International Journal of Speech Technology (2018) 21:233–250
https://doi.org/10.1007/s10772-018-9498-5

Manner of articulation based Bengali phoneme classification

Tanmay Bhowmik[1,2] · Shyamal Kumar Das Mandal[2]

Received: 29 April 2017 / Accepted: 4 March 2018 / Published online: 9 March 2018

Abstract
A phoneme classification model has been developed for Bengali continuous speech in this experiment. The analysis was conducted using a deep neural network based classification model. In the first phase, phoneme classification task has been performed using the deep-structured classification model along with two baseline models. The deep-structured model provided better overall classification accuracy than the baseline systems which were designed using hidden Markov model and multilayer Perceptron respectively. The confusion matrix of all the Bengali phonemes generated by the classification model is observed, and the phonemes are divided into nine groups. These nine groups provided better overall classification accuracy of 98.7%. In the next phase of this study, the place and manner of articulation based phonological features are detected and classified. The phonemes are regrouped into 15 groups using the manner of articulation based knowledge, and the deep-structured model is retrained. The system provided 98.9% of overall classification accuracy this time. This is almost equal to the overall classification accuracy which was observed for nine phoneme groups. But as the nine phoneme groups are redivided into 15 groups, the phoneme confusion in a single group became less which leads to a better phoneme classification model.

Keywords Phoneme classification · Manner of articulation · Phoneme confusion matrix · Deep neural network · Autoencoder

1 Introduction

Phoneme classification is one of the most important section of continuous speech recognition task. In the Bengali language, there are 16 stop consonants and four affricates. For the pronunciation of these phonemes, the vocal tract is blocked for a duration to stop the airflow (Hayes and Lahiri 1991) which is known as occlusion period of the stop consonants and the affricates. So the occlusion period is nothing but just like a silence and it is common to all the stop consonants and the affricates. All of these consonants are differed from each other based on only their transitory parts which are very small in duration in comparison to the occlusion period. So classification of phonemes in Bengali continuous speech is not a trivial task. Framewise classification of phoneme was performed by bidirectional Long Short Term Memory (LSTM) networks which are a complete gradient format of LSTM learning algorithm (Graves and Schmidhuber 2005). In this study, the bidirectional approach performed better than the unidirectional approach. The importance of contextual information in classification task is obtained from this experiment. Phoneme classification was performed to classify vowels with three different types of input analysis and altering numbers of radial basis functions (Renals and Rohwer 1989). A combined approach of large margin kernel method with the Bayesian analysis was applied for hierarchical classification of phonemes (Dekel et al. 2004). A Semicontinuous HMM (SCHMM) based system with state duration was used for significant improvement in phoneme classification task over the discrete and continuous HMM-based systems (Huang 1992). The error rate in classification process was dropped by 30 and 20% while using the SCHMM based systems comparing to the discrete and continuous HMM-based systems respectively.

In continuous Bengali speech, sometimes consecutive phonemes contain almost same coarticulatory information, so it becomes difficult to pronounce them separately and

✉ Tanmay Bhowmik
tanmaybhowmik@gmail.com

[1] School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India

[2] CET, Indian Institute of Technology Kharagpur, Kharagpur, India

distinguish from each other (Das Mandal 2007). Regarding this problem, we derive confusion matrix for all the recognized phonemes. Phoneme confusion is not only found in ASR, but it is very regular in Human also. Meyer et al. studies about phoneme confusion in Human Speech Recognition (HSR) and ASR (Meyer et al. 2007). Deng et al. designed a Convolutional Neural Network (CNN) framework using heterogeneous pooling to trade about phonetic confusion with acoustic invariance in TIMIT dataset (Deng et al. 2013). Xu et al. generates Expectation–Maximization (EM)-based phoneme confusion matrix for Spoken Document Retrieval (SDR) and Spoken Term Detection (STD) task (Xu et al. 2014). Priorly, Srinivasan and Petkovic, and Moreau et al. utilized phoneme confusion matrix to retrieve spoken document (Srinivasan and Petkovic 2000; Moreau et al. 2004). Phoneme confusion also improves speech recognition accuracy (Žgank et al. 2005; Morales and Cox 2007). Zhang et al. applied phoneme confusion matrix resolve the confusions in Mandarin Chinese due to different dialects (Zhang et al. 2006).

Different speech research groups found several advantages of using phonological features in speech recognition systems during last few decades. Goldberg and Reddy incorporated some of the acoustic features in Harpy and Hearsay-II system (Goldberg and Reddy 1976). Harrington made some use of them in consonant recognition (Harrington 1987). These were the type of systems that tried to utilize knowledge-based techniques to extract features and incorporated them into speech recognition systems. On a short time scale, the rapid rate of change of vocal tract causes coarticulation which is the blurring of acoustic features. Bitar and Espy-Wilson measured some signal properties like the energy in certain frequency bands, formant frequencies to define phonological features as a function of these measurements (Bitar and Espy-Wilson 1995a, b, 1996).

Ali et al. used auditory models with explicit analysis to detect different phoneme classes (Ali et al. 1998, 1999, 2000, 2001, 2002). They proposed an acoustic–phonetic feature and knowledge-based system to identify some features based on the place of articulation for certain segment of phonetic classes (Ali et al. 1999). They also classified the fricative and stop consonants using acoustic features (Ali et al. 1998, 2001). The fricative detection task was comprised of two sections, detection of voicing regions of speech and detection of the place of articulation. They achieved accuracy as 93% in voicing detection and 91% in place of articulation detection task. Overall fricative detection accuracy was 87% (Ali et al. 2001). The researchers executed their study on stop consonant detection task with initial and medial stop consonants of TIMIT corpus. They obtained stop consonant detection accuracy as 86% (Ali et al. 2001). Three features of voicing were analyzed as voicing during the closure, voice onset time (VOT), and duration of the closure. For robust detection of formants, they proposed average localized synchrony detection (ALSD) method (Ali et al. 2000, 2002). ALSD produced better performance in vowel recognition task with an accuracy of 81% for clean speech and 79% for noisy speech with 10 dB of Signal-to-Noise Ratio (SNR). Phonological features were also extracted from the speech signal using a bottom-up, rule-based approach and decoded into lexical words (Lahiri 1999; Reetz 1999).

King and Taylor conducted their experiments of phonological feature detection (King and Taylor 2000) by comparing the features of the sound pattern of English (SPE) with multivalued features. They also compared the Government phonology system with a set of structured primes (Harris 1994). They achieved 92% accuracy in case of SPE features whereas 86% for multivalued features. King et al. also compared the performance of HMM and ANN-based system in SPE and multivalued feature detection task and observed that ANN is performing better than HMM for both of SPE and multivalued feature detection task (King et al. 2000).

Frankel and King proposed a hybrid approach for articulatory feature recognition (Frankel and King 2005). They discussed the modeling of inter feature dependencies. Six features as manner, place, voicing, rounding, front-back, and static were used, and probabilities of each feature were evaluated by Recurrent Neural Network. The Artificial Neural Network (ANN) and Dynamic Bayesian Network based hybrid model performed better than the ANN/HMM-based hybrid system in probability estimation of the features (Frankel and King 2005).

Automatic Speech Attribute Transcription (ASAT) was another ASR model with a bottom-up structure that first observed a group of speech attributes and combined them for linguistic validation (Lee et al. 2007). The goal of ASAT was to incorporate different information related to speech waveform, linguistic, spectral, and temporal knowledge in the speech recognition system to improve the performances of existing ASR systems. The group of information is collectively known as the speech attributes. The speech attributes are not only important to deliver high performance in speech recognition but also useful for many applications like speaker recognition, speech perception, speech synthesis, language identification, etc. (Lee et al. 2007). ASAT model tried to combine different knowledge sources in the bottom-up structure. ASAT was also applied to various tasks as rescoring of HMM detection output (Siniscalchi and Lee 2009), estimating of speaker specific height and vocal tract length from speech sound (Dusan 2005), recognition of continuous phoneme (Siniscalchi et al. 2007), cross-language attribute detection and phoneme recognition using minimal target specific training data (Siniscalchi et al. 2012), and spoken language recognition (Siniscalchi et al. 2009).

Application of Deep Neural Network (DNN) is a recent trend in ASAT project. Siniscalchi et al. developed an

MLP-based, bottom-up stepwise knowledge integration technique in large vocabulary continuous speech recognition (LVCSR) (Siniscalchi et al. 2011) and replaced the MLP method with DNN (Siniscalchi et al. 2013) to achieve superior performance. DNN based speech recognition systems provided good results in phoneme (Mohamed et al. 2010, 2012) and word recognition (Dahl et al. 2011; Seide et al. 2011; Yu et al. 2010). About the ASAT model, a group of speech attribute detectors was designed to identify place and manner based phonological features (Yu et al. 2012; Siniscalchi et al. 2013). A DNN was modeled then to combine all the outputs of the attribute detector together and produce phoneme posterior probability.

In this study, first, the Bengali phonemes are recognized and classified form continuous Bengali speech and the overall phoneme confusion matrix is derived. Then the phonological features are incorporated into the classification module to obtain better phoneme classification result. A DNN based classification module has been developed where the DNN based model is pre-trained by a stacked autoencoder. Three autoencoders are stacked to form the deep network.

Described method of manner based phoneme classification will be used in prosodic and phonological feature based Bengali speech recognition system. It is a multilevel approach for speech recognition. In the first stage, the continuous speech signal is broken into sub-segments (prosodic word) based on prosodic parameters (F0 contour). After that, each prosodic word is labelled based on the manner of articulation of phoneme to generate some pseudo word representation of the prosodic word (Bhowmik 2017). A Lexical Expert System (Das Mandal 2007) will be used offline for the classification of the pseudo words during the continuous speech recognition procedure.

2 Speech material

This present study deals with the Bengali language, the part of the Indo-Aryan (IA) or Indic group of languages which is the dominant language group in the Indian subcontinent and a branch of Indo-European language family also. Bengali is one of the most important Indo-Aryan (IA) languages. It is the official state language of the Eastern Indian state West Bengal and the national language of the country Bangladesh. Bengali is the fifth largest spoken languages in the world with nearly 250 million speakers (Lewis et al. 2016). In India, most of the Bengali speaking population is found in West Bengali (85%), Tripura (67%), Jharkhand (40%), Assam (34%), Andaman and Nicobar Islands (26%), Arunachal Pradesh (10%), Mizoram (9%), and Meghalaya (8%) (online census data). Dialect wise Bengali language is divided into two main branches; eastern and western. Eastern branch is mainly used in Bangladesh while the western branch Bengali is mostly used in West Bengal. The western branch Bengali is further clustered into Rarha, Varendra, and Kamrupa based on dialects. These dialects are most commonly used in southern, north-central, and northern region of West Bengal respectively. Rarha is further subdivided into South-Western Bengali (SWB) and Standard Colloquial Bengali (SCB). The SCB is spoken around Kolkata (Bhattacharya 1988). The present study is based on Standard Colloquial Bengali (SCB).

