Circuits, Systems, and Signal Processing manuscript No.
(will be inserted by the editor)

A COMPARATIVE STUDY OF DEEP LEARNING TECHNIQUES ON FRAME-LEVEL SPEECH DATA CLASSIFICATION

Abdolreza Sabzi Shahrebabaki, Ali Shariq Imran, Negar Olfati, Torbjørn Svendsen

Pre-print version. Accepted version available at https://doi.org/10.1007/s00034-019-01130-0

Abstract This paper provides a comprehensive analysis of the effect of speaking rate on frame classification accuracy. Different speaking rates may affect the performance of the automatic speech recognition (ASR) system yielding poor recognition accuracy. A model trained on a normal speaking rate is better able to recognize speech at a normal pace but fails to achieve similar performance when tested on slow or fast speaking rates. Our recent study has shown that a drop of almost ten percentage points in the classification accuracy is observed when a deep feed-forward network is trained on the normal speaking rate and evaluated on slow and fast speaking rates. In this paper, we extend our work to convolutional neural networks (CNN) to see if this model can reduce the accuracy gap between different speaking rates. Filter bank energies (FBE) and Mel frequency cepstral coefficients (MFCC) are evaluated on multiple configurations of the CNN where the networks are trained on normal speaking rate and evaluated on slow and fast speaking rates. The results are compared to those obtained by a deep neural network (DNN). A breakdown of phoneme level classification results and the confusion between vowels and consonants is also presented. The experiments show that the CNN architecture when used with FBE features performs better on both slow and fast speaking rates. An improvement of nearly 2% in case of fast, and 3% in case of slow speaking rates is observed.

Keywords speech recognition · phoneme classification · speaking rate · deep learning

1 Introduction

Deep learning has seen a massive growth in the past decade with the advancement of high performance computational devices and dynamic programming. The decades worth of knowledge that has gone into conventional state-of-the-art recognition systems is seeing a paradigm shift, and speech recognition is no exception. Traditional speech recognition systems based on Gausian mixture model (GMM)/hidden Markov

Abdolreza Sabzi Shahrebabaki,
Department of Electronic Systems, NTNU, Trondheim, Norway
Tel.: +47-48680765
ORCiD.: 0000-0002-0877-9456
E-mail: abdolreza.sabzi@ntnu.no

Ali Shariq Imran
Department of Electronic Systems, NTNU, Trondheim, Norway
E-mail: ali.imran@ntnu.no

Negar Olfati
Department of Electronic Systems, NTNU, Trondheim, Norway
E-mail: negar.olfati@ntnu.no

Torbjørn Svendsen
Department of Electronic Systems, NTNU, Trondheim, Norway
E-mail: torbjorn.svendsen@ntnu.no

model (HMM) are now being replaced by deep architectures [6], as the gains in accuracy outweigh the computational cost. Deep architectures having multiple hidden layers are now an essential part of automatic speech recognition (ASR) systems - not only for feature representation [21] but also for language [2] and acoustic modeling [21, 18].

ASR systems such as those employing supervised machine learning techniques and deep learning methods can efficiently learn phonetic patterns. Having said that, speaking rate variability can drastically decrease the performance of the ASR systems if they are not tuned for it. Variations in speaking rate affect the mapping between the acoustic properties of speech and the linguistic interpretation of the utterances [9]. Humans can naturally account for variations in speaking rate and can adapt to it while maintaining phonetic constancy [4]; however, this can still be a challenging task for ASR systems.

Performance evaluation of ASR can be carried out across different intrinsic variabilities such as speaking rate, effort, style, and dialect by decomposing the speech signal into phonetic features that describe aspects of speech production such as voicing, manner, and place of articulation. Voicing deals with the glottal vibration, i.e., whether glottal vibration is present or absent. Manner deals with the mode of articulatory production such as nasal, stop, and fricative. Place of articulation deals with the locus of articulatory constriction such as anterior, medial, or posterior. At the phonetic level, the consonant phonemes and vowel phonemes can be grouped based on phonetic features for evaluating various speech intrinsic variabilities.

This paper extends our earlier work reported in [22] where we compared four different acoustic features namely filter bank energies (FBE), Mel frequency cepstral coefficients (MFCC), line spectral frequencies (LSF), and linear predictive coefficients (LPC). We provided a comparison between these features in terms of different speaking rates, training the model on the normal speaking rate and evaluating slow and fast speaking rates. Our findings showed that FBE results were far better compared to the other features. In this paper, we further evaluate the FBE and MFCC on different deep neural network (DNN) models and architectures. The contribution of the paper is threefold:

1. Evaluation of multiple configurations of both DNN and convolutional neural network (CNN) architectures with respect to the number of hidden layers and the number of neurons in each layer.
2. Utilization of temporal context for creating context-dependent feature vectors for both FBE and MFCC features.
3. Studying the impact of speaking rate variabilities on different deep learning architectures.

The remainder of the paper is structured as follows. Section 2 presents the related work, followed by Section 3 where we describe both DNN and CNN. Section 4 gives an overview of the experimental setup. Results and their analysis are presented in section 5, while section 6 concludes the paper.

2 Related work

Speech recognition is an area of wide interest and has undergone massive growth over the last couple of decades. One particular topic that attracted attention in the late 1990s was the effect of speaking rate on the performance of speech recognition [13, 5, 12]. It was observed that speech recognition performance dropped considerably on fast speaking rates [19, 26]. Faltlhauser et al. in their work "why has speaking rate such an impact on speech recognition performance" tried to figure out how speaking rate affects system performance by showing a direct correlation between local average HMM score and local speech rate [8]. Similar studies also showed that variations in speaking rate also affect speaker recognition [27, 10] and authentication [20] performance, specifically as a result of distorted spectrum [28].

Meyer and his colleagues designed the logotome speech corpus to study the effect of different intrinsic variabilities by comparing the performance of human speech recognition to automatic recognition systems [15]. The corpus was designed to facilitate studies of various intrinsic variabilities including speaking rate, effort, and dialect. The three state HMM model used in their study described each phoneme by a binary voicing feature and a ternary feature defining manner and place of articulation. They showed that misclassification of voicing and manner of articulation were the major causes for recognition errors. Later, they tried to reduce the gap between the performance of automatic systems and human listeners [16]. This work was further extended to use a DNN model in which the authors compared three acoustic features namely MFCC, FBE, and PLP [7].

