Locally Normalized Filter Banks Applied to Deep Neural-Network-Based Robust Speech Recognition

Josué Fredes, José Novoa, Simon King, Richard M. Stern, *Fellow, IEEE*, and Nestor Becerra Yoma, *Member, IEEE*

***Abstract*—This letter describes modifications to locally normalized filter banks (LNFB), which substantially improve their performance on the Aurora-4 robust speech recognition task using a Deep Neural Network-Hidden Markov Model (DNN-HMM)-based speech recognition system. The modified coefficients, referred to as LNFB features, are a filter-bank version of locally normalized cepstral coefficients (LNCC), which have been described previously. The ability of the LNFB features is enhanced through the use of newly proposed dynamic versions of them, which are developed using an approach that differs somewhat from the traditional development of delta and delta–delta features. Further enhancements are obtained through the use of mean normalization and mean–variance normalization, which is evaluated both on a per-speaker and a per-utterance basis. The best performing feature combination (typically LNFB combined with LNFB delta and delta–delta features and mean–variance normalization) provides an average relative reduction in word error rate of 11.4% and 9.4%, respectively, compared to comparable features derived from Mel filter banks when clean and multinoise training are used for the Aurora-4 evaluation. The results presented here suggest that the proposed technique is more robust to channel mismatches between training and testing data than MFCC-derived features and is more effective in dealing with channel diversity.**

***Index Terms*—Automatic speech recognition (ASR), Aurora-4, channel mismatch, deep neural network (DNN), locally normalized filter bank (LNFB).**

I. Introduction

The use of deep neural networks (DNNs) has produced major improvements in the recognition accuracy of automatic speech recognition (ASR) systems. DNNs have the ability to learn internal features, which are robust to many sources of variability in speech signals (e.g., [1] and [2]). In this context, simple features like log-Mel filter banks (MelFB) favor the DNN learning process and provide greater recognition accuracy than traditional MFCC features [3], [4]. Nevertheless, when mismatches between training and testing conditions are too large, the learning ability of DNNs is limited and the recognition accuracy degrades significantly [1]. Well-known techniques for reducing the train-test mismatch include the application of input normalization such as mean normalization (MN) and mean and variance normalization (MVN). MN assumes that the mean of data is invariant, and MVN uses the stronger assumption that the mean and variance of data are invariant, so standardizing mean and/or variance removes irrelevant information [5]. These techniques reduce the mismatch between features representing clean and noisy utterances [6], [7], and they reduce distortion introduced by linear filtering or spectral tilt [5]. MN and MVN are also used for DNN input normalization [8] because DNN training is sensitive to input scale [9]. Because un-normalized features with greater variance would dominate the DNN learning process, scaling each dimension of the input data to a similar range improves DNN performance [8]. Another effective approach for reducing training–testing mismatches is to employ multicondition training with noisy or distorted data, so noise is present in both the training and testing data. However, this strategy is not practical in some applications where the channel between the speaker and the ASR system may vary over time. Examples of such applications include human–robot interaction, meeting transcription, lecture transcription, etc.

Locally normalized cepstral coefficients (LNCC) [10] were designed to be robust to channel mismatches, and they provided better accuracy than traditional MFCC features on a speaker verification task with mismatches between training and testing conditions [10], [11]. The development of locally normalized filter bank (LNFB) features is motivated by the observations that the performance of DNN-HMM ASR systems is typically better when spectrogram-like features (such as MelFB parameters) are used, rather than features in a pseudo-cepstral domain (such as MFCC parameters), despite the fact that MFCC coefficients are simply the discrete cosine transform (DCT) of the MelFB parameters. Similarly, LNCC features are the DCT of LNFB features.

In this letter we consider the use of LNFB parameters for the Aurora-4 ASR task. The sections below discuss the application of LNFB features to robust DNN-HMM-based ASR, the development of LNFB delta and delta–delta coefficients, the combination of static and dynamic LNFB parameters, and an analysis and comparison of results obtained using speaker-based and utterance-based based MN and MVN input normalization.

II. LNFB Coefficients and Delta Coefficients

A. LNFB Features

LNCCs are a set of cepstral-type features, inspired by Seneff's generalized synchrony detector (GSD) [12], which

Manuscript received June 30, 2016; revised November 4, 2016; accepted January 14, 2017. Date of publication January 31, 2017; date of current version February 24, 2017. This work was supported by CONICYT-FONDECYT under Project 1151306 and CONICYT-ANILLO under Project ACT 1120. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Peter K. Willett.

