TOWARDS ONLINE BANGLA HANDWRITING RECOGNITION

K. Roy* and S. Sen*

*West Bengal State University, Barasat
*Future Institute of Technology, Kolkata

Abstract - This paper describes a procedure to recognize online bangla handwriting recognition in unconstrained domain. For recognition purpose we are using two phase Neural Network. Thinking is that if the number of output class is being reduced then recognition rate can be enhanced or not. Experimented result of the system after testing 200 bangla word is x%.

Keywords: stroke, special character, Valid character set, May be valid character set, modifier character set, not character set, Neural Network, classifier.

I. INTRODUCTION

The goal of handwriting recognition is to identify an input character’s image correctly and it is an application area of pattern recognition, in addition the conversion of handwritten text image to editable form is important to many automated systems. Two main techniques are used for character recognition depending on the intended application. The first is off-line optical recognition (OCR), which accepts it’s input from a digital scanner or from a picture using some image processing algorithm needed before classification step. The alternative approach is online character recognition, which accepts input data in real time and then computes the relationship between points to extract the features in real time. This work is based on online approach. Any handwriting recognition system typically consists of the following phases: scanning, digitization, preprocessing, segmentation to single character or segments related to character, feature extraction, recognition using classifier, post processing for verification using lexicon and the last step of the system is evaluation. Digitization consists noise elimination, gaps filling, size translation, normalization and binarization, while preprocessing phase convert the original form of input image to another form skeleton to simplify the feature extraction process. In this work total 59 basic strokes are considered to recognize purpose. By stroke I mean collection of pen points between a pen down and pen up. In other words the number of sample points collected by a continuous writing of pen without lifting it. Recognition is based on neural network. After segmentation of bangla word and feature extraction when a segmented part goes to neural network as input for recognition, it falls under any one of possible 59 classes. In that case success rate is not too good. So, in this work it has been tried to recognize bangla word using two-phase neural network. Main target is to improve success rate by reducing the number of output classes.

II. DATA COLLECTION & SEGMENTATION

2.1. DATA COLLECTION: On-line handwriting recognition involves the automatic conversion of text as it is written on a special digitizer or A4 take note where a sensor picks up the pen-tip movements X (t), Y (t) as well as pen-up/pen-down switching. That kind of data is known as digital ink and can be regarded as a dynamic representation of handwriting. The ink signal is captured by either:

- A paper based capture device
- A digital pen on patterned paper
- A pen-sensitive surface such as a touch screen

In this work we use pen positions (x, y) and pen pressure (z) sampled at a certain interval from the pen tablet. If the stroke is continuing i.e. pen pressure is ‘on’ i.e. ‘pen-down’, the value of z at a particular point or pixel will be 1 and x, y denotes the pixel’s x coordinate values and y coordinate values when using pen we write any bangla word or character.

The information on strokes and trajectories is mathematically represented in an ink signal composed of a sequence of 2D points ordered by time. No matter what the handwriting surface may be, the digital ink is always plotted according to a matrix with x and y-axes and a point of origin.

Online data acquisition captures just the information needed, which is trajectory and strokes, to obtain a clear signal. This effective information makes the data easier to process. To collect the data (Word) we use Wacom tablet, A4 take note and the datasheets. The pen pressure represents pen ups and downs in a continuous manner. In our conventional research, we did not use pen-up-down information
whether the pen leaves (pen-up) or touches (pen-down) the tablet surface or A4 take note surface. For online data collection, the sampling rate of the signal is considered fixed for all the samples of all the classes of character. Thus the number of points M in the series of co-ordinates samples of all the classes of character. Thus the number of points M in the series of co-ordinates for a particular sample is not fixed and depends on the time taken to write the sample on the pad. As the number of points in actual trace of the characters are generally large and varies greatly due to high variation in writing speed, a fixed lesser number of points, regularly spaced in time are selected for further processing. The digitizer output is represented in the format of

\[ p^M \in \mathbb{R}^2 \times \{0, 1\} \], where \( p_i \) is the pen position having x-coordinate (\( x_i \)) and y-coordinate (\( y_i \)) and M is the total number of sample points. Let (\( p_i \)) and (\( p_j \)) be two consecutive pen points. We retain both of these two consecutive pen points (\( p_i \)) and (\( p_j \)) if the following condition is satisfied:

\[ x^2 + y^2 > m^2 \quad \ldots \ldots . \quad (i) \]