Various acoustic features have been revisited for the task of frame level phoneme classification with the development of deeper and wider neural networks. For instance, authors in [24] used MFFC and relative spectral transformation perceptual linear prediction (RASTA-PLP) for predicting the phoneme classes. They applied a reservoir computing technique in a two-layer recurrent neural network to classify 39 phoneme classes of English and found that MFCC performs slightly better compared to RASTA-PLP. MFCC, PLP, and RASTA-PLP were also evaluated by authors in [11] using fuzzy logic and deep belief networks (DBN) for the same task but for a different language. Their results for the African language Fongbe similarly showed that MFCC produced better classification accuracy. Authors in [14] used a more traditional approach by modeling a five state HMM to compare six acoustic features including LPC, MFCC, PLP, FBE, linear prediction reflection coefficients and Mel-filter bank coefficients on the task of frame classification for phoneme recognition for Arabic. In their study, they found that the FBE representation attained the highest frame level classification accuracy. From the previous work reported in the literature and the work presented in this section, it is evident that the FBE and MFCC are the two feature sets that stand out from the rest for correctly classifying phoneme classes. However, it should be noted that all of these studies were conducted on normal speaking rates.

Not much literature exists regarding the performance of different acoustic features for variable speaking rates. Evaluating performance for variable speaking rates means that a system should be trained on regular speaking rates, and tested on slow or fast speaking rates. Our earlier work, on evaluating acoustic features for variable speaking rates [22], showed that FBE performs slightly better than MFCC when trained and tested on a three-hidden layer DNN with 1024 neurons in each hidden layer. In this paper, we extend our work to evaluate and compare how these speech features perform on convolutional deep neural network and feed-forward deep neural network architectures. Therefore, we build and train our neural network models on deep architectures with context-dependent acoustic features on normal speaking rate data and test them on fast and slow speaking rates.

3 Deep neural networks

3.1 DNN

State-of-the-art speech recognition systems use DNN for acoustic modeling as an alternative to GMM [6]. DNN is a feed-forward, artificial neural network consisting of one input layer, one output layer, and at least two hidden layers. Each hidden unit contains two inner parts: a sum operand adding up all the information from the previous layer, and a nonlinear function, typically logistic regression (1) or tangent hyperbolic (depending on the range of the target values) mapping the total input y_i from the previous layer to a scalar x_j:

$$x_j = b_j + \sum_i y_i \theta_{ij}, \quad y_j = \text{logistic}(x_j) = \frac{1}{1 + \exp(-x_j)}, \tag{1}$$

where b_j is the bias of j^{th} unit, θ_{ij} is the weight on the edge connecting unit i to unit j, and i is the index corresponding to all input units to the current layer. In the last layer of nodes, a softmax function will enforce the total outputs to be mapped as probabilities, by making them add to one. Training the network means to adjusting the set of parameters such that for each input vector, the output vector is as close as possible to the desired output.

DNN training consists of two phases. In the first phase, the layers are pre-trained sequentially, in an unsupervised manner to initialize the weights, and in the second phase a supervised fine tuning is applied to the network for finalizing the weights. The most popular pre-training scheme is built upon a special type of Boltzmann machines known as a restricted Boltzmann machines (RBM). These are probabilistic graphical models consisting of two layers. The variables in the same layer do not have edges linking them (this is why they are called restricted), while there are undirected edges connecting the nodes of one layer to the other. The upper level corresponds to hidden variables, and the lower level constitutes the visible nodes. The joint energy of the visible and hidden units is defined as:

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{I}\sum_{j=1}^{J} \theta_{ij} h_i v_j - \sum_{i=1}^{I} b_i h_i - \sum_{j=1}^{J} c_j v_j, \tag{2}$$

where h_i and v_j are the binary hidden and visible state variables, respectively, and b_i and c_j are the bias terms for the hidden and visible nodes, respectively. The joint probability assigned to each pair of hidden and observed variables is:

$$P(\boldsymbol{v},\boldsymbol{h}) = \frac{\exp\big(-E(\boldsymbol{v},\boldsymbol{h})\big)}{Z}, \tag{3}$$

where Z is the normalizing constant:

$$Z = \sum_{\boldsymbol{v}}\sum_{\boldsymbol{h}} \exp\big(-E(\boldsymbol{v},\boldsymbol{h})\big). \tag{4}$$

Training an RBM involves learning the set of unknown parameters θ_{ij}, b_i and c_j. This is obtained by maximizing the average log-likelihood:

$$\begin{aligned} L(\Theta,\boldsymbol{b},\boldsymbol{c}) &= \frac{1}{N}\sum_{n=1}^{N} \ln P(\boldsymbol{v}_n;\Theta,\boldsymbol{b},\boldsymbol{c}) = \frac{1}{N}\sum_{n=1}^{N} \ln\Big(\frac{1}{Z}\sum_{h}\exp(-E(\boldsymbol{v}_n,\boldsymbol{h};\Theta,\boldsymbol{b},\boldsymbol{c}))\Big) \\ &= \frac{1}{N}\sum_{n=1}^{N} \ln\Big(\sum_{\boldsymbol{h}}\exp\big(-E(\boldsymbol{v}_n,\boldsymbol{h};\Theta,\boldsymbol{b},\boldsymbol{c})\big)\Big) - \ln\sum_{\boldsymbol{h}}\sum_{\boldsymbol{v}}\exp\big(-E(\boldsymbol{v},\boldsymbol{h})\big), \end{aligned} \tag{5}$$

where Θ, $\boldsymbol{b}$, and $\boldsymbol{c}$ refer to the collection of the unknown parameters. Taking the derivative of the log-likelihood with respect to one of the weights, we have:

$$\frac{\partial L(\Theta,\boldsymbol{b},\boldsymbol{c})}{\partial \theta_{ij}} = \frac{1}{N}\sum_{n=1}^{N}\Big(\sum_{\boldsymbol{h}} P(\boldsymbol{h}|\boldsymbol{v}_n) h_i v_{jn}\Big) - \sum_{\boldsymbol{v}}\sum_{\boldsymbol{h}} P(\boldsymbol{v},\boldsymbol{h}) h_i v_j, \tag{6}$$

where N is the size of the training set. A gradient ascent scheme for maximizing the log-likelihood will be of the form:

$$\theta_{ij}(new) = \theta_{ij}(old) + \mu\big(< h_i v_j >_{\text{data}} - < h_i v_j >_{\text{reconstruction}}\big), \tag{7}$$

where the $< ... >$ denotes the expectation over the containing variables, the $< h_i v_j >_{\text{data}}$ corresponds to the first term of (6) and represents the correlation between the visible units v_j which are the data samples and the hidden units h_i which are generated by (8), $< h_i v_j >_{\text{reconstruction}}$ corresponds to the second term of (6) and represents the correlation between the visible units v_j which are reconstructed by (9) and the hidden units h_i which are generated by (8) conditioned to the reconstructed visible units, and μ shows the learning rate. For calculating $< h_i v_j >$ the conditional probabilities should be computed:

$$p(h_i|\boldsymbol{v}) = \text{logistic}\Big(b_i + \sum_{j=1}^{J}(\theta_{ij} v_j)\Big), \tag{8}$$

and because of the symmetry in the previous equations the following equation can easily be derived:

$$p(v_j|\boldsymbol{h}) = \text{logistic}\Big(c_i + \sum_{i=1}^{I}(\theta_{ij} h_i)\Big). \tag{9}$$

Based on these equations, the correlations can be calculated by applying Gibbs sampling sequentially. The sampling will be done by the following steps, which are known as contrastive divergence (CD):

- Given the visible layer $\boldsymbol{v}^{(1)}$, samples for the hidden layers are generated by a Gibbs sampler according to $\boldsymbol{h}^{(1)} \sim p(\boldsymbol{h}|\boldsymbol{v}^{(1)})$.
- From the hidden samples $\boldsymbol{h}^{(1)}$ the visible samples can be generated as $\boldsymbol{v}^{(2)} \sim p(\boldsymbol{v}|\boldsymbol{h}^{(1)})$.
- Use the recently generated visible samples $\boldsymbol{v}^{(2)}$ for generating the next hidden samples $h^{(2)} \sim p(\boldsymbol{h}|\boldsymbol{v}^{(2)})$.

