International Journal of Engineering Innovation & Research
Volume 1. Issue 3. ISSN : 2277 – 5668

Generalized Hidden Markov Models To Handwritten Devanagari Word Recognition

Mr. Pradeep Singh Thakur
M.E. (Communication)
SSCET, BHILAI (C.G), INDIA
pradeep.singh4u@gmail.com

Mr. Sandeep Patil
Electronics & Telecommunication Department
SSCET, BHILAI (C.G), INDIA
sandeep.patil@sscet.ac.in

***Abstract* - Hidden Markov Models (HMM) have long been a popular choice for Western cursive handwriting recognition following their success in speech recognition. Even for the recognition of Oriental scripts such as Chinese, Japanese and Korean, Hidden Markov Models are increasingly being used to model substrokes of characters. However, when it comes to Indic script recognition, the published work employing HMMs is limited, and generally focused on isolated character recognition. In this effort, a data-driven HMM-based handwritten word recognition system for Hindi, an Indic script, is proposed. Though Devanagari is the script for Hindi, which is the official language of India, its character and word recognition pose great challenges due to large variety of symbols and their proximity in appearance. The accuracies obtained ranged from 30% to 60% with lexicon. These initial results are promising and warrant further research in this direction. The results are also encouraging to explore possibilities for adopting the approach to other Indic scripts as well.**

***Keywords* - CDVDHMM, CJK, DAR, GHMM, HMM, HWR, OCR, IT.**

I. 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. Hidden Markov Models are suitable for handwriting recognition for a number of reasons [1]. Since these are stochastic models, they can cope with noise and variations in the handwriting. The observation sequence that corresponds to features of an input word can be of variable length, and most importantly, word HMMs can solve the problem of segmentation implicitly. In this work, Hidden Markov Models, which are shown to be successful for western cursive recognition, and CJK script recognition to some extent, are applied to model Hindi words.

Handwritten word recognition is an important area of Document Analysis and Recognition (DAR). DAR is a mechanism in which the document images are processed to obtain text and graphics features. The main objective of DAR is to read the intended information from the document using computer as much as a human would do. The outcome of DAR system is usually in the ASCII format [2]. The applications of DAR [3-4] include such as office and library automation, Publishing houses, help to the visually handicapped when interfaced with a voice synthesizer, postal service assistance, reading entrance examination forms, processing of applications of victims and criminal records in police station, etc. A slight mistake in interpreting the characters can lead to mistake in the automation process such as wrong dispatch in postal service or wrong entry in entrance examination forms. Handwriting recognition can be achieved by character, word and sentence level. A character recognizer needs to be trained with sample characters from the alphabets used in the language. There are two approaches for the recognition of isolated handwritten Devanagari words [5]. The first is to segment the word into its character parts, individually recognize each character, and then reconstruct the word. The major drawback of this approach for the Devanagari script is that the words contain Matra, Shirorekha, conjunct characters, modifiers and lack of standard benchmark database for training the classifier. The second scheme is to recognize the word in its entirety. Word recognizers are complex if they are general purpose but are simpler if it is based on specific lexicon. This approach of word recognition avoids the overhead of character segmentation.

While significant advances have been achieved in recognizing Roman based scripts like English, ideographic characters (Chinese, Japanese, Korean, etc) and Arabic to some extent, OCR research on Indian scripts is very less. Only few works on some of the major scripts like Devanagari, Bangla, Gurumukhi, Tamil, Telugu, etc. are available in the literature.

The era of handwritten Devanagari character recognition was started in the early days of OCR research by Sethi et al. [6]. The research in offline Devanagari word recognition was started by Parui et al. proposed a HMM based holistic approach for the word recognition [4]. Later, Shaw et al. published a segmentation based approach [7]. 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.

