Recognition of Unconstrained Malayalam Handwritten Numeral

U. Pal, S. Kundu, Y. Ali, H. Islam and N. Tripathy
C VPR Unit, Indian Statistical Institute, Kolkata-108, India
Email: umapada@isical.ac.in

Abstract

Main problem in handwritten recognition is the huge variability and distortion of patterns. To take care of writing variability of different individuals, a recognition scheme for isolated off-line unconstrained Malayalam handwritten numeral is proposed here. Main features used in the scheme are based on water-reservoir concept. A reservoir is a metaphor to illustrate the cavity region of the numeral where water can store if water is poured from a side of the numeral. The important reservoir based features used in the scheme are: (i) number of reservoirs (ii) positions of reservoirs with respect to bounding box of the touching pattern (iii) height and width of the reservoirs (iv) water flow direction, etc. Topological and structural features are also used for the recognition along with water reservoir concept based features. Close loop features (number of close loop, position of loops with respect to the bounding box of the component) are the main topological features used here. In the structural feature we consider the morphological pattern of the numeral. At present we obtained 96.34% overall recognition accuracy.

1. Introduction

There are many pieces of work on the recognition of unconstrained handwritten numerals of Roman, Chinese and Arabic script [1]. Although there are twelve different scripts in India, only a few research papers have been published on handwritten numeral recognition and these papers are mainly on Devnagari, Bangla and Oriya scripts [2,3,4,5]. Various approaches have been proposed by the researchers towards the recognition of non-Indian numerals [1]. One of the widely used approaches is based on neural network [4]. Some researches used structural approach, where each pattern class is defined by structural description and the recognition is performed according to structural similarities [1]. Statistical approach is also applied to numeral recognition [6]. Among others, Support vector machines [7], Fourier and Wavelet description [8], Fuzzy rules [9], tolerant rough set [10], free automatic scheme for unconstrained off-line are reported in the literatures.

In this paper, we propose a normalization and thinning Malayalam isolated handwritten numeral recognition. Malayalam is a popular Indian script and language. The propose scheme is mainly based on features obtained from water *reservoir* concept as well as topological and structural features of the numerals. Reservoir based features like number of reservoirs, their size and positions, water flow direction, topological feature like number of loops, position of loops, the ratio of reservoir/loop height to the numeral height, profile based features, features based on jump discontinuity etc. are main features used in the recognition scheme along with other features. To the best of our knowledge, this is the first work on handwritten Malayalam numeral recognition.

2. Malayalam numerals and pre-processing

In India there are twelve scripts and nineteen languages. Malayalam is one of the popular language and script of India. Like other scripts Malayalam has also 10 numerals. To get an idea of Malayalam numerals and their variability six sets of handwritten numerals are shown in Fig.1.

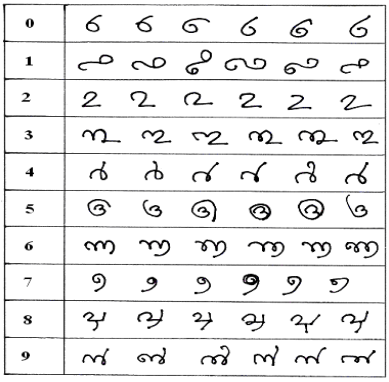

Fig.1: Example of Malayalam handwritten numerals.

The images are digitized by a HP scanner at 300 DPI. The digitized images are in gray tone and we have used a histogram based thresholding approach to convert them into two-tone images [3]. For a document 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 the label 1 represents the object and 0 represents the background.

The digitized image may contain spurious noise pixels and irregularities on the boundary of the numerals, leading to undesired effects on the system. For removing these noise pixels we have used a simple and efficient method described in [3].