Because of only one up-down-up Gibbs sampling is used, this method is known as CD-1 [23]. The new weights can be calculated from the generated samples as:

$$\theta_{ij}(n) = \theta_{ij}(n-1) + \mu\big(h_i^{(1)}v_j^{(1)} - h_i^{(2)}v_j^{(2)}\big) \tag{10}$$

After training the first RBM on the data, the estimated samples for the hidden units are used as the visible samples for training the next RBM. In this way, the RBMs can be stacked on top of each other to make a multilayer generative model that is called a deep belief network (DBN). This process is a pre-training phase which is carried out in an unsupervised manner. The inferred DBN is then used as the initial values for the final DNN, which is trained in a supervised manner by adding a softmax layer at the end and having the label samples as the output. This process is known as fine-tuning. Figure 1 shows the whole procedure of training a DNN by using the DBN.

3.2 CNN

CNN is a slightly modified version of a standard neural network and is a combination of the feature extracting layers with the conventional deep neural network. The feature extracting layers contain convolution and pooling layers, which are not fully connected layers. These layers try to extract the information which is invariant to variations in the input data. The important part of how to use CNN is presenting the input data to the convolution layer. The representation of data as features is often called a feature map. It represents the features along different locations, for example, time and frequency in speech processing. In the case of speech processing, in particular, a feature map mostly constitutes the information along the frequency axis [1]. As the feature maps are one-dimensional for a speech signal, each one of them shows the values of a different frequency band. When the feature maps are ready, the convolution and pooling are applied to extract the features. The input feature maps $X_i, (i = 1, \ldots, I)$ are transformed to another set of feature maps $Y_j, (j = 1, \ldots, J)$ by the convolution layers based on the number of the local filters $\theta_{ij}, (i = 1, \ldots, I; j = 1, \ldots, J)$. The calculation of the new feature maps is done by the convolution as follows:

$$Y_j = \sigma\Big(\sum_i X_i * \theta_{ij} + \theta_{0,j}\Big), \tag{11}$$

which is in a matrix form. Here, σ is the activation function. For more clarification, the convolution operation for the m^{th} element of the feature map $x_{i,m}$ is of the following form:

$$y_{j,m} = \sigma\Big(\sum_i \sum_{k=1}^{F} x_{i,k+m-1}\theta_{i,j,k} + \theta_{0,j}\Big), \tag{12}$$

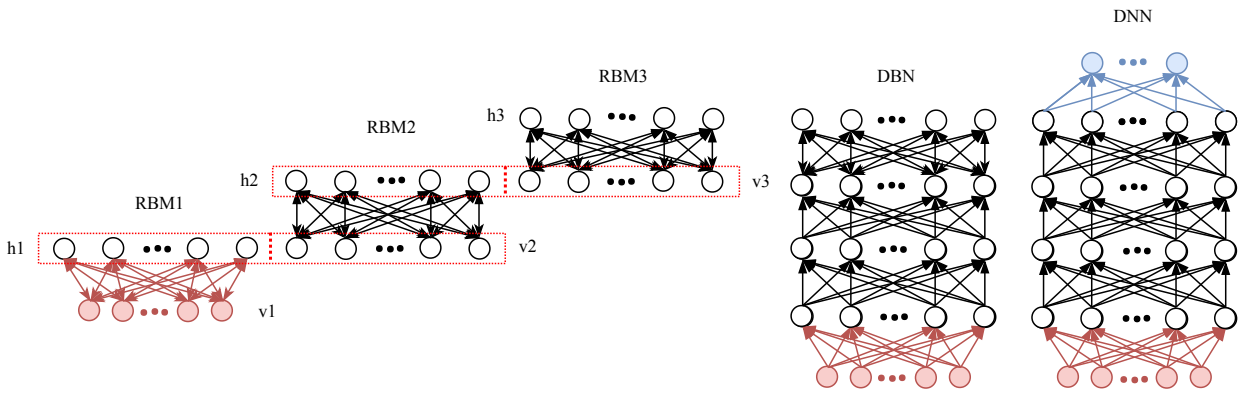

Fig. 1: Training a DNN based on unsupervised learning of DBN with the stacked RBMs.

where $y_{j,m}$ is the m^{th} element of the j^{th} feature map, $\theta_{i,j,k}$ is the k^{th} element of the weight vector $\theta_{i,j}$ which connects the i^{th} input feature map to the j^{th} feature map of the convolution layer and F is the filter size. The filter size specifies the number of input bands which are convolved with each element of the convolution layer. After computations of the feature map on the convolution layer, a pooling layer is applied to reduce the dimension of the feature maps and also to remove small variations in the location [1]. There are two common ways of pooling, maximum and average pooling. If the maximum pooling is applied, the output is calculated as:

$$z_{i,m} = \max_{k=1}^{P} y_{i,(m-1)\times s+k}, \tag{13}$$

where P is the pooling size and s is the shift between the pooling regions. In the same way, the average pooling output is calculated as:

$$z_{i,m} = \frac{1}{P}\sum_{k=1}^{P} y_{i,(m-1)\times s+k}. \tag{14}$$

The pooling output can be fed to a nonlinear function or fully connected layers and then all of the weights can be updated by the backpropagation algorithm.

4 Experimental setup

This section describes the speech dataset, presents the data preparation step, and explains the feature representation process used for data analysis, presented in section 5.

4.1 OLLO dataset

The Oldenburg logatome (OLLO) corpus [25] was used for this study. It is a speech database that contains simple, non-sense combinations of consonants (C) and vowels (V), which are referred to as logatomes. These different phonemes are listed as below:

- Vowels: /a/, /ɛ/, /ɪ/, /ɔ/, /ʊ/, /aː/, /e/, /i/, /o/, /u/.
- Consonants: /d/, /t/, /g/, /k/, /f/, /s/, /b/, /p/, /v/, /ts/, /m/, /n/, /ʃ/, /l/.