II. Features of Devanagari Script

Devanagari is the script used for writing Hindi which is the official language of India [8]. It is also the script for Sanskrit, Marathi and Nepali languages. Devanagari script consists of 13 vowels and 33 consonants characters. These characters are called the basic characters. The characters may also have a half form. A half character in most of the cases touches the following character, resulting in a composite character. Some characters of Devanagari script take the next character in their shadow because of their shape. The script has a set of modifier symbols which are placed either on top, at the bottom, on the left, to the right or a combination of these. Top modifiers are placed above the shirorekha (Head line), which is a horizontal line drawn on the top of the word. The lower modifiers are placed below the character which may or may not touch the characters. More than one lower modifier may also be placed below one character. A character may be in shadow of another character, either due to a lower modifier or due the shapes of two adjacent characters. Upper and lower modifiers with basic character modifiers make OCR with Devanagari script very challenging. OCR is further complicated by compound characters that make character separation and identification very difficult.

III. Hidden Markov Models

HMMs are an extension of Markov chains, for which symbol (or observation) production along states (symbol generation along transitions is another variant that is widely used) is no longer deterministic but occurs according to an output probabilistic function, hence their description as a double stochastic process. The Markov chain serves as an abstract representation of structural constraints on data causality. Such constraints are usually derived from our knowledge about the problem and the data as well as from taking into account the way data are transformed into sequential strings. The output probabilistic functions embody the second stochastic process and model the inherent variability of characters or of any basic unit of the language we are dealing with. The variability results from various distortions inherent to image acquisition, binarization, intra- and interscriptor diversity of writing styles, and so on. The underlying distribution of these functions defines the HMM type. If it is discrete (nonparametric), the model is called discrete HMM. The discrete alphabet (codebook), in this case, is usually obtained through vector quantization [9, 10]. Continuous HMMs [11] model the output function by a continuous probability density function, usually a mixture of Gaussians. Semicontinuous HMMs [12] can be thought of as a compromise of these two extremes and have the output functions sharing the same set of Gaussians (but model Gaussian coefficients as state-dependent parameters).

An HMM is formally defined by the following parameters:

• $A = \{aij\}$, the Markov chain matrix, where aij is the probability of transition from state i to state j, with $i, j \in \{1, 2. . . N\}$, 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. $k \in \{1, 2, . . . , M\}$, M being the size of the symbol alphabet.

• $\pi = \{\pi i\}$, 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, $\pi 1 = 1$ and $\pi i = 0$ for $i \neq 1$. In the same way, we assume that N is the only terminating state. $\lambda = (A, B)$ 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 full covariance matrices,

$sigma((1 + (i - 1) \square P): (i \square P), \square)$

(Where 'P' is the dimension of the observation vectors) is the covariance matrix corresponding to the i-th state; for diagonal covariance matrices, Sigma(i, :) contains the diagonal of the covariance matrix for the i-th state (i.e. the diagonal coefficients stored as row vectors). A Gaussian mixture model, is rather similar except that as the underlying jump process being i.i.d., pi0 and A are replaced by a single row vector containing the mixture weights **w** defined by

$w(i) = P(s_t = i)$

Most functions (those that have mu and Sigma among their input arguments) are able to determine the dimensions of the model (size of observation vectors and number of states) and the type of covariance matrices (full or diagonal) from the size of their input arguments. This is achieved by the two functions hmm_chk and mix_chk.

IV.HMM In Handwritten Word Recognition

Hidden Markov models (HMM's) have been applied to handwritten word recognition by many researchers during the last decade. Chen *et al.* [13] used discrete HMM to recognize words using lexicons. The input word image is first segmented into a sequence of segments in which an individual segment may be a complete character, a partial character, or joint characters. The HMM parameters are estimated from the lexicon and the training image segments. A modified Viterbi algorithm is used to find the best state sequences. One HMM is constructed for the whole language and the optimal paths are utilized to perform classification. When tested on a set of 98 words, this system is able to achieve 72.3% success for a lexicon of size 271 using 35 features. Another application of HMM in character segmentation-based word recognition was presented by Chen *et al.* [14]. Chen used continuous density, variable duration hidden Markov model (CDVDHMM) to build character models. The character models are left-to-right HMM's in which it is possible to skip any number of states. A major disadvantage of this technique is that it is slow to train and in operation because of introducing more parameters and computation for the state duration statistics. Another interesting technique was developed by Gillies [15] for cursive word recognition using left-to-right discrete hidden Markov models. In this technique, each column of the word binary image is represented as a feature vector. A series of morphological operations are used to label each pixel in the image according to its location in strokes, holes, and concavities located above, within and below the core region. A separate model is trained for each individual character using word images where the character boundaries have been identified. The word matching process uses models for each word in a supplied lexicon. The advantage of this technique is that the word need not be segmented into characters for the matching process. When tested on a set of 269 cursive words this technique is able to achieve 72.6% success for a lexicon of size 100.