3 Phoneme set of Bengali

Altogether 32 consonants and seven vowels are found in Bengali phoneme set (Chatterji 1926). The phoneme set of SCB is shown in Fig. 1. The consonants /e̯/ and /w/ are treated as the semivowels (Bhattacharya 1988). There exist total 20 stop consonants in Bengali (Chatterji 1926). The phonemic variation among the stop consonants depends only on the transitory parts of the corresponding phonemes. The duration of the occlusion part for the unvoiced stop consonant is much higher than the length of the transitory part for the stop consonants. So these phonemes only has difference in their transitory part with smaller duration. The spectrogram of four unvoiced phoneme /k/, /ʈ/, /t̪/, /p/ is found in Fig. 2, from where it is found that the occlusion period spans over a significant portion of the total duration of each unvoiced stop consonant. The transitory part comes after the occlusion period and it spans over a very small duration. So sometimes, it becomes difficult to distinguish the unvoiced stop consonants. The same goes for voiced consonants. For voiced stop consonants, the voiced segment of the corresponding phoneme is expanded for the higher duration than the transitory part. As a result, confusion occurred between voiced stop consonants also in the case of continuous speech recognition. In case of affricates, the sound of phoneme /ʧ/ is very close to phoneme /ʃ/ because both have similar place and manner of articulation as post-alveolar and unvoiced respectively. The phoneme /ʧ/ starts with a complete stoppage of airflow at the post-alveolar point of articulation. From the IPA notation, it might be thought that /ʧ/ has pronunciation similarity to that of /t̪/ in that stoppage segment. However, /ʧ/ is pronounced more at post-alveolar point of articulation. As it is also an unvoiced, stop consonant, so it also has occlusion period which spans a significant duration. Sometimes phoneme recognizers treat this occlusion period as silence and the rest of the section of /ʧ/ is recognized as /ʃ/. This results a confusion between /ʧ/ and /ʃ/ in Bengali continuous speech. In Bengali there are four affricates /ʧ/, /ʧʰ/, /ʤ/ and /ʤʰ/. The segments of all the affricates are shown in Fig. 3. Apart from this, all the phonemes from nasal, lateral, trill and tap/flap manners are voiced phonemes whereas the fricatives fall in unvoiced class. All the place

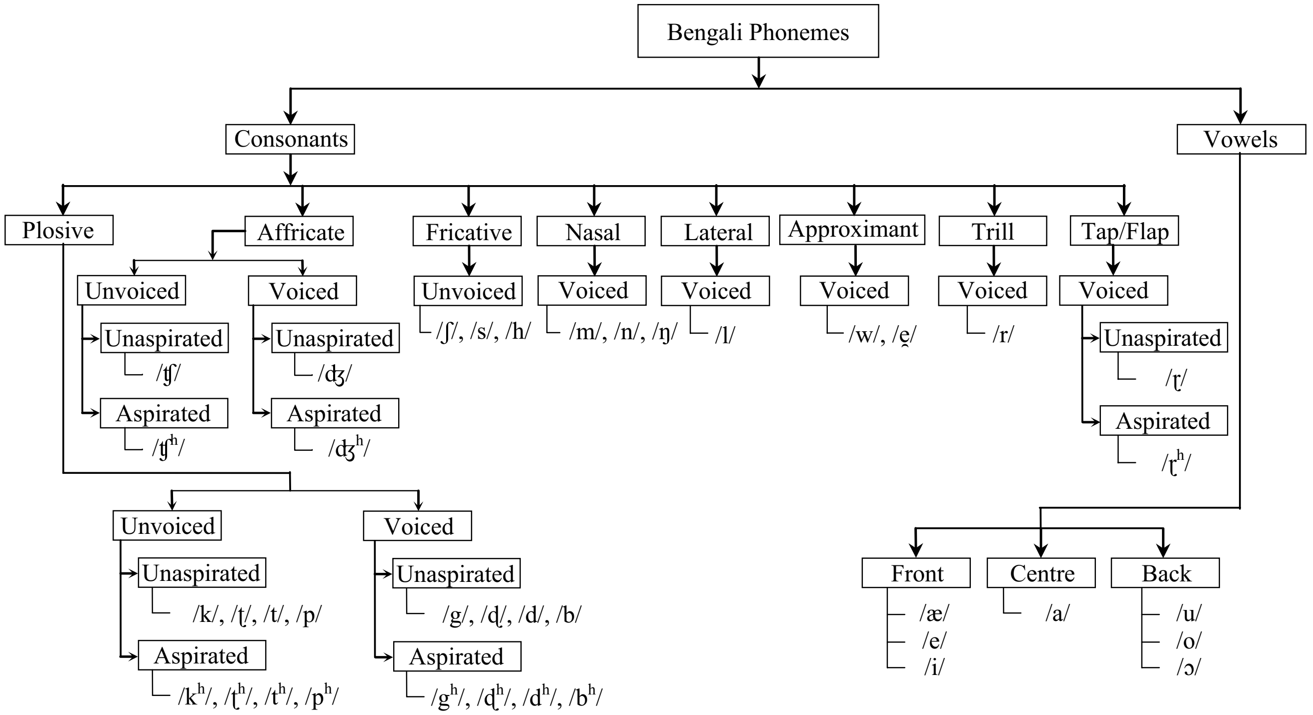


Fig. 1 Phoneme set of Bengali

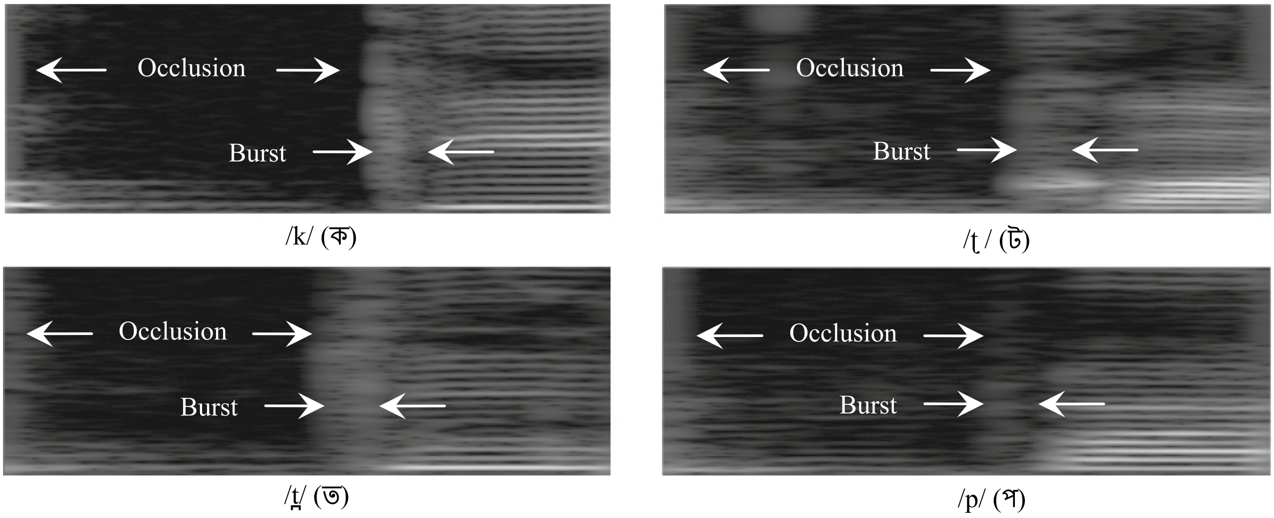


Fig. 2 Occlusion period of unvoiced, unaspirated, stop consonants

and manner of articulation and the corresponding Bengali consonants are given in Table 1.

There are seven vowels in the Bengali phoneme set. In Fig. 1 the vowels are categorized on the basis of tongue position. The vowels are also classified based on the lip roundness. Some of the vowels like /u/, /ɔ/ and /o/ are rounded and the remaining four vowels are unrounded.

Phonological features are categorized in term of place and manner of articulation of different speech sound. The place where the obstruction occurs is called the place of articulation. In Human speech production system the primary source of energy for speech production is the air in lungs. At the time of articulation, the lungs with the aid of diaphragm and other muscles force the air to pass through the glottis between the vocal cords and the larynx to the three primary cavities of the vocal tract, the pharyngeal, oral, and nasal cavities. The airflow exits through the mouth from the oral cavity whereas the same occurs through the nose from the

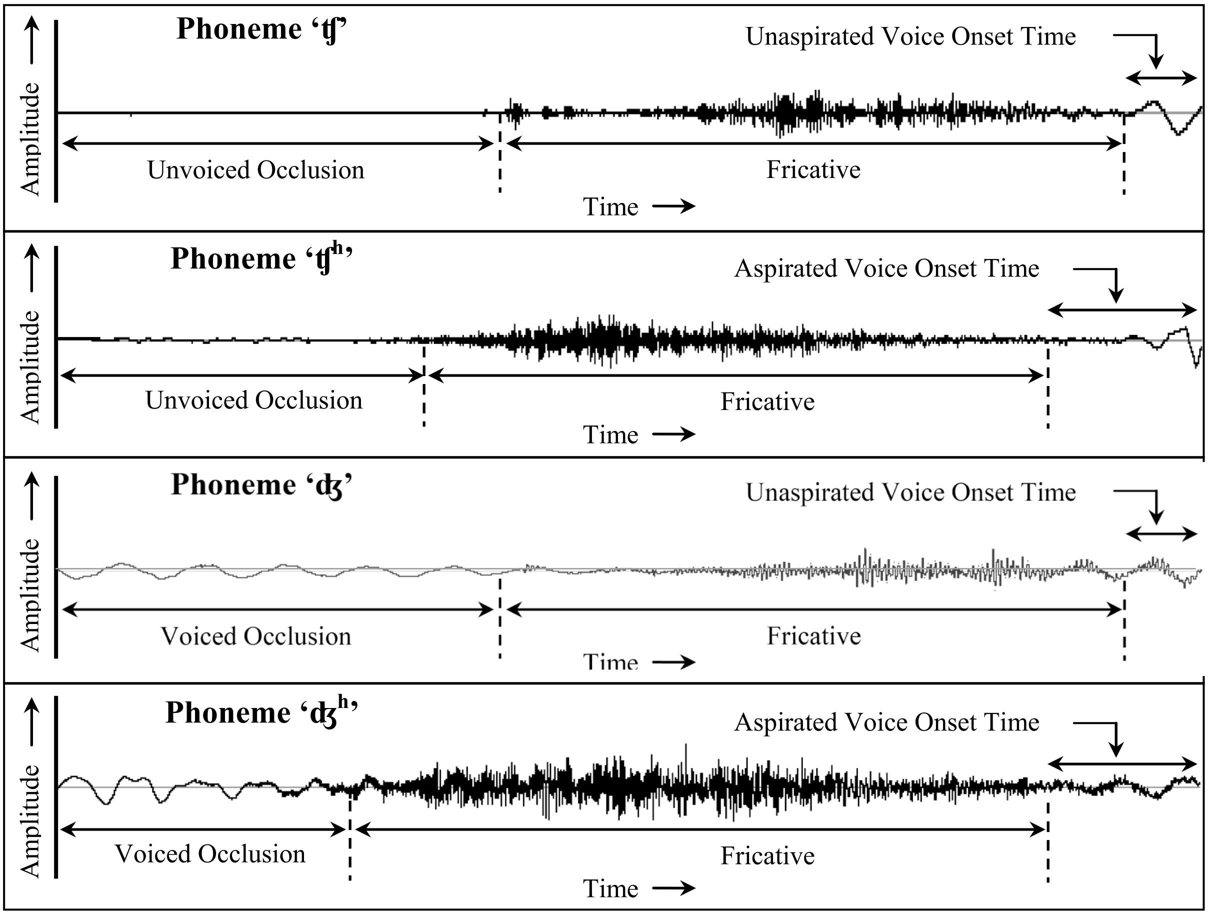


Fig. 3 Example of segments of different affricates

nasal cavity. So depending on the state of the vocal cord speech sound are two type voiced and unvoiced.