3. Feature detection and recognition

3.1 Feature detection

Recognition result of a system mainly depends on the robustness of the features to be used in the system. To take care of variability involved in the handwriting, the features are chosen with the consideration of (a) Independence of various writing styles of different individuals (b) Simplicity of detection (c) Independence of size (d) Independence of stroke-width of writing element

The stroke width (R_L) is the length of most frequently occurring black run of a component. In other words, R_L is the statistical mode of the black run lengths of the component. R_L is an important feature independent of the pen used for the writing. As a result, our system can recognize numerals written in different types of pen. In this work, we use R_L as a threshold in computation of many features. The value of R_L is calculated as follows. The component is scanned both horizontally and vertically and the lengths of different black runs are noted. If from a component we get n different runs of lengths r_1, r_2,..r_n with frequencies f_1, f_2 ...f_n, respectively, then the value of $R_L = r_i$ where $f_i = \max(f_j)$, j = 1...n.

Because of page limitation of this conference only a few principal features used in the recognition scheme are described below:

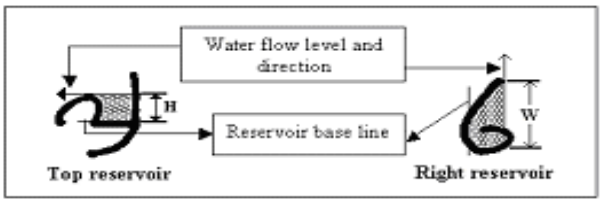

Fig.2: Illustration of different features obtained from water reservoir principle. Here shaded area denotes the reservoir. 'H' denotes the height of top reservoir & 'W' denotes the width of right reservoir.

(i) Feature based on water reservoir principle:

The water reservoir principle is as follows. If water is poured from a side of a numeral, the cavity regions of the numerals where water will be stored are considered as reservoirs [11]. For an illustration see Fig.2. Now, we will discuss here some terms on water reservoir that will be used in feature extraction.

Top reservoir: By top reservoir of a component we mean the reservoir obtained when water is poured from top of the component. See the left component of Fig.2 where top reservoir is shown.

Bottom reservoir: By bottom reservoir of a component we mean the reservoir obtained when water is poured from bottom of the component. A bottom reservoir of a component is visualized as a top reservoir when water will be poured from top after rotating the component by 180°.

Right (left) reservoir: If water is poured from right (left) side of a component, the cavity regions of the component where water will be stored are considered as right (left) reservoirs. See the right component of Fig.2 where right reservoir is shown.

Water reservoir area: By area of a reservoir we mean the area of the cavity region where water can be stored if water is poured from a particular side of the component. The number of pixels inside a reservoir is computed and this number is considered as the area of the reservoir.

Water flow level: The level from which water overflows from a reservoir is called as water flow level of the reservoir (see Fig.2, where water flow level and direction is shown).

Reservoir base-line and the base-point: A line passing through the deepest point of a reservoir and parallel to water flow level of the reservoir is called as reservoir base-line (see Fig.2). Reservoir boarder points lie on the base-line are noted and the left most point of such boarder points is called as base-point.

Height of a reservoir: By height of a reservoir we mean the depth of water in the reservoir. In other words, height of a reservoir is the normal distance between reservoir base-line and water flow level of the reservoir. (In Fig: 2 it is denoted by 'H').

Width of a reservoir: By width of a reservoir we mean the maximum distance between two boundaries of a reservoir. For top/bottom reservoir it is the distance between rightmost and leftmost boarder point of the reservoir. Width of a right reservoir is marked by 'W' in Fig.2.

All reservoirs obtained in a component are not considered for further processing. Those reservoirs whose heights are greater than a threshold T_1 are considered for future processing. The threshold value T_1 is obtained from the experiment.

Some of the numerals show different behaviour in reservoir based features and these behaviours are used for their recognition purpose. The reservoir based features used in the recognition are number different reservoirs, their positions and height, reservoir width, reservoir area, flow direction etc.

