Recognition of Spoken Bengali Numerals Using MLP, SVM, RF Based Models with PCA Based Feature Summarization

Avisek Gupta and Kamal Sarkar
Department of Computer Science and Engineering, Jadavpur University, India

Abstract: *This paper presents a method of automatic recognition of Bengali numerals spoken in noise-free and noisy environments by multiple speakers with different dialects. Mel Frequency Cepstral Coefficients (MFCC) are used for feature extraction, and Principal Component Analysis is used as a feature summarizer to form the feature vector from the MFCC data for each digit utterance. Finally, we use Support Vector Machines, Multi-Layer Perceptrons, and Random Forests to recognize the Bengali digits and compare their performance. In our approach, we treat each digit utterance as a single indivisible entity, and we attempt to recognize it using features of the digit utterance as a whole. This approach can therefore be easily applied to spoken digit recognition tasks for other languages as well.*

Keywords: *Speech recognition, isolated digits, principal component analysis, support vector machines, multi-layered perceptrons, random forests.*

Received July 2, 2014; accepted March 15, 2015

1. Introduction

Speech recognition has always been a challenging area, with some of its earliest work dating back to the 1950s [8], where one of the first experiments in the field was recognising isolated digits spoken by a single speaker. Since then, a lot of work has been done in many languages to develop a speaker dependent or independent system of automatic speech recognition for both isolated and continuous speech. Since the early 2000s, the use of Mel Frequency Cepstral Coefficients (MFCC) in speech recognition has been quite popular [14, 15, 16, 19]. A study of recent work in Bengali speech recognition shows that some preliminary studies have been done and there is a scope for doing more work for improving performance of a speech recognition system.

This work focuses on the speaker-independent recognition of isolated Bengali digits in noise-free and noisy environments. Bengali is an eastern Indo-Aryan language. It is native to the region of eastern South Asia known as Bengal, which comprises present-day Bangladesh, the Indian state of West Bengal, and parts of the Indian states of Tripura and Assam. With about 220 million native and about 250 million total speakers, Bengali is one of the most spoken languages, ranked seventh in the world.

The development of a Bengali spoken digit recognition system would be useful in a number of areas. In railway stations, announcing the train numbers of arriving or departing trains could lead to automatic display of the train number. For people who find it difficult to dial numbers on phones, an alternate system could exist where an individual could simply read out the number to be dialled. These systems must be speaker-independent, and must give high recognition rates even in noisy environments.

It has been seen [15] that in the area of isolated digit recognition, high recognition results can occur when speech is recorded under quiet conditions. Our study investigates the recognition performance under noisy and noise free conditions. Some methods that have been used recently for speech recognition are Hidden Markov Models (HMM) [1, 13], Dynamic Time Warping [9], and Gaussian Mixture Models [2]. Though HMM is the widely used method for automatic speech recognition, the performance of Artificial Neural Networks (ANN) has been reported to be comparable to the HMM-based spoken digit recognition [3, 17].

The traditional HMM-based approach treats a spoken word as a collection of simpler subunits such as phones or sub-phones, and proceeds by segmenting the word into these units. Each word is modelled using a separate HMM.

The chief problems of this approach are:

1. Segmentation ambiguity: deciding where to segment the utterance.
2. Variable phone duration.
3. Portability to a new language domain.

In our work, we treat the digit utterance as a single, indivisible entity and attempt to recognize it using features of the digit utterance as a whole. We make no attempt to segment the word into subunits. The main

motivation behind using the word based approach for recognizing isolated Bengali spoken digits is that the number of classes is small and fixed and so, it becomes possible to collect a large number of training samples for each class. Since our proposed approach does not require the use of HMM lexicons and a phoneme dictionary while recognizing the spoken digits, this approach can be easily ported to a new language domain.

In this paper, we develop a system of recognizing Bengali digit utterances by multiple speakers in noise-free and noisy environments. The layout in Figure 1 describes the various units of the system.

