Representation and Extraction of Nodal Features of DevNagri Letters

Rajiv Kapoor
Research Scholar
Punjab University, Chandigarh
raj_himani@hotmail.com

Deepak Bagai
Assistant Professor
PEC, Chandigarh
dbagai@yahoo.com

T.S.Kamal
Professor, SLIET,
Longowal, Sangrur, Punjab
tskamal@yahoo.com

Abstract

This paper proposes a method for the recognition of handwritten DevNagri characters. Recognition has been done using an HMM-based approach, using junction points of a character as the main feature. The character has been divided into 3 major Zones (Upper, Middle, and Lower). This paper explains the method for middle zone which depends upon three major features i.e. number of paths, directions of paths, & region of the node. Tolerance of feature points has been proposed as an additional important parameter. Heuristic features taken up from the existing methods have been redefined and refined for this application. The connectivity features which give meaning to the whole word have been redefined, excluding number of Matra endpoints, Matra type (curve/line). Finally every character has been recognized using HMM. The proposed method has been tested for the sample-words taken from 100 persons.

Keywords: - *Character Recognition Software; Feature point; State Transition Probability Matrix; Hidden Markov Method.*

1. Introduction

The process of Character Recognition is to understand hand-written characters from scanned image files and storing in the ASCII format, so that it could be processed in the editor.

1.1 Fundamental Steps in Character Processing

Fundamental steps required to perform character-processing and understanding task have been discussed below.

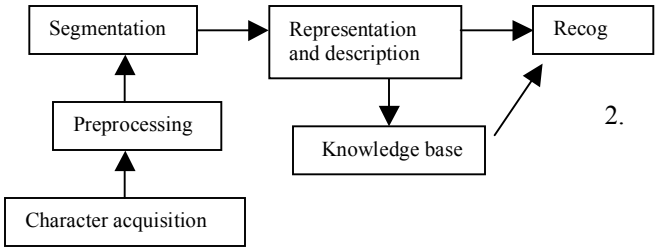

Figure-1 (Sequence of Operation in a Character/Image Recognition Cycle)

1.1.1 Character acquisition

The first step in the process is to acquire hand written characters. Generally we use Scanner to do the above job. Other sensors can be Camera, Video camera etc. Resultant file is stored in memory.

1.1.2 Pre-processing

The key function of preprocessing is to improve the image for other processes. It typically deals with techniques for enhancing contrast, removing noise etc.

1.1.3 Segmentation

Segmentation is generally an extremely important & difficult task but in DevNagri character recognition problem , the probability of finding significant ligatures is very less and hence is out of the scope of the present work. In this paper, we have taken isolated characters.

1.1.4 Representation and Description

Description is also called as feature selection which deals with features-extraction and further results in some quantitative information of interest or features that are basic for differentiating one character from another. The features selected as descriptors should be as insensitive as possible to variation such as change in size, translation, and rotation.

1.1.5 Recognition and Interpretation

Recognition is the process that assigns a label to a character based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized characters. In this paper, isolated characters will be processed to extract the pre-conceived features of the character for recognition purpose. Extracting and defining the features of DevNagri characters is our major concern here.

2. PREVIOUS WORK

H.S.Park and Lee [2] suggested an efficient scheme for off-line recognition of handwritten characters using HMM (Hidden Markov Model). Four types of feature vectors based on the regional projection contour transformation (RPCT) [1] were employed. The recognition

system consists of two phases. For each character, in the training phase, multiple HMMs corresponding to different feature types of RPCT are built. In the classification phase, the results of individual classifiers to produce the final recognition result for an input character are integrated, where each individual HMM classifier produces one score that is the probability of generating the test observation sequence for each character model. Finally, we, the character model that gives the highest probability are chosen.

