Speech Recognition Based on Deep Tensor Neural Network and Multifactor Feature

Yahui Shan* and Min Liu* and Qingran Zhan* and Shixuan Du* and Jing Wang* and Xiang Xie *
* Beijing Institute of Technology, Beijing, China
E-mail: yahui_shan@163.com, lqblmin@163.com, asrzhanqingran@gmail.com, jfhz4689@gmail.com, wangjing@bit.edu.cn, xiexiang@bit.edu.cn

Abstract—**This paper presents a speech recognition system based on deep tensor neural network which uses multifactor feature as input feature of acoustic model. First, a deep neural network is trained to estimate articulatory feature from input speech, where the training data is MOCHA database[1]. Mel frequency cepstrum coefficients in conjunction with articulatory feature are used as multifactor feature. Deep tensor neural network which involves tensor interactions among neurons is used as the acoustic model in this system. Speech recognition results indicate that the multifactor feature helps in improving speech recognition performance not only under clean conditions but also under noisy background conditions; deep tensor neural network is more capable of modeling multifactor features because of its tensor interactions than deep neural network.**

I. Introduction

At present, a speech recognizer works well in the quiet environment. But in the noisy environment, the performance of speech recognition system will be greatly degraded. The articulatory features (AFs) reflect the physical location of the articulation organ, and it is less affected by acoustic noise and external environment. Therefore, the AFs are more robust than the acoustic feature naturely. Over the years, the study of the AFs has received extensive attention. According to Liberman [2] from Yale university in the United States, the language sensing system and the language-generating motion system develop together, interact and work together, and the Gesture of pronunciation is the common feature used in the process of language generation and perception. In [3], Mesgarani from University of Calif San Francisco points out that the brain can encode AFs in the superior temporal gyrus through regional response. K. E. Manjunath [4] obtains 74.7% phoneme recognition rate on the TIMIT database using the AF in conjunction with Mel Frequency Cepstrum Coefficient (MFCC). [5] uses articulatory information as an additional input to a fused deep neural network (DNN) and convolutional deep neural network (CNN) acoustic model and reduces the error rate by 12% relative to the baseline in both Switchboard subset and CallHome subset. Ioannis K. Douros [6] demonstrates that articulatory information is helpful for phone recognition. Other experiments have also confirmed that the use of AFs can improve the performance of speech recognition systems [7], [8], [9], [10], [11].

In recent years, tensor modeling as a multifactor analysis method has shown its potential in processing high-order signals and improving neural network modeling capabilities. In terms of features, [12] uses the techniques of nonnegative tensor factorization to propose convolutive nonnegative tensor factorization (CNTF). This algorithm provides considerable improvements for a clean-trained speech recognition system. Qiang Wu etc.[13] proposes a novel speech feature extraction method based on Gabor filtering and tensor factorization which is able to improve the speech recognition performance. Multifactor analysis using tensor provides a potential approch for generating robust features. For neural network modeling using tensor, [14] replaces one of the sigmoid hidden layers in the neural network with a tensor layer, and achieves better results in frame-level phoneme classification. [15] designs a tensor-based DNN that the hidden speaker and environment factors and tied triphone states are jointly approximated. Dong Y. et al. [16] proposes that deep tensor neural network (DTNN) is used for Large Vocabulary Continuous Speech Recognition (LVCSR) and has achieved good results. Considering the tensor in neural network design has the potential to enhance the capability of neural network modeling.

The AFs and acoustic features belong to two types of features in different dimensions, while the tensor neural network is suitable for analyzing the relationship between multiple factors. Inspired by this idea, AFs and tensor are considered to add into a speech recognition system. First, a DNN is trained to estimate AFs of speech signals. We then use it to generate AFs. The AFs in conjunction with MFCCs are used as features for training and testing English LVCSR systems. Experimental results show that AFs can provide complementary information that improves speech recognition performance not only under clean conditions but also under noisy background conditions. And for DTNN, it has a better capability to model multifactor features and noisy speech features than DNN. When the LVCSR system uses the MFCCs along with AFs as input feature, the DTNN provides larger improvement in speech recognition performance than the system that only uses MFCCs as input feature.