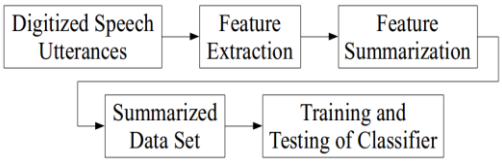


Figure 1. Layout of the Speech Recognition system.

The functions of the various units are described below.

- *Digitized speech utterances*: This is the data set of all digitized audio recordings of Bengali digits spoken by multiple speakers.
- *Feature Extraction*: Each digitized speech utterance is divided into multiple overlapping windows. From each window, 39 MFCC are extracted.
- *Feature Summarization*: The MFCC features over all windows for the utterance of a digit is summarized to a single feature vector using Principal Component Analysis.
- *Data Set*: This is the collection of all feature vectors, each labelled according to the digit utterance it represents.
- *Classifiers*: The classifiers are trained with the labelled data set for recognizing the digits. We have used Multi-Layer Perceptrons, Support Vector Machines and Random Forests as the classifiers for our spoken Bengali digit recognition task.

2. Data Collection

A corpus for isolated Bengali digits has been developed due to lack of availability of a standard Bengali speech corpus. A total number of 48 speakers were appointed to record digits for the speech corpus. Among them there were 24 male and 24 female speakers. Our aim is to develop a speech corpus that will contain speech from people of various locations across the state of West Bengal, India. Due to the availability of higher number of people from Kolkata, the number of speakers from Kolkata is higher than the rest. The number of speakers with different dialects from various districts in the state of West Bengal in India is given below in Table 1.

All recordings were done at 44.1 kHz and saved with an uncompressed 32-bit float bit rate, with 2 channels. Out of all the recordings in the speech corpus, 17 were recorded in a noise-free environment, and the rest of the 31 were recorded in a noisier setting of a normal room, where the windows were open and the fans were on.

For the recordings done in the quiet environment, the speakers spoke each digit approximately 20 times. For the recordings done in the noisy environment, speakers were requested to select 10 digits, as in the format of a phone number, and requested to speak those 10 digits 5 times.

Table 1. Number of speakers from various districts in west bengal, India.

Location	Number of speakers
Kolkata	25
North 34 Parganas	7
Bardhaman	3
Midnapore	2
Hooghly	2
Howrah	1
South 24 Parganas	1
Cooch Bihar	1
Bankura	1
Murshidabad	1
Birbhum	1
Jalpaiguri	1
Purulia	1
Nadia	1

3. Methodology

From the digitised Bengali digit utterances, Mel Frequency Cepstral Coefficients are extracted (Feature Extraction), following which Principal Component Analysis is done (Feature Summarization), and a data set is prepared, after which various classifiers are used for recognition. Each step is discussed next in detail.

3.1. Feature Extraction

From each speech recording, the entire utterance of a digit was manually isolated and saved separately. The computation of MFCC [12] involves dividing the speech utterance into small overlapping windows (around 25ms). The length of a window is kept short enough to get a reliable spectral estimate. The power spectrum of each window is calculated, after which a Mel filterbank is used to compute the quantity of energy present in various frequency regions. The Mel filterbank contains 26 filters that are spaced using the Mel scale. The Mel scale spaces filter by making them wider as the frequencies increase. This is modelled on how the human ear works. As frequencies keep increasing, the human ear cannot differentiate between the increasingly wider ranges of closely placed frequencies. The logarithm of the filterbank energies is taken, which is based on the fact that the human ear perceives loudness on a logarithmic scale. The Discrete Cosine Transform (DCT) of these log-filterbank energies are taken and 26 coefficients are obtained.