(ii) Loop Feature:

By loop we mean the white region enclosed by black pixels. The loop (hole) features are mainly the number of

loops, their height, their area, and their positions in the character. The maximum of height and width of a loop is computed. If any of height or width of a loop is less than the stroke width (R_L) the corresponding loop is ignored. By height of a loop we mean the distance between the topmost and bottommost rows of the loop. The area of a loop is the number of white pixels within the loop. Position of a loop is noted with reference to the bounding box position of the component. For illustration see Fig.3.

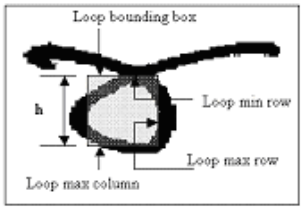

Fig.3: Loop features of a component are shown ('h' is loop height)

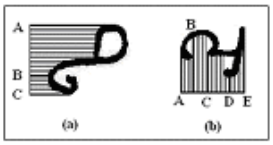

Fig.4: (a) Left and (b) bottom profile features of a component

(iii) Profile based feature:

Suppose each numeral is located within a rectangular boundary like a frame. The horizontal or vertical distances from any one side of the frame to the numeral edge are a group of parallel lines, which we call the *profile*. Left and bottom profiles of a numeral are shown in Fig.4. If we compute left or bottom profile of the numerals, we can notice some distinct difference among the numerals. For example, some numerals have one transition while some other numerals have two or more transition points. By transition we mean change of the length of the profiles from increasing mode to decreasing mode or vice-versa. In the left profile of the numeral shown in Fig.4(a), the profiles from A to B are in decreasing mode (decreases or remain constant), and from B to C the profiles are in increasing mode. Thus, the left profile of this numeral has two transition points where as the bottom profile of the numeral shown in Fig:4(b) has five transitions. The profile-based features used in the recognition scheme are number of transitions, and distance of the profiles from the sides of the numeral boundary. For all four sides (i.e. left, right, top and bottom) the profile features are extracted.

(iv) Feature based on jump discontinuity:

In this feature we check jump discontinuity between two consecutive boarder pixels of a numeral from a particular side of the numeral. We use the presence of jump discontinuity as a feature for identification. To compute this feature from left (The feature also can be computed from other three sides. i.e. from bottom, from top or from right side), starting from top each horizontal row (P_i) of a numeral is scanned (left-to-right) until it reaches a black pixel (Q_i) (Similar with the profile based feature). Thus for a numeral of row-size (height) M, we get M such P_is. To check the presence of jump discontinuity, we compute the difference of the scanned value of two consecutive columns. In Fig.5 the two consecutive pixels (having the row value P_i and P_{i+1}) are denoted by Q_1 and Q_2, and the distance between Q_1 and Q_2 is denoted by D_1. Now if D_1 is greater than 2.5 times of the stock-width (R_L) (described at the loop feature) of the numeral then it is treated as a jump and the number of the jump value is increased by one otherwise it is discarded and continue processing for the next two consecutive rows (P_{i+1} and P_{i+2}). The positions where jump discontinuity occur are noted. Also, number of such jump continuity is noted. Position and number of jump discontinuity are used as the features.

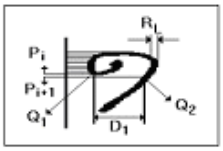

Fig.5: Example of Jump discontinuity feature

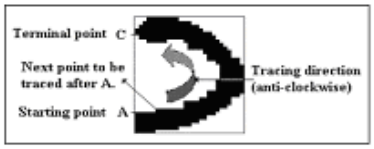

Fig.6: Illustration of tracing feature

(v) Contour tracing feature:

Contour tracing feature is an useful feature to observe the behavior of the contour of the numeral. Computation of this feature is done as follows. Starting from a particular point on the contour of the numeral, the contour pixel tracing is made in clockwise/anti-clockwise direction until it reaches the terminal point of tracing. The direction of tracing (clockwise or anti-clockwise) and terminal points are decided based on the numeral on which this feature will be applied for recognition. During contour tracing we calculate the distance of each traced pixel to a side of the character's bounding box. The side to be used for the distance computation also depends on the numeral. Noting the number of transition of the distance sequence we have recognized the numerals. By transition we mean change of values from increasing mode to decreasing mode or vice-versa.