J. Fredes, J. Novoa, and N. B. Yoma are with the Speech Processing and Transmission Laboratory, Electrical Engineering Department, University of Chile, Santiago, Chile (e-mail: jfredes@ing.uchile.cl; jose.novoa@ing.uchile.cl; nbecerra@ing.uchile.cl).

S. King is with the Centre for Speech Technology Research, University of Edinburgh, Edinburgh, EH8 9YL, U.K. (e-mail: Simon.King@ed.ac.uk).

R. M. Stern is with the Department of Electrical and Computer Engineering and Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: rms@cs.cmu.edu).

Digital Object Identifier 10.1109/LSP.2017.2661699

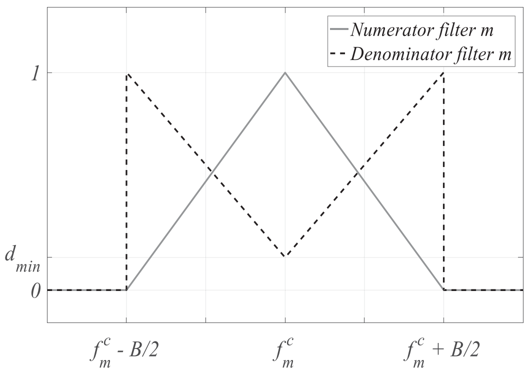

Fig. 1. Graphical representation of the *m*th numerator filter (solid line) and the *m*th denominator filter (dashed line). The frequency axis is in the Bark scale [13] as used in [10].

perform a local normalization in the frequency domain in each auditory channel, and hence are relatively invariant to changes in the frequency response of the transmission channel [10]. Accordingly, LNFB features are LNCC features before the final DCT computation. The local normalization is achieved in the filter-bank space by dividing the output of a triangular frequency-weighting filter (which is similar to the triangular filter in conventional MFCC coefficients) by the output of a second frequency-weighting filter [10]. This normalization removes very coarse variations in the spectral shape that can be considered constant within both filters, such as overall tilt, which we assume arise mostly from channel variability. We refer to these two filters as the "numerator filter" and the "denominator filter," and their shape is an approximation to the frequency response of the numerator and denominator of the Seneff GSD operator

$$\mathrm{Num}_m(f) = \begin{cases} -\frac{2}{B}\left|f - f_m^C\right|, & \text{if } \left|f - f_m^C\right| \le \frac{B}{2} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$\mathrm{Den}_m(f) =$$

$$\begin{cases} \frac{2}{B}\left(1 - d_{\min}\right)\left|f - f_m^C\right| + d_{\min}, & \text{if } \left|f - f_m^C\right| \le \frac{B}{2} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where the frequency variable *f* is in the Bark scale [13]. The shapes of these filters are shown in Fig. 1. Given a channel *m* with center frequency f_m^C and bandwidth *B*, the LNFB feature *m* is defined as the log of the locally normalized energy for channel *m*, LN_m

$$\mathrm{LNFB}_m = \log\left(\mathrm{LN}_m\right) = \log(\mathrm{LNNum}_m/\mathrm{LNDen}_m) \tag{3}$$

where LNNum_m is the numerator filter energy and LNDen_m is the denominator filter energy.

B. Dynamic Features with LNFB

Delta and delta–delta features [14] represent the most common method for capturing the temporal evolution of the short-term spectrum or cepstrum [15]. Delta and delta–delta dynamic features are computed as polynomial approximations of the first- and second-order time derivatives of the static features. If $C_m(n)$ is the static feature *m* at discrete time *n*, the delta features can be expressed as [14]

$$\Delta C_m(n) = \frac{\sum_{k=-K}^{K} kC_m(n+k)}{\sum_{k=-K}^{K} k^2} \tag{4}$$

where *2K + 1* frames centered around frame *n* are used to compute the numeric time derivative at frame *n*. The delta–delta coefficients are obtained by repeating (4) using the delta coefficients as input. Delta and delta–delta features represent the dynamic characteristics of the speech spectra over time, and they are usually employed in combination with static coefficients such as MFCC, LPC, PLP, MelFB, etc.