The proposed features in general, include counts of topological features (crossings, endpoints, holes, etc.) and various mathematical moments. While these ad hoc features have performed well in many tests, but these are not generally applicable to other character sets like DevNagri. A number of deformable models for digit recognition have been applied. Nishida [3] proposed a grammar-like model for applying deformations to structures composed of primitive strokes. This does not work with the DevNagri Character set because of their complex structure. Lam and Suen [4] used a two-stage method for recognition, in which samples have been classified as per their structure using a tree classifier. Samples which can not be satisfactorily assigned to a class in such a manner, are passed to a slower relaxation matching algorithm which uses deformation to match the sample to each template. This method is not much useful because classifying the characters fully will be extremely difficult task and more over the character is attached with Matras. Cheung [5] used model characters with a spline, and assumed that the spline parameters have a multivariate Gaussian distribution. A Bayesian approach is then used to determine the character class. Simard[6] presents a digit recognition system based on an efficient distance measure that is locally invariant to transformations such as translation, rotation,scaling,stroke thickness. Casey[7] gives a method for linear transformation of digit images, based on moment normalization, for removing some skew and orientation variations. Wakahara[8] uses iterated local affine transformation(LAT) operations to deform images to match prototype digit images. This method correctly identified 96.8 percent of characters in a 2400-samples database. Anil K. Jain and Douglas Zongker[9] proposed to deform the template of the image so that it closely matches the other image. Deformation and then template matching may not help in the DevNagri character set because the character set is not understandable in isolation but in association. Nishida [10] suggested an algorithm for On-Line recognition. The algorithm is based on the structural analysis of thin-line pictures and the class description, which is essential in structural approaches. This method works well again for the individual character but not the characters in association.

Hwang and Bang [11] suggested RBF Neural Network Model for pattern recognition task. Wu [12] suggested a verification scheme based on split-and-merge matching mechanism. Each word in the signature is specified with static and dynamic features: a sequence of(x,y) coordinates and a sequence of (x,y) velocities. The velocity feature does not give good results with the increase in the number of samples. Senior [13] suggested the structural features like dots, junctions, endpoints, turning points, loops. Each of these features can be encoded by a single number. This can give good results in printed set of characters but not in the hand written character-set of DevNagri. Sin and Kim [14] proposed a HMM based modeling of letters and Inter-connecting boundaries. HMM based approach results in better recognition rate but because of the structure of the DevNagri character set; it is difficult to distinguish between the inter-connecting boundary and the Matra or Head-Line (which is inherently there in DevNagri Character set). Kim and Givindraju[15] proposed a new Lexicon driven approach for Real Time Word recognition, for Real Time applications. This can be the additional approach to confirm that the right character has been recognized. Basit[16] suggested a feature-recognition-network pattern recognition that learns the patterns .Such approach is valid when the right kind of features have been used for training the network that is being proposed in the paper.

3. PROPOSED METHOD

3.1 Features of DevNagri Script

Based on various approaches discussed above and their utility and efficiency, a new feature point method has been proposed to define the DevNagri character in it's totality. A feature is a point of human interest in an image, a place where something happens. It could be an intersection between two lines, or it could be a corner, or it could be just a dot surrounded by space. These relationships are used for character

identification, and hence the feature points are exploited for this task of character recognition. Here we are using nodes as feature points as shown below.

Fig-2(Character Kaa of DevNagri Script- depicting various Nodes/Junctures)

3.1.1 First Feature -Node

Nodal junctures are important in getting the shape of the DevNagri character as shown in Fig-2. Formation of node has been defined as given below:

3.1.1.1 Process of forming nodes

A node is formed :->

- At the starting of character (Upper left corner).
- Whenever there is discontinuity in the graph (uniformity) ie if an expected line could be written in two parts because of the pen problem.
- Path breaks into two or more directions.
- When path ends.