The vocal tract is the above air passages of the larynx. The vocal tract consists of the pharynx, oral cavity within the mouth, and the nasal cavity within the nose. The sections of the vocal tract, which are utilized to produce different speech utterances, are called articulators. The articulators that form the lower surface of the vocal tract often move towards those that form the upper surface. There are different principle parts of the upper surface of the vocal tract. Those are the lip, teeth, alveolar ridge, hard palate, soft palate, velum, and uvula. Similarly, there are different parts of the lower surface; the lip and the blade, the tip, front, centre and back of the tongue. During the articulation of the consonants, the airstreams through the vocal tract must be obstructed in some way. The manner of articulation is concerned with airflow i.e. the paths it takes and the degree to which vocal tract constrictions impede it.

In the case of vowel sound production, none of the articulators come very close together, and the passage of the airstream is relatively unobstructed. Thus vowels are described regarding three factors (a) the height of the raised part of the tongue, (b) the front-back position of the tongue, (c) the roundness of lip.

4 Methodology

In this experiment, first, the phonemes are recognized and classified from Bengali continuous speech. Then the phonological features are incorporated into the classification module to enhance the system performance. To execute this, the manners of articulation are detected and classified, and the phonemes are classified based on the detected manners to produce better performance. The basic block diagram is shown in Fig. 4.

Table 1 Phonological features and associated Bengali consonants in IPA notation (Bhattacharya 1988; Hayes and Lahiri 1991)

Attributes	Phonemes
Place of articulation	
Velar	/k/, /k^h/, /g/, /g^h/, /ŋ/
Post-alveolar	/ʧ/, /ʧ^h/, /ʤ/, /ʤ^h/, /ɽ/, /ɽ^h/, /ʃ/
Alveolar	/ʈ/, /ʈ^h/, /ɖ/, /ɖ^h/, /n/, /l/, /r/, /s/
Dental	/t̪/, /t̪^{h}/, /d̪/, /d̪^{h}/, /n/
Bilabial	/p/, /p^h/, /b/, /b^h/, /m/
Glottal	/h/
Palatal	/ɳ/, /e̯/
Manner of articulation	
Plosive	/k/, /k^h/, /g/, /g^h/, /ʈ/, /ʈ^h/, /ɖ/, /ɖ^h/, /t̪/, /t̪^{h}/, /d̪/, /d̪^{h}/, /p/, /p^h/, /b/, /b^h/
Affricate	/ʧ/, /ʧ^h/, /ʤ/, /ʤ^h/
Fricative	/ʃ/, /s/, /h/
Nasal murmur	/m/, /n/, /ŋ/, /ɳ/
Lateral	/l/
Trill	/r/
Retroflex	/ʈ/, /ʈ^h/, /ɖ/, /ɖ^h/, /ɳ/
Flap/tap	/ɽ/, /ɽ^h/
Approximant	/e̯/, /w/
Voicing	
Unvoiced	/k/, /k^h/, /ʧ/, /ʧ^h/, /ʈ/, /ʈ^h/, /t̪/, /t̪^{h}/, /p/, /p^h/, /ʃ/, /s/, /h/
Voiced	/g/, /g^h/, /ʤ/, /ʤ^h/, /ɖ/, /ɖ^h/, /d̪/, /d̪^{h}/, /b/, /b^h/, /m/, /n/, /ŋ/, /ɳ/, /l/, /r/, /ɽ/, /ɽ^h/, /e̯/, /w/
Aspiration	
Unaspirated	/k/, /g/, /ʧ/, /ʤ/, /ʈ/, /ɖ/, / t̪/, /d̪/, /p/, /b/
Aspirated	/k^h/, /g^h/, /ʧ^h/, /ʤ^h/, /ʈ^h/, /ɖ^h/, /t^h/, /d^h/, /p^h/, /b^h/

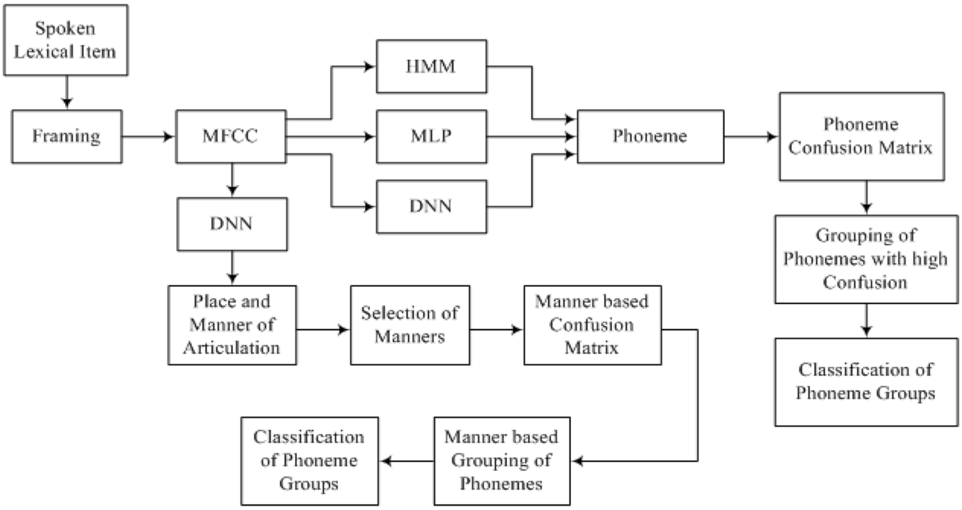

Spoken Lexical Item
Framing
MFCC
HMM
MLP
DNN
Phoneme
Phoneme Confusion Matrix
DNN
Grouping of Phonemes with high Confusion
Place and Manner of Articulation
Selection of Manners
Manner based Confusion Matrix
Classification of Phoneme Groups
Classification of Phoneme Groups
Manner based Grouping of Phonemes

Fig. 4 Proposed basic block diagram of the system

4.1 Detection and classification

Detection and Classification are two different aspects of ASR. In detection process, sequential, frame-by-frame processing of speech waveform is performed first, and the posterior probability is calculated. The possibility of detecting a particular feature as present is determined based on the probability value is above a predecided threshold or not. In this method, there is no need to have any preliminary knowledge about sentence (Hou 2009) but the detector needs to be trained with entire training data to achieve good detection result.

In the classification task, first the speech signal is segmented, and then the segment is classified within some group of speech features or phonemes. In this thesis, seven numbers of consecutive speech frames have been considered to form a context of speech segment. The duration of each speech frame was 25 ms. 10 ms of frame shift was used to prevent data loss. It needs to train the classifier with the speech parameters related to the set of features inside a class.

4.2 Evaluation metric

Phoneme Error Rate (PER) is used to estimate the recognition accuracy for the entire phoneme recognition model. PER is calculated as PER = [(S + I + D)/N] × 100%, where S, I, D, N represents the number of substitution errors (phonemes being recognized as another), number of insertion errors (insertion of wrong extra phonemes), number of deletion errors (correct phonemes overlooked), and total number of phonemes respectively.

The classification accuracy of a phoneme class is calculated with the numbers of correctly classified samples divided by the total number of samples. If a particular phoneme class is considered as 'P_i' then the classification accuracy for P_i will be measured as P_i = (t/n) × 100% where t stands for number of correctly classified samples and n stands for total number of sample cases. The confusion matrix is generated between predicted class and actual class with all the true positive (TP), false positive (FP), false negative (FN), and true negative (TN) values. Precision and Recall values for all the phonemes are calculated. Precision is measured as the percentage of predicted items which are correct, and it is calculated as TP/(TP + FP). The recall is obtained as the proportion of the exact items which are predicted and calculated as TP/(TP + FN). The f-score combines the precision and the recall values. It is used to calculate the weighted average of the precision and the recall. The overall accuracy of the system is calculated as (TP + TN)/(TP + TN + FP + FN) (Fawcett 2006). The conditions related to the confusion matrix are shown in Table 2.

5 Deep neural network

A DNN is a feed-forward, artificial neural network consists of more than one layer of hidden units between its input and output. In general, each hidden unit uses a logistic function to map its entire input to output layer (Hinton et al. 2012):

$$Y_j = logistic\left(x_j\right) = \frac{1}{1 + e^{-x_j}}$$

where,

$$x_j = b_j + \sum_i y_i w_{ij}$$

In the above equations, b_j is the bias of unit j, i is the index of the input layer, and w_{ij} is the weight of the connection from unit i to unit j. In our experiment, we are about to recognize multiple phonemes. For multi-class classification, the total input x_j of unit j is converted into a class probability p_j in the output layer with the use of softmax nonlinearity (Hinton et al. 2012).

$$p_j = \frac{\mathrm{e}^{\mathrm{x}_j}}{\sum_k \mathrm{e}^{\mathrm{x}_k}}$$

here k is an index of all classes.