C. Deltas Delta–Deltas for LNFB Features

Direct application of (4) to the LNFB features in (3) would produce dynamic features according to the following equations:

$$\begin{aligned} \Delta\mathrm{LNFB1}_m(n) &= \Delta\log(\mathrm{LNNum}_m(n)) \\ &\quad - \Delta\log(\mathrm{LNDen}_m(n)) \end{aligned} \tag{5}$$

$$\begin{aligned} \Delta\Delta\mathrm{LNFB1}_m(n) &= \Delta\Delta\log(\mathrm{LNNum}_m(n)) \\ &\quad - \Delta\Delta\log(\mathrm{LNDen}_m(n)). \end{aligned} \tag{6}$$

Because LNFB features have already been normalized, we believe that computing delta and delta–delta LNFB coefficients using (5) and (6), as mentioned earlier, would not represent spectral evolution properly, because independent delta operations would be applied to both the numerator and denominator filter of (1) and (2). It is not clear what the result of such an operation might represent, but it would not be consistent with the original motivation of dynamic features [14], [15]. Instead, we believe that computation of dynamic versions of the LNFB features should be accomplished by applying the linear regression of [14] to the numerator filter only, using the following equations:

$$\Delta\mathrm{LNFB2}_m(n) = \Delta\log(\mathrm{LNNum}_m(n)) \tag{7}$$

$$\Delta\Delta\mathrm{LNFB2}_m(n) = \Delta\Delta\log(\mathrm{LNNum}_m(n)). \tag{8}$$

The superiority of this approach to computing the $\triangle$LNFB features is confirmed by experimental results described later. This strategy could be generalized easily to other self-normalizing features.

III. Data Normalization in DNN-HMM-Based ASR

MN and MVN are widely used to achieve robust ASR. In applying these approaches to a DNN-HMM based system, the means and/or variances could be evaluated over the training set only or over both the training and testing data, and on a per-speaker or per-utterance basis (e.g., [3], [4], [16], and [17]). The optimal normalization could be dependent on the task and on the degree of mismatch between training and testing conditions due to the combined effect of these normalizations (acoustic compensation and scale normalization). In this letter, four input normalization schemes are considered: MN and MVN applied on a per-speaker and on a per-utterance basis, normalizing both the training and testing data in all cases.

Fig. 2 compares the amplitude distributions of MelFB and LNFB coefficients with no normalization, with MN, and with MVN. The plots compare the normalized histograms for filters with center frequencies of approximately 470 Hz using the

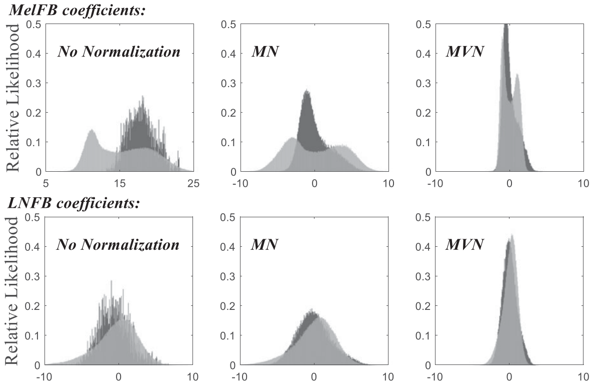

Fig. 2. Histograms for filters with center frequencies of approximately 470 Hz, from the Aurora-4 clean train set (in lighter gray) and the Aurora-4 Group D test subset (in darker gray), for (upper row) MelFB and (lower row) LNFB features. No normalization (left column), mean normalization (center column), and MVN (right column) were applied to training and test data on a per-speaker basis.

Aurora-4 clean train set and the Aurora-4 Group D degraded test set (as defined later), which includes both additive noise and channel distortion. The MN and MVN normalizations were applied on an utterance-by-utterance basis. As can be seen in Fig. 2, the clean and noisy histograms of MelFB coefficients are bimodal or multimodal while those of the LNFB coefficients are unimodal. After applying MN or MVN, the clean and noisy histograms remain clearly distinguishable from each other using MelFB features, despite the fact that they are scaled to similar ranges of variation. In contrast, the clean and noisy histograms from LNFB coefficients are unimodal and very similar after applying MN or MVN. While not shown in this letter, this behavior is observed for the vast majority of frequency channels. This result suggests that LNFB should be more robust than MelFB to additive noise and channel distortion and that MN and MVN should be more effective for the LNFB coefficients. We believe that the unimodal nature of the LNFB histograms is a consequence of the normalization of the numerator by the denominator filter energy according to (3).