The DCT de-correlates the energies so that the diagonal covariance matrices can be used to model the features. Among the obtained 26 coefficients only the lower 12 are kept. The remaining higher 14 DCT coefficients represent fast changes in the filterbank energies and can degrade the recognition performance of the system. The 0th coefficient is added which is the sum over time of the power of the samples in each window. Also, delta features capture the change in each MFCC coefficient, and delta-delta features capture the change in each delta feature. So for each of the 13 coefficients (0th and 12 MFCC coefficients), 13 more delta features and another 13 delta-delta features are added, to get a total of 39 features. VOICEBOX [6], a toolbox for speech processing is used in GNU Octave [10] to extract the MFCC features from the recorded speech utterances. From each window of a speech utterance, 12 MFCCs and the 0th coefficient, along with their 13 delta and 13 delta-delta features are extracted, for a total of 39 features. All other parameters are set to their default values while extracting the MFCC features.

3.2. Feature Summarization

The duration of the speech utterance of a digit varies with each utterance. On extraction of MFCC for each utterance, we get a matrix of size m×39, where m is the number of windows and 39 features are extracted from each window of the digit utterance, as shown in Figure 2.

Figure 2. Generation of MFCC from digitized speech utterance.

Principal Component Analysis (PCA) is used for feature summarization, which reduces a collection of vectors represented as the m×39 matrix to a 1×39 vector. The process of feature summarization using PCA is described below.

In this feature summarization process, for a digit utterance, a matrix *A* of size m×39 is created, where *m* is the number of windows the speech utterance was divided into, and for each window, 39 MFCCs were generated. Equation (1) shows the matrix A.

$$A_{m\times 39} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,39} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,39} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,39} \end{pmatrix}, \quad (1)$$

The matrix *A* is normalised to obtain the matrix $\hat{A}$ ={ $\hat{a}_{ij}$ },where i=1 to m, j=1 to 39 and $\hat{a}_{ij}$ is calculated using the Equation (2):

$$\hat{a}_{ij} = \frac{a_{ij} - min_j}{max_j - min_j} \quad (2)$$

Where:

min_j and max_j are respectively the minimum and the maximum of elements in the j^{th} column of the matrix A. a_{ij} is the j^{th} element in the i^{th} row of the matrix A.

Then the covariance between two rows a_i and a_j of the matrix is computed using the Equation (3).

$$cov(a_i, a_j) = \frac{\sum(a_{ik} - \bar{a}_i)(a_{jk} - \bar{a}_j)}{m-1} \quad (3)$$

Here, a_{ik} and a_{jk} are the values of the *k*-th column of the rows a_i and a_j respectively, $\bar{a}_i$ and $\bar{a}_j$ are the mean of a_i and a_j respectively, and *m* is the number of rows in the matrix. Using the Equation (3), the following covariance matrix *C* of the normalized matrix $\hat{A}$ is computed, which is of size $m \times m$ (refer to the Equation (4).

$$C_{m\times m} = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,m} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,m} \end{pmatrix} \quad (4)$$

In the covariance matrix *C*, each entry $c_{i,j}$ is the covariance between two rows a_i and a_j.

After computing the covariance matrix *C*, the roots of the Equation (5) is computed, which give the eigenvalues of the covariance matrix *C*.

$$| C - \lambda I | = 0 \quad (5)$$

If there are *N* eigenvalues, then for each eigenvalue, a corresponding N×1 eigenvector can be found. As the size of the covariance matrix *C* is m×m, there will be *m* eigenvalues and, for each eigenvalue, an eigenvector of size m×1 will exist. The eigenvector corresponding to the largest eigenvalue is called the Principal Component *P*. From the Principal Component, the summarized feature vector *F* for the utterance is computed using the Equation (6).

$$F = P^T \times \hat{A} \quad (6)$$

Here *P* is the principal component, which is basically an eigenvector of size m×1. The size of the normalised matrix is m×39. Therefore the summarised feature vector *F* is of size 1×39, representing a summarized view of 39 features (refer to the Equation (7).

$$F = \begin{pmatrix} f_1 & f_2 & \cdots & f_{39} \end{pmatrix} \quad (7)$$

Hence, a summarized feature vector of size 1×39 is obtained for each speech utterance corresponding to a digit.