3.2 Numeral Recognition

A binary tree classifier is employed for the recognition of Malayalam numerals. At first, using some principal features (e.g. reservoir based features like number of reservoirs, their size and positions, water flow direction, topological feature like number of loops, their positions etc., the ratio of reservoir/loop height to the numeral height, profile based features, features based on jump discontinuity, etc.) we generate a binary tree where a leaf node of the tree may contain up to two numerals. Next, we use more specific features to identify numerals of different leaf nodes of the tree. A part of the classification tree is shown in Fig.7. In the tree, only one feature is tested at each non-terminal node. The features used for

decision in a node are mostly binary e.g. presence/absence of the feature. (Some of the features used for the decision tree are marked by $\mathbf{P_1}$, $\mathbf{P_2}$, $\mathbf{P_3}$ etc. in Fig.7 and the meaning of these features are given at the side of the tree.). We have chosen the features at non-terminal nodes so that the classification tree can be represented in an optimum way. A tree will be optimum if every node of the tree can divide the elements into two groups of equal number of elements. We have noticed that out of ten numerals of Malayalam script five numerals have loops like structure. Keeping this in mind, we have used the loop feature ($\mathbf{P_1}$) at the beginning of the tree because in general this feature can divide the numeral set into two groups of equal size. One of these two subsets contains those numerals having loops(S_1) and the other without loops(S_2). Other features of the tree have been used in similar ways.

Because of page limitation, here we cannot provide detail discussion about the recognition methods of all the numerals of the leaf nodes. Here, we shall discuss about the identification techniques of some of the leaf nodes only. Because of the recognition scheme used here, the numerals of a leaf node are similar shaped. Some of the numeral pair obtained in the leaf nodes of the recognition tree are shown in Fig.8.

For recognition of the two similar shaped numerals shown in Fig.8(a), we note the number of black pixels in a specific region of the numerals. This specific region is shown by gray shade in Fig.8(a). Depending on the reservoir feature this specific region is detected as follows. From Fig.8(a) it can be noted that both the numerals have two bottom reservoirs and we choose the rightmost reservoir for the detection of the specific region. From the base-point (base-pints is defined in Section 3) of the rightmost reservoir we go upwards until we get the first white pixel. Let this white pixel is 'X'. See Fig.8(a) where the point 'X' is shown. If we draw a co-ordinate system, considering the point 'X' as origin, then the first quadrant is the specific region used for recognition. This first quadrant is marked by gray shade in Fig.8 (a). From the figure, it can be noted that there is no black part of the component in this specific region for the numeral shown in left side of Fig.8 (a), whereas for the numeral shown in right side of Fig.8 (a) there is some black part of the component in this specific region. We compute the number of black pixels in the shaded region. If for a numeral, the number of black pixels in the shaded region is more than $2*R_L$ (R_L is the stroke width discussed earlier) then we identify that numeral as nine. Else, it is numeral three.

Similar technique is used to identify the numerals two and four (shown in Fig.8 (b)).

For identification of the numerals three and six, shown in Fig.8(c), we use reservoir based information. It can be noted that both the numerals have two bottom reservoirs and we choose the rightmost reservoir for the purpose. The boarder pixels of the chosen reservoir are computed and the rightmost boarder pixel is noted. The rightmost boarder pixel is marked by 'Y' in Fig.8(c). A vertical line passing thorough 'Y' is drawn. Also, the right edge of the image frame is noted. The normal distance between the line passing through 'Y' and the right edge of the image frame is noted. Let this distance is 'd' (See Fig.8(c)). If for a numeral, this distance (d) is greater than 3 times of the stroke-width (R_L) then the numeral is three. Else, it is numeral six.