Discriminative training is found in the case of DNN with backpropagation of derivatives of the cost function which measure the deviation of actual output from the target output for each training case (Rumelhart et al. 1986). At the time of softmax normalization, the natural cost function C acts like the cross entropy between target probability and softmax output.

$$C = -\sum_j d_j \log p_j$$

where d, the target probability usually takes the value of one or zero, and this is the supervised information used to train the DNN classifier. The softmax output is denoted by p. Usually, for the large training sets, computation of derivatives on a small, random mini batch of training cases is more efficient than the whole training set before the update of weights in gradient scale (Hinton et al. 2012). The biases are updated by considering them as the weights on connections coming

Table 2 Outcomes of confusion matrix

		Target class	
	Total population	Target condition positive	Target condition negative
Predicted class	Predicted condition positive	True positive (TP)	False positive (FP)
	Predicted condition negative	False negative (FN)	True negative (TN)

		Precision = TP/(TP + FP), Recall = TP/(TP + FN)
		F-score = (2 × Precision × Recall)/(Precision + Recall)
		Overall accuracy = (TP + TN)/(TP + TN + FP + FN)	

from units with a state of one. DNNs with numbers of hidden layers are difficult to optimize. The initial weights of a fully connected DNN are given small random values to prevent from having an exactly same gradient to all the hidden units in a layer. Glorot and Bengio state that gradient descent from a starting point which is very close to the origin is not the best way to find a better set of weights, and to obtain that initial scales of weights are deliberately chosen (Glorot and Bengio 2010). Due to the presence of large numbers of hidden layers and hidden units, the DNNs are a very adjustable model with huge numbers of parameters. That is why DNNs are capable of designing complex and non-linear relations between inputs and outputs. This ability is crucial in better acoustic modeling (Hinton et al. 2012). The deep model needs generative pre-training to ensure effective training of the complex and non-linear relationship.

5.1 Generative pre-training

The idea of generative pre-training is to learn one layer of input features at a time where the feature states that layer will act as the data to train the next layer. After this pre-training phase, the deep-structured network finds a better starting point for the discriminative fine-tuning phase. During the fine-tuning, the backpropagation procedure through the DNN slightly modifies the weights which are observed in pre-training phase (Hinton and Salakhutdinov 2006). The generative pre-training finds a set of weight vectors through which the fine-tuning process can have rapid progress and pre-training also reduces overfitting (Larochelle et al. 2007). In this study, the Stacked Autoencoder (SAE) is used for pre-training purpose.

5.2 Classical autoencoder

Figure 5 depicts a basic Autoencoder (AE) structure. It consists of three layers as an input layer, hidden layer, and output layer (Vincent et al. 2010). The AE first takes the input and maps it to a hidden representation with a nonlinear function. In Fig. 5, the input data is represented by x. When the input is mapped to the hidden layer with a nonlinear function, the input is encoded in the hidden layer, and the output y is generated. W represents the weight matrix, and b stands for the input bias. In this study, the Rectified Linear Unit (ReLU) is used as the nonlinear function. So the generated output in the hidden layer is represented as

$$y = s(Wx + b)$$

where ‘s’ stands for the nonlinear function. So in this experiment s will be

$$s = ReLU\left(x_i\right) = \max(0, x_i)$$

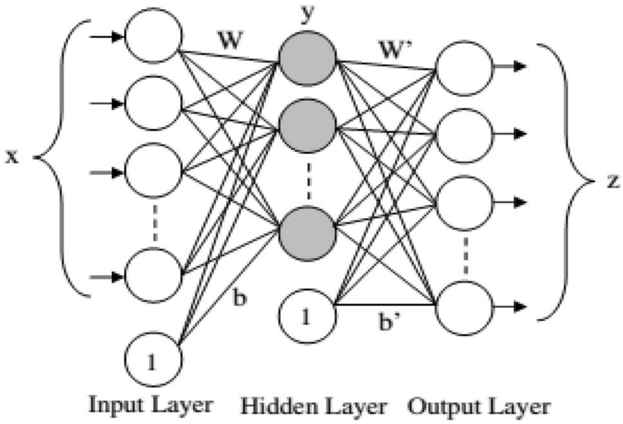


Fig. 5 Basic structure of an autoencoder

Now the output ‘y’ acts as the input in the next level to produce the output ‘z’ of the system. So z is represented as

$$z = s(W'y + b')$$

In an AE, the target output is the input itself. Here, z is the reconstructed form of input x. z is derived by decoding y which is an encoded form of input x. So an AE consists of an encoder and a decoder in a single structure, and that is why it is called as an Autoencoder. The reconstructed version z is not an exact representation of x, rather z is a probabilistic distribution that can produce x with high probability. As a result, a reconstruction error is generated, and a loss function measures that. For binary input (0, 1), the loss function is represented as

$$L(x, z) = -\sum_k x_k \log z_k + (1 - x_k)\log(1 - z_k)$$

When the input is real valued, the loss function is measured as the sum of squared error

$$L(x, z) = \frac{1}{2}\sum_k (z - x)^2$$

A traditional Autoencoder has the parameter set consists of W, b, W′, b′. Here, W′ is the weight matrix of the decoding process and $W' = W^T$. This is cited as tied weights (Bengio 2009). This parameter set needs to be optimized to minimize the average reconstruction error. The encoded form y is considered as a lossy compression of x. So for all x, it can not be a very good compression. Due to optimization, it becomes a good compression technique for training.

The encoded form of y is considered as a distributed representation which can capture the coordinates along the important factors related to the variation of data like the projection of principal components would handle the main variational factors of input data (Yu and Deng 2014). If the mean squared error criterion is used to train a network with one hidden layer consists of k hidden units, then the k number of units learn to project the input in the span of first k numbers of principal units of the data. In a case of a nonlinear hidden layer, the AE acts differently from PCA. AEs can capture multimodal aspects of input distributions. The separation from PCA is more important when it needs to be stacked up and forms Stacked Autoencoder.

5.3 Denoising autoencoder

In a Denoising Autoencoder (DAE) model, the Autoencoders are trained to reconstruct the input from a corrupted version of the same. The structure of a DAE is same as a traditional AE model. The only difference is that the noisy input data is fed into the input layer of the Autoencoder. So in reference to Fig. 5, the input data x becomes $x + r$ where r is noise. Hence, the output in the hidden layer will be $y = s[W(x + r) + b]$ and from this, the reconstruction of uncorrupted input data can be possible as $z = s(W'y + b')$. Here the parameter set (W, b) and (W′, b′) are trained to minimize the reconstruction error instead of minimizing the loss function $L(x + r, z)$. The output z needs to be as close as possible to the uncorrupted input x.

In general, a DAE performs two tasks. It tries to preserve the information about the input and tries to undo the effect of corruption process into the input of the AE by measuring the statistical dependencies between the inputs (Vincent et al. 2008).

5.4 Stacked autoencoder

The Autoencoders are stacked to create the deeper structure, the Stacked Autoencoder (SAE). The output from one AE is fed as input into next AE. Unsupervised training is done for one layer at a time to minimize the reconstruction error (Bengio 2009). After the completion of generative pre-training of all the layers, the supervised fine-tuning phase starts for the deep network. Pre-training is performed by following the efficient approximation learning algorithm (Vincent et al. 2010). Pre-training process adjusts hidden weights in a way such that the system keeps away from local minima at the time of supervised fine-tuning. During pre-training, each of the AE layers is learned using the one step Contrastive Divergence (CD-1) algorithm (Hinton et al. 2006).

Stacked Denoising Autoencoder (SDAE) is an extension of Stacked Autoencoder. With the availability of both the noisy and clean speech data, an SDAE can be pre-trained and fine-tuned by noisy and clean speech features respectively. The SDAE can be used to remove noise from speech and to reconstruct various speech features with sufficient phonetic information (Feng et al. 2014).

Random noises are included in the input data in SDAE architecture. This leads to several advantages for the model. It helps the model to avoid to learn the trivial identity mapping function. Due to the addition of random noise, the learning of the model would be robust, and it can handle the same kind of distortions in the test data. The chance of overfitting can be reduced since the corrupted input increases the training size (Deng and Yu 2013).

5.5 Fine-tuning

The final fine-tuning of the deep network is executed by incorporating a softmax or multinomial regression layer on top of the network. The softmax regression is used when it needs to classify between multiple classes. The training of softmax layer is executed in supervised way (Hinton and Salakhutdinov 2006). In this experiment, softmax regression is necessary as the classification is evaluated between 49 phoneme classes. The output of DNN is improved by performing backpropagation on the whole multilayer network. This process is referred to as Fine-tuning. In this process, the entire network is retrained on the training data in a supervised manner (Hinton et al. 2012).

6 Experimental setup

6.1 Speech corpus

The Bengali speech corpus from Centre for Development of Advanced Computing (CDAC), India has been used for continuous spoken Bengali speech data (Mandal et al. 2005). That is a high quality Bengali speech corpus labeled at both the phone level and the word level. Total 13 types of sentences, read by ten speakers, and are randomly selected from the speech corpus for this study. There are six male speakers and four female speakers aged about 12–53 years, and their speech rate varies from 4 to 6 syllables per second.

The system is also trained in the English language to validate the experiment. For training in English, the TIMIT corpus (Garofolo and Consortium 1993) has been used. This corpus consists of a total of 6300 sentences. The sampling frequency of all the recorded sentences was 16,000 Hz. Due to computational limitation, a subset was selected from TIMIT corpus. The details about the selected speech corpora about duration and number of sentences are mentioned in Table 3.

6.2 Input features

The Mel Frequency Cepstral Coefficient (MFCC) features have been used as input features for this study. The Mel frequency scale is better resembled to human auditory system compared to other parameters (Davis and Mermelstein 1980). 12 MFCC features plus the 0th cepstral coefficient is

Table 3 Duration and number of sentences in selected speech corpora

Corpus	Duration of sentences (in min)				Number of sentences			
	Selected subset	Training	Validation	Testing	Selected subset	Training	Validation	Testing
CDAC	90	66	14	10	1480	1130	200	150
TIMIT	85	65	10	10	1400	1100	180	120

Table 4 Detail information about MFCC extraction

Sampling frequency (f_s)	16,000 Hz
Window type	Hamming
Analysis frame duration	25 ms
Analysis frame shift	10 ms
Pre-emphasis	Yes
Number of filters in filterbank	Floor $(3\times\log(f_s))=12$
Number of Cepstral coefficients	13

computed for each frame. 13 numbers of D and DD coefficients are also computed as $\Delta(n) = [x(n+1)–x(n–1)]$ and $\Delta\Delta(n) = [0.5x(n+1)–2x(n)+0.5x(n–1)]$ respectively to get a complete 39 dimensional input dataset. A contextual representation of seven frames per context has been taken to prevent data loss; three frames in the back and three frames ahead. In the absence of a speech frame, zero is appended to complete the context. Due to this, input data dimension becomes $39\times7=273$. The input feature detail is described in Table 4. Here pre-emphasis is executed to flatten the magnitude spectrum and balance the high and low frequency components. In general, it is a first order, high pass filter with the time domain equation $y[n]=x[n]–\alpha x[n–1]$ where y[n] is the output, x[n] is the input, and $0.9 \le \alpha \le 1.0$. Default values are kept for other parameters. The matlab toolkit 'voicebox' is utilized to use the 'melcepst()' function (MATLAB 2015) for mfcc feature extraction.

