PREPROCESSING PHASE FOR HANDWRITTEN DEVANAGIRI WORD RECOGNITION

Mr.Pradeep Singh Thakur*

Mr.SandeepPatil**

Abstract

In this paper we focused the importance of the pattern classification and its application. We list the characteristics of Hindi language writing style, furthermore focused onthe preprocessing step of the recognition system. We describe a complete system for the recognition of isolated handwritten Devanagari word using Fourier Descriptor and Hidden-Markov Model (HMM). The HMM has the property that its states are not defined as a priory information, but are determined automatically based on a database of handwritten word images. In this work the image database consist of 500 images of handwritten Devanagari words from 50 different writers. Before extracting the features, the images are normalized using image isometrics such as translation, rotation and scaling. After normalization the Fourier features are extracted using Fourier Descriptor. An automatic system trained 500 images of image database and word model form with multivariate Gaussian state conditional distribution. A separate set of 100 words was used to test the system. The recognition accuracy for individual words varies from 90% to 98% for number of states per model N=3.

Keywords: IT, CJK, HMM, Multivariate, Gaussian, mu, sigma.

*M.E. (Communication), SSCET, BHILAI (C.G), INDIA

**Electronics & Telecommunication Department, SSCET, BHILAI (C.G), INDIA

1. INTRODUCTION

The penetration of Information Technology (IT) becomes harder in a country such as India where the majority read and writes in their native language. Therefore, enabling interaction with computers in the native language and in a natural way such as handwriting is absolutely necessary. Indic script recognition poses different challenges when compared to Western, and Chinese, Japanese and Korean (CJK) scripts. When compared to Western scripts, Indic scripts exhibit a large number of classes, stroke order/ number variation and two dimensional natures. Indic script recognition also differs from that of CJK in a few significant ways. In the case of CJK scripts, the shape of each stroke in a character is generally a straight line and hence stroke direction based features are often sufficient. But in the case of Indic scripts, the basic strokes are often nonlinear or curved, and hence features that provide more information than just the directional properties are required. Moreover, in CJK scripts, a word is generally written discretely and hence segmenting it into characters is much easier when compared to Indic scripts, where the most common style of writing is run-on. Due to these differences, the techniques employed for other scripts may not be readily applicable for Indic script recognition. In our present work for word recognition, we have applied the holistic approach to avoid the overhead of segmentation and due to lack of standard benchmark database for training the classifier. Since a standard benchmark database was not available for Indian script so we created a word database for Devanagari to test the performance of our system. In the present report, training and test results of the proposed approach are presented on the basis of this database.

2. DEVANAGARI SCRIPT

Devanagari is the script used for writing Hindi which is the official language of India. The Devnagari script is a mixture of syllabic and alphabetic scripts. It is written in a left to right manner and has no capital letters like those present in Latin scripts. Each Devnagari consonant has an inherent vowel (A). Vowels can be written as independent letters, or by using a variety of diacritical marks which are written above, below, before or after the consonant they belong to. Devnagari has 13 independent vowels, 33 independent consonants and 12 dependent vowel signs (Fig. 1.1).

Independent Vowels												
अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ	अं	अः
A	AA	I	II	U	UU	R	E	AI	O	AU	ANG	AHAA
Consonants												
क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट	ठ	ड
KA	KHA	GA	GHA	NGA	CA	CHA	JA	JHA	NYA	TTA	TTHA	DDA
ढ	ण	त	थ	द	ध	न	प	फ	ब	भ	म	य
DDHA	NNA	TA	THA	DA	DHA	NA	PA	PHA	BA	BHA	MA	YA
र	ल	व	श	ष	स	ह						
RA	LA	VA	SHA	SSA	SA	HA	KSHA	TRA	GYAN	DDHA	RHA	
Dependent Vowel signs												
ा	ि	ी	ु	ू	ृ	े	ै	ो	ौ	ं	ः	
AAKAR	IKAR	IIKAR	UKAR	UUKAR	RKAR	EKAR	AIKAR	OKAR	AUKAR	ANUSVARA	BISARGA	

Fig. 1.1 Devnagari character set.