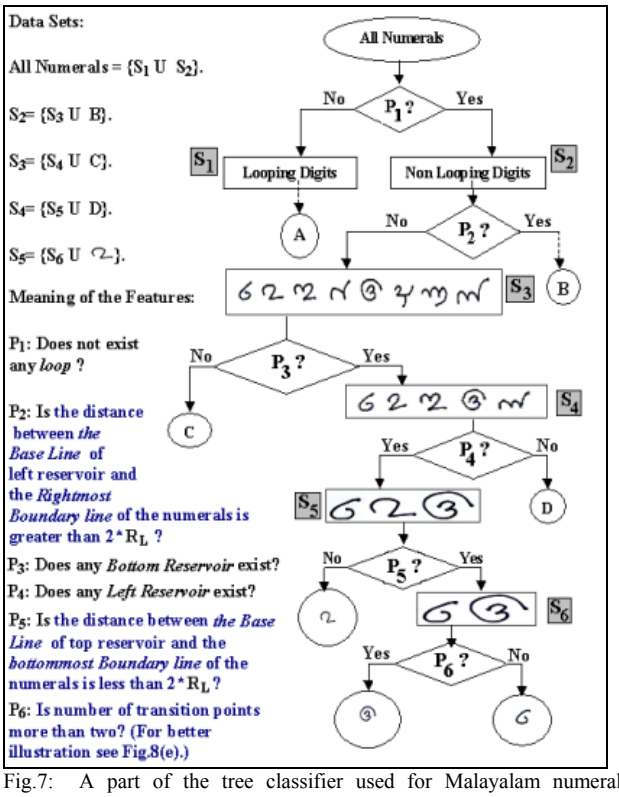

Fig.7: A part of the tree classifier used for Malayalam numerals recognition. Different features used in the nodes of the tree are described in the left side of the tree. Here S_1, S_2, S_3 …etc. are different numeral sets defined in the left side of the tree.

Sometimes the numerals three and two fall in the same leaf node of the tree. For the distinction of these numerals we simply use the number of bottom reservoir. For example, see Fig.8 (d). Here the numeral three has two bottom reservoirs whereas the numeral two has only one.

For distinction of two numerals zero and five shown in Fig.8 (e), we use the features obtain from left reservoir and anti-clockwise tracing. These two numerals appear in the same leaf node of the tree because both the numerals have the following properties: (a) no loop (b) both have top and left reservoirs (c) the distance between the base line of left reservoir and the rightmost boundary line of the numerals is less than $2*R_L$ (d) the distance between the base line of top reservoir and the bottommost boundary line of the numerals is less than $2*R_L$. For

their identification, at first, we detect the position of the left reservoir of the numeral and we use the portion of the numeral where left reservoir appears. A zoomed version of the portion of the numerals where left reservoir appears is also shown in Fig.8 (e). Starting from the lowermost row of the left reservoir we anti-clockwise trace the reservoir boarder pixels until the uppermost row of the left reservoir is reached. During tracing the column value of each traced pixel is noted. At last we compute the number of transition points depending on these column values. Based on the number of transition points these two characters are identified. For the numeral zero, we notice that the column value first increases from 'A' to 'B' and then decreases from 'B' to 'C'. So number the transitions point of this numeral in the traced position is two. For the numeral five, we get four transition points since the column value first increases from 'A' to 'B' and then decreases from 'B' to 'C' again increases from 'C' to 'D' and then decreases from 'D' to 'E'. For illustrations, see Fig.8 (e).