3.1.2 Directions of the nodal juncture (Second Feature -Direction)

We are considering that each node is 8 connected as follows:

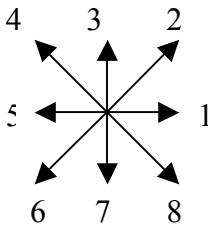

Fig-3(Directions at a Node)

This feature is similar to chain code feature. While tracing curve for getting the nodes, direction of the segment is related with the sequence of tracing the nodes. For example, if we have traced node 2 after Node 1 then the direction of segment will be the directions given in Figure-3. If the exact direction as per fig-3 is not coming while tracing then the direction out of fig-3 which is nearly equal to the segment direction will be chosen for further processing.

3.1.3 Region where the node lies (Third Feature -Region)

First Feature Point "Node" will lie in one of the sixteen regions and hence this feature. Region will give a good idea about the structure of the character.

Hence, a grid of 16 regions as follows: -

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Fig-4(Node Region)

3.2 Numbering the Nodes

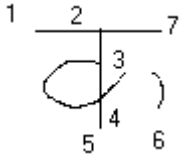

Fig-5(DevNagri Character Kaa with Node Numbering)

Here we number the nodes using Pre-order traversal (node, left, right) of a tree is done and as along the path of the character.

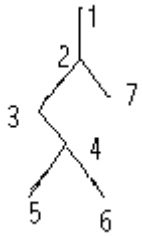

Fig-6 (Pre-Order Tree Traversal)

Following is the method of numbering the nodes:

- Start numbering from the upper left corner of the grid or cell.
- Number the starting point (node) as 1.
- If there are more than two successive nodes then give next number to the current node and move to the left node and put other node in the stack.
- Repeat the above process till the end node found and put the right nodes on the stack.
- At last Pop Up the stack and number the remaining nodes present in the stack.

3.3 Procedure for writing Descriptors (features) for character recognition

Here we will discuss the method to write the descriptors of a character to be recognized. These descriptors will be matched with the dictionary references. For all this we will write a series that will give the idea about the structure of the character but initially fit the character in the well-defined region (cell or grid).

S → for starting node

E → for end node

Descriptor (node) is represented three dimensionally. Three parts represent description of each node: →(p, Φ, and r)

p → This part consists of number of paths in which character proceeds.

S and E are written for starting and ending node.

Φ → This part gives the direction of paths from the given node.

r → This part gives the region in which Node is present.

Let there be a model which expresses the specified features of the DevNagri Characters completely so that the characteristics of a DevNagri character could be easily identified by below mentioned Nodal Model.

Sample Node can be expressed as given below:

Node (no. of paths, directions of paths, region of node).

Or

Node (p, Φ, r).

Where 'p' is fixed, 'Φ' will vary hence will contribute to the STP matrix and 'r' can vary and hence will contribute to STP Matrix. STP Matrix is State Transition Probability Matrix which is useful in understanding the DevNagri Character using the Hidden Markov Method as a tool. The states of the HMM and further processing have been deliberately skipped. Hence the suggested Nodal-model has been expressed as below:

$$\sum_{i=1}^{3} \text{Node}(p, \Phi_i, r_i)$$

The value of 'i' will depend upon the number of states. So in this case we have taken 3 to accommodate nine state transitions. This matrix completely depicts all the possibilities, in which the character can behave. Detailed tutorial on this has been already given by Rabiner[17].

Tolerances help in the formation of state transition probability matrix. For example, while tracing the curve, (and moving from node x to node y) there is always probability that a particular node may lie in two regions still keeping the meaning of the character same.

Example:

(2, 5/6&7, 6/7)

This node has 2 forward paths, one path in direction 5 and other is in direction 7, and it is in region 6. The values after / indicates the ***tolerance***. I.e. in above example path in direction 5 may have direction 6; similarly node may be in region 7 instead of 6.

3.4 Traversing through the image

Here, top line or first row of the array is considered to be the first line and number of lines increase downward in the array; gap corresponds to a column of array. Node found in the character image is stored in the structure called node having integer members line and gap for storing position of node, branches for storing number of further nodes and region for storing the value of region in which node lies in an image. An array of self referential pointers to store addresses of further nodes found in the image and an integer array for storing the directions of corresponding nodes.