The Devnagari alphabet is used for writing Hindi, Sanskrit, Marathi, Nepali languages. The Devnagari script has a moderately large symbol set and shapes of characters are extremely cursive even when written separately. Also, there are few characters, which are similar in shape in their handwritten form. These pose as various points of difficulty when it comes to automatic recognition of handwritten Devnagari characters\words. One of the characteristic features of Devnagari script is that there is a horizontal line on the top of most characters. This line is called head line or Matraor Shirorekha. A Devnagari word is divided into three zones, namely, upper zone middle zone and lower zone. Matradivides the upper and middle zones (Fig. 1.2).

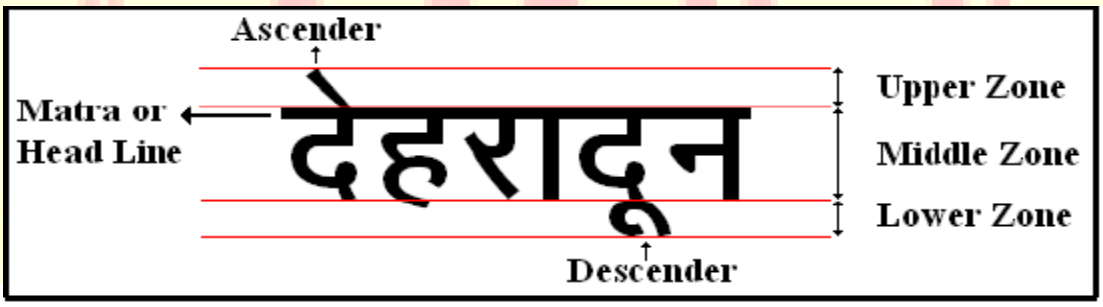

Fig 1.2 Zones of Devnagari word.

3.WORD RECOGNITION SYSTEMS

The process of handwriting recognition involves extraction of some defined characteristics called features to classify an unknown handwritten character into one of the known classes. A typical handwriting recognition system consists of several steps, namely: preprocessing, segmentation, feature extraction, and classification. Several types of decision methods, including statistical methods, neural networks, structural matching (on trees, chains, etc.) The stochastic processing (Markov chains, etc.) have been used along with different types of features [1-5]. Many recent approaches combine several of these techniques together in order to obtain improved reliability, despite wide variation in handwriting. A generic word-recognition system has two inputs: a digital image, assumed to be an image of a word and a list of strings called a lexicon, representing possible identities for the word image. In general, before looking for features, some preprocessing techniques are applied to the word image to avoid recognition mistakes due to the processing of irrelevant data (seeFig.1.3). The goal of the word recognition system is to assign a match score to each candidate in the lexicon. The match score assigned to a string in the lexicon represents the degree to which the image "looks like" the string. The output from this matching process is usually followed by a post processing step to check for highly unlikely decisions. Finally, a sorted lexicon is the output from the word recognition system.