Performing word recognition without segmenting into characters is an attractive feature of a word recognition system since segmentation is ambiguous and prone to failure for a significant portion of the handwritten words coming from the mail stream. In an attempt to avoid some limitations of the existing segmentation-based handwritten word recognition techniques we designed a segmentation-free model-based system. In this technique, the training process needs representative images of each word in a lexicon. Our proposed segmentation-free system computes features from each column in a word image and uses a continuous density HMM for each word class.

The continuous density HMM has a significant advantage over the discrete density HMM. Using a continuous density HMM is attractive in the sense that the observations are encoded as continuous signals. Although it is possible to quantize such continuous signals there might be a serious degradation associated with such quantization. In the case of the discrete HMM, a codebook must be constructed prior to training the models for any class. The codebook is constructed using vector quantization or clustering applied to the set of all observations from all classes. By contrast, in the continuous case, the clusters of observations are created for each model separately (e.g., by estimating Gaussian mixture parameters for each model). Thus, continuous density HMM does provide more flexibility for representing the different levels of complexity and feature attributes of the individual word classes. Furthermore, in the discrete case, training a new model may require creating a new codebook since the features required by the new model may be quite different from those represented in the old codebook. Since the continuous density models perform clustering independently for each class, there is no such requirement.

One computational difficulty associated with using continuous density HMM's the inversion of the covariance matrices. We overcame this difficulty by reducing the dimensionality using principal component analysis and approximating the densities as a mixture of Gaussian densities with diagonal covariance matrices. The results provided in this paper demonstrate that the approach performs reasonably well when confronted with various styles of handwritten words.

V. Word Recognition Systems

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). 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.

The approach considered for our research is based on the use of the classical and generalized hidden Markov models. The first task accomplished in this research is the development and demonstration of a classical HMM handwritten word recognition system using novel image processing and feature extraction techniques. We used a segmentation-free continuous density hidden Markov modeling approach to improve the performance of the existing techniques in the literature. Our approach is the first to use continuous density hidden Markov models for segmentation-free handwritten word recognition. The second task is the development and demonstration of a generalized HMM system. In this paper, we describe our implementation for the classical and generalized models in detail.

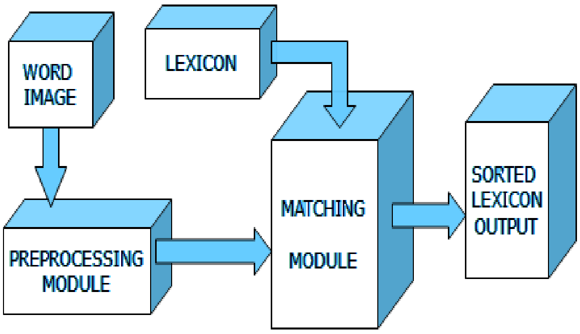

Fig.1. Over view of word recognition system

The remainder of this paper is organized to describe our research approach in the following manner. In Section VI, a complete description of the implemented classical HMM handwritten word recognition systems is provided. This section describes the feature extraction, dimensionality reduction, training and matching strategies. In Section VII, a complete description of the database is provided. We describe the database and the preprocessing steps we applied to the raw images contained in the standard training and testing sets. Section VIII shows the experimental results and analysis of the performance of the implemented systems. Finally, Section IX is dedicated to the summary of this study and the suggestions for future research.