One hundred and fifty different CVC and VCV combinations were spoken by 40 German and 10 French speakers. The VCVs are the combination of 14 central consonants and 5 outer vowels. Also eight consonants and 10 vowels are combined to make the CVCs. In both combinations the outer phonemes are the same.

Four different dialects are covered by the German speakers containing no dialect, Bavarian, East Frisian, and East Phalian. The database contains logatomes spoken normally, followed by variabilities such as 'fast', 'slow', 'loud', 'soft', and 'questioning'. These variabilities can be grouped together into three categories: i) speaking rate (fast, slow and normal), ii) speaking style (question and statement), and iii) speaking effort (loud, soft and normal). Each of the 150 logatomes was repeated three times by each speaker. The same number of male and female speakers were used to record the database to cover gender variabilities. The sampling frequency of the utterances is 16 kHz.

OLLO is mostly used for comparison between human speech recognition (HSR) and ASR [15, 17]. We primarily chose to use this dataset for the following reasons:

(a) This database enables the evaluation of different variabilities and their effects on ASR systems;
(b) OLLO may be useful in studying the influence of dialect or accent on speech recognitition performance.

In the following experiments 10 speakers with no dialect were chosen. The variations fast, slow, and normal were used.

Table 1: Number of training and test samples for fast, normal, and slow speaking rate utterances.

Speaking rate	**Fast**	**Normal**	**Slow**
No. training samples	127100 $\simeq$ 20 minutes	174200 $\simeq$ 30 minutes	258200 $\simeq$ 45 minutes
No. test samples	14100 $\simeq$ 2.5 minutes	19300 $\simeq$ 3.5 minutes	31700 $\simeq$ 5.5 minutes

Table 2: Leave-one-out cross validation train and test data samples for each speaker.

	S01	**S02**	**S03**	**S04**	**S05**	**S06**	**S07**	**S08**	**S09**	**S10**
Train	110945	115529	117133	113956	117639	119083	116433	117454	115817	116749
Test	18025	13441	11837	15014	11331	9887	12537	11516	13153	12221

4.2 Data preparation

For the experiments, different speaking rate utterances such as fast, normal, and slow were chosen from no-dialect speakers, as shown in Table 1. Then leave-one-out cross validation is performed on each of the ten speakers (S01 - S10). Each speaker is chosen iteratively for the test and the remaining nine speakers are used for training. Table 2 shows the total number of train and test samples for each of the ten speakers.

As mentioned in section 4.1, 150 logatomes are available with three repetitions each, which equals 450 utterances for each speaking rate. The average length of utterances for slow, normal, and fast speaking rates are 700 ms, 400 ms, and 300 ms respectively. The frame length and frame shift were set to 25 ms and 10 ms respectively. For the validation data, we used 5% of the training data. The total number of training and test samples, and the amount of data in minutes are shown in Table 1.

4.3 Speech Features

Two different feature representations were used for the experiments: MFCC and FBE. All of the features were extracted from a windowed signal of length 25ms with a 15ms overlap between consecutive frames.

4.3.1 FBE and MFCC

FBE features were extracted by a uniform filter bank of size 40 on the Mel-scaled frequency axis which resembles the frequency characteristics of the human auditory system. The logarithm of the filter bank energies yield the FBE features. MFCCs were then obtained by using the DCT transformation on the FBE features. As a result of this transformation, the features were decorrelated. To have the same information in both FBE and MFCC representations, all of the 40 MFCC features were preserved, and there was no dimensionality reduction.

4.3.2 Feature representation

The features extracted in section 4.3.1 were represented as a temporal context produced by the DCT vector C_v. Different sizes of C_v were generated by concatenating M prceeding and N succeeding frames: $C_v = [C^T_{(f-M)}, \ldots, C^T_f, \ldots, C^T_{(f+N)}]^T$. Each frame is a single feature vector of size 1×40 and C_f is the context frame under consideration which leads to C_v having size $40 \times (M + 1 + N)$. We set $M = N$. Context sizes of M $(i.e., M = 0, \ldots, 10)$ were used. This resulted in a context vector of size 40 for no-context and 840 for $M = 10$ corresponding to time durations of $25ms$ to $225ms$. Figure 2 shows the frame classification accuracy for different M. These were initial experiments to determine context size, and based on results, context size $M = 10$ was used for the remaining experiments. It is obvious that the larger the context size, the better the accuracy. The empirical results do, however, suggest that the accuracy rate becomes saturated for $M \geq 9$. The C_v is the input to various deep networks, as shown in Figure 3.

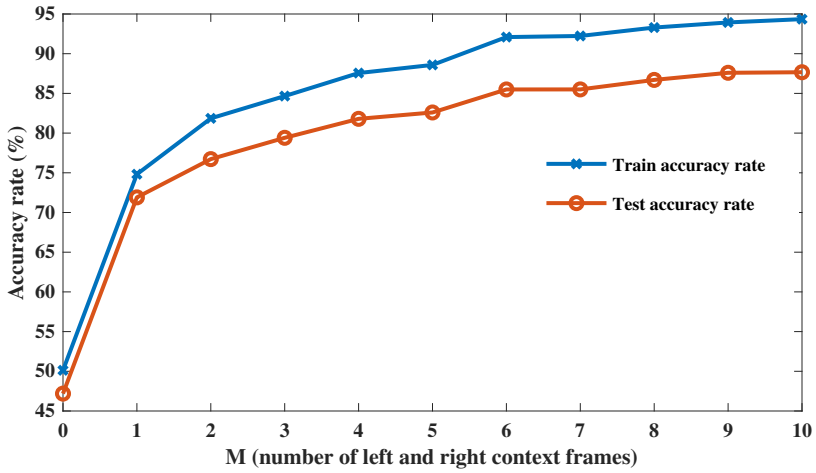

Fig. 2: Effect of context size on the accuracy rate for train and test data.

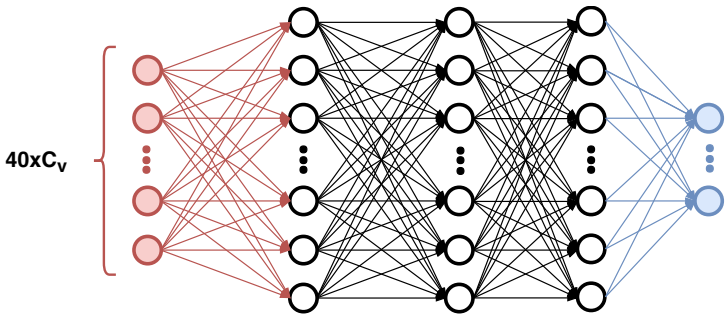

Fig. 3: A 3 hidden layer DNN with the context window size of C_v.

Table 3: Topology of the CNN.