IV. Experiments

The effectiveness of the LNFB features was validated by performing ASR experiments on the Aurora-4 corpus [18] using the Kaldi Speech Recognition Toolkit [19]. Three training sets from Aurora-4 were employed: the clean, multinoise and multiconditions. Each training set contains 7137 utterances from 83 speakers. The clean training set contains only clean data recorded with the high-quality Sennheiser HMD-414 microphone. The multinoise set contains clean and artificially degraded utterances with 6 different noises added at SNRs between 10 and 20 dB. All data in this training set were recorded with the Sennheiser HMD-414 microphone. Finally, half the multicondition training set was recorded by the Sennheiser HMD-414 microphone, while the other half was recorded by one out of 18 different microphones, with noise added as in the multinoise data. The evaluation database is composed of 14 testing sets with 330 utterances each, clustered in four groups, as summarized in Table I [18].

TABLE I
Description of Aurora-4 Testing Data Sets [18]

Test Set	Microphone	Noise	Group
1	Sennheiser HMD-414	Clean	A
2	Sennheiser HMD-414	Car	B
3		Babble	
4		Restaurant	
5		Street	
6		Airport	
7		Train	
8	different types of mic.	Clean	C
9	different types of mic.	Car	D
10		Babble	
11		Restaurant	
12		Street	
13		Airport	
14		Train	

Three sets of features were evaluated: static and dynamic Mel Filterbank features labeled as MelFB+ΔMelFB; static and dynamic LNFB parameters, using (5) and (6), labeled as LNFB+ΔLNFB1; and static LNFB features combined with dynamic LNFB parameters based on the numerator filters only, using (7) and (8), labeled as LNFB+ΔLNFB2. Note that for the rest of this letter, we use the labels $\triangle$MelFB, $\triangle$LNFB1, and $\triangle$LNFB2 for compactness; the appropriate delta–delta coefficients are always incorporated into all the three features. The DNN-HMM models were trained making use of the same data alignment obtained with a GMM-HMM recognition system trained in clean conditions employing MFCC features, linear discriminant analysis (LDA), and maximum likelihood linear transforms (MLLT), according to the tri2b Kaldi Aurora-4 recipe [19]. This recipe begins by training a uniphone system, uses alignments from that system for an initial triphone system, and finally uses those triphone alignments to train the final triphone system.

In a previous optimization step, the DNN-HMM baseline system with multicondition training was tested with 24, 32, 40, and 56 MelFB filters. The lowest WER, 10.9%, was obtained with 40 filters using MN on a per-speaker basis. This WER is very competitive with current results in the literature for the same task (e.g., [3], [4], and [20]–[25]). The number of LNFB filters was also set to 40. The filter bandwidth for each channel, *B*, was set equal to 5.2 Barks. The spacing between contiguous filters is a function of the number of filters and is approximately equal to 0.40 Barks. Each DNN in the DNN-HMM systems consisted of seven hidden layers and 2048 units per layer. The number of units of the output layer equaled the number of pdfs of the corresponding GMM-HMM system. The Aurora-4 databases dev_330_01 and dev_330 were used as DNN cross-validation data for clean and multicondition training, respectively. For noisy training, datasets from dev_330_01 to dev_330_07 were employed for cross validation. Results are shown in Tables II, III, and IV.

V. Discussion

A. Dynamic Coefficients Based on the Numerator Filters

Table II compares the WER averaged over all test data for the three feature sets described above. MN was applied on a per-speaker basis. In all cases, the combination of static LNFB features with delta and delta–delta parameters based

TABLE II
COMPARISON OF ALGORITHMS FOR DELTA FEATURES FOR AURORA-4 DATA

Training	MelFB + ΔMelFB	LNFB + ΔLNFB1	LNFB + ΔLNFB2
Clean	32.66	49.21	36.73
Multinoise	16.00	18.42	16.06
Multicondition	10.90	13.97	12.26

TABLE III
DEPENDENCE OF WER FOR AURORA-4 ON MN AND MVN

Train Type	Feature Set	No Norm	MN per Spk	MVN per Spk	MN per Utt	MVN per Utt
Clean	MelFB + ΔMelFB	49.95	32.66	30.74	28.74	27.12
	LNFB + ΔLNFB2	51.10	36.73	28.63	33.38	24.03
Multinoise	MelFB + ΔMelFB	18.11	16.00	17.12	13.97	14.73
	LNFB + ΔLNFB2	16.13	16.06	14.85	14.55	13.35
Multicond.	MelFB + ΔMelFB	11.95	10.90	11.93	10.23	10.62
	LNFB + ΔLNFB2	12.27	12.26	12.11	11.62	11.18