II. Dataset

To train a model for estimating AFs from speech, we require a speech dataset containing truth AFs. Very few libraries marked with AFs. The MOCHA (Multi-CHannel Articulatory) database [1] is created to provide a resource for training speaker-independent continuous ASR systems and for general co-articulatory studies. The articulatory channels include Electromagnetic Articulograph (EMA) sensors directly attached to the seven positions which are three positions of tongue (tip, body, dorsum), upper and lower lip, jaw and the velum. The EMA data consists of x and y co-ordinates making 14 coefficients in total. The speech is recorded simultaneously with these articulatory measures. The recording is done at the same studio at the Queen Margaret University College, Edinburgh.

There are 920 English sentences that 1 male and 1 female native speaker record 460 sentences respectively which have been checked. Also, the dataset includes unchecked speech sentences that 3 male and 4 female record. The speech signals are at a sampling rate of 16kHz. 4010 recordings are determined for training the DNN model to estimate AFs. 80% of the data is used as the training set, 10% is used as the cross validation set, and the remaining 10% is used as the test set.

For LVCSR experiments, we use the popular speech database TIMIT. The first to 12th-order MFCCs and energy, along with their first and second temporal derivatives of the speech is extracted for connecting with AFs later. The training set consists of 462 speakers. The development set which is used for tuning contains 50 speakers. Results are reported using the standard 24-speaker core test set consisting of 192 sentences.

III. AF ESTIMATOR

Estimating the AFs of the speech signal is a task where the acoustic feature is used to predict the AFs. DNN has been used for the task [17], [10], [18]. DNN has a strong non-linear transformation capability so that it can learn the correlation between the input and the output. In the training process, the nets are trained with a greedy layer-wise learning. The algorithm means training one layer at a time. Then we use back propagation to fine-tune the network.

We represent the speech using first to 12th-order MFCCs and energy, along with their first and second temporal derivatives as input feature. The speech data is analyzed using a 25-ms Hamming window with a 10-ms fixed frame rate. The AFs of the speech in MOCHA are downsampled to 100Hz to temporally synchronize with MFCCs. The experiment uses a context window of 5 frames. Hence, for the DNN model of AF estimator, the number of input nodes is 195 and the number of output nodes is 14 for EMA data.

IV. ACOUSTIC MODEL

For LVCSR system, DTNN that combined with Hidden Markov Model (HMM) is used as the acoustic model. The architecture of DTNN is shown in subgraph (a) of the Fig. 1. The DTNN develops the regular DNN by replacing one or more layers with double-projection and tensor layer.

It can be seen from subgraph (a) of the Fig. 1, the hidden layer of DTNN h^{l-1} is separated into two parts: h_1^{l-1} and h_2^{l-1}. The dimension of h_1^{l-1} is N_1^{l-1} and h_2^{l-1} is N_2^{l-1}. These two parts are connected with the next hidden layer h^l which is $N^l \times 1$ vector through the three-way tensor u^l of dimension $N_1^{l-1} \times N_2^{l-1} \times N^l$. In the (a), the three-way tensor is represented with a cube. The formula is as follows,

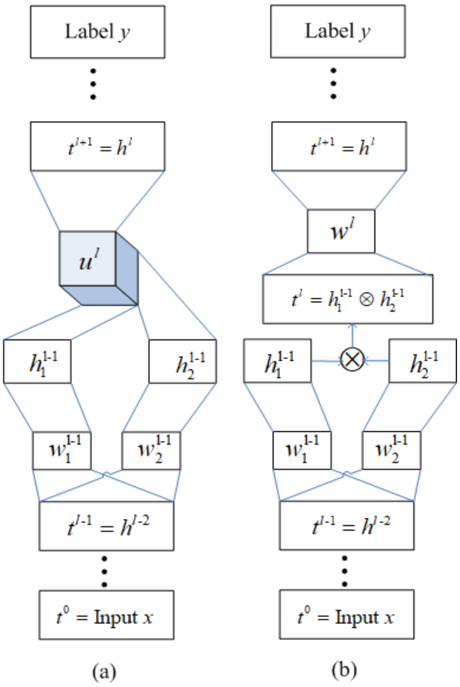

Fig. 1. The architecture of DTNN.