Where \( x = x_i - x_j \) and \( y = y_i - y_j \). The parameter \( m \) is empirically chosen. We have set \( m \) equal to zero in Equation (i) to removes all consecutive repeated points. Analyzing a total of 22,000 Bangla characters it was found that, for writing Bangla characters, the number of points varies from 14 (\( \mathbb{R}_3 \)) to 189 (\( \mathbb{R}_6 \)) points. The average number of points in a Bangla character is 72. It was also noted that the character (\( \mathbb{R}_6 \)) uses the maximum number of points in average and its value is 115. It is closely followed by ‘ং’ (108), ‘ঃ’ (105), & ‘ঁ’ (104). The minimum number of points in an average is used by the character ‘ং’ (47) and is closely followed by ‘ঁ’ (49) ‘ঃ’ (51).

2.2 PREPROCESSING: Most of the classification techniques assume that the data is given in a predetermined form, which satisfy certain requirements as to quality, size, invariance, etc. However, these characteristics are commonly not satisfied by on-line handwritten data. The low quality of the data is due basically to the combination if three facts. One is the addition of noise during digitalization, which is generally generated by a badly configured digital tablet. The other is the irregularity generated by inexperienced users having an erratic handwriting. The last are variations in handwriting styles.

To overcome these problems, we use preprocessing, which involves the substitution, removal, reordering, and/or extraction of the data. Preprocessing eliminates noise, normalizes handwriting, and reduces the amount of redundant information, in order to fix the variations of handwriting and facilitate encoding of raw data into feature vectors. (I) Noise and Data reduction, (II) Smoothing, (III) Dehooking, (IV) Feature Extraction.

SEGMENTATION: In case of online character segmentation we have designed an algorithm and the procedure to design the algorithm is as follows:

Each and every online word document file contains x and y coordinates and third value (as pen-up or pen-down) as 1 or 0. If the third value is 0 then a new stroke begins, if it is 1 the stroke continues. In case of online data segmentation the data can be segmented by changing the value of pen stroke feature from 1 to 0. But the major challenges here to identify the coordinate where to segment the word and find out the complete character according to bangle script. In the process of character segmentation there is a possibility of improper segmentation, which is called either over segmentation or under segmentation. In case of under segmentation the coordinates where the segmentation is required could not be chosen and in case of over segmentation the coordinates where it is not required but has been segmented.

In Bengali handwriting the movement of each stroke is generally downside. By keeping this concept in mind it has been seen that in a downside movement stroke the point from where that downside movement starts at that point we have to split that stroke. This should be done only in the upper zone i.e. first 33% portion of the total height of the image. In the remaining 67% of the image segmentation is not needed. Generally people write any word in a manner where more than one alphabet is joined with one another. This joining is generally found in the upper 1/3rd. portion of the image (exception in few cases).

Step 1: Store each pixel of the online data in three variables corresponds to X and Y coordinates and pen feature value of 0 or 1 in third variables for identifying strokes.

Step 2: For each third variable value 0 separates each strokes scanning pixels of the word. Calculate the 30% of the height of the entire word image.

Step 3: Select at which point of stroke segmentation is needed. We have to finally segment those points of same or different strokes, which required to be segmented. So, we use one function to check at which pixel it is feasible to segment a stroke. We have to check few features of bangla characters for this process such as (i) each pixel’s distance from the start and end of the stroke, (ii) the width of the stroke up to the pixel in question from the start and end of the stroke, (iii) the height of the stroke up to the pixel in question, (iv) Total stroke distance, (v)
Total width of the word. After finding these features we have to take some ratio of (a) each pixel’s distance & Total stroke distance, (b) the width of the stroke upto the pixel in question & Total width of the word and thus to decide at which pixel of a particular stroke segmentation is feasible.