6.3 Proposed model

A recognition model has been designed using DNN to execute the phoneme recognition process on Bengali continuous speech. A functional block diagram of this DNN-based model is given in Fig. 6. While considering the deep architecture, during pre-training, denoising autoencoders are trained. In this experiment, three autoencoders are stacked to form the deep architecture. For each autoencoder, the Rectified Linear Unit (ReLU) function has been used as the nonlinear activation function. 200 hidden units are used in each hidden layer. In output layer, classification has been done between 49 phonetic classes. In Table 1, all the Bengali phonemes are represented. In continuous Bengali speech there is almost no difference in pronunciation between the phoneme 'ɽ' and 'ɽ^{h}'. That is why these two phonemes have been considered as a single class in this experiment. Apart from this, we found 'silence' as another class as each continuous spoken sentence possesses some silent regions in the initial and final position of the sentence. Continuous utterances also may have some small pauses in between utterances. This is considered as another class named 'short pause'. There exist diphthongs in Bengali. A diphthong is a vowel–vowel combination. Two of them are found in the selected sentences of the Bengali speech corpus. So in the output layer, there are total 49 nodes have been considered. The learning rate is kept as one, and the mini batch size is fixed at 120. The input zero masked fraction value is taken as 0.5. In input and output layer, there are the MFCC features and the posterior probabilities respectively. A deep learning toolbox has been used for this recognition model (Palm 2012).

6.4 Hardware

A DELL Precision T3600 workstation is used for this experiment. This workstation is a six-core computer with a CPU clock speed of 3.2 GHz, 12 MB of L3 cache memory and

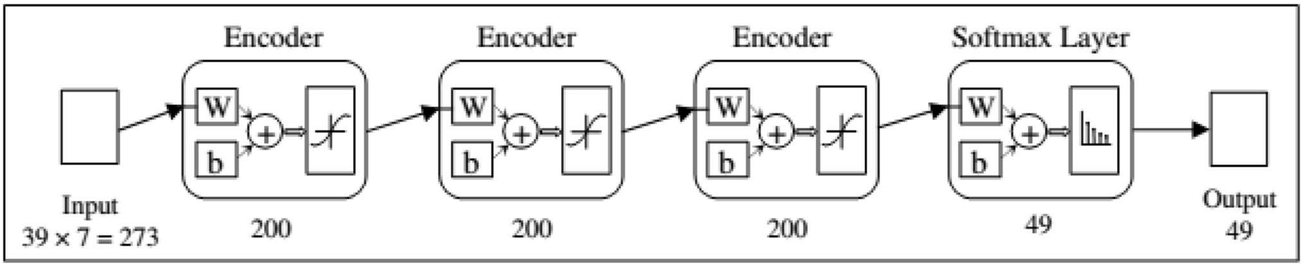


Fig. 6 Functional block diagram of the DNN-based system

64 GB DDR3 RAM and a NVIDIA Quadro 4000 General Purpose Graphical Processing Unit (GPGPU).

7 Results and discussion

The overall recognition and classification results are shown in Tables 5 and 6 respectively. The performance for training, validation, and testing for both the CDAC and TIMIT corpus are depicted here. The leave-one-out cross-validation method has been used for this study as it is reasonably unbiased though it may suffer from some high variance in some cases (Efron and Tibshirani 1997). In both the cases of CDAC and TIMIT corpus, the deep-structured model performed better than the baseline methods.

The details about phoneme classification result are represented in Table 7. The overall classification result with the precision, recall and f-score values of all the Bengali phonemes are reported here. Most of the phonemes are classified with good precision and recall values. Analysis and observation of the speech corpus reveal that some of the phonemes occur fewer numbers of times. Classification result for those phonemes is not so satisfactory. Either the precision or the recall values are found to be low for them which results in a less f-score result. Some of the nasal vowels are not found in the Bengali sentences of the speech corpus. So no results are produced for them.

Based on the results, the phoneme confusion matrix (PCM) for all the 49 phoneme classes is generated. Some segments of the overall PCM is depicted in Fig. 7.

From the confusion matrices of Fig. 7a, b it can be observed that almost each unaspirated consonant has confusion with the aspirated one irrespective of voiced or unvoiced. This is because in Bengali continuous speech sometimes it becomes quite difficult to separate the unaspirated and aspirated stop consonants. The duration of glottal aspiration is very less for continuous speech data, and that is why sometimes it becomes difficult to separate them. As a result sometimes the Bengali word 'sat̪'(in English: seven) is confused with the word 'sat̪^{h}'(in English: together with). There are more examples like this also.

Regarding all the unvoiced phonemes it is observed from the spectrogram that each of them contains an occlusion period which is a silence due to the blockage of nasal and air passage of mouth (Mandal et al. 2011). The occlusion is nothing but silence. The difference between these phonemes are observed only in the transitory part that comes after the occlusion, and it spans along a very little duration. As a result, sometimes it becomes difficult for a recognizer to find differences between them in continuous speech.

In continuous Bengali speech it is hard to pronounce separately trill /r/ and the flap/tap consonants /ɽ/, /ɽ^{h}/. That is why the phoneme 'ɽ' has confusion with 'r'. The lateral phoneme /l/ and the trill /r/ has confusion between them also.

All the Bengali phonemes are separated into some groups based on all of these confusions are found between them in Bengali continuous speech. The groups are listed in Table 8.

Nine groups of phonemes are created based on the confusion matrices. The classification task is performed once again to classify these nine groups. So this time the output node becomes nine. After performing the classification task, this time the overall classification accuracy becomes 98.7%. The confusion matrices are shown in Fig. 8. Considering the classification accuracy of individual groups, it is found that GR1 to GR7 which consist almost all of most happening phonemes of continuous Bengali speech,

Table 5 Phoneme error rate for different models

Methodologies	Phoneme error rate (PER) (%)					
	CDAC corpus			TIMIT corpus		
	Training	Validation	Testing	Training	Validation	Testing
DNN	22.54	22.87	23.37	19.17	20.79	21.46
HMM	26.43	27.31	27.85	20.97	21.33	22.39
MLP	26.96	27.63	28.24	21.53	22.87	23.28

Table 6 Phoneme classification accuracy for different models

Methodologies	Phoneme classification accuracy (%)					
	CDAC corpus			TIMIT corpus		
	Training	Validation	Testing	Training	Validation	Testing
DNN	87.80	86.12	82.53	85.31	84.67	83.42
HMM	77.64	75.43	74.37	79.23	78.34	76.82
MLP	76.89	75.71	72.48	78.64	77.29	76.26

Table 7 Individual precision, recall and f-score for all Bengali phonemes

Phoneme class	Precision (%)	Recall (%)	F-score (%)	Phoneme class	Precision (%)	Recall (%)	F-score (%)	Phoneme class	Precision (%)	Recall (%)	F-score (%)
k	87.9	88.8	88.35	p	89.5	91.6	90.54	sp	87.8	89	88.40
k^h	88.6	85.7	87.13	p^h	67.1	49.6	67.04	ɔ	91.9	90	90.94
g	76.5	63.2	69.22	b	91.1	92.4	91.75	a	94.9	95.6	95.25
g^h	71.7	53.3	61.15	b^h	78.8	77.9	78.35	i	89.4	91.4	90.39
ʧ	87.9	83.6	85.7	ʃ	98.8	98.8	98.80	u	79.9	78.4	79.14
$ʧ^h$	94.7	95.5	95.10	s	97.2	97.2	97.20	e	90.0	91.5	90.74
ʤ	91.5	92.4	91.95	h	90.8	91.7	91.25	o	89.5	92.7	91.07
$ʤ^h$	66.7	29.3	40.71	m	88.7	86.1	87.38	æ	88.5	83.7	86.03
ʈ	79.0	69.7	74.06	n	89.1	93.0	91.01	ɔx			
$ʈ^h$	89.2	88.0	88.60	ŋ	86.2	81.2	83.63	ax	83.3	54.5	65.89
ɖ	74.1	56.2	63.92	r	87.1	89.2	88.14	ix			
$ɖ^h$	43.2	31.8	36.63	l	90.9	85	87.85	ux	83.8	50.9	63.33
t̪	87.8	91.0	89.37	ɽ	81.8	72.8	77.04	ex			
$t̪^h$	70.3	54.3	61.27	e̯	89.2	79.5	84.07	ox			
d̪	85.6	88.4	86.98	w	77.2	81.4	79.24	æx			
$d̪^h$	80.5	82.9	81.68	sil	89.9	89.0	89.45	oi	84.5	69.4	76.21
								ou	82.1	48.4	60.90

GR1	ক /k/	খ /kʰ/	ট /ʈ/	ঠ /ʈʰ/	ত /t̪/	থ /t̪ʰ/	প /p/	ফ /pʰ/	Precision (%)
ক /k/	**7537**	185	106	8	237	38	91	21	87.9
খ /kʰ/	64	**1985**	6	9	7	28	1	33	88.6
ট /ʈ/	81	13	**1407**	43	52	5	24	3	79.0
ঠ /ʈʰ/	8	5	36	**761**	1	5	1	3	89.2
ত /t̪/	251	18	170	13	**8068**	204	116	10	87.8
থ /t̪ʰ/	13	19	5	6	58	**538**	3	9	70.3
প /p/	159	6	77	13	114	9	**4806**	17	89.5
ফ /pʰ/	5	6	5	1	5	5	6	**167**	67.1
Recall (%)	88.8	85.7	69.7	88	91.0	54.3	91.6	49.6	

(a)

GR2	গ /g/	ঘ /gʰ/	ড /ɖ/	ঢ /ɖʰ/	দ /d̪/	ধ /d̪ʰ/	ব /b/	ভ /bʰ/	Precision (%)
গ /g/	**803**	3	5	0	24	5	35	17	76.5
ঘ /gʰ/	6	**243**	0	0	0	5	1	26	71.7
ড /ɖ/	4	1	**186**	4	6	2	5	0	74.1
ঢ /ɖʰ/	6	6	3	**48**	1	8	0	2	43.2
দ /d̪/	69	1	37	8	**3110**	31	67	10	85.6
ধ /d̪ʰ/	6	7	0	9	34	**1168**	0	28	80.5
ব /b/	105	2	14	0	56	0	**5000**	37	91.1
ভ /bʰ/	6	53	2	15	4	14	19	**1004**	78.8
Recall (%)	63.2	53.3	56.2	31.8	88.4	82.9	92.4	77.9	

(b)