3.3. Preparation of Data Set

The summarized feature vector corresponding to each digit utterance is labelled with the corresponding class label. In Bengali, there are 10 digits. Hence one of 10

class labels is assigned to each summarised feature vector.

After assigning the class labels to the feature vectors, the labelled feature vectors are stored in a data file. Each row in the data file is of the form as shown in the Equation (8).

$$d_i = (f_1 \quad f_2 \quad \cdots \quad f_{39} C) \tag{8}$$

Here d_i is the *i*-th row in the labelled data file, containing 39 feature values - f_1 , f_2, ..., , f_{39}forming a feature vector and a class label *C* for the feature vector (*C* can be one of 10 possible classes).

3.4. Classifiers

The three classifiers that we have used for recognition of spoken digits are Multi-Layer Perceptrons, Support Vector Machines and Random Forests. We have used these three classifiers for our spoken digit recognition task because these classifiers have proven to be successful for many pattern recognition tasks. A brief description of each classifier is given below.

3.4.1. Multi-Layer Perceptrons

One efficient way of solving a complex problem is to decompose into simpler elements, in order to be able to manage it. Neural Networks provide us an approach which solves a complex problem by combining solutions of relatively simple sub-problems. Multi-Layer Perceptron (MLP) neural network consists of a network of neurons (called nodes) arranged in separate layers. There is an input and output layer, along with one or more number of hidden layers of neurons.

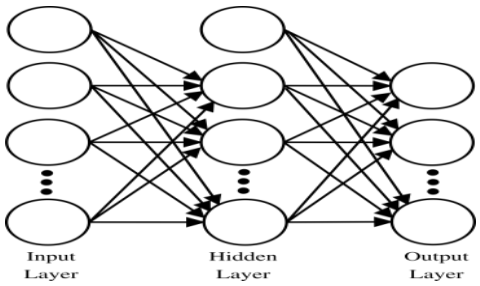


Figure 3. A Multi-Layer perceptron with one hidden layer.

According to the architecture given in Figure 3, the hidden nodes are used for solving the simpler elements of a complex problem and the solutions are combined at the output nodes. Every neuron in one layer is connected to all neurons in the next layer, as shown in Figure 3. Weights are assigned to the transition between neurons. Activation functions exist on neurons of all layers except the input layer. Activation functions decide what the output of a neuron will be, depending on what the input to the neuron was. The possible activation function for one neuron is shown in the Equation (9).

$$a_i^{(j)} = (\theta_{i0}^{(j-1)} x_0 + \theta_{i1}^{(j-1)} x_1 + \cdots + \theta_{in}^{(j-1)} x_n) \tag{9}$$

Here, $a_i^{(j)}$ is the activation of the *i*-th node in the *j*-th layer of the network. In the (j-1)-th layer, there are *n* nodes x_1, x_2, ..., x_n , plus a bias node, which is considered here as node x_o. Each node from the *(j-1)*-th layer is connected to the *i*-th node in the *j*-th layer. $\theta_{ik}^{(j-1)}$ Is the weight assigned to the transition from the *k*-th node in the *(j-1)*-th layer to the *i*-th node in the *j*-th layer. The activation at a node is therefore the summation of the products of the value input from the neurons of the previous layer, and the weights of the transitions. The output response of a neuron is a function *g(x)* for a given activation *x*. A sigmoid function is usually used for implementing *g(x)*.

Each layer of an MLP consists of several neurons. Since each neuron of one layer is connected to all neurons of the next layer, one layer in an MLP can be thought of as a function *f(x)* that maps an input vector *x* to an *n*-dimensional vector *y*, as shown in the Equation (10).

$$y = f(x) = (\, g_1(x)\, g_2(x) \ldots g_n(x)\,) \tag{10}$$

Here $g_i(x)$ is the output of the *i*-th neuron of the layer. If an MLP contains *d* number of layers, then it maps its input *x* to its output *z* through the composition of the function defined by each layer, as shown in the Equation (11).