Step 4: Now if at a particular pixel it is feasible to segment the stroke, then first we check whether that pixel’s y co-ordinate value is 30% of the height or not. If it is not then there will be no segmentation. If it is, then we check whether at that pixel downside movement of the stroke starts or not. For this checking I am taking two points \( p_{i-1} \) and \( p_{i-2} \) before the point in question and similarly two points \( p_{i+1} \) and \( p_{i+2} \) after that point. Then we have to determine the angle between two line segments \( p_{i-2} \) to \( p_i \) and \( p_i \) to \( p_{i+2} \). If this angle satisfies certain range then only at point \( p_i \) stroke value will be made as 0, that is, in this point the stroke will be splitted. If the y-coordinate of \( p_{i-1} \) is \( \leq \) \( p_{i-2} \) and \( p_i \) \( \leq \) \( p_{i-1} \) and simultaneously if the y-coordinate of \( p_{i+1} \) \( \geq \) \( p_i \) and \( p_{i+2} \) \( \geq \) \( p_i \) (i.e. downside movement of stroke) then only angle is calculated. If at a particular point stroke is splitted then we skip next 9 or 10 pixels for checking of feasibility of segmentation.

Step 5: Repeat step 3 and 4 for each pixels and each strokes of the entire word.

Step 6: Now after the final segmentation find out the final strokes of the words and all the strokes need to be added and converted to offline file for compare with original word file. By this approach I have done segmentation on all the words covering all the vowel and consonant modifiers and also covering all the alphabets in Bengali language.

III. FEATURE GENERATION

Any online feature is very much sensitive to writing stroke sequence and size variation. total of 233 features (90+15+128) are used for recognition.

The features used are

1. Structural features (15)
2. Point based feature (90)
3. Directional feature (128)

The processed character is transformed into a sequence \( t = [t_1, \ldots, t_N, b_{N+1}, \ldots, b_{N+15}, b_{N+15+1}, \ldots, b_{N+15+128}] \) of feature vectors \( t_i = (t_{i1}, t_{i2}, t_{i3})^T \) (Where \( I <= N \)). We calculated the following features:

3.1. Structural features:

Gradient (\( t_{N+1} \)):

\[
t_{N+1} = \sum_{i=0}^{N} x_i y_i - \sum_{i=0}^{N} x_i \sum_{j=0}^{N} y_j n \sum_{i=0}^{N} x_i^2 - \sum_{i=0}^{N} y_i^2
\]

Length by Width ratio (\( t_{N+3} \)):

\[
t_{N+3} = (\max (x_i) - \min(x_i)) / (\max(y_i) - \min(y_i)) \quad i = 0, 1, \ldots, N
\]

Standard Deviation (\( t_{N+4} \)):

\[
\text{The standard deviation measures the spread of the data about the mean value. It is useful in comparing sets of data which may have the same mean but a different range. Here the deviation of each co-ordinate is calculated with respect to its mean value.}
\]

Normalized Start Co-ordinates and End Co-ordinates (\( t_{N+4} \)):

In this feature only the first and last co-ordinates in the strokes of a character considered. Taking the first and last co-ordinates normalized them and stored them as feature.

Crossing of the lines: Here the co-ordinate position of the crossing of the stroke is stored with itself as shown in figure 2. In this system only first two crossing are considered.

Fig. 1: Length and Width of a character

Fig. 2: Crossing points of a stroke.
3.2. Point Based Feature

The strokes are first normalized into 30 points. The normalization is done in two stages. First the points are re-sampled to fixed number points and then they are converted from equal time sample to equal distant points. The processed character is transformed into a sequence \( t = [t_1, \ldots, t_N] \) of feature vectors \( t_i = (t_{i1}, t_{i2}, t_{i3})^T \) \[4\]. The following features were calculated:

\[ t_{i1} = \frac{x_i - \mu_x}{\sigma_x}, \quad t_{i2} = \frac{y_i - \mu_y}{\sigma_y} \]

where \( \mu_x = \frac{1}{n} \sum_{i=1}^{n} x_i \) and standard deviation, \( \sigma_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu_x)^2} \) of the character’s sample points.

Tangent slope angle \( (t_{i3}) \):

\[ t_{i3} = \text{arg} \left( (x_{i+1} - x_{i-1}) + j(y_{i+1} - y_{i-1}) \right) \]

with \( j^2 = -1 \) and “arg” the phase of the complex number above, is an approximation of the tangent slope angle at point \( i \).