GR3	য় /e̯/	/w/	উ /u/	ও /o/	অ /ɔ/	আ /a/	অ্যা /æ/	এ /e/	ই /i/	Precision (%)
য় /e̯/	**3168**	16	2	11	4	109	26	149	45	89.2
/w/	21	**254**	1	18	3	28	3	2	1	77.2
উ /u/	0	0	**1782**	201	12	2	0	50	22	79.9
ও /o/	18	23	252	**15276**	631	133	8	119	17	89.5
অ /ɔ/	2	1	10	367	**8156**	297	1	4	0	91.9
আ /a/	138	14	2	102	193	**19982**	157	123	0	94.9
অ্যা /æ/	46	0	0	9	0	108	**1918**	64	11	88.5
এ /e/	438	2	35	93	17	141	140	**19615**	930	90.0
ই /i/	124	2	22	18	0	3	17	1052	**13444**	89.4
Recall (%)	79.5	81.4	78.4	92	90.0	95.6	83.7	91.5	91.4	

(c)

GR4	চ /ʧ/	ছ /ʧʰ/	জ /ʤ/	ঝ /ʤʰ/	Precision (%)
চ /ʧ/	**1918**	125	46	9	87.9
ছ /ʧʰ/	158	**4551**	24	27	94.7
জ /ʤ/	52	25	**3954**	57	91.5
ঝ /ʤʰ/	5	3	4	**44**	66.7
Recall (%)	83.6	95.5	92.4	29.3	

(d)

GR5	ল /l/	র /r/	ড় /ɽ/	Precision (%)
ল /l/	**3339**	27	2	90.9
র /r/	76	**7372**	102	87.1
ড় /ɽ/	5	53	**386**	81.8
Recall (%)	85.0	89.2	72.8	

(e)

GR6	শ /ʃ/	স /s/	হ /h/	Precision (%)
শ /ʃ/	**9839**	48	24	98.8
স /s/	24	**2963**	0	97.2
হ /h/	16	0	**2813**	90.2
Recall (%)	98.8	97.2	91.7	

(f)

GR7	ম /m/	ন /n/	ঙ /ŋ/	Precision (%)
ম /m/	**3413**	168	67	88.7
ন /n/	328	**9130**	107	89.1
ঙ /ŋ/	57	42	**996**	86.2
Recall (%)	86.1	93.0	81.2	

(g)

Fig. 7 Confusion matrices of different phonemes using DNN based classification model

obtained high precision and recall value of more than 96%. So the phonemes which were classified with low precision and recall value earlier, after the grouping they obtained good precision and recall value. For example, the phoneme /ʤʰ/ has only 66.7% of precision and 29.3% of recall value, resulting in an f-score of 40.71% while classifying as an individual phoneme. But after grouping with other affricates, the corresponding group 'GR3' achieved 99% precision and 98.7% recall value which results in an F-score of 98.85%.

The precision and recall values for each of the group is visible in the confusion matrix in Fig. 8 and they are listed along with the f-score values in Table 9 also.

In the second phase of the experiment, the phonological features are classified first. The same classification model

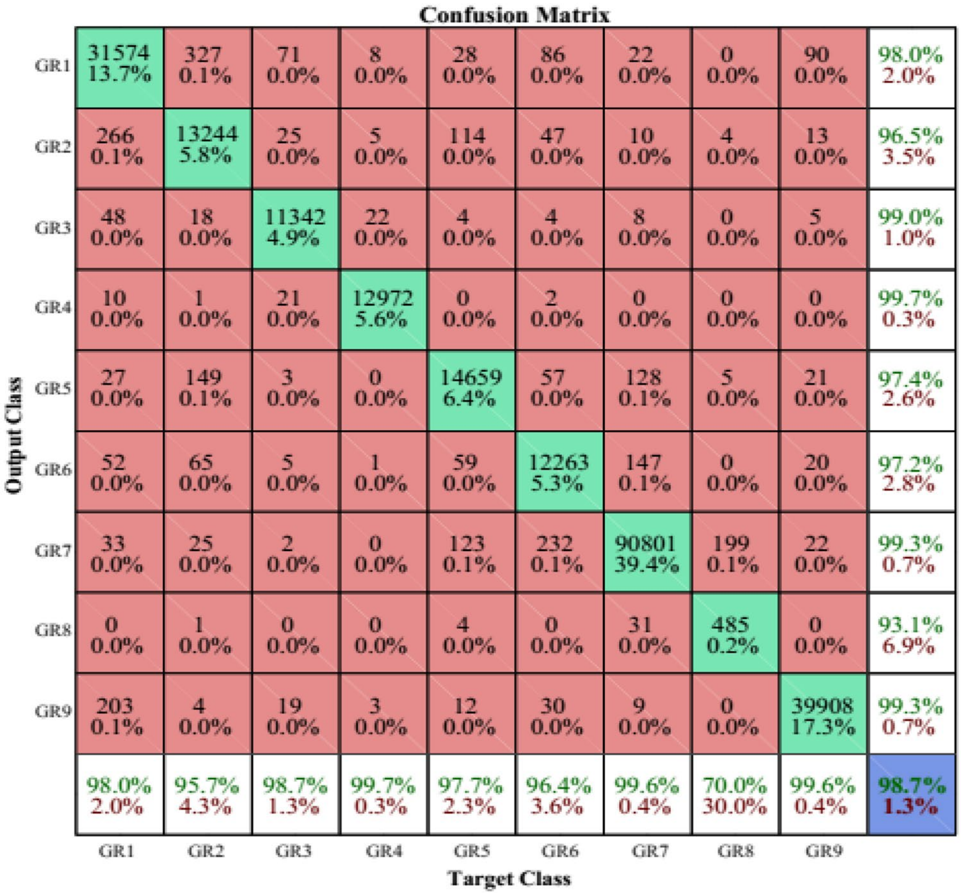
Table 8 Grouping of Bengali phonemes

Group	Phonemes
GR1	k, k^{h}, ʈ, ʈ^{h}, t̪, t̪^{h}, p, p^{h}
GR2	g, g^{h}, ɖ, ɖ^{h}, d̪, d̪^{h}, b, b^{h}
GR3	ʧ, ʧ^{h}, ʤ, ʤ^{h}
GR4	ʃ, s, h
GR5	m, n, ŋ
GR6	l, r, ɽ
GR7	e̯, w, u, o, ɔ, a, æ, e, i
GR8	Nasal vowels
GR9	Diphthongs and others

which was used for phoneme classification is used for this purpose also. The only change was done on the output node. Some robustly ideltified group of manners (Das Mandal 2007) are applied to rearrange the manner of articulation. This combination produced 15 manners of articulation based groups which were kept as output nodes in this classification process. The confusion matrix which is developed for this classification process is shown in Fig. 9.

This classification procedure produces an impressive overall classification accuracy of 98.9%. The manners and corresponding phonemes are listed in Table 10. The individual precision, recall, and f-score for every group are shown in Table 11. It is observed that almost each group possesses the precision and recall values of above 90%. In the case of ‘GR8’, the precision and recall value is found as 81.8 and 51.5%. So the f-score value becomes 63.21%. So the classification performance of this group is not satisfactory. From Table 10 it is found that only the phoneme ‘ʤ^{h}’ is situated in ‘GR8’. In general the phoneme ‘ʤ^{h}’ is occurred for less number of times in continuous Bengali speech. That is why ‘GR8’ is classified with low recall value.

Now, it is important to discuss how the incorporation of phonological features improves the system performance. We compare two confusion matrices of Figs. 8 and 9. The overall classification accuracy observed in two systems as 98.7 and 98.9% respectively. These two results are almost equal. However, when we follow the phoneme sets of these two systems in Tables 8 and 10, we found that the number of phonemes in a single group is less in Table 10 comparing to Table 8. It is very helpful to have increased recognition accuracy for an individual phoneme. Let us explain with an example. From the confusion matrix of Figs. 8 and 9, it is observed that the group GR1 has 98 and 98.3% classification accuracy respectively. However, in Table 8, there are eight phonemes fall in GR1 class whereas Table 10 show only



	GR1	GR2	GR3	GR4	GR5	GR6	GR7	GR8	GR9	
GR1	31574 13.7%	327 0.1%	71 0.0%	8 0.0%	28 0.0%	86 0.0%	22 0.0%	0 0.0%	90 0.0%	98.0% 2.0%
GR2	266 0.1%	13244 5.8%	25 0.0%	5 0.0%	114 0.0%	47 0.0%	10 0.0%	4 0.0%	13 0.0%	96.5% 3.5%
GR3	48 0.0%	18 0.0%	11342 4.9%	22 0.0%	4 0.0%	4 0.0%	8 0.0%	0 0.0%	5 0.0%	99.0% 1.0%
GR4	10 0.0%	1 0.0%	21 0.0%	12972 5.6%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	99.7% 0.3%
GR5	27 0.0%	149 0.1%	3 0.0%	0 0.0%	14659 6.4%	57 0.0%	128 0.1%	5 0.0%	21 0.0%	97.4% 2.6%
GR6	52 0.0%	65 0.0%	5 0.0%	1 0.0%	59 0.0%	12263 5.3%	147 0.1%	0 0.0%	20 0.0%	97.2% 2.8%
GR7	33 0.0%	25 0.0%	2 0.0%	0 0.0%	123 0.1%	232 0.1%	90801 39.4%	199 0.1%	22 0.0%	99.3% 0.7%
GR8	0 0.0%	1 0.0%	0 0.0%	0 0.0%	4 0.0%	0 0.0%	31 0.0%	485 0.2%	0 0.0%	93.1% 6.9%
GR9	203 0.1%	4 0.0%	19 0.0%	3 0.0%	12 0.0%	30 0.0%	9 0.0%	0 0.0%	39908 17.3%	99.3% 0.7%
	98.0% 2.0%	95.7% 4.3%	98.7% 1.3%	99.7% 0.3%	97.7% 2.3%	96.4% 3.6%	99.6% 0.4%	70.0% 30.0%	99.6% 0.4%	98.7% 1.3%