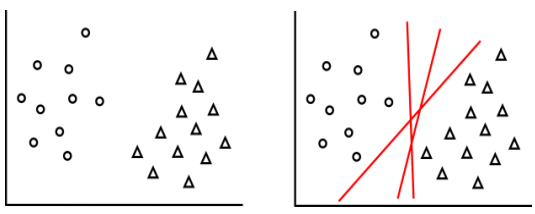
$$z(x) = f_d(f_{d-1}(f_{d-2}(\ldots(f_1(x))\ldots))) \tag{11}$$

Here f_i is the function represented by the*i*-th layer. In this study, an MLP with one hidden layer was implemented in GNU Octave. The Backpropagation algorithm is used to train the network.

3.4.2. Support Vector Machines

Support Vector Machines (SVMs) [18] is a popular machine learning technique and it has been successfully applied to many real-world classification tasks in various domains. Due to its solid mathematical background, high generalization capability and ability to find global and non-linear classification solutions, SVMs have been popular among the researchers.

SVM find the optimal decision boundary between two classes of data. Given two classes of data, there are an infinite number of choices on how to draw a decision boundary between the two classes of data, as shown in Figure 4.



a) Two classes of data. b) Possible hyperplanes.

Figure 4. Many possible separating hyperplanes.

If the data is *n*-dimensional, then an *(n-1)*-dimensional hyperplane needs to be drawn as the decision boundary between the two classes.

The objective of SVM is to draw a separating hyperplane that is maximally distant from both classes, as shown in Figure 5. The intuition behind this is that if a new data point arrives, it will be assigned to the class that it is closer to. SVM always computes the maximum-margin hyperplane for two classes of data, due to which SVM is called the maximum margin classifier.

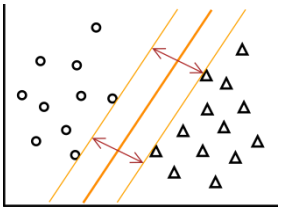


Figure 5. Maximum-margin hyperplane between classes.

This maximum-margin hyperplane can be drawn only if the data is linearly separable. If the data is not linearly separable, then kernel functions are used to map the data to a higher dimensional space where the data can be separated linearly. In that higher dimensional space, the maximum-margin hyperplane is constructed.

For implementation of our approach to recognition of spoken Bengali numerals using SVM, we have used the SVM tool included in a machine learning toolkit called WEKA [11].

3.4.3. Random Forest

Random forest [5] is an ensemble classifier that combines the predictions of many decision trees using majority voting to output the class for an input vector. Each decision tree participating in the ensemble process chooses a subset of features randomly to find the best split at each node of the decision tree. The method combines the ideas of *bagging* [4] and random selection of features. The random forest learning algorithm has two phases- training phase and testing phase. The training phase has the following steps**.**

For each of *N* decision trees to be built,

- Select a new bootstrap sample from training set.
- Grow an un-pruned decision tree on this bootstrap.
- While growing a decision tree, at each internal node, randomly select m_{try} predictors (features) and determine the best split using only these predictors.
- Do not perform pruning. Save the decision tree.

In the testing phase**,** the learned model is tested on each input test vector and a test vector is assigned the class label that is the mode of the class labels produced by all the individually trained decision trees.

For our experiments, we have used the *Random Forest* classifier included in Waikato Environment for Knowledge Analysis (WEKA).

We have used this algorithm for our recognition task for following several reasons:

- For many data sets, it produces a highly accurate classifier [7].
- It runs efficiently on large databases, performs well consistently across all dimensions.
- It generates an internal unbiased estimate of the generalization error.

4. Experiments and Results

For recognition of isolated Bengali spoken digits, the data is shuffled randomly, and 10 fold cross validation is performed. The overall recognition performance of a classifier is obtained by averaging the results over 10 folds. The various parameters associated with each classifier are tuned for obtaining the best results. We obtain the best results for our used classifiers with the parameter settings which are described below in detail. For our experiments, we have implemented MLP using GNU Octave and the SVM and Random Forests are chosen from the WEKA toolkit.