3.4.1 Finding next point

After starting from beginning of top line which is Start point of image, for finding next pixel of image (i.e. 0) box1 is made around the present pixel.Box1 is a box consisting of eight immediate neighbors of present pixel. Every pixel in the box is checked whether it lies on the image or not, if it lies on the image (i.e. 0) it is made present position and previous pixel is changed in color (i.e. 0 to 2) so that this path is not detected again.

3.4.2 Check node condition & finding node center

At every pixel box2 is made i.e. leaving the first eight immediate neighbors of a pixel, next sixteen neighbors of pixel are considered and checked if there are pixels present in the box with the background color between them if yes, than node is present at that point. Number of branches at that point is found by number of continuous set of pixels present in box2. This procedure is repeated the next five pixels found by next point method and for each pixel number of branches is found. Node center is the pixel having largest number of branches and its position is stored in the node structure and its address is given to the pointer called next in the previous node if present(start pointer is used for pointing to first node).

3.4.3 Finding paths to traverse around the node

For finding paths around the node box5 is made similar to box1 and box2 containing 40 points

around the node at a difference of four lines or four columns from it. All the continuous sets of pixels of image are found and each pixel is used as the starting point of next branch for traversal. All the paths are traversed in depth first order starting from left most branches and if further nodes are found then this procedure is repeated recursively.

3.4.4 Checking condition for repeated node

When node is found it should be checked if it has been found earlier. For checking the repeatability condition lines and gap parameters of given node is checked with corresponding lines and gap parameters of all the deducted nodes if, the difference between these parameters of given node and any other is less than equal to 8 i.e. node lies in box8 around any other node than it is a repeated node.

3.4.5 Starting and End point node

First node is made at the pixel at the beginning of the top line called as starting node and end node is made at the point no branch at some pixel in the box2; this is the returning condition for our recursive function traversing and finding nodes in the character image.

3.4.6 Finding region of node

Array containing the character image is divided in 16 equal regions according to its dimensions and according to position of node region is assigned to it.

3.4.7 Finding directions of further nodes

Slopes of all further nodes from a node are found by there corresponding line and gap parameters and according to slope value direction is assigned to a node.

3.5 Process of writing the Series in Reference table

First write down the S, all E and number of directions in which a node breaks in serial order as shown below. Start the series with the S. Note down the directions in which the character proceeds at a particular node. Write the descriptions of each node as discussed above. Series has been written in the sequence of numbering of nodes.

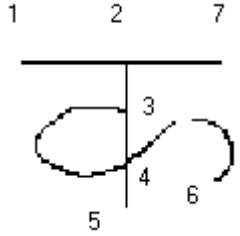

Fig-7(DevNagri Character Kaa-for descriptor Model)

1→ S
2→ 2
3→ 2 (S, 1, 1) (2, 1&7, 3/2) (2, 5/6&7, 6/7) (2, 2&7, 10/11) (E, 3, 14/15) (E, 4/3, 15/16) (E, 5, 4).
4→2
5→E
6→E
7→E

These DevNagri Character set-features are in the sequence which is universally followed. The way the above mentioned characters have been expressed using the model, is to help in formation of state transition matrix of HMMs [17]. Every character will have a well defined HMM of its own where the HMM has been trained by analyzing the sample data from 100 writers using our model. Original characters have not been taken and in place, symbolic representations of characters have been taken because we intended to explain the method. After training HMMs, we tested the method by taking test samples from twenty persons and the results were 100%. Samples have not been given keeping in view the length constraints of the paper. The variation in the character due to the change in the writer or pre-process cycle has been taken care. For example, we understand that the nodes are the major feature and the direction and region will vary and hence a tolerance based representation has been suggested. This helps in the formation and therefore implementation of the HMM based recognition process.