VI. Design and Implementation of HMM Handwritten Word Recognition System

In our handwritten word-recognition systems, an HMM is constructed for each word class using the training data. We used left-to-right HMM's with continuous probability densities to model the word classes. The approach is a Segmentation-free technique. Classification is performed according to the matching scores computed from the optimal state sequence using the Viterbi algorithm in the classical HMM and the fuzzy Viterbi algorithm in the generalized hidden Markov model (GHMM). The inputs to the word recognition algorithm are a binary word image and a lexicon (see Fig. 2). After preprocessing the input binary word image, the resultant image is subjected to a feature extraction process. The output from this process is a sequence of observation vectors; each corresponds to an image column. A word model is constructed for each string inside the lexicon. The string matching process computes a matching score between the sequence of observation vectors and each word model using the Viterbi algorithm. After post processing, a lexicon sorted by the matching score is the final output of the word recognition system.

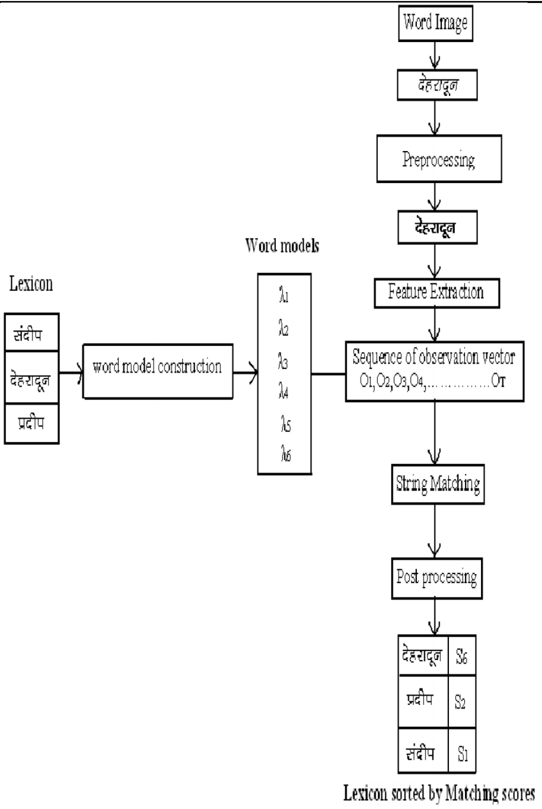

Fig.2. System Overview

The following sections describe the system components and related implementation issues.

A. Feature Extraction

The performance of any classification or recognition algorithm depends, to a large extent, on the representation chosen, i.e., the features or primitives that are extracted from the inputs. These characteristics must, as far as possible, summarize the information, which is pertinent and useful for clustering and at the same time eliminate useless or irrelevant information, i.e., the randomness due to variability and nondiscriminant information.

The description of a binary word image as an ordered list of observation vectors is accomplished by encoding a combination of features computed from each column in the given preprocessed image. This representation uses what we refer to as the Fourier descriptor (FD) a fundamental need is to retrieve a particular shape from a given image. In principle, it represents the shape of the object in the frequency domain. By doing so, Fourier descriptors enjoy multiple advantages over their counterparts. These advantages include strong discrimination ability, low noise sensitivity, easy normalization, and information preservation.

B. Word Models

For the purpose of isolated handwritten word recognition, it is useful to consider left-to-right models. In a left-to-right model transition from state to state is only allowed if, resulting in a smaller number of transition probabilities to be learned. The left-to-right HMM has the desired property that it can readily model signals whose properties change over time.

C. Training word HMM's

Before using an HMM in evaluation or decoding, we first need to estimate its parameters. That is, given a set of training data, we want to derive a model that is able to perform reasonably well on future unseen data. Unfortunately, no satisfactory and efficient method has been devised so far to simultaneously optimize the model structure (topology) and the model parameters.