Thus finally, a feature vector sequence is defined as \( t = [t_1, \ldots, t_{N+1}, \ldots, t_{N+15}] \), each vector of it as \( t_i = (t_{i1}, t_{i2}, t_{i3})^T \) is obtained. Here the number of points in which the character is normalized is (N) 30. So a total of 105 (30 X 3 [3 for each point] + 15 [15 local features based on Stroke]) features are used.

3.3 Directional Feature

Given a stroke point \( P_j \), its direction vector \( \vec{V}_j \) is defined as follows \[7\]:

\[
\vec{V}_j = \begin{cases} 
\frac{P_{j+1} - P_j}{||P_{j+1} - P_j||} & \text{if } P_j \text{ is a start point} \\
\frac{P_j - P_{j-1}}{||P_j - P_{j-1}||} & \text{if } P_j \text{ is a non-end point} \\
\frac{P_{j+1} - P_j}{||P_{j+1} - P_j||} & \text{if } P_j \text{ is an end point}
\end{cases}
\]

For a non-end point \( P_j \), if its two neighbors \( P_{j-1} \) and \( P_{j+1} \) are in the same position, the point \( P_j \) is ignored and no directional features are extracted at this point. Given \( \vec{V}_j \), its normalized version, \( \vec{V}_j/||\vec{V}_j|| \), can be projected onto two of the 8 directional axes.

**Figure 4**: 8 Directional features of handwritten data

one is from the direction set of \( \{D_1, D_3, D_5, D_7\} \) and denoted as \( d_{1j} \), and the other is from the set of \( \{D_2, D_4, D_6, D_8\} \) and denoted as \( d_{2j} \). If we define the coordinate system for the original online point \( P_j = (x_j, y_j) \) as follows: the x-axis is from left to right, and the y-axis is from top down; then \( d_{1j} \) and \( d_{2j} \) for a non-end point \( P_j \) can be identified as follows:

\[
d_{1j} = \begin{cases} 
D_1 & \text{if } x_{j-1} \leq x_{j+1} \& |y_{j+1} - y_{j-1}| \leq |x_{j+1} - x_{j-1}| \\
D_2 & \text{if } x_{j-1} > x_{j+1} \& |y_{j+1} - y_{j-1}| \leq |x_{j+1} - x_{j-1}| \\
D_3 & \text{if } y_{j-1} \leq y_{j+1} \& |y_{j+1} - y_{j-1}| > |x_{j+1} - x_{j-1}| \\
D_4 & \text{if } y_{j-1} > y_{j+1} \& |y_{j+1} - y_{j-1}| > |x_{j+1} - x_{j-1}|
\end{cases}
\]

\[
d_{2j} = \begin{cases} 
D_5 & \text{if } x_{j-1} \leq x_{j+1} \& |y_{j+1} - y_{j-1}| \leq |x_{j+1} - x_{j-1}| \\
D_6 & \text{if } x_{j-1} > x_{j+1} \& |y_{j+1} - y_{j-1}| \leq |x_{j+1} - x_{j-1}| \\
D_7 & \text{if } y_{j-1} \leq y_{j+1} \& |y_{j+1} - y_{j-1}| > |x_{j+1} - x_{j-1}| \\
D_8 & \text{if } y_{j-1} > y_{j+1} \& |y_{j+1} - y_{j-1}| > |x_{j+1} - x_{j-1}|
\end{cases}
\]
In feature extraction part directional features are extracted by dividing total charter area by 16 equal rectangle and then calculate how many number in which direction.

Thus finally, a feature vector sequence is defined as \( t = [t_1, t_2, t_3, ..., t_N, t_{N+1}, t_{N+2}, ..., t_{N+15}, t_{N+16}] \), each vector of it as \( t_i = (t_{i1}, t_{i2}, t_{i3})^T \) is obtained. Here the number of points in which the character is normalized is \( (N) 30 \). So a total of 233 \( (30 \times 3 + 15 \times 8 + 16 \times 8) \) features are used in our experiment.

IV. RECOGNITION

For the recognition of bangla handwritten words, firstly by closely looking the shapes and characteristics of basic bangla strokes, that is if the stroke alone can describe any valid bangla character, can describe valid character or also may be a part of any valid character, can be a modifier character, or can be part of a character or a special character, depending on that, basic strokes are categorized or grouped into following five classes.