TABLE IV
SUMMARY OF RESULTS FROM AURORA-4 TEST SETS

Training	Aurora Group	Mel FB + ΔMelFB	LNFB + ΔLNFB2
Clean (as in Group A)	A	2.39	2.65
	B	19.70	19.26
	C	21.69	**14.10**
	D	39.55	**34.02**
	Average	27.12	24.03
Multinoise (as in Group B)	A	2.56	3.19
	B	**6.11**	6.94
	C	16.57	**11.84**
	D	25.06	**21.70**
	Average	14.73	13.35
Multicondition (as in Group D)	A	3.42	3.62
	B	**6.35**	7.19
	C	7.12	7.38
	D	16.68	17.06
	Average	10.62	11.18
All-Condition Average		17.5	16.2

on the numerator filter energy, as defined in (6) and (7), LNFB+ΔLNFB2, leads to average relative reductions in WER equal to 25.4%, 12.8% and 12.2% for clean, noisy, and multicondition training, respectively, when compared with LNFB+ΔLNFB1. These results indicate clearly that the dynamic $\triangle$LNFB2 coefficients based only on the numerator filters provide more useful information than coefficients based directly on the original $\triangle$LNFB1 features.

B. Effect of Input Normalization

Table III compares averaged WERs over all of the Aurora-4 test sets, broken out by training type, obtained using MVN or MN applied by utterance and by speaker. As can be seen, MN and MVN always lead to lower WERs when applied utterance by utterance rather than speaker by speaker. (The Kaldi toolkit applies MN speaker by speaker by default.) The average reduction in WER provided by normalization per utterance compared to normalization per speaker is 9.1% for MN and 11.8% for MVN. This could be due to the fact that the SNR in the Aurora-4 database changes from one utterance to the next.

Table III also shows that the use of MVN rather than MN provides more effective normalization in the DNN-HMM system with the LNFB+$\triangle$LNFB2 features, with the difference being particularly dramatic in the case of clean training. In contrast, MN was more effective than MVN when MelFB features were used in conjunction with multinoise and multicondition training. This may be because LNFB features always produce unimodal histograms (see Fig. 2), and hence, the use of MVN is more helpful than with MelFB features.

C. Comparison With MelFB Features

Table IV compares results obtained using the LNFB + $\triangle$LNFB2 and MelFB+$\triangle$MelFB features, broken out according to training and testing conditions, all with MVN invoked per utterance. The best feature set for a pair of train/test conditions is indicated in bold when the difference is statistically significant at the level of $p = .001$ or better, according to the NIST matched pairs test for sentence segment word error (MAPSSWE) [26].

In interpreting Table IV, we focus on the subset of comparisons that are statistically significant. With clean and multinoise training, we believe that the LNFB+$\triangle$LNFB2 features provided superior performance for Aurora Groups C and D because of differences in the microphones used between the training and testing data. The only train/test combinations for which MelFB+$\triangle$MelFB features provided significantly better performance than LNFB+$\triangle$LNFB2 features were for the Group B testing data with multinoise and multicondition training. For the former case, training and testing conditions were completely matched, so no benefit was expected from LNFB+$\triangle$LNFB2 features. In the latter case, the training data had multiple microphones and additive noise while the testing data had additive noise only. In this case, we suspect that any potential benefit from channel mismatch would be vitiated by the dominance of the matching noise. In general, the results of Table IV confirm our previous observations that LNFB coefficients are especially effective in compensating for the effects of channel mismatches, as had been demonstrated previously for the related LNCC features in speaker verification [10].

The global WER for LNFB-based features 7.4% relative smaller than the average WER obtained from MelFB features.

VI. CONCLUSION

A filter-bank version of LNCCs, LNFB, is described and applied to the Aurora-4 robust DNN-HMM-based speech recognition task. It is shown that the “Delta” and “Delta–Delta” versions of the LNFB features should be developed from the numerator term only in the LNFB expression. In addition, it is shown that MVN is more effective than MN for the LNFB features. The relative global WER over all conditions for LNFB features was 7.4% smaller than the average WER obtained using MelFB features. These results indicate that LNFB features provide better recognition accuracy for DNN-HMM ASR systems compared to Mel filterbank features, and that they are especially helpful in providing robustness to channel mismatches between training and testing data.