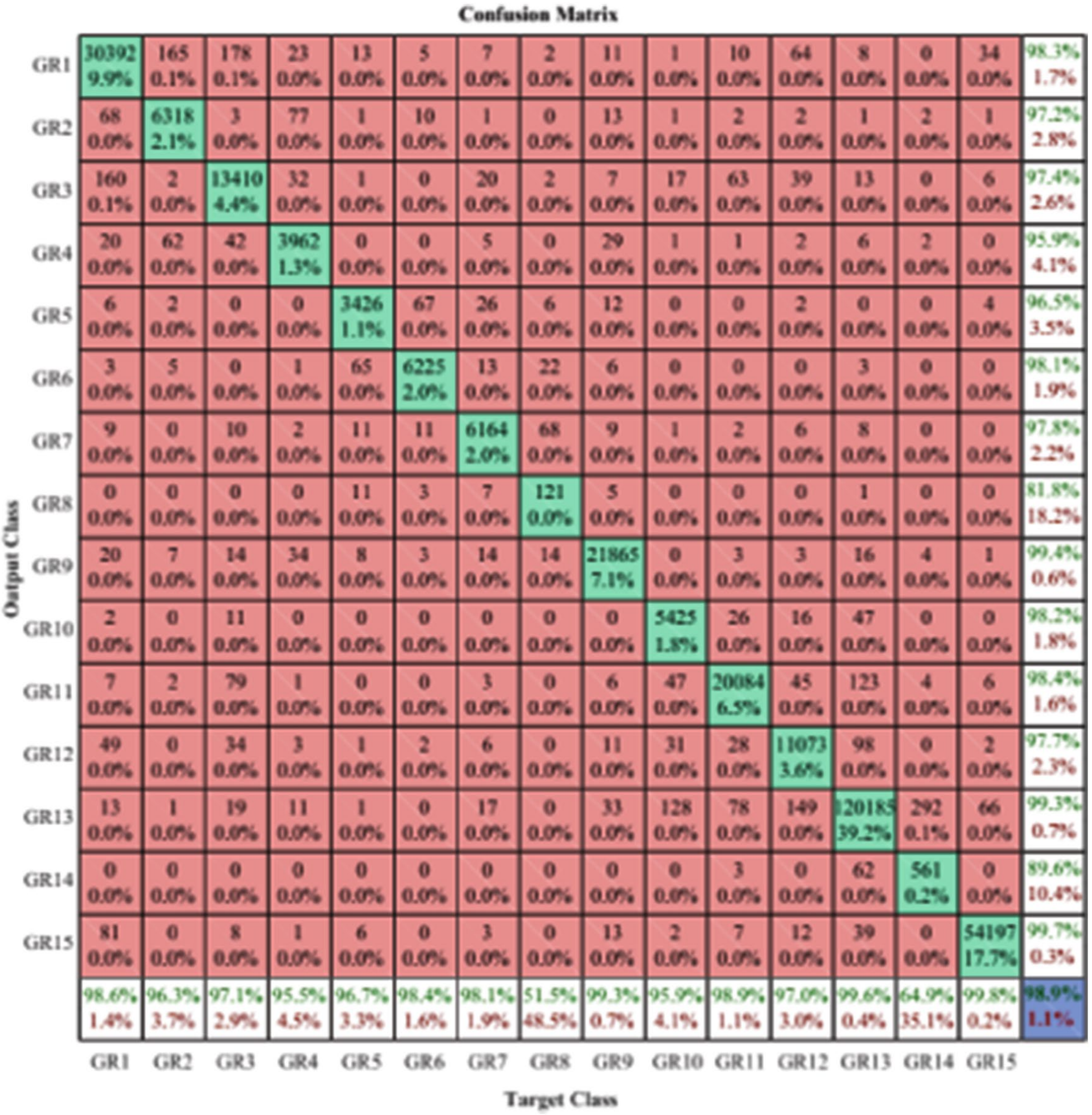
Fig. 8 Confusion matrix of phoneme groups. (Color figure online)

Table 9 Precision, recall and F-score for phoneme groups

Groups	Precision (%)	Recall (%)	F-score (%)
GR1	98.0	98.0	98.00
GR2	96.5	95.7	96.10
GR3	99.0	98.7	98.85
GR4	99.7	99.7	99.70
GR5	97.4	97.7	97.55
GR6	97.2	96.4	96.80
GR7	99.3	99.6	99.45
GR8	93.1	70.0	79.91
GR9	99.3	99.6	99.45

four phonemes in GR1. So when it needs to get individual phoneme classification accuracy for the class GR1, we will have a (8 × 8) confusion matrix in earlier system whereas a (4 × 4) confusion will be observed in the manner based system. This will increase the system performance concerning phoneme recognition and classification accuracy. The Same thing is applicable for GR2 also. GR3 of Table 8 is redivided into four groups in Table 10. This leads to improved classification accuracies for individual phonemes.

Improvement of classification result after incorporation of manner of articulation is more clearly observed when the classification results for ‘ʧ’, ‘ʧʰ’, ‘ʤ’ and ‘ʤʰ’ from Tables 7 and 11 are compared. All of the precision, recall and f-score results are improved when the classification



	GR1	GR2	GR3	GR4	GR5	GR6	GR7	GR8	GR9	GR10	GR11	GR12	GR13	GR14	GR15	
GR1	30392 9.9%	165 0.1%	178 0.1%	23 0.0%	13 0.0%	5 0.0%	7 0.0%	2 0.0%	11 0.0%	1 0.0%	10 0.0%	64 0.0%	8 0.0%	0 0.0%	34 0.0%	98.3% 1.7%
GR2	68 0.0%	6318 2.1%	3 0.0%	77 0.0%	1 0.0%	10 0.0%	1 0.0%	0 0.0%	13 0.0%	1 0.0%	2 0.0%	2 0.0%	1 0.0%	2 0.0%	1 0.0%	97.2% 2.8%
GR3	160 0.1%	2 0.0%	13410 4.4%	32 0.0%	1 0.0%	0 0.0%	20 0.0%	2 0.0%	7 0.0%	17 0.0%	63 0.0%	39 0.0%	13 0.0%	0 0.0%	6 0.0%	97.4% 2.6%
GR4	20 0.0%	62 0.0%	42 0.0%	3962 1.3%	0 0.0%	0 0.0%	5 0.0%	0 0.0%	29 0.0%	1 0.0%	1 0.0%	2 0.0%	6 0.0%	2 0.0%	0 0.0%	95.9% 4.1%
GR5	6 0.0%	2 0.0%	0 0.0%	0 0.0%	3426 1.1%	67 0.0%	26 0.0%	6 0.0%	12 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	4 0.0%	96.5% 3.5%
GR6	3 0.0%	5 0.0%	0 0.0%	1 0.0%	65 0.0%	6225 2.0%	13 0.0%	22 0.0%	6 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	0 0.0%	98.1% 1.9%
GR7	9 0.0%	0 0.0%	10 0.0%	2 0.0%	11 0.0%	11 0.0%	6164 2.0%	68 0.0%	9 0.0%	1 0.0%	2 0.0%	6 0.0%	8 0.0%	0 0.0%	0 0.0%	97.8% 2.2%
GR8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	11 0.0%	3 0.0%	7 0.0%	121 0.0%	5 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	81.8% 18.2%
GR9	20 0.0%	7 0.0%	14 0.0%	34 0.0%	8 0.0%	3 0.0%	14 0.0%	14 0.0%	21865 7.1%	0 0.0%	3 0.0%	3 0.0%	16 0.0%	4 0.0%	1 0.0%	99.4% 0.6%
GR10	2 0.0%	0 0.0%	11 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5425 1.8%	26 0.0%	16 0.0%	47 0.0%	0 0.0%	0 0.0%	98.2% 1.8%
GR11	7 0.0%	2 0.0%	79 0.0%	1 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	6 0.0%	47 0.0%	20084 6.5%	45 0.0%	123 0.0%	4 0.0%	6 0.0%	98.4% 1.6%
GR12	49 0.0%	0 0.0%	34 0.0%	3 0.0%	1 0.0%	2 0.0%	6 0.0%	0 0.0%	11 0.0%	31 0.0%	28 0.0%	11073 3.6%	98 0.0%	0 0.0%	2 0.0%	97.7% 2.3%
GR13	13 0.0%	1 0.0%	19 0.0%	11 0.0%	1 0.0%	0 0.0%	17 0.0%	0 0.0%	33 0.0%	128 0.0%	78 0.0%	149 0.0%	120185 39.2%	292 0.1%	66 0.0%	99.3% 0.7%
GR14	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	62 0.0%	561 0.2%	0 0.0%	89.6% 10.4%
GR15	81 0.0%	0 0.0%	8 0.0%	1 0.0%	6 0.0%	0 0.0%	3 0.0%	0 0.0%	13 0.0%	2 0.0%	7 0.0%	12 0.0%	39 0.0%	0 0.0%	54197 17.7%	99.7% 0.3%
	98.6% 1.4%	96.3% 3.7%	97.1% 2.9%	95.5% 4.5%	96.7% 3.3%	98.4% 1.6%	98.1% 1.9%	51.5% 48.5%	99.3% 0.7%	95.9% 4.1%	98.9% 1.1%	97.0% 3.0%	99.6% 0.4%	64.9% 35.1%	99.8% 0.2%	98.9% 1.1%

Fig. 9 Confusion matrix of group of manners. (Color figure online)

Table 10 Grouping of Bengali phonemes with respect to group of manners

Group	Manners	Phonemes
GR1	Unvoiced, unaspirated, stop	k, ʈ, t̪, p
GR2	Unvoiced, aspirated, stop	kʰ, ʈʰ, t̪ʰ, pʰ
GR3	Voiced, unaspirated, stop	g, ɖ, d̪, b
GR4	Voiced, aspirated, stop	gʰ, ɖʰ, d̪ʰ, bʰ
GR5	Unvoiced, unaspirated, affricate	ʧ
GR6	Unvoiced, aspirated, affricate	ʧʰ
GR7	Voiced, unaspirated, affricate	ʤ
GR8	Voiced, aspirated, affricate	ʤʰ
GR9	Fricative	ʃ, s, h
GR10	Lateral	l
GR11	Nasal murmur	m, n, ŋ
GR12	Trill, tap/flap	r, ɽ, ɽʰ
GR13	Vowel and semivowel	u, o, ɔ, a, æ, e, i, e̯, w
GR14	Nasal vowel	Nasal vowels
GR15	Dipthong and others	Dipthongs, short pause, silence

Table 11 Precision, recall, and F-score for the group of manners

Manner based groups	Precision (%)	Recall (%)	F-score (%)
GR1	98.3	98.6	98.45
GR2	97.2	96.3	96.75
GR3	97.4	97.1	97.25
GR4	95.9	95.5	95.70
GR5	96.5	96.7	96.60
GR6	98.1	98.4	98.25
GR7	97.8	98.1	97.95
GR8	81.8	51.5	63.21
GR9	99.4	99.3	99.35
GR10	98.2	95.9	97.04
GR11	98.4	98.9	98.65
GR12	97.7	97	97.35
GR13	99.3	99.6	99.45
GR14	89.6	64.9	75.28
GR15	99.7	99.8	99.75

system is incorporated with manner of articulation based phonological features.

8 Conclusion

From the result of this experiment, it is very clear that inclusion of manner of articulation is very much useful to obtain improved performance in phoneme classification task. Both in the first and second phase of this experiment the vowels are kept in a single class. The duration of the transitory part of each phoneme is very small. For unvoiced phonemes, the occlusion period consists most of the duration of the phonemes whereas for voiced phoneme the voicebar does the same thing. That is why it becomes very difficult to classify the place of articulation based phonological features. The inclusion of different places of articulation in the Bengali phoneme classification model is the next target.