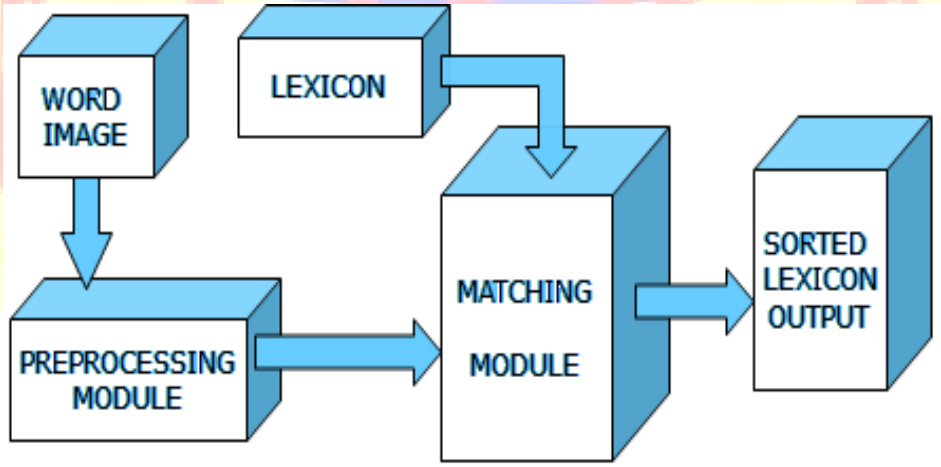

Fig.1.3 over view of word recognition system

4. PREPROCESSING PHASE: -IMAGE NORMALIZATION AND FEATURE EXTRACTION

The binary image written in MS-paint consists of a black foreground in front of a large white background. Hence the image is inverted such that the background is black and the foreground is white. This is done by subtracting the binary image from a matrix of 1s of the same size. Moreover, smaller number 1s will mean lesser calculations in correlation. To extract features, which are invariant to translation and scaling, it is necessary to normalize images. For the process of normalization the method of moment normalization is used, which is proposed by Parantonis and Lisboa [8]. The regular geometrical moment of order zero and one is used to find the centre of gravity of centroid. The (p+ q) thorder geometrical moment [9] of a digital image f (x1 , y1) of size MxMis given as

$$M_{p,q} = \sum_{x_1=0}^{M-1}\sum_{y_1=0}^{M-1} {x_1}^p {y_1}^q f(x_1, y_1). \qquad (1)$$

Where p and q positive integers. The zeroth order moment can be obtained by putting $p = q = 0$ in (1) It will be then transformed into

$$M_{0,0} = \sum_{x_1=0}^{M-1}\sum_{y_1=0}^{M-1} {x_1}^0 {y_1}^0 f(x_1, y_1). \qquad (2)$$

We can get first order moments in two ways either by putting $p = 1 \; and \; q = 0$ or $p = 0 \; and \; q = 1$ .With $p = 1 \; and \; q = 0$ we get first order moment as

$$M_{1,0} = \sum_{x_1=0}^{M-1}\sum_{y_1=0}^{M-1} {x_1}^1 {y_1}^0 f(x_1, y_1). \qquad (3)$$

From first order moment we get important information about the location of object in the image. It is called as center of gravity or centroid. If it is assumed to be $(\bar{x}, \bar{y})$, then centroid of the object is calculated as

$$\bar{x} = \frac{M_{1,0}}{M_{0,0}} \quad \text{and} \quad \bar{y} = \frac{M_{0,1}}{M_{0,0}} \qquad (4)$$

In the database objects in the images can have centroid anywhere in the image frame. But the Fourier Descriptor is used to resize the image into 65X65 and then translate it to the center of the image frame (33,33).These features are extracted from all the images in the data base and are applied to HMM for training. The Fig.1.4 shows the original image for word 'Praveen' and image after translation (33, 33) and scaling by factor β=400.

(a)Original image (b)Image after translation and scaling

Fig.1.4 Image Normalization

5. HMM IN HANDWRITTEN WORD RECOGNITION

A hidden Markov model is a doubly stochastic process, with an underlying stochastic process that is not observable (hence the word hidden), but can be observed through another stochastic process that produces the sequence of observations [10-14]. The hidden process consists of a set of states connected to each other by transitions with probabilities, while the observed process consists of a set of outputs or observations, each of which may be emitted by each state

according to some output probability density function (PDF) [15-17]. Depending on the nature of this PDF function several kinds of HMMs can be distinguished.

An HMM is formally defined by the following parameters:

• A = {aij}, the Markov chain matrix, where aijis the probability of transition from state i to state j, with i, j ∈ {1, 2, . . . , N}, N being the number of states.