REFERENCES

[1] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks—Studies on speech recognition tasks," in *Proc. Int. Conf. Learning Representations*, 2013, Scottsdale, AZ, USA, pp. 1–9. [Online]. Available: https://arxiv.org/abs/1301.3605

[2] S. Haykin, "Multilayer perceptrons," in *Neural Networks and Learning Machines*, 3rd ed. London, U.K.: Pearson Prentice Hall, 2009.

[3] T. Drugman, Y. Stylianou, L. Chen, X. Chen, and M. Gales, "Robust excitation-based features for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, 2015, Brisbane, Qld., Australia, pp. 4664–4668.

[4] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Processing*, 2013, Vancouver, BC, Canada, pp. 7398–7402.

[5] Y. Obuchi and R. M. Stern, "Normalization of time-derivative parameters using histogram equalization," in *Proc. INTERSPEECH*, 2003, Lyon, France, pp. 665–668.

[6] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Commun.*, vol. 25, no. 1, pp. 133–147, 1998.

[7] X. Xiao, J. Li, E. S. Chng, H. Li, and C.-H. Lee, "A study on the generalization capability of acoustic models for robust speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 6, pp. 1158–1169, Aug. 2010.

[8] D. Yu and L. Deng, "Deep neural networks," in *Automatic Speech Recognition: A Deep Learning Approach*. London, U.K.: Springer, 2015, pp. 57–76.

[9] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade,* 2nd ed. Berlin, Germany: Springer, 2012, pp. 9–48.

[10] V. Poblete *et al.*, "A perceptually-motivated low-complexity instantaneous linear channel normalization technique applied to speaker verification," *Comput. Speech Language*, vol. 31, no. 1, pp. 1–27, 2015.

[11] J. Fredes, J. Novoa, V. Poblete, S. King, R. Stern, and N. B. Yoma, "Robustness to additive noise of locally-normalized cepstral coefficients in speaker verification," in *Proc. INTERSPEECH*, 2015, Dresden, Germany, pp. 3011–3015.

[12] S. Seneff, "A joint synchrony/mean-rate model of auditory speech processing," *J. Phonetics*, vol. 16, pp. 55–76, 1988.

[13] E. Zwicker and E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency," *J. Acoustical Soc. Amer.*, vol. 68, pp. 1523–1525, 1980.

[14] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoustics, Speech Signal Process.*, vol. 34, no. 1, pp. 52–59, Feb. 1986.

[15] B. Gold, N. Morgan, and D. Ellis, "Feature extraction for ASR," in *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, 2nd ed. Hoboken, NJ, USA: Wiley, 2011, pp. 301–318 [Online Library].

[16] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 12, pp. 1713–1725, Dec. 2014.

[17] Y. Miao, H. Zhang, and F. Metze, "Distributed learning of multilingual DNN feature extractors using GPUs," in *Proc. INTERSPEECH*, 2014, Singapore, pp. 830–834.

[18] G. Hirsch, "Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task, version 2.0, AU/417/02," *ETSI STQ Aurora DSR Working Group*, 2002.

[19] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, Dec. 2011, Waikoloa, HI, USA, N° EPFL-CONF-192584.

[20] B. Li and K. C. Sim, "A spectral masking approach to noise-robust speech recognition using deep neural networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 8, pp. 1296–1305, Aug. 2014.

[21] Z. Tang, D. Wang, and Z. Zhang, "Recurrent neural network training with dark knowledge transfer," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, 2016, Shanghai, China, pp. 5900–5904.

[22] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 12, pp. 2263–2276, Dec. 2016.

[23] A. Bayestehtashk, I. Shafran, and A. Babaeian, "Robust speech recognition using multivariate copula models," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, 2016, Shanghai, China, pp. 5890–5894.

[24] H. B. Sailor and H. A. Patil, "Unsupervised deep auditory model using stack of convolutional RBMs for speech recognition," in *Proc. INTERSPEECH*, 2016, San Francisco, CA, USA, pp. 3379–3383.

[25] S. Kundu, K. C. Sim, and M. Gales, "Incorporating a generative front-end layer to deep neural network for noise robust automatic speech recognition," in *Proc. INTERSPEECH*, 2016, San Francisco, CA, USA, pp. 2359–2363.

[26] D. S. Pallett, W. M. Fisher, and J. G. Fiscus, "Tools for the analysis of benchmark speech recognition tests," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 1990, Albuquerque, NM, USA, pp. 97–100.