Layer type	Input shape	Output shape	No. of filters	Filter size	Activation function
Convolution 1D	840	4224	128	8×21	Linear
Max pooling	4224	1408	––	––	Sigmoid
Convolution 1D	1408	2048	256	512×1	––
Max pooling	2048	2048	––	––	Sigmoid
Fully connected	2048	1024	––	––	Sigmoid
Fully connected	1024	1024	––	––	Sigmoid
Fully connected	1024	24	––	––	Softmax

4.4 Networks topology

Two types of deep learning models are utilized in this study, each containing an architecture of four hidden layers, as shown in Fig. 4. For the DNN architecture, each fully-connected layer containing 1024 nodes is placed one after another. The hyperparameters of the DNN topology and the network configurations are set according to [22]. In the case of CNN, the network topology consists of two complex convolution layers where each complex layer consists of a convolution followed by a pooling operation. A *Max pooling* of size 3 and a pool stride of 1 was used for the first pooling layer and a pool size of 1 for the second pooling layer in the experiments. The other two hidden layers are fully-connected layers. The network configuration and the total trainable parameters of the CNN are given in Table 3. In both models, the input layer contains 840 nodes (i.e., $40F_v \times 21C_v$) and the output layer contains 24 nodes.

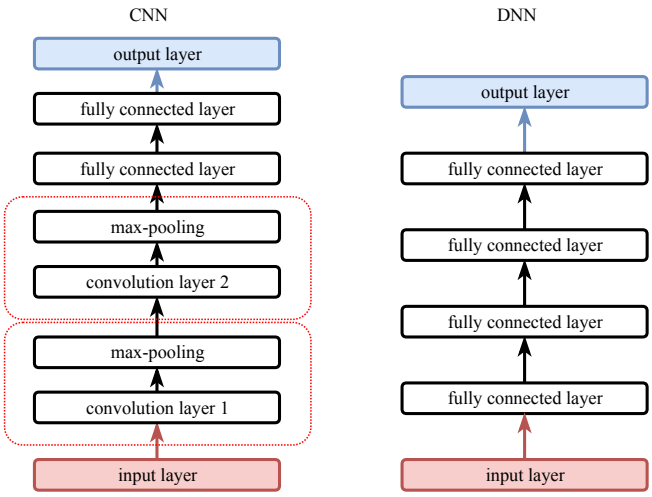

Fig. 4: Architecture of CNN and DNN.

	/a/	/ɛ/	/ɪ/	/ɔ/	/ʊ/	/aː/	/e/	/i/	/o/	/u/
/a/	**0.84**	0.02	0	0.02	0	0	0	0	0	0
/ɛ/	0	**0.81**	0.06	0	0	0	0	0	0	0
/ɪ/	0	0.05	**0.85**	0	0	0	0.02	0	0	0
/ɔ/	0.1	0	0	**0.76**	0.04	0	0	0	0	0
/ʊ/	0	0	0	0.06	**0.79**	0	0	0	0	0.02
/aː/	0.17	0	0	0.02	0	**0.74**	0	0	0	0
/e/	0	0.02	0.34	0	0	0	**0.5**	0.07	0	0
/i/	0	0	0.11	0	0	0	0.02	**0.82**	0	0
/o/	0	0	0	0.03	0.14	0	0	0	**0.71**	0.05
/u/	0	0	0	0	0.15	0	0	0	0.10	**0.69**

(a) Fast speaking rate

	/a/	/ɛ/	/ɪ/	/ɔ/	/ʊ/	/aː/	/e/	/i/	/o/	/u/
/a/	**0.83**	0	0	0.03	0	0.05	0	0	0	0
/ɛ/	0	**0.86**	0.06	0	0	0	0	0	0	0
/ɪ/	0	0.06	**0.78**	0	0	0	0.04	0.05	0	0
/ɔ/	0.03	0	0	**0.83**	0.04	0	0	0	0	0
/ʊ/	0	0	0	0.05	**0.78**	0	0	0	0.03	0.04
/aː/	0.03	0	0	0.03	0	**0.87**	0	0	0	0
/e/	0	0	0.11	0	0	0	**0.73**	0.1	0	0
/i/	0	0	0.03	0	0	0	0.03	**0.91**	0	0
/o/	0	0	0	0	0.02	0	0	0	**0.86**	0.09
/u/	0	0	0	0	0.03	0	0	0	0.09	**0.83**

(b) Normal speaking rate

	/a/	/ɛ/	/ɪ/	/ɔ/	/ʊ/	/aː/	/e/	/i/	/o/	/u/
/a/	**0.58**	0	0	0.04	0	0.29	0	0	0	0
/ɛ/	0	**0.77**	0.06	0	0	0	0.03	0	0	0
/ɪ/	0	0.06	**0.53**	0	0	0	0.12	0.16	0	0
/ɔ/	0	0	0	**0.76**	0.05	0.03	0	0	0.08	0
/ʊ/	0	0	0	0.07	**0.52**	0	0	0	0.13	0.19
/aː/	0	0	0	0.02	0	**0.88**	0	0	0	0
/e/	0	0	0.06	0	0	0	**0.6**	0.23	0	0
/i/	0	0	0	0	0	0	0.02	**0.91**	0	0
/o/	0	0	0	0	0	0	0	0	**0.80**	0.14
/u/	0	0	0	0	0.04	0	0	0	0.12	**0.79**

(c) Slow speaking rate

Fig. 5: Vowel part of the confusion matrix for different speaking rates with DNN trained on FBE features.

5 Deep learning for frame-level classification

This section presents an analysis of the results for frame-level classification in terms of classification accuracy of correctly classified phonemes. It also provides in-depth error analysis and presents confusion matrices for various phonemes. An F_1 score showing a weighted average of precision and recall was computed for the full confusion matrix to compare the performance between the DNN model and the CNN. Additionally, a Kappa κ value that represents classification accuracy normalized by the imbalance of the classes in the data was also computed for a fair comparison.