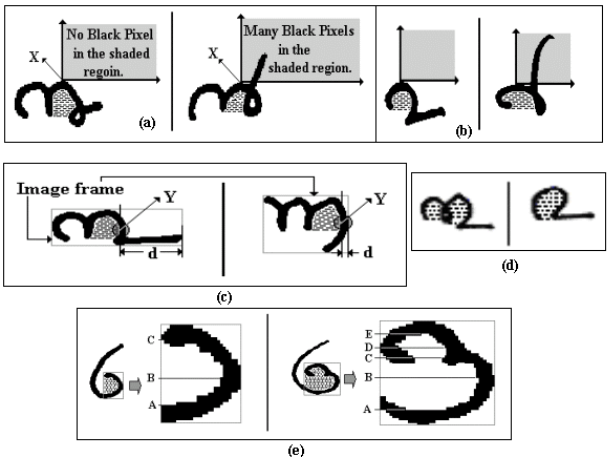

Fig.8: Similar shaped numerals and their recognition techniques are illustrated. Examples of five pair of similar shaped numerals are shown here. (a) numerals three and nine (b) numerals two and four (c) numerals three and six (d) numerals three and two (e) numeral zero and five; (left and right side, respectively) (A zoomed vision of a portion of the numerals of Fig.8 (e) is shown in the right side of the respective numerals).

4. Results and discussion

4.1 Data set

The data used for the experiment were collected from different individuals of various professions like students and teachers, bank and post office employees, businessmen etc. There was no restriction or guidance in the writing style as a result we noted that the data sets contain varieties of writing styles. We collected 180 different samples for each numeral class for the experiment. As a result total number of data was 1800. Out of 180 samples of each class we use 46 samples to train the recognition tree classifier. Rest 134 samples of each class have been used for testing the classifier.

4.2 Results

From the experiment we noted that the overall accuracy of the proposed recognition scheme was about 96.34%. Detail distributions of the results obtained from the above data set is given in Table 1. From the experiment we noted that numeral three has the highest recognition rate (98.5%). This is because of its distinct shape from other numerals. We also noted that second highest recognition rate obtained from the numeral seven (97.76%). From the experiment we noted that the maximum error (6.71%) obtained from numeral nine and most of the times it miss-recognized as numerals six. Sometimes it confuses with numeral three also. This is because of their similar behaviour in water reservoir based features.

Table 1. Distributions of the recognition results obtained from the proposed scheme.

Numeral [Data size 1340 (134*10)]	Number of numerals recognized as →									
	1	2	3	4	5	6	7	8	9	0
One	127			3	2		2			
Two		129		1				4		
Three			132			1			1	
Four	1	2		130				1		
Five	1				128	1	3		1	
Six			2			130			2	
Seven	2	1					131			
Eight		2	1			1		129	1	
Nine			3			5		1	125	
Zero		1		2	1					130

Due to handwriting styles of different individuals sometimes two different numerals in Malayalam may look in similar shape fashion. As a result some miss-recognition occurs. For example see, Fig.9 where numeral two(left side) and seven(right side) are shown. Due to addition of a small stroke by the writer in the numeral seven these two numerals look similar. This additional stroke is marked by dotted box in the numeral seven of Fig.9.

From the experiment we noted because of shapes of the numerals and the features used for the recognition sometimes a numeral may fall in both the subset of a node. As a result, accuracy of this classifier increases although the depth of the classification tree increases. The depth of our proposed classification tree is 8. However, the depth in an ideal classification tree will be 4 for 10 numerals.

Main advantage of the proposed method is its flexibility. Because of the feature used in the recognition process, the proposed method can handle varieties of handwriting. For example if an individual writes the Malayalam numeral three in any of the fashions(shown in Fig.10 (a)) then our method is able to identify it.

Similarly, if someone writes the numeral four in any of the form shown in Fig.10 (b) our method is also able to identify it.

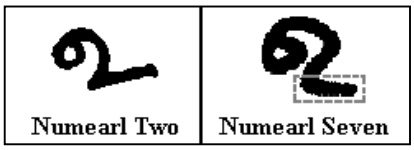

Fig.9: Confusing numerals where miss-recognition occurs

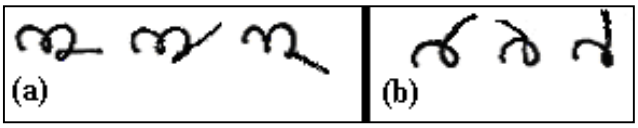

Fig.10. Example of three different styles of Malayalam numerals (a) Numeral three (b) Numeral four.

