineering and Technology(BUET), Dhaka-1205, Bangladesh, 2013.

[21] P. Soille. *Morphological Image Analysis; Principles and Applications*. 1999, 2003.

[22] P. Upadhyay, S. Barman, D. Bhatacharyyaa, and M. Dixit. Enhanced Bangla Character Recognition using ANN. *International Conference on Communication Systems and Network Technologies*, 2011.