5.1 Classification accuracy

Results in Table 4 show the frame accuracy rate for the normal speaking rate for both training and test data. There is a wide gap between training and test accuracy rates for MFCC as compared to FBE for both DNN and CNN, especially in the case of DNN with MFCC features. This huge gap might be a result of over-fitting of the training data, which failed to generalize well on the test set in the case of MFCC for DNN. CNN, on the other hand, has improved the test classification accuracy by 0.5% and 2% for FBE and MFCC respectively. Table 5 shows the frame classification accuracy for fast (V1) and slow (V2) speaking rates for the networks trained on normal (V6) speaking style. Networks trained on FBE achieved better

	/a/	/ɛ/	/ɪ/	/ɔ/	/ʊ/	/aː/	/e/	/i/	/o/	/u/
/a/	**0.85**	0.02	0	0.02	0	0	0	0	0	0
/ɛ/	0	**0.86**	0.03	0	0	0	0	0	0	0
/ɪ/	0	0.07	**0.81**	0	0	0	0.03	0	0	0
/ɔ/	0.08	0	0	**0.77**	0.02	0	0	0	0	0
/ʊ/	0	0	0	0.05	**0.81**	0	0	0	0	0.03
/aː/	0.16	0	0	0	0	**0.78**	0	0	0	0
/e/	0	0.05	0.32	0	0	0	**0.53**	0.03	0	0
/i/	0	0	0.12	0	0	0	0.03	**0.81**	0	0
/o/	0	0	0	0.03	0.14	0	0	0	**0.68**	0.07
/u/	0	0	0	0	0.14	0	0	0	0.08	**0.73**

(a) Fast speaking rate

	/a/	/ɛ/	/ɪ/	/ɔ/	/ʊ/	/aː/	/e/	/i/	/o/	/u/
/a/	**0.83**	0	0	0.02	0	0.06	0	0	0	0
/ɛ/	0	**0.9**	0.02	0	0	0	0	0	0	0
/ɪ/	0	0.08	**0.76**	0	0	0	0.04	0.02	0	0
/ɔ/	0.02	0	0	**0.83**	0.02	0.02	0	0	0	0
/ʊ/	0	0	0	0.05	**0.77**	0	0	0	0.03	0.05
/aː/	0.02	0	0	0.02	0	**0.92**	0	0	0	0
/e/	0	0.02	0.09	0	0	0	**0.76**	0.07	0	0
/i/	0	0	0.03	0	0	0	0	**0.93**	0	0
/o/	0	0	0	0	0.02	0	0	0	**0.85**	0.09
/u/	0	0	0	0	0.02	0	0	0	0.06	**0.87**

(b) Normal speaking rate

	/a/	/ɛ/	/ɪ/	/ɔ/	/ʊ/	/aː/	/e/	/i/	/o/	/u/
/a/	**0.54**	0	0	0.04	0	0.33	0	0	0	0
/ɛ/	0	**0.83**	0.03	0	0	0	0.02	0	0	0
/ɪ/	0	0.06	**0.54**	0	0	0	0.13	0.12	0	0
/ɔ/	0	0	0	**0.78**	0.04	0.05	0	0	0.03	0
/ʊ/	0	0	0	0.04	**0.56**	0	0	0	0.14	0.18
/aː/	0	0	0	0	0	**0.93**	0	0	0	0
/e/	0	0	0.06	0	0	0	**0.7**	0.15	0	0
/i/	0	0	0	0	0	0	0	**0.93**	0	0
/o/	0	0	0	0	0	0	0	0	**0.78**	0.16
/u/	0	0	0	0	0	0	0	0	0.08	**0.88**

(c) Slow speaking rate

Fig. 6: Vowel part of the confusion matrix for different speaking rates with CNN trained on FBE features.

classification accuracy when tested on variable speaking rates. The best performing model was CNN with FBE, which had better accuracy rates for both slow and fast speaking rates. This implies that the CNN can better incorporate the variations in speaking rate, as is evident from Table 5.

Table 4: Frame accuracy rate for training and test of normal speaking style (V6), context size $M = 10$.

	DNN		**CNN**	
	FBE	MFCC	FBE	MFCC
Train	94.81	97.01	95.41	93.96
Test	83.22	74.75	83.76	77.06

Table 5: Frame accuracy rate for fast (V1) and slow (V2) speaking styles on the networks trained on normal speaking style with context size $M = 10$.

SR	**DNN**		**CNN**	
	FBE	MFCC	FBE	MFCC
V1	75.59	67.90	76.65	69.23
V2	74.85	68.97	77.87	71.81

5.2 Confusion matrix

A confusion matrix depicts the performance of the classifier in terms of correctly classified and misclassified labels, which are phonemes in this case. 24 phonemes inventory consists of 10 vowels and 14 consonants. Figures 5 and 6 show the confusion matrices for fast, normal, and slow speaking rates on the FBE features from both DNN and CNN architecture. A quick glimpse at the confusion matrices shows that the DNN and CNN have almost the same performance for different speaking rates. However, if we compare and analyze the confusion matrices for fast and slow speaking rates between DNN in Fig. 5 and CNN in Fig. 6 we can see that, except for /i/, /ɪ/, and /o/ in the case of fast, and /a/ and /o/ in the case of slow, CNN's performance is marginally better than the DNN. Also, CNN better captured (correctly classified) short vowels at the fast speaking rate. This is noticeable when we compare Fig. 6a and Fig. 6c.

Furthermore, by comparing the confusion matrices for the same network but with different speaking rates of the same network, it can be inferred that variation in the speaking rate will lead to confusion between short and long vowels. Looking at Fig. 8a and Fig. 6b, it is obvious that the trained network on MFCC often gets more confused between short and long vowels, for example, vowel /e/ gets confused with /ɪ/ more than

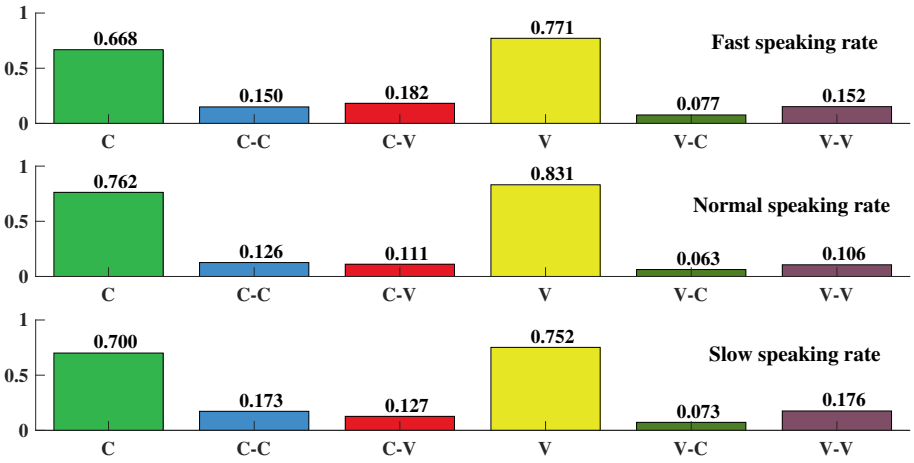

Fig. 7: Rates for correct and incorrect classification of consonants and vowels. Bar (C) shows the correct classification rate for consonants, bar (C-C) shows the misclassification of a consonant as another consonant, bar (C-V) shows the confusion of the consonants with vowels. Bar (V) shows the correct classification rate of vowels, bar (V-C) shows the misclassification of a vowel as a consonant and bar (V-V) shows the confusion of the vowels with another vowel.