$$h_{(n)}^l = f(\sum_{i,j} u_{i,j,n}^l h_{1(i)}^{l-1} h_{2(j)}^{l-1} + b_{(n)}^l) \tag{1}$$

where i, j, k are indexes of the hidden units in layers h_1^{l-1}, h_2^{l-1}, and h^l, respectively. $f(\cdot)$ is the activation function. The two parts $h_{1(i)}^{l-1}$ and $h_{2(j)}^{l-1}$ learn different information. [16] calls the hidden layer h^{l-1} a double-projection (DP) layer because the information of the h^{l-2} is projected into two separate subspaces at layer h^{l-1} as h_1^{l-1} and h_2^{l-1}. The h^l is connected with h^{l-1} which is DP layer through the tensor u^l. In order to illustrate the operation of DTNN, the input to layer l is represented as t^l.

$$t^l = vec(h_1^{l-1} \otimes h_2^{l-1}) = vec(h_1^{l-1} (h_2^{l-1})^T), \tag{2}$$

where $\otimes$ is the Kronecker product, $vec(\cdot)$ represents the column-vectorized representation of the matrix, and T means transpose. Finally, the tensor layer is shown as follows,

$$h_{(n)}^l = f(\sum_{i} w_{(i,n)}^l t_{(i)}^l + b_{(k)}^l), \tag{3}$$

where w^l is the weight matrix that the tensor u^l is rewritten. b^l is the bias. So the alternative structure of DTNN is shown in subgraph (b) of the Fig. 1.

Fig. 1 shows the DTNN with one DP layer. However, each layer can be DP layer. Fig. 2 shows the structure of the DTNN in which hidden layers are all DP layers.

For DP layers, if the given input is t^l, the activation vector is illustated as (4)

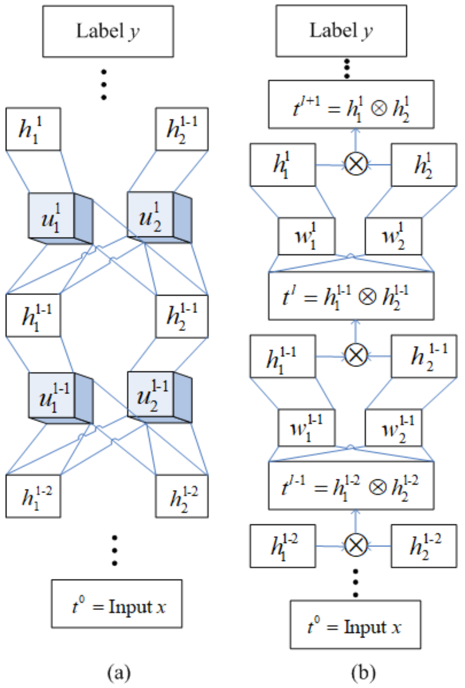

Fig. 2. The alternative architecture of DTNN.

$$z^l(t^l) = (w^l)^T t^l + b^l. \tag{4}$$

The output for the DP layer has two parts,

$$h_i^l = f(z_i^l(t^l)) = f((w_i^l)^T t^l + b_i^l), \tag{5}$$

where $i = 1, 2$ which represents the part number.

The loss function to optimize the DTNN model is (6).

$$E = \frac{1}{N}\sum_x E(x) = \frac{1}{N}\sum_x\sum_y \tilde{p}(y|x) logp(y|x), \tag{6}$$

where N is the number of samples in the training set. $\tilde{p}(y|x)$ is the target probability and $p(y|x)$ is the model's predicted probability. The parameters can be learned using the backpropagation (BP) algorithm. The DTNN can have the conventional layer and DP layer at the same time. For the conventional layer, the error signal

$$e^{l-1}(x) = \frac{\partial E(x)}{\partial t^l} = w^l diag(f'(z^l(t^l)))e^l(x), \tag{7}$$

where $f'(\cdot)$ is the derivative of the activation function.

$$\frac{\partial E(x)}{\partial w^l} = t^l(diag(f'(z^l(t^l)))e^l(x))^T, \tag{8}$$

$$\frac{\partial E(x)}{\partial b^l} = diag(f'(z^l(t^l)))e^l(x), \tag{9}$$

where $diag(\cdot)$ is the diagonal matrix determined by the operand and T represents transposition.