The neural network architecture that we have used has one hidden layer along with the input and output layers. The hidden layer contained 100 nodes. The input layer contained 39 nodes due to the dimensions of the feature vector being 39, and the output layer contained 10 nodes due to 10 classes representing the 10 different Bengali digits. Ten-fold cross-validation was done for 1000 iterations with a regularization parameter set to 0.1. All other parameters were set to default values. The accuracy is calculated as percentage of correctly classified digits. The results reported in this paper are obtained by averaging the accuracy over all 10 folds.

Table 2. Overall recognition performances of MLP, SVM and RF (Random Forest) on our dataset of Bengali isolated spoken digits.

Classifier	Accuracy (%)
Multi-Layered Perceptrons	89.5
Support Vector Machines	91.67
Random Forests	84.06

We have chosen SVM from WEKA with the kernel function set to the radial basis function. Among the parameters that were set for SVM, the cost parameter was set to 25, the tolerance of the termination criteria was set to 0.00001, gamma set to 0.07, the coefficient used was set to 1, and all other parameters were set to default values

Random Forests chosen from WEKA is configured by setting the number of trees to 105, and all other parameters to default values.

The overall recognition performances of the three classifiers are shown in Table 2. Table 2 shows that SVM outperforms both MLP and Random Forests for Bengali spoken digit recognition. The performance of MLP is also comparable to SVM.

The detailed digit-wise recognition performances of our used three different classifiers are shown in Tables 3, 4 and 5 respectively. From the results, it can be seen that both MLP and SVM outperform Random Forests. Recognition of digits 0 and 1 was much higher in accuracy compared to other digits. The digit-wise performance analysis shows that the lowest accuracy obtained while recognizing the digit 4. Compared to MLP and Random Forests, SVM gives the best results when recognizing the digit 4.

Table 3. Digit wise Recognition performance achieved by Multi-Layered Perceptron

Digit	Accuracy (%)
0 (shunyo)	95.55
1 (ek)	95.7
2 (dui)	90.5
3 (tin)	91.68
4 (char)	77.9
5 (panch)	92.06
6 (chhoy)	84.75
7 (saat)	87.55
8 (Ath)	86.18
9 (noy)	93.32

Table 4. Digit wise Recognition performance achieved by Support Vector Machines

Digit	Accuracy (%)
0 (shunyo)	96
1 (ek)	95.2
2 (dui)	90.73
3 (tin)	93.2
4 (char)	86.3
5 (panch)	91.2
6 (chhoy)	89.2
7 (saat)	91.9
8 (Ath)	88.6
9 (noy)	94.3

Table 5. Digit wise Recognition performance achieved by Random Forests.

Digit	Accuracy (%)
0 (shunyo)	89.5
1 (ek)	86.9
2 (dui)	79.7
3 (tin)	89.9
4 (char)	74.3
5 (panch)	83.2
6 (chhoy)	79.3
7 (saat)	85.2
8 (Ath)	82.5
9 (noy)	89

5. Conclusions

This study examines the recognition performance of a system that has as its input, isolated digits spoken by multiple speakers with different dialects. Our developed spoken digit recognition system has obtained more than 90% accuracy, which is highly encouraging. Since phoneme-level modelling is not required for implementing our proposed approach, our proposed spoken digit recognition approach can be easily applied to other languages like Hindi, Punjabi etc.

Since we isolate digits from spoken mobile numbers, an accurate module for automatically isolating digits from mobile numbers in continuous speech forms is required to apply our system to automatic recognition of spoken mobile numbers.

References

[1] Abushariah M., Ainon R., Zainuddin R., Elshafei M., and Khalifa O., "Arabic Speaker-Independent Continuous Automatic Speech Recognition Based on a Phonetically Rich and Balanced Speech Corpus," *The International Arab Journal of Information Technology*, vol. 9, no. 1, pp. 84-93, 2012.