	/ɑ/	/ɛ/	/ɪ/	/ɔ/	/ʊ/	/ɑː/	/e/	/i/	/o/	/u/
/ɑ/	**0.73**	0	0	0	0	0.15	0	0	0	0
/ɛ/	0	**0.84**	0.05	0	0	0	0	0	0	0
/ɪ/	0	0.06	**0.67**	0	0	0	0.08	0.08	0	0
/ɔ/	0	0	0	**0.84**	0.02	0.02	0	0	0	0
/ʊ/	0	0	0	0.04	**0.65**	0	0	0	0.09	0.11
/ɑː/	0.06	0	0	0.02	0	**0.87**	0	0	0	0
/e/	0	0.02	0.23	0	0	0	**0.61**	0.09	0	0
/i/	0	0	0.06	0	0	0	0.02	**0.87**	0	0
/o/	0	0	0	0.02	0.07	0	0	0	**0.78**	0.06
/u/	0	0	0	0	0.08	0	0	0	0.07	**0.77**

(a) Vowel part of the confusion matrix for the normal speaking rate with CNN trained on MFCC features.

	/d/	/t/	/g/	/k/	/f/	/s/	/b/	/p/	/v/	/ts/	/m/	/n/	/ʃ/	/l/
/d/	**0.71**	0	0.07	0	0	0	0.06	0.03	0	0	0	0	0	0
/t/	0	**0.87**	0	0	0	0	0	0.03	0	0	0	0	0	0
/g/	0.03	0	**0.7**	0.07	0	0	0.03	0.04	0	0	0	0	0	0
/k/	0	0	0.03	**0.85**	0	0	0	0.03	0	0	0	0	0	0
/f/	0	0	0	0	**0.91**	0.02	0	0	0	0	0	0	0	0
/s/	0	0	0	0	0.05	**0.87**	0	0	0	0	0	0	0.03	0
/b/	0.05	0	0.07	0	0	0	**0.69**	0	0	0	0	0	0	0
/p/	0	0.03	0	0.03	0	0	0	**0.82**	0	0	0	0	0	0
/v/	0	0	0	0	0	0	0	0	**0.74**	0	0	0	0	0
/ts/	0	0	0	0	0	0	0	0	0	**0.91**	0	0	0	0
/m/	0	0	0	0	0	0	0	0	0	0	**0.58**	0.1	0	0.1
/n/	0	0	0	0	0	0	0	0	0	0	0.09	**0.61**	0	0
/ʃ/	0	0	0	0	0	0.07	0	0	0	0	0	0	**0.88**	0
/l/	0	0	0	0	0	0	0	0	0	0	0	0	0	**0.76**

(b) Confusion matrix for the 14 consonants from the CNN trained on the FBE.

Fig. 8: Confusion matrix for the normal speaking rate.

the /i/. Similarly, there is a higher confusion rate for /ʊ/ with /o/ and /u/ in the MFCC case. Confusions was typically between phonemes belonging to the same broad phonetic class, for example between the nasals /m/ and /n/, or the fricatives /s/ and /ʃ/.

In addition, the misclassifications were broken down into two separate categories for both vowels and consonants: confusions within the broad class (e.g., a consonant misclassified as another consonant) and confusions between the classes (e.g., a vowel misclassified as a consonant). Figure 7 shows these performance measures for test data with different speaking rates using FBE features as the input vector with context size $M = 10$ to the CNN. By considering the normal speaking rate as the reference point, we can see that the true classification rate of the consonants in the fast speaking rate is the lowest one, and it is confused more

with vowels. Also, the true vowel classification rate was lowest at the slow speaking rate, with increased vowel confusion.

5.3 F_1 and Kappa score

To evaluate the overall performance of the deep learning models and to compare them across different features, we compute the F_1 and the Cohen's Kappa measure.

F_1 is a harmonic mean of precision and recall and is defined as:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{15}$$

where Precision is given as:

$$Precision = \sum_{i=1}^{L} \frac{w_i \times TP_i}{TP_i + \sum_{j=1}^{L-1} FP_{ij}}, \tag{16}$$

and Recall is given as:

$$Recall = \sum_{i=1}^{L} \frac{w_i \times TP_i}{TP_i + \sum_{j=1}^{L-1} FN_{ij}}, \tag{17}$$

where w_i is:

$$w_i = \frac{\#N_i}{\sum_{j=1}^{L} \#N_j}. \tag{18}$$

L is the number of labels, w_i is the weight for a given class label which are calculated as the number of samples ($\#N_i$) that belongs to the class i divided by the total number of samples, and TP, FP, and FN are the numbers of true positives, false positives, and false negatives respectively,

Table 6 shows various statistical performance measures for the DNN and CNN architectures. If we look at the F_1 score for fast (V1) and slow (V2) speaking rates, we observe that both networks performed better on the slow speaking rate compared to the fast, and FBE on both occasions has an improvement of 7-8% over the MFCC.

Table 6: Precision, Recall, F1, and Kapp score of DNN and CNN for V1, V2 and V6 speaking rates.

SR	**Performance Measure**	**DNN**		**CNN**	
		FBE	MFCC	FBE	MFCC
V1	Precision	78.22	70.42	78.96	71.40
...	Recall	76.31	68.18	77.42	69.60
...	F_1	76.62	68.44	77.68	69.91
...	Kappa	74.95	66.34	76.11	67.84
V2	Precision	76.21	69.54	79.11	72.81
...	Recall	74.86	68.99	78.03	71.97
...	F_1	74.83	68.77	77.87	71.87
...	Kappa	73.29	67.05	76.66	70.23
V6	Precision	83.27	75.27	83.81	77.25
...	Recall	83.13	74.80	83.75	77.07
...	F_1	83.16	74.80	83.74	77.05
...	Kappa	82.13	73.30	82.78	75.70

Since the number of training samples across different categories is not uniform in our dataset, we therefore also presents the Kappa score in addition to F_1 score. Kappa κ represents the classification accuracy normalized by the imbalance of the classes in the data. It is given as:

$$\kappa = \frac{(p_o - p_e)}{1 - p_e}, \tag{19}$$

where p_o is the observed agreement ratio and p_e is the expected agreement, given the labels are assigned randomly. The value of p_e is estimated as given in [3] using a per-annotator empirical prior over the class labels.

The obtained κ scores for various networks configuration as shown in Table 6 were fairly consistent with the F_1 scores. In the case of CNN with FBE there was a drop of only 1%, which shows that the classifiers are able to efficiently handle the data imbalance for all three speaking rates.

6 Conclusion/ future work

One of the challenges facing ASR systems is the variation in the speaking rate which causes changes both in formant frequencies and in their transition tracks. It directly affects the performance of the systems as a word spoken at a normal pace has a better chance of being recognized compared to words spoken at slower or faster paces by the same speaker. To better understand the effects of speaking variabilities, we studied two acoustic features on the task of phoneme classification using different deep learning models. In particular, we used a four hidden-layer feed-forward DNN and CNN. The study also provides an in-depth analysis of the MFCC and FBE features trained using different context sizes on two deep learning models on the normal speaking rate and tested on slow and fast speaking rates. Results were analyzed at micro-level.