• B = {bj(k)}, the output distribution matrix, where bj(k) is the probability of producing symbol k, when the Markov process is in state j. k ∈ {1, 2, . . . , M}, M being the size of the symbol alphabet.

• π= {πi}, the probability that the Markov process starts in state i. Without loss of generality, it will be assumed in the remaining of this section that state 1 is the only initial state. Thus, π1 = 1 and πi = 0 for i ≠1. In the same way, we assume that N is the only terminating state. λ = (A,B) is a compact representation of the HMM.

An HMM with multivariate Gaussian state conditional distribution consists of:

'π_0' Row vector containing the probability distribution for the first (unobserved) state:

$$\pi_0(i) = P(s_1 = i)$$

'A' Transition matrix:

$$a_{ij} = P(s_t + 1 = j | s_t = i)$$

Mu: Mean vectors (of the state-conditional distributions) stacked as row vectors, such that mu(i, :) is the mean (row) vector corresponding to the i-th state of the HMM.

Sigma: Covariance matrices. These are stored one above the other in two different way depending on whether full or diagonal covariance matrices are used: For diagonal covariance matrices, sigma (**i**, :) contain the diagonal of the covariance matrix for the **i**-th state. For the purpose of isolated handwritten character recognition, it is useful to consider left- to right models. In left-to-right model transition from state i to state j is only allowed if j ≥ i, resulting in smaller number of transition probabilities to be learned. The clusters of observation are created

for each model separately by estimating Gaussian mixture parameter for each model. The function **mu** and **sigma** able to determine the dimension of the model and the type of covariance matrices i.e. size of the observation vector and the number of states. The matching process computes a matching score between the sequence of observation vector and each character model using the Viterbi algorithm [12]. After post processing, a lexicon sorted by matching score is the final output of the character recognition system. For transition matrix **A**, the row vector summation must be equal to 1 for any number of states N. The transition matrices (3x3) for N=3 shows in Table1 and the N-mean vector and the covariance matrices for word 'Praveen' is calculated for N=3 as shown in Table 2.

0.85	**0.15**	**0.0**
0.0	**0.85**	**0.15**
0.0	**0.0**	**1.0**

Table 1: A transition Matrix

N mean vector of word "Praveen"				
-23.8999	2.0253	0.5573	0.7112	0.2102
-27.7756	2.8529	0.246	0.2196	0.0545
-38.784	5.2439	-0.5321	0.897	-0.5022
Diagonal of covariance matrices for word "Praveen"				
13.6664	1.6745	0.2569	0.1057	0.1361

Table 2:The N-mean vector and the covariance matrices for word 'Praveen'

6. EXPERIMENTAL RESULT

As there does not exist any standard database of word images. In the present work the image database is collected from 50 different persons, A total of 500 genuine words were collected from a population of 50 human subjects which included 25 women and 25 men .To make the size of images in the database constant, these images are then edited with the help of Microsoft paint, available in Windows operating system. They are in arbitrary translation, rotation and scale.Out of these 500 digital images 400 images are used for training purpose and remaining 100 images are used for testing purpose. When we applied first 100 images ten samples of each words then the recognition result varies for individual's words. The recognition percentage for different word sample is obtaining in between 90-98%.

Applied Images words	Number of Applied images	Image correctly recognized	% Rate of recognition
vÒh	50	47	94
v¨e	50	48	96
gfj	50	46	92
jke	50	48	96
Áoh.k	50	45	90
Ádk'k	50	46	92
lanhi	50	49	98
f'kojkt	50	47	94

Table 3:Showing recognition percentage for each 50 database sample of Devanagari word

7. CONCLUSION

In this work, a writer-independent handwritten Hindi word recognition system that employs HMMs for word modeling was discussed. We used a segmentation-free continuous density

hidden morkov modeling approach to improve the performance of the existing techniques in the literature. The recognition system was trained and tested on the Devnagari data. The recognition result obtained from this work varies for individual's words. The overall recognition percentage for different word sample is obtaining 94%. There are several possible improvements to the system. The relatively low performance in the case of high lexicon size can be improved by the use of statistical language models, which are commonly applied in Western cursive recognition. To increase the recognition percentage to obtained a maximum result. For this purpose we can used the combined method i.e. both (Analytical & Holistic) which can reduce the drawback of this method and have the advantage of combined method.