A Hidden Markov Model is a doubly stochastic model [16]. The underlying stochastic process corresponds to state transitions that are hidden, but the state changes are observed through another set of stochastic processes that produce the output symbols. The output symbol is said to be discrete if it is from a finite alphabet, and it is continuous if it has real-valued attributes. Accordingly, the model is called discrete or continuous HMM. In our experiment, continuous HMMs were used to model the Hindi words since the features are real-valued. An HMM state is said to generate feature vectors following a probabilistic distribution, usually a mixture of Gaussians. The number of Gaussians in the mixture and the number of states in the HMM were determined empirically. HMM training was done using ML-based Baum–Welch algorithm, which is an implementation of the EM algorithm [17, 18] in the case of HMMs. Fig.3.shows the singular values for the covariance matrix of particular word samples used for training the system.

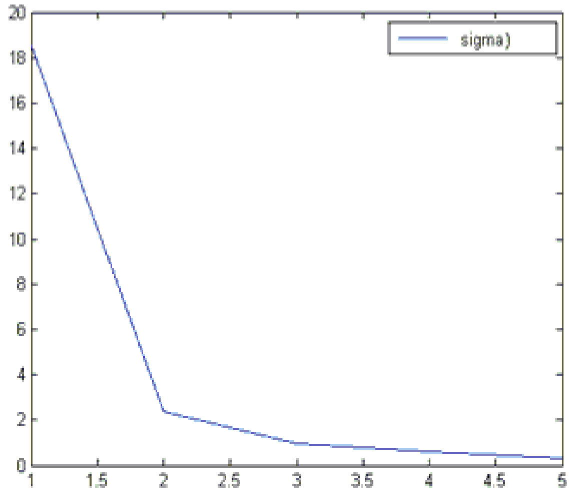

Fig.3.Singular values for the training covariance matrix of transition and gradient feature vectors.

D. Lexicon Pruning

A lexicon pruning component is used to improve the recognition process by limiting the number of lexicon entries, especially when dealing with large size lexicons. Indeed, an exhaustive search through all the lexicon entries is likely to have side effects on the recognition robustness in addition to being very time consuming. Lexicon pruning is actually a global recognition process. Since the words to be recognized possess some discriminant features like ascenders, decsenders, loops, and overall length, this process usually builds a coarse description of the character segments inside words along a set of features. For now, our system uses the overall length of the word to avoid matching against some of the strings in the provided lexicons.

E. Matching Strategies and Post Processing

We have different models for the input thus matching process needed the different models and lexicon. We match against each string in the lexicon using the Viterbi algorithm [19, 20] and the maximum is taken as the confidence for the given string.

VII. Database and Preprocessing

A. The Database

The database used for our experiments consists of handwritten words collected from people domain with 10 samples of each word from about 15 persons from different field and age. Data acquisition is done manually as well as automatically, i.e.,data collection for the experiment has been done from the different individuals. For manually data collection the writers were provided with the plain A4 sheet and each writer has asked to write Devanagari words and then collected document are scanned using scanner which is usually a low noise and good quality images. For automatically data collection is done by MS-paint. The binary image written in MS-paint consists of a black foreground in front of a large white background. The digitized images are stored as binary image in BMP format.

A sample of Devanagari handwritten words from the data set is shown in fig.4.

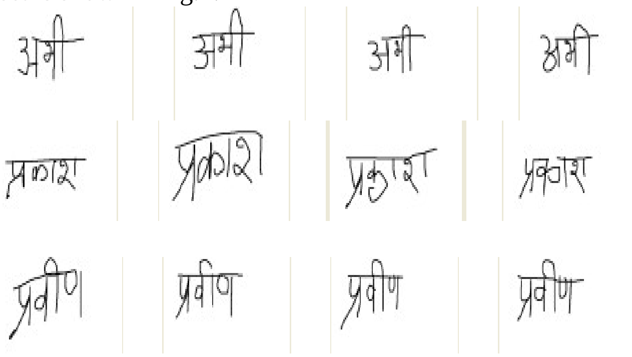

Fig.4.Sample database (Words)

B. Preprocessing

The following preprocessing steps have been implemented, fine tuned, and applied to all word images in the database: binarization, line removal, border cleaning, tilt correction, slant correction, smoothing and scaling. Binarization was performed using Otsu's thresholding methods [21].