[2] Ali A., Hossain M., and Bhuiyan N., "Automatic Speech Recognition Technique for Bangla Words," *International Journal of Advanced Science and Technology*, vol. 50, pp. 51-60, 2013.

[3] Alotaibi A., "Comparative Study of ANN and HMM to Arabic Digit Recognition Systems," *Engineering Sciences*, vol. 19, no. 1, pp. 43-60, 2008.

[4] Breiman L., "Bagging Predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.

[5] Breiman L., "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[6] Brookes M., *VOICEBOX: Speech Processing Toolbox for MATLAB*, http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html, Last Visited 2014.

[7] Caruana R., Karampatziakis N., and Yessenalina A., "An Empirical Evaluation of Supervised Learning in High Dimensions," *in Proceedings of the 25th International Conference on Machine Learning*, Finland, pp. 96-103, 2008.

[8] Davis H., Biddulph R., and Balashek S., "Automatic Recognition of Spoken Digits," *The Journal of the Acoustical Society of America*, vol. 24, no. 6, pp. 637-642, 1952.

[9] Ghanty K., Shaikh H., and Chaki N., "On Recognition of Spoken Bengali Numerals," *in Proceedings of Computer Information Systems and Industrial Management Applications*, Poland, pp. 54-59, 2010.

[10] Eaton W., Bateman D., and Hauberg S., *GNU Octave Version 4 3.0.1 Manual: a High-Level Interactive Language for Numerical Computations*, CreateSpace Independent Publishing Platform, 2009.

[11] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten H., "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10-18, 2009.

[12] Jurafsky D. and Martin H., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational*

Linguistics, and Speech Recognition, *2nd Edition*, Prentice Hall, 2008.

[13] Kumar K. and Aggarwal K., "Hindi Speech Recognition System using HTK," *International Journal of Computing and Business Research*, vol. 2, no. 2, 2011.

[14] Martin A., Charlet D., and Mauuary L., "Robust Speech/non-speech Detection using LDA Applied to MFCC," *in Proceedings of the 2001 IEEE International Conference on Acoustics, Speech and Signal Processing*, Salt Lake, pp. 237-240, 2001.

[15] Muhammad G., Alotaibi A., and Huda N., "Automatic Speech Recognition for Bangla Digits," *in Proceedings of the 12th International Conference on Computers and Information Technology*, Dhaka, pp. 379-383, 2009.

[16] Muhammad G. and Alghathbar K., "Environment Recognition for Digital Audio Forensics using MPEG-7 and Mel Cepstral Features," *The International Arab Journal of Information Technology*, vol. 10, no. 1, pp. 43-50, 2013.

[17] Othman Z., Abdullah N., Razak Z., and Mohd-Yusoff M., "Speech to Text Engine for Jawi Language," *The International Arab Journal of Information Technology*, vol. 11, no. 5, pp. 507-513, 2014.

[18] Vapnik V., *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[19] Zheng F., Zhang G., and Song Z., "Comparison of Different Implementations of MFCC," *Journal of Computer Science and Technology*, vol. 16, no. 6, pp. 582-589, 2001.

Avisek Gupta He has obtained an M.E. degree from Jadavpur University, Kolkata, India, and has previously obtained a B.Tech. Degree from Future Institute of Engineering and Management, Kolkata, India. His research interests include Speech Recognition, Information Retrieval, and Machine Learning.

Kamal Sarkar He received his B.E degree in Computer Science and Engineering from the Faculty of Engineering, Jadavpur University in 1996. He received the M.E degree and Ph.D. (Engg) in Computer Science and Engg. From the same University in 1999 and 2011 respectively. In 2001, he joined as a lecturer in the Department of Computer Science & Engineering, Jadavpur University, Kolkata, where he is currently a professor. His research interest includes Natural Language Processing, Machine Learning, Text Summarization, Text Mining, Speech Recognition.