TABLE I

Average R value of different posotion

Position	R
Tongue body	0.9013
Tongue tip	0.8801
Tongue dorsum	0.8832
Upper lip	0.8591
Lower lip	0.8602
Jaw	0.9001
Velum	0.8994

The learning algorithms is more complicated for the DP layers. The gradients needed for BP algorithm in the DP layers are

$$\frac{\partial E(x)}{\partial w_i^l} = t^l(diag(\delta'(z_i^l(t^l)))e_i^l(x))^T, \tag{10}$$

$$\frac{\partial E(x)}{\partial b_i^l} = diag(\delta'(z_i^l(t^l)))e_i^l(x), \tag{11}$$

and

$$e^{l-1}(x) = \sum_{j\in 1,2} w_i^l diag(\delta'(z_i^l(t^l)))e_i^l(x). \tag{12}$$

The derivation process can be found in [16].

We implement the LVCSR system on the Kaldi platform [19]. The input feature is multifactor feature which consists of 13 MFCCs (including energy), along with their first and second temporal derivatives and AFs. Before acoustic model training, multifactor feature is processed using speak-level mean and variance normalization. Triphone HMMs with decision-tree-based state clustering are used to train the acoustic model. The number of fully tied states is 2072. The model uses three left-to-right states per phone and is trained with maximum likelihood estimation. Language model (LM) is used statistical bigram model.

V. Experimental Results

A. The performance of AF estimator

For the AF estimator, the DNN was trained with a greedy layer-wise learning algorithm, where we trained one hidden layer at a time with different numbers of nodes two times and computed the average performance on the cross validation set. The number of nodes in each layer was determined according to average performance as specified before. After the number of nodes in each layer was determined, a final training pass for all the layers was performed. There were four hidden layers with number of nodes, 150, 200, 80 and 40. The activation function of the network was tanh. TABLE 1 shows the average Pearson's product-moment correlation coefficient (R) for each of the AFs from the DNN. The x and y co-ordinates of the same position are averaged to represent the value of the position.

The R values are all above 0.85. Considering the limitation of the amount of data when training the model, the DNN mapping model has a good effect and could be used to extract the AFs of other speech data with unlabeled AF parameters.

B. Benefits of using AFs

For the LVCSR experiments, we used MFCCs in conjunction with AFs as the input feature. Different acoustic models and input features were used to observe the benefits of using AFs. We use one-layer DP for the experiment. The reason will be explained later. The second column of TABLE 2 shows the WER obtained from the different systems on the clean corpus. The results of the first four models show that for GMM-HMM model, using MFCCs+AFs achieves 9.8% relative WER reduction compared with that using MFCCs. And the value is 6.7% for DNN-HMM model. The results means that when using AFs combined with MFCC as the input feature of acoustic model, the system has a better performance. This illustrates that AFs have a positive effect on the improvement of speech recognition performance. And DNN model has a better modeling capability than GMM.

AURORA-4 noises at different SNRs of 0dB, 5dB, 10dB and 20dB are added into the test set for the robust experiment. We choose four kinds of noise. Two are stationary noise, street and babble, and other two are car and airport which are stationary noise. TABLE 2 and TABLE 3 show the WER obtained from the different systems at different SNR.

It can be seen from the first four models in TABLE 2 and TABLE 3, take the babble as an example, using AFs helps to reduce the relative WER by 6.2% for GMM-HMM and 11.7% for DNN-HMM in average. Obviously, the AFs help in improving speech recognition performance not only under clean conditions but also under noisy background conditions when combined with MFCCs, which can in turn prove the AFs are less affected by acoustic noise and external environment.

C. The modeling capability of DTNN

For DTNN, we used the notation in [16] to represent the DP and tensor layers. Take $(32 : 32)$ for an example, $(32 : 32)$ denotes a DP layer with 32 units in each of the two parts. TABLE 4 compares the performance of different DTNN configurations on the word error rate (WER).

From the TABLE 4, we can see that the larger the size of the DP layer, the better the performance. However, as the size of the DP layer increases, the number of the network parameters will increase, so the $(64 : 64)$ is finally chosen. Also, the performance of the model which includes two DP layers is worse than that with one DP layer. This is because much of the information is lost when the features is transformed into 128 dimension in DP layer which is much smaller than 2048 in DNN. We can observe that replacing the top hidden layer with DP layer achieve 4.3% relative WER reduction over the DNN. That's why we choose the configuration $2048 - 2048 - (64 : 64)$ for the experiment.