REFERENCES

[1]H. Bunke, M. Roth, and E. G. Schukat-Talamazzini. Offline Cursive Handwriting Recognition usingHiddenMarkovModels.Pattern Recognition, 28(9):1399–1413, 1995.

[2] S. Marinai "Introduction to document analysis and recognition", Studies in Computational Intelligence (SCI),Vol. 90, pp. 1–20, 2008.

[3] Y.Y. Tang, C.Y. Suen, C.D. Yan, and M.Cheriet, "Document analysis and understanding: a brief survey" First Int. Conf. on Document Analysis and Recognition, Saint-Malo, France, pp. 17-31, October 1991.

[4] R. Plamondon and S. N. Srihari, "On-line and off-line handwritten recognition: A comprehensive survey", IEEE Trans on PAMI, Vol.22, pp.62-84, 2000.

[5] Swapan Kr. Parui and Bikash Shaw, "Offline handwritten Devanagari word recognition: An HMM based approach", LNCS 4815, Springer-Verlag, (PReMI-2007), 2007, pp. 528- 535.

[6] I. K. Sethi and B. Chatterjee, "Machine recognition of constrained hand printed Devanagari", Pattern Recognition, Vol. 9, pp. 69-75,1977.

[7] Bikash Shaw, Swapan Kumar Parui and MalayappanShridhar, "A segmentation based approach to offline handwritten Devanagari word recognition," PReMI, IEEE, pp. 528-35.

[8]S.J. Parantonis and P.J.G. Lisboa, Translation, rotation and scale invariant pattern recognition by high-order neural network and moment classifiers, IEEE Trans. Neuralnetworks, 3(2) (1992), 241-251.

[9]H2M: A set of MATLAB/OCTAVE functions for the EM estimation of mixture and hiddenmorkov model by Olivier Cappe ENST. Dpt. TSI/LTCI (CNRS-URA 820),France, August 24, 2001.

[10]Sandeep B. Patil, G.R. Sinha and Vaishali S. Patil, Isolated handwritten Devnagrinumeral recognition usingHMM, IEEE Trans. eait, Second International Conference on Emerging Applications of InformationTechnology, (2011), 185-189.

[11]C. Rafael Gonzalez and E. Richard Woods, Digital Image Processing, Addison-Wesley Publishing Company.

[12]A. Magdi Mohamed and Paul Gader, Generalized hidden markov models-part II: application to handwrittenword recognition, On fuzzy system, IEEE Trans., 8(1) (February 2000).

[13]L.R. Rabiner, A tutorial on hidden markov models and selected application in speech recognition, Proceedings of the IEEE, 77(2) (February 1989), 257-286.

[14]M. Christopher Bishop, Pattern recognition and machine learning, Information Science and Statistic Series, Springer, 423-455.

[15]JiaZeng and Zhi-Qiang Liu, Markov random field-based statistical character structure modeling for handwritten Chinese character recognition, On pattern analysis and machine intelligence, IEEE Trans., 30(5), May(2008).

[16]R. Stephan Veltman and Ramjee Prasad, Hidden markov models applied to on-line handwritten isolated character recognition, On image processing, IEEE Trans., 3(3) (1994).

[17]Thierry Artie, SanparithMarukatat and Patrick Gallinari, Online handwritten shape recognition using segmentalhidden markov models, On pattern analysis and machine intelligence, IEEE Trans., 29(2), February (2007).