VIII. Performance Evaluation

The recognition system was trained and tested on the Devnagari data, collected following the procedure described in Section VI. The recognition result obtained from this work varies for individual's words. The recognition percentage for different word sample is obtaining in between 30-60%.

IX. Conclusions and Future Work

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 Markov modeling approach to improve the performance of the existing techniques in the literature. One of the advantages of using HMM's is that the parameters do have meaning associated with them such as the expectation of transitions among states, the means and covariance matrices of clusters of feature vectors inside each state. This provides useful information for choosing initial values, which is very helpful for finding good approximation of the modeling parameters by applying the reestimation formulas inside the training algorithm.

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. A lot of efforts have been made to get higher accuracy but still there is tremendous scope of improving recognition accuracy by developing new feature extraction techniques or modifying the existing feature extraction techniques.

References

[1] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini. Offline Cursive Handwriting Recognition using Hidden Markov Models. 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 Malayappan Shridhar, "A segmentation based approach to offline handwritten Devanagari word recognition," PReMI, IEEE, pp. 528-35.

[8] P.S. Deshpande, L. Malik and S. Arora, "Characterizing handwritten Devanagari characters using evolved regular expressions", in Proceeding of TENCON, 2006, pp. 1-4.

[9] R. M. Gray. Vector quantization. IEEE ASSP Magazine, 4–29, 1984.

[10] J. Makhoul, S. Roucos, and H. Gish. Vector quantization in speech coding. Proceedings of IEEE, 73, 1551– 1588,1985.

[11] L. A. Liporace. Maximum likelihood estimation for multivariate observation of Markov sources. IEEE Transactions on Information Theory, 28(5), 729–734, 1982.

[12] X. D. Huang and M. A. Jack. Semi-continuous hidden Markov models for speech signals. Computer Speech and Language, 3, 239–251, 1989.

[13] M. Chen, "Off-line handwritten word recognition using hidden Markov models," in Proc. U.S. Postal Service Adv. Technol. Conf., Washington, DC, Nov. 1992, pp. 563–579.

[14] M. Chen and A. Kundu, "An alternative to variable duration HMM in handwritten word recognition," in Proc. 3rd Int. Workshop Frontiers Handwriting Recognition, Buffalo, NY, May 1993, pp. 48–54.

[15] A. Gillies, "Cursive word recognition using hidden Markov models," in Proc. U.S. Postal Service Adv. Technol. Conf., Washington, DC, Nov.1992, pp. 557–563.

[16] Rabiner R. "A tutorial on Hidden Markov Models and selected applications in speech recognition," Proceedings of IEEE, 1989, 79(2). pp. 257-286.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society B, 39, 1–38, 1977.

[18] T. K. Moon. The expectation-maximization algorithm. IEEE Signal Processing Magazine, 13(6), 47–60, 1996.

[19] A. Lifchitz and F. Maire. A fast lexically constrained Viterbi algorithm for on-line handwriting recognition. In Proceedings of the 7th International Workshop on Frontiers of Handwriting Recognition, Amsterdam, the Netherlands, pp. 313–322, 2000.

[20] L. Rabiner and B.-H. Juang. Fundamentals of Speech Recognition. Prentice-Hall, 1993.

[21] O. Nobuyuki, "A threshold selection method from gray-level histograms," IEEE Trans. Syst., Man, Cybern., vol. SMC-9, no. 1, pp.62–66, Jan. 1979.

Author's Profile

Sandeep Patil

Senior Associate Professor in the Department of Electronics & Telecommunication Engineering in Shri Shankaracharya College of Engineering & Technology (SSCET), Bhilai, Chhattisgarh, India. His interests are in the field of Digital Signal Processing and its Applications. He has published multiple articles in several National and International Journals.

Pradeep Singh Thakur

received B.E. in Electronics & Telecommunication Engg. in year 2005 and in pursuit for M.E. in Communication Engg. from Shri Shankaracharya College of Engineering & Technology (SSCET), Bhilai, Chhattisgarh, India. His interests are in the field of Digital Signal Processing and its Applications.