In the next phase of this experiment the manner based labeling of the phonemes will be performed on the speech corpora and a lexical expert system (Das Mandal 2007) will be used offline to complete the speech recognition task.

References

Ali, A. A., Van Der Speigel, J., & Mueller, P. (2000). Auditory-based speech processing based on the average localized synchrony detection. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP'00* (Vol. 3, pp. 1623–1626). IEEE.

Ali, A. A., Van der Spiegel, J., & Mueller, P. (1998). An acoustic-phonetic featurebased system for the automatic recognition of fricative consonants. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing* (Vol. 2, pp. 961–964). IEEE.

Ali, A. A., Van der Spiegel, J., Mueller, P., Haentjens, G., & Berman, J. (1999). An acoustic-phonetic feature-based system for automatic phoneme recognition in continuous speech. In *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems* (Vol. 3, pp. 118–121). IEEE.

Ali, A. M. A., Van der Spiegel, J., & Mueller, P. (2001). Acoustic-phonetic features for the automatic classification of fricatives. *The Journal of the Acoustical Society of America, 109*(5), 2217–2235.

Ali, A. M. A., Van der Spiegel, J., & Mueller, P. (2002). Robust auditory-based speech processing using the average localized synchrony detection. *IEEE Transactions on Speech and Audio Processing, 10*(5), 279–292.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning, 2*(1), 1–127.

Bhattacharya, K. (1988). *Bengali phonetic reader* (Vol. 28). Mysuru: Central Institute of Indian Languages.

Bhowmik, T. (2017). Prosodic and phonological feature based speech recognition system for Bengali (Doctoral dissertation, IIT Kharagpur).

Bitar, N., & Espy-Wilson, C. Y. (1995a). A signal representation of speech based on phonetic features. In *Proceedings of 5th Annual Dual Use Technologies and Applications Conference* (pp. 310–315).

Bitar, N. N., & Espy-Wilson, C. Y. (1995b). Speech parameterization based on phonetic features: Application to speech recognition. In *Fourth European Conference on Speech Communication and Technology* (pp. 1411–1414).

Bitar, N. N., Espy-Wilson, C. Y. (1996). A knowledge-based signal representation for speech recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Processing* (pp. 29–32). IEEE.

Chatterji, S. (1926). *The origin and development of the Bengali language*. Calcutta: Calcutta University Press.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2011). Large vocabulary continuous speech recognition with context dependent dbn-hmms.

In *IEEE International Conference on Acoustics, Speech And Signal Processing (ICASSP)* (pp. 4688–4691). IEEE.

Das Mandal, S. (2007). Role of shape parameters in speech recognition: A study on standard colloquial Bengali (SCB). PhD thesis.

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 28*(4), 357–366.

Dekel, O., Keshet, J., & Singer, Y. (2004). An online algorithm for hierarchical phoneme classification. In *International Workshop on Machine Learning for Multimodal Interaction* (pp. 146–158). Berlin: Springer.

Deng, L., Abdel-Hamid, O., & Yu, D. (2013). A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6669–6673). IEEE.

Deng, L., & Yu, D. (2013). *Deep learning for signal and information processing*. Redmond, WA: Microsoft Research Monograph.

Dusan, S. (2005). Estimation of speaker's height and vocal tract length from speech signal. In *Ninth European Conference on Speech Communication and Technology*.

Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: The 632 + bootstrap method. *Journal of the American Statistical Association, 92*(438), 548–560.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters, 27*(8), 861–874.

Feng, X., Zhang, Y., & Glass, J. (2014). Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1759–1763). IEEE.

Frankel, J., & King, S. (2005). A hybrid ann/dbn approach to articulatory feature recognition. In *Proceedings of Eurospeech*. Lisbon: CD-ROM.

Garofolo, J., Consortium, L. D., et al. (1993). *TIMIT: Acoustic-phonetic continuous speech corpus*. Philadelphia, PA: Linguistic Data Consortium.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Vol. 9, pp. 249–256).

Goldberg, H., & Reddy, D. (1976). Feature extraction segmentation and labeling in the harpy and hearsay-ii systems. *The Journal of the Acoustical Society of America, 60*(S1), S11–S11.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks, 18*(5), 602–610.

Harrington, J. (1987). Acoustic cues for automatic recognition of English consonants. In *Speech Technology: A Survey*. (pp. 19–74). Edinburgh: Edinburgh University Press

Harris, J. (1994). *English sound structure*. Oxford: Wiley.

Hayes, B., & Lahiri, A. (1991). Bengali intonational phonology. *Natural Language & Linguistic Theory, 9*(1), 47–96.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-R., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine, 29*(6), 82–97.

Hinton, G., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507.

Hou, J. (2009). On the use of frame and segment-based methods for the detection and classification of speech sounds and features. PhD thesis, Rutgers University Graduate School, New Brunswick.

Huang, X. (1992). Phoneme classification using semicontinuous hidden markov models. *IEEE Transactions on Signal Processing, 40*(5), 1062–1067.

King, S., & Taylor, P. (2000). Detection of phonological features in continuous speech using neural networks. *Computer Speech & Language, 14*(4), 333–353.

King, S., Taylor, P., Frankel, J., & Richmond, K. (2000). Speech recognition via phonetically featured syllables. University of the Saarland.

Lahiri, A. (1999). Speech recognition with phonological features. In *Proceedings of the XIVth International Congress of Phonetic Sciences* (Vol. 99, pp. 715–718).

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., & Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th International Conference on Machine learning* (pp. 473–480). ACM.

Lee, C.-H., Clements, M., Dusan, S., Fosler-Lussier, E., Johnson, K., Juang, B.-H., & Rabiner, L. (2007). An overview on automatic speech attribute transcription (ASAT). In *INTERSPEECH* (pp. 1825–1828) Antwerp.

Lewis, M. P., Simons, G. F., & Fennig, C. D. (2016). *Ethnologue: Languages of the world* (Vol. 19). Dallas, TX: SIL International Dallas.

Mandal, S., Chandra, S., Lata, S., & Datta, A. (2011). Places and manner of articulation of Bangla consonants: An epg based study. In *INTERSPEECH* (pp. 3149–3152) Florence.

Mandal, S. D., Saha, A., & Datta, A. (2005). Annotated speech corpora development in Indian languages. *Vishwa Bharat, 6*, 49–64.

MATLAB. (2015). MATLAB version 8.5.0.197613 (R2015b). Natick: The Mathworks, Inc..

Meyer, B. T., Wächter, M., Brand, T., & Kollmeier, B. (2007). Phoneme confusions in human and automatic speech recognition. In *INTERSPEECH* (pp. 1485–1488).

Mohamed, A.-R., Dahl, G. E., & Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing, 20*(1), 14–22.

Mohamed, A.-R., Yu, D., & Deng, L. (2010). Investigation of full sequence training of deep belief networks for speech recognition. In *INTERSPEECH* (Vol. 10, pp. 2846–2849).

Morales, S. O. C., & Cox, S. J. (2007). Modelling confusion matrices to improve speech recognition accuracy, with an application to dysarthric speech. In INTERSPEECH (pp. 1565–1568).

Moreau, N., Kim, H.-G., & Sikora, T. (2004). Phonetic confusion based document expansion for spoken document retrieval. In *INTERSPEECH*.

Online census data (2016). Retrieved July 20, 2016, from http://censusindia.gov.in/Census_Data_2001/ Census_Data_Online/Language/Statement3.htm.

Palm, R. B. (2012). Prediction as a candidate for learning deep hierarchical models of data. Master's thesis.

Reetz, H. (1999). Converting speech signals to phonological features. In *Proceedings of the XIVth International Congress of Phonetic Sciences* (Vol. 99, pp. 1733–1736).

Renals, S., & Rohwer, R. (1989). Phoneme classification experiments using radial basis functions. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNNâ˘A´Z89)* (Vol. 1, pp. 461–467).

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning representations by backpropagating errors. *Nature, 323*, 533–536.

Seide, F., Li, G., & Yu, D. (2011). Conversational speech transcription using context-dependent deep neural networks. In *INTERSPEECH* (pp. 437–440). Florence.

Siniscalchi, S., & Lee, C.-H. (2009). A study on integrating acoustic-phonetic information into lattice rescoring for automatic speech recognition. *Speech Communication, 51*(11), 1139–1153.

Siniscalchi, S., Lyu, D.-C., Svendsen, T., Lee, C.-H. (2012). Experiments on cross-language attribute detection and phone recognition with minimal targetspecific training data. *IEEE Transactions on Audio, Speech, and Language Processing, 20*(3), 875–887.

Siniscalchi, S., Svendsen, T., & Lee, C.-H. (2007). Towards bottom-up continuous phone recognition. In *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)* (pp. 566–569). IEEE.

Siniscalchi, S., Yu, D., Deng, L., & Lee, C.-H. (2013). Exploiting deep neural networks for detection based speech recognition. *Neurocomputing, 106*, 148–157.

Siniscalchi, S. M., & Reed, J., Svendsen, T., & Lee, C.-H. (2009). Exploring universal attribute characterization of spoken languages for spoken language recognition. In *INTERSPEECH* (pp. 168–171). Brighton.

Siniscalchi, S. M., Svendsen, T., & Lee, C.-H. (2011). A bottom-up stepwise knowledge integration approach to large vocabulary continuous speech recognition using weighted finite state machines. In *INTERSPEECH* (pp. 901–904). Florence.

Srinivasan, S., & Petkovic, D. (2000). Phonetic confusion matrix based spoken document retrieval. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 81–87). ACM.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 1096–1103). ACM.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research, 11*, 3371–3408.

Xu, D., Wang, Y., & Metze, F. (2014). EM-based phoneme confusion matrix generation for low-resource spoken term detection. *IEEE Spoken Language Technology Workshop (SLT)* (pp. 424–429). IEEE.

Yu, D., & Deng, L. (2014). Automatic speech recognition: A deep learning approach. London: Springer.

Yu, D., Deng, L., & Dahl, G. (2010). Roles of pre-training and fine tuning in context dependent dbn-hmms for real world speech recognition. In *Proceedings of NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Yu, D., Siniscalchi, S., Deng, L., & Lee, C.-H. (2012). Boosting attribute and phone estimation accuracies with deep neural networks for detection based speech recognition. In *ICASSP* (pp. 4169–4172). IEEE.

Žgank, A., Horvat, B., & Kačič Z. (2005). Data driven generation of phonetic broad classes, based on phoneme confusion matrix similarity. *Speech Communication, 47*(3), 379–393.

Zhang, P., Shao, J., Han, J., Liu, Z., & Yan, Y. (2006). Keyword spotting based on phoneme confusion matrix. *Proceedings of ICSLP* (Vol. 2, pp. 408–419).