In some cases due to use of poor quality ink or paper, there may appear some discontinuity in the binary image. If the area of this discontinuous region is small our smoothing techniques can take care of this discontinuity. But if the area is big then miss-recognition occurs. Some errors also occur due to double handwriting. By double handwriting we mean repetition of the pen once more on the numeral.

The proposed method does not depend on the size of the numerals. This is one of the most important advantages of the method. Since there is no work on the recognition of unconstrained off-line Malayalam handwritten numeral, so we cannot compare our results.

4.3 Drawbacks of the proposed system

The main drawback of the proposed method is that it will fail sometime if there is a big break (discontinuity) on the contour portion used as the boundary of the reservoir. In that case water cannot be filled up properly to get reservoir (hole) and hence miss-recognition will occur. But many other methods (for example those based on contour following) will also fail for this type of situation. We note that such cases are very rare (1.1%) We use smearing technique to remove some of these situations where size of the break point region is small.

The data set used for the experiment is not large. Currently, we are collecting more data and we have a plan to collect about 2000 samples for each numeral class. After development of the database we plan to compare the results of the proposed system with that of the neural network based technique. At present we did not implement any rejection options. In future we also plan to add a rejection module.

5. Conclusion

This paper deals with a normalization and thinning free automatic scheme for the recognition of unconstrained off-line Malayalam isolated handwritten numerals. To take care of variability involved in the writing style of different individuals, a recognition scheme based on water reservoir concept is proposed here. The water reservoir based features described here can be used in other work of pattern recognition. To the best of our knowledge this is the first report on unconstrained off-line Malayalam handwritten numerals. At present, the recognition scheme considers only isolated numeral recognition. In future we plan to modify it so that it can handle handwritten touching numeral strings.

References:

[1] R. Plamondon and S. N. Srihari, “On-line and off-line handwritten recognition: A comprehensive survey”, IEEE PAMI, vol. 22, pp 62-84, 2000.

[2] K. Dutta and S. Chaudhuri, “Bengali alpha-numeric numeral recognition using curvature features”, Pattern Recognition, vol. 26, pp 1757-1770, 1993.

[3] U. Pal and B. B. Chaudhuri, “Automatic Recognition of Unconstrained Off-line Bangla Hand-written Numerals”, Proc. Advances in Multimodal Interfaces, Springer Verlag Lecture Notes on Computer Science (LNCS-1948), pp 371-378, 2000.

[4] Bhattacharya et al., “Self-organizing neural network-based system for recognition of hand printed Bangla numerals”, Proc. Annual Convention of Computer Society of India, pp C92-C96, 2001.

[5] N. Tripathy, M. Panda and U. Pal, “A System for Oriya Handwritten Numeral Recognition”, SPIE Proceedings, Vol.-5296, Eds. E. H. Barney Smith, J. Hu and J. Allan, pp. 174-181,2004.

[6] J. Cai and Z. Q. Liu, “Integration of structural and statistical information for unconstrained handwritten numeral recognition”, IEEE PAMI, vol. 21, pp. 263-270, 1999.

[7] H. Byan and S. W. Lee, “A survey on pattern recognition applications of support vector machines”, IJPRAI, 17, 459-486 2003.

[8] P. Wunsch and A. F. Laine, “Wavelet Descriptors for Multi-resolution Recognition of Hand-printed Digits”, Pattern Recognition, vol. 28, pp 1237-1249, 1995.

[9] Z. Chi and H. Yan, “Handwritten numeral recognition using self-organizing maps and fuzzy rules”, Pattern Recognition, vol 28, pp 56-66, 1995.

[10] K. Kim and S. Y. Bang, “A handwritten numeral character classification using tolerant Rough set”, IEEE PAMI, vol. 22, pp 923-937, 2000.

[11] U. Pal, A. Belaid and Ch. Choisy, “Water Reservoir Based Approach for Touching Numeral Segmentation”, In Proc. Sixth ICDAR, pp 892-896, 2001.