Our findings suggest that a four-hidden-layer CNN can classify both short and long vowels slightly better than the DNN with similar network configurations, though the short vowels were better captured by the CNN for the fast speaking utterances. Comparing confusion matrices between different speaking rates also revealed that the variations in speaking rate lead to a higher confusion between short and long vowels even on the same network. The true classification rate of consonants at slow speaking rates is higher than the rate for vowels. For fast speaking rates, consonants become confused more often with vowels. On the other hand, on both CNN and DNN, FBE results were far better than the MFCC. Both models, however, favoured the slow speaking rate where a slight improvement can be observed in terms of both F_1 and Kappa κ scores. Overall, CNN with FBE features performed best for all three speaking rates.

The speaking rate variations may be better treated by recurrent neural networks which handle dynamic temporal behaviour. This may reduce the context size dependent pre-processing step on the speech data. Future work, therefore, may focus on using recurrent neural networks, such as long-short term memory, for analyzing variations in speaking rate.

7 Acknowledgements

This work has been supported by the Research Council of Norway through the project AULUS, and by NTNU through the project ArtiFutt.

References

1. Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G., Yu, D.: Convolutional neural networks for speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing **22**(10), 1533–1545 (2014)
2. Arisoy, E., Sainath, T.N., Kingsbury, B., Ramabhadran, B.: Deep neural network language models. In: Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT, WLM '12, pp. 20–28. Association for Computational Linguistics, Stroudsburg, PA, USA (2012)
3. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. Computational Linguistics **34**(4), 555–596 (2008)

4. Best, C.T., Tyler, M.D., Gooding, T.N., Orlando, C.B., Quann, C.A.: Development of phonological constancy: Toddlers' perception of native- and jamaican-accented words. Psychological Science **20**(5), 539–542 (2009)
5. Brondsted, T., Madsen, J.P.: Analysis of speaking rate variations in stress-timed languages. In: EUROSPEECH–1997, pp. 481–484 (1997)
6. Dahl, G.E., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing **20**(1), 30–42 (2012)
7. Exter, M., Meyer, B.T.: Dnn-based automatic speech recognition as a model for human phoneme perception. In: INTERSPEECH–2016, pp. 615–619 (2016)
8. Faltlhauser, R., Ruske, G., Thomae, M.: Towards the question: Why has speaking rate such an impact on speech recognition performance? In: ICSLP–2002, pp. 2429–2432 (2002)
9. Francis, A.L., Nusbaum, H.C.: Paying attention to speaking rate. In: ICSLP–1996, pp. 1537–1540 vol.3 (1996)
10. Grimaldi, M., Cummins, F.: Speech style and speaker recognition: a case study. In: INTERSPEECH–2009, pp. 920–923 (2009)
11. Laleye, F.A., Ezin, E.C., Motamed, C.: Adaptive decision-level fusion for Fongbe phoneme classification using fuzzy logic and Deep Belief Networks. In: Informatics in Control, Automation and Robotics (ICINCO), 2015 12th International Conference on, vol. 1, pp. 15–24. IEEE (2015)
12. Martinez, F., Tapias, D., Alvarez, J.: Towards speech rate independence in large vocabulary continuous speech recognition. In: Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on, vol. 2, pp. 725–728 vol.2 (1998). DOI 10.1109/ICASSP.1998.675367
13. Martinez, F., Tapias, D., Alvarez, J., Leon, P.: Characteristics of slow, average and fast speech and their effects in large vocabulary continuous speech recognition. In: EUROSPEECH–1997, pp. 469–472 (1997)
14. Meftah, A., Alotaibi, Y.A., Selouani, S.: A comparative study of different speech features for arabic phonemes classification. In: 2016 European Modelling Symposium (EMS), pp. 47–52 (2016). DOI 10.1109/EMS.2016.018
15. Meyer, B., Wesker, T., Brand, T., Mertins, A., Kollmeier, B.: A human-machine comparison in speech recognition based on a logatome corpus. In: Speech Recognition and Intrinsic Variation Workshop (2006)
16. Meyer, B.T., Brand, T., Kollmeier, B.: Effect of speech-intrinsic variations on human and automatic recognition of spoken phonemes. The Journal of the Acoustical Society of America **129**(1), 388–403 (2011)
17. Meyer, B.T., Wächter, M., Brand, T., Kollmeier, B.: Phoneme confusions in human and automatic speech recognition. In: Eighth Annual Conference of the International Speech Communication Association (2007)
18. Mohamed, A.R., Dahl, G.E., Hinton, G.: Acoustic modeling using deep belief networks. IEEE Transactions on Audio, Speech, and Language Processing **20**(1), 14–22 (2012)
19. Pfau, T., Ruske, G.: Creating hidden markov models for fast speech. In: ICSLP–1998, pp. 205–208 (1998)
20. Rozi, A., Li, L., Wang, D., Zheng, T.F.: Feature transformation for speaker verification under speaking rate mismatch condition. In: Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2016 Asia-Pacific, pp. 1–4. IEEE (2016)
21. Seide, F., Li, G., Chen, X., Yu, D.: Feature engineering in context-dependent deep neural networks for conversational speech transcription. In: 2011 IEEE Workshop on Automatic Speech Recognition Understanding, pp. 24–29 (2011). DOI 10.1109/ASRU.2011.6163899
22. Shahrebabaki, A.S., Imran, A.S., Olfati, N., Svendsen, T.: Acoustic feature comparison for different speaking rates. In: Human-Computer Interaction. Interaction Technologies, pp. 176–189. Springer International Publishing (2018)
23. Theodoridis, S.: Chapter 18 - neural networks and deep learning. In: Machine Learning, pp. 875 – 936. Academic Press, Oxford (2015)
24. Varghese, D., Mathew, D.: Phoneme classification using reservoirs with MFCC and RASTA-PLP features. In: 2016 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–6 (2016)
25. Wesker, T., Meyer, B., Wagener, K., Anemüller, J., Mertins, A., Kollmeier, B.: Oldenburg logatome speech corpus (ollo) for speech recognition experiments with humans and machines. In: Ninth European

Conference on Speech Communication and Technology (2005)
26. Wrede, B., Fink, G.A., Sagerer, G.: An investigation of modelling aspects for ratedependent speech recognition. In: EUROSPEECH–2001, pp. 2527–2530 (2001)
27. Xu, M., Zhang, L., Wang, L.: Database collection for study on speech variation robust speaker recognition. Proc. O-COCOSDA (2008)
28. Zeng, X., Yin, S., Wang, D.: Learning speech rate in speech recognition. In: INTERSPEECH-2015, pp. 528–532 (2015)