- **Special_character_class**: only matra (‘’’) is within this class.
- **Valid_character_class**: strokes that can represent a valid bangla character such as ‘ক’.
- **May_be_valid_character_class**: strokes that can represent a valid character and also can be a part of any valid character such as ‘ব’, ‘য’.
- **Modifier_character_class**: ‘.’ , ‘’ etc are the examples of modifier character class.
- **Not_character_class**: strokes that are the part of any character such as ‘।’.

After scanning, digitization, preprocessing, segmentation to single character or segments related to character, and feature extraction, strokes goes to two-phase neural network for recognition purpose, where, In the first phase of neural network, constituent strokes of the input word are identified in terms of its corresponding correct class category and in the second phase the actual character id of the corresponding strokes are determined.

V. RESULT AND DISCUSSION

Statistical result analysis of the first phase of neural network where the constituent strokes of the given bangla word are categorized in terms of their corresponding class is shown in Table 1.

<table>
<thead>
<tr>
<th>Classification</th>
<th>Training set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success rate</td>
<td>80.4</td>
<td>78.6</td>
</tr>
<tr>
<td>Confidence</td>
<td>84.2</td>
<td>82.7</td>
</tr>
<tr>
<td>Misclassification</td>
<td>2498</td>
<td>1360</td>
</tr>
<tr>
<td>Rejection rate</td>
<td>15</td>
<td>16.4</td>
</tr>
<tr>
<td>Rejection</td>
<td>747</td>
<td>409</td>
</tr>
<tr>
<td>Rejection rate</td>
<td>4.5</td>
<td>4.9</td>
</tr>
</tbody>
</table>

Now the Statistical analysis of result after testing 200 bangla words is shown below as Table 3:

<table>
<thead>
<tr>
<th>Words (no. of instance)</th>
<th>Correctly recognized</th>
<th>One stroke error</th>
<th>Two stroke error</th>
</tr>
</thead>
<tbody>
<tr>
<td>ওল (50)</td>
<td>27</td>
<td>19</td>
<td>4</td>
</tr>
<tr>
<td>ফল (50)</td>
<td>19</td>
<td>23</td>
<td>8</td>
</tr>
<tr>
<td>জল (50)</td>
<td>18</td>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td>ঢাক (50)</td>
<td>17</td>
<td>13</td>
<td>20</td>
</tr>
</tbody>
</table>

Some problems occur due to improper segmentation (Over or Under-Segmentation), in some cases segmentation is proper i.e. correct basic strokes are generated from segmentation but recognizer can’t recognize properly because in the second phase it fails to recognize the correct character id from the correct class and it mainly happens in modifier and not character class category. Valid character, may be valid character and special class category strokes are recognized well. Few misclassifications between strokes are noticed between ‘ফ’ and ‘থ’, ‘ন’ and ‘ল’ etc. The features for modifier and not character class category is also not satisfactory. So there is a scope for further study on modifier and not character class category type strokes and its feature generation to increase recognition rate.

VI. ACKNOWLEDGEMENT

The author likes to acknowledge the partial financial support of the AICTE for this project under RPS (a). The author is also thankful to Mr.
Rajib Ghosh and Mr. Anupam Ghosh for their help in the work.

VII. REFERENCE


<table>
<thead>
<tr>
<th>Special char set</th>
<th>Valid char set</th>
<th>May be valid char set</th>
<th>Modifier char set</th>
<th>Not char set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classifier</td>
<td>3285</td>
<td>906</td>
<td>4044</td>
<td>1917</td>
</tr>
<tr>
<td>Success rate</td>
<td>1</td>
<td>1</td>
<td>94.8</td>
<td>90</td>
</tr>
<tr>
<td>Confidence</td>
<td>1</td>
<td>1</td>
<td>96.9</td>
<td>99.9</td>
</tr>
<tr>
<td>Misclassification</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>62</td>
</tr>
<tr>
<td>Classification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rate</td>
<td>0</td>
<td>0</td>
<td>2.9</td>
<td>0.1</td>
</tr>
<tr>
<td>Rejection</td>
<td>0</td>
<td>0</td>
<td>221</td>
<td>151</td>
</tr>
<tr>
<td>Rejection rate</td>
<td>0</td>
<td>0</td>
<td>5.1</td>
<td>7.0</td>
</tr>
</tbody>
</table>