3.6 Results & Conclusion

The table provides the essence of the DevNagri characters. To recognize the hand-written Hindi script, the above mentioned method has been applied to extract the features, and thereafter the finding from the table have been used to express each character using HMM . This method provides fast recognition which is less than 100 ms per character and the recognition rate was 100% when the method was applied to the sample text from 100 writers and the characters were isolated from each other.

REFERENCES

[1] Y.Y.Tang, S.W.Lee and C.Y.Suen, "VLSI architecture for pattern recognition based on algorithm array mapping" , Proc. 2nd Pacific Rim int. Conference Artificial

Intelligence. Pp. 1036-1043, Seoul, Korea (September 1992).

[2] HEE-SEON Park and SEONG-WHAN Lee, "Off-Line Recognition of large-set handwritten characters with multiple hidden Markov models", Pattern Recognition, Vol. 29, No.2, pp231-244, 1996, Elsevier Science Ltd. GB.

[3] H. Nishida, "A Structural Model of shape Deformation," Pattern Recognition, Vol 28,no. 10,pp. 1,611-1,620,1995.

[4] L.Lam and C.Y.Suen, " Structural Classification and Relaxation Matching of Totally Unconstrained Handwritten Zip-Code numbers," Pattern Recognition, Vol. 21, no. 1, pp. 19-31, 1988.

[5] K.W.Cheung, D.Y.Yeung and R.T.Chin, " A United Framework for Hand Written Character Recognition using deformable models," Proc. Second Asian Conf. Computer Vision, vol.1, pp. 344-348, 1995

[6] P.Y.Simard, Y.Le. Cun and J.S.Denker, "Memory based Character Recognition using a Transformation Invariant Metrics, "Proc. 12th Int, Conf. On Pattern Recognition, pp. 262-267, Oct. 1994.

[7] R.G. Casey, "Moments Normalization of Hand printed Characters, "IBM Research and Development, pp. 548-557, Nov. 1970.

[8] T.Wakashara, "Shape Matching Using LAT and it's application to Handwritten Numeral Recognition, "IEEE Trans. Pattern Analysis and Machine Intelligence, vol.16, no. 6, pp. 618-629, June 1994.

[9] Anil K. Jain and Douglas Zongker , "Representation and Recognition of Handwritten Digits using Deformable Templates", IEEE Trans. On PAMI, vol.19, no. 12, December 1997.

[10] Hirobumi Nishida, "An approach to integration of off-line and on-line recognition of handwriting", Pattern Recognition Letters 16 (1995) 1213-1219.

[11] Young-Sup Hwang and Sung-Yang Bang, "Recognition of unconstrained handwritten numerals by a radial basis function neural network classifier", Pattern Recognition Letters 18(1997) 657-664.

[12] Quen-Zong Wu , Suh-Yin Lee, I-Chang Jou , "On-Line signature verification based on split-and-merge matching mechanism", Pattern Recognition Letters 18(1997)665-673

[13] Andrew W.Senior, A.J.Robinson, "An off-line cursive handwriting recognition system", IEEE, TPAMI, Vol.20 No.3, March 1998.

[14] Bong-Kee Sin and Jin H. Kim , "Ligature Modeling for On-Line Cursive Script Recognition", IEEE TPAMI, Vol. 19., No., 6, June 1997

[15] Gyeonghwan Kim and Venu Govindraju, " A lexicon driven approach to Handwritten word recognition for real time applications, IEEE, TPAMI, VOL.19 ,NO.4,APRIL 1997.

[16] Basit Hussain and M.R.Kabuka, "A Novel Feature Recognition Neural Network and it's application to character recognition", IEEE,TPAMI,Vol.16 ,No. 1 Jan 1994.

[17] Rabiner Lawrence R and Juang, "An Introduction to HMM", IEEE ASSP, Vol. 3, No. 1, pp. 4-16, Jan 1986.