For the experiment on the clean corpus, the second column of TABLE 2 includes the WER of DTNN-HMM system. DTNN helps to reduce the relative WER by 1% and 4.3% under MFCC and MFCC+AFs features compared with DNN. The DTNN-HMM model with MFCC along with AFs as input feature has the best performance in the all models listed in this paper.

For the anti-noise experiment, we list the WER of DTNN-HMM model in the last two lines in TABLE 2 and TABLE 3. Take the average results of Babble noise with the least performance improvement as an example, the DTNN-HMM model using MFCC+AFs achieves 3.0% relative WER reduction compared with DNN-HMM model using the same feature. However, if the feature is MFCC alone, the value of WER reduction becomes 2.4%. The situation is similar under other conditions. It illustrates that DTNN has potential in modeling multifactor features.

VI. Conclusions

In this paper, we present a speech recognition system based on deep tensor neural network which uses multifactor feature as input feature of acoustic model. A DNN is used to estimate articulatory features from the speech signal. The well-trained DNN model for estimating articulatory features have a good performance to extract the AFs of other speech data with unlabeled AFs. Also, the DTNN which involves tensor interactions among neurons is used as acoustic model of speech recognition. We have presented LVCSR experiments using multifactor feature and DTNN. In terms of input feature of acoustic model, when combined with MFCC, AFs help in improving speech recognition performance under clean as well as noise conditions. In terms of acoustic model, DTNN is a powerful deep architecture capable of modeling multifactor features and noisy speech. The experimental results also demonstrate when the DP layer is placed at the top hidden layer, the DTNN performs best. Our future work will extend multifactor feature and DTNN to more languages.

VII. Acknowledgements

The authors would like to thank the reviewers for their suggestions. This work is partly supported by Natural Science Foundation in China (No.11590772, No.61571044, No.61473041).

References

[1] A. A.Wrench and K. Richmond, "Continuous speech recognition using articulatory data," *Proc Icslp*, pp. 145–148, 2000.

[2] A. M. Liberman, F. S. Cooper, D. P. Shankweiler, and M. Studdertkennedy, "Perception of the speech code," *Psychological Review*, vol. 74, no. 6, p. 431, 1967.

[3] M. Nima, C. Connie, J. Keith, and E. F. Chang, "Phonetic feature encoding in human superior temporal gyrus," *Science*, vol. 343, no. 6174, p. 1006, 2014.

[4] K. E. Manjunath and K. S. Rao, "Improvement of phone recognition accuracy using articulatory features," *Circuits Systems & Signal Processing*, no. 4, pp. 1–25, 2017.

[5] V. Mitra, W. Wen, C. Bartels, H. Franco, and D. Vergyri, "Articulatory information and multiview features for large vocabulary continuous speech recognition," 2018.

[6] I. Douros, I. Douros, A. Katsamanis, and P. Maragos, "Multi-view audio-articulatory features for phonetic recognition on rtmri-timit database," pp. 5514–5518, 2018.

[7] O. Cetin, A. Kantor, S. King, C. Bartels, M. Magimai-Doss, J. Frankel, and K. Livescu, "An articulatory feature-based tandem approach and factored observation modeling," in *IEEE International Conference on Acoustics*, 2007.

[8] J. Frankel, M. Magimai-Doss, S. King, K. Livescu, and zgr etin, "Articulatory feature classifiers trained on 2000 hours of telephone speech," *Proc Interspeech*, pp. 2485–2488, 2007.

TABLE II
WER% OBTAINED FROM THE DIFFERENT SYSTEMS WITH STATIONARY NOISE VS. CLEAN SPEECH

Features and acoustic models	Clean	20dB		10dB		5dB		0dB		Average	
	-	Bab	Street	Bab	Street	Bab	Street	Bab	Street	Bab	Street
MFCC GMM-HMM	24.4	34.5	34.5	54.4	51.8	68.0	62.1	75.8	71.8	58.175	55.05
MFCC+AFs GMM-HMM	22.0	33.1	29.3	50.1	43.3	63.1	55.8	71.9	62.5	54.55	47.75
MFCC DNN-HMM	22.5	25.1	23.7	39.1	34.2	52.6	46.3	65.5	60.3	45.575	41.125
MFCC+AFs DNN-HMM	21.0	23.9	22.1	33.8	29.5	44.1	38.0	59.1	49.5	40.225	34.775
MFCC DTNN-HMM	22.3	24.0	22.8	37.9	33.1	51.3	45.1	64.7	58.7	44.475	39.925
MFCC+AFs DTNN-HMM	20.1	22.7	20.9	33.0	28.4	43.0	36.8	57.4	48.6	39.025	33.675

TABLE III
WER% OBTAINED FROM THE DIFFERENT SYSTEMS WITH NON-STATIONARY NOISE

Features and acoustic models	20dB		10dB		5dB		0dB		Average	
	Car	Airport	Car	Airport	Car	Airport	Car	Airport	Car	Airport
MFCC GMM-HMM	34.3	34.6	54.7	52.7	66.6	62.8	75.8	72.3	57.85	55.6
MFCC+AFs GMM-HMM	29.4	28.7	45.1	45.3	59.4	56.0	66.9	64.8	50.2	48.7
MFCC DNN-HMM	24.4	22.8	36.3	34.2	50.8	46.8	64.9	61.1	44.1	41.225
MFCC+AFs DNN-HMM	23.1	22.0	31.1	30.3	39.4	39.2	52.8	52.3	36.6	35.95
MFCC DTNN-HMM	23.8	22.1	35.1	33.3	49.2	45.3	61.2	59.1	42.325	39.95
MFCC+AFs DTNN-HMM	21.1	20.9	29.5	29.1	37.9	38.0	51.0	51.2	34.875	34.8

TABLE IV
COMPARING THE PERFORMANCE OF DIFFERENT DTNN CONFIGURATIONS ON THE TIMIT

Configuration	WER
CD-GMM-HMM	22.0
DNN 2048-2048-2048	21.0
DTNN (32:32)-2048-2048	20.5
DTNN (32:32)×2-2048	20.7
DTNN (64:64)-2048-2048	20.4
DTNN 2048-(64:64)-2048	20.3
DTNN 2048-2048-(64:64)	20.1
DTNN (96:96)-2048-2048	20.2

[9] K. Kirchhoff, G. A. Fink, and G. Sagerer, "Combining acoustic and articulatory feature information for robust speech recognition," *Speech Communication*, vol. 37, no. 3, pp. 303–319, 2002.

[10] V. Mitra, W. Wen, A. Stolcke, H. Nam, C. Richey, J. Yuan, and M. Liberman, "Articulatory trajectories for large-vocabulary speech recognition," in *IEEE International Conference on Acoustics*, 2013.

[11] V. Mitra, G. Sivaraman, C. Bartels, H. Nam, W. Wen, C. Espy-Wilson, D. Vergyri, and H. Franco, "Joint modeling of articulatory and acoustic spaces for continuous speech recognition tasks," in *IEEE International Conference on Acoustics*, 2017.

[12] S. Mirsamadi and J. H. L. Hansen, "A generalized nonnegative tensor factorization approach for distant speech recognition with distributed microphones," *IEEE/ACM Transactions on Audio Speech & Language Processing*, vol. 24, no. 10, pp. 1721–1731, 2016.

[13] W. Qiang, L. Zhang, and G. Shi, "Robust multifactor speech feature extraction based on gabor analysis," *IEEE Transactions on Audio Speech & Language Processing*, vol. 19, no. 4, pp. 927–936, 2011.

[14] B. Hutchinson, L. Deng, and D. Yu, "A deep architecture with bilinear modeling of hidden representations: Applications to phonetic recognition," in *IEEE International Conference on Acoustics*, 2012.

[15] D. Yu, X. Chen, and L. Deng, "Factorized deep neural networks for adaptive speech recognition," *Proc.int.workshop Statist.mach.learn.speech Process*, 2012.

[16] Y. Dong, D. Li, and F. Seide, "The deep tensor neural network with applications to large vocabulary speech recognition," *IEEE Transactions on Audio Speech & Language Processing*, vol. 21, no. 2, pp. 388–396, 2013.

[17] V. Mitra, H. Nam, C. Y. Espy-Wilson, E. Saltzman, and L. Goldstein, "Articulatory information for noise robust speech recognition," *IEEE Transactions on Audio Speech & Language Processing*, vol. 19, no. 7, pp. 1913–1924, 2011.

[18] V. Mitra, G. Sivaraman, H. Nam, C. Espy-Wilson, and E. Saltzman, "Articulatory features from deep neural networks and their role in speech recognition," in *IEEE International Conference on Acoustics*, 2014.

[19] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.