INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Generative Noise Modeling and Channel Simulation for Robust Speech Recognition in Unseen Conditions

Meet Soni, Sonal Joshi, Ashish Panda

TCS Innovations Lab
Yantra Park, Thane, Mumbai, India
{meet.soni, sonals.joshi, ashish.panda}@tcs.com

Abstract

Multi-conditioned training is a state-of-the-art approach to achieve robustness in Automatic Speech Recognition (ASR) systems. This approach works well in practice for seen degradation conditions. However, the performance of such system is still an issue for unseen degradation conditions. In this work we consider distortions due to additive noise and channel mismatch. To achieve the robustness to additive noise, we propose a parametric generative model for noise signals. By changing the parameters of the proposed generative model, various noise signals can be generated and used to develop a multi-conditioned dataset for ASR system training. The generative model is designed to span the feature space of Mel Filterbank Energies by using band-limited white noise signals as basis. To simulate channel distortions, we propose to shift the mean of log spectral magnitude using utterances with estimated channel distortions. Experiments performed on the Aurora 4 noisy speech database show that using noise types generated from the proposed generative model for multi-conditioned training provides significant performance gain for additive noise in unseen conditions. We compare our results with those from multi-conditioning by various real noise databases including environmental and other real life noises.

Index Terms: speech recognition, noise robustness, noise design, multi-conditioned training

1. Introduction

Modern ASR systems still perform poorly in various degradation conditions such as channel distortion, presence of additive noise and reverberation etc. [1]. Such performance degradation limits their deployment in real world environments. Various researchers have studied the robustness of modern ASR systems either by employing enhancement front-ends [2–7] or various model adaptation techniques [8–11].

Front-end techniques employ an enhancement front-end that is trained on a parallel corpus of degraded-clean speech utterances. Many neural network architectures such as Deep Neural Network (DNN) [12], Recurrent Neural Network (RNN) [4, 5], Time-Delay Neural Network (TDNN) [6] etc. have been used as enhancement front-ends. Such front-ends have been shown to perform well in the case of seen noisy conditions. However, they perform poorly when unseen noisy conditions are encountered [10, 12], especially if they are not trained jointly with the acoustic model. Joint training of the acoustic model and the enhancement front-end is reported to provide better robustness in the case of unseen noise conditions [13]. Such approaches require parallel noisy-clean speech corpora to pre-train the enhancement front-end, which may not be available in many realistic scenarios and the training time of the resultant complex network is formidable.

Another approach is to train the acoustic model using degraded speech data, generally referred to as multi-conditioned training. This approach is widely used to achieve robustness against additive noise, channel mismatch [10, 12, 14] and reverberation [15, 16] etc. As is noted in [17], adding noise to the training data is a form of regularization during training, which provides better generalization capabilities. Multi-conditioned training has been shown to provide better performance in the case of unseen degradation conditions than the enhancement front-ends [10, 12]. However, the performance in unseen conditions lags behind the performance obtained in seen conditions [12, 13]. In [17], it has been shown that the performance in the unseen condition is dependent on the types of noises used in the training phase. Hence, multi-conditioning with one set of degradation conditions does not guarantee a good performance in all test conditions. This leads to the question: what kind of degradation conditions should one consider while designing the multi-conditioned training dataset to yield good performance in unseen conditions? Generally, this question has been addressed by using a large number of noise signals to create multi-conditioned database to obtain robustness against additive noise [18]. Similarly, a large number of room impulse responses have been used during training, to obtain robustness against reverberation [16]. However, these studies do not report how the performance of the ASR system changes as the degradation conditions are changed in the multi-conditioned database. The large number of degradation conditions used to create the multi-conditioned database renders such a study impractical.

In this work, we address the above question in a novel way without using a large number of degradation conditions. The focus of this work is performance degradation due to unseen additive noise and channel mismatch. We handle additive noise by designing a parametric model for noise generation and by studying the effect of the parameters of the model on ASR performance. The parametric model is designed to span the entire feature-space of Mel Filterbank Energies (MFBEs) by using band-limited white noise signals as the bases. Since the recognition is performed in the feature space, it is possible to generate any signal in the MFBE domain by changing the model parameters. We can sample time-domain signals using the generative model by putting certain constraints on the model parameters. Since the model is parametric and spans the entire feature space, it is relatively easy to study how differently designed multi-conditioned datasets affect the performance in unseen conditions simply by changing the model parameters. For channel degradations simulation, we estimate channel response means using pairs of clean and channel distorted utterances. We add the estimated channel means to clean utterances to simulate channel distortions. We generate a multi-conditioned training set using the additive noise generated by the proposed

http://dx.doi.org/10.21437/Interspeech.2019-2090

model and by simulating channel distortion using the proposed method. The beauty of this work lies in using a set of 24 different band-limited white noise signals to obtain a performance in unseen conditions that surpasses the performance of systems trained on even 100 different noise types.

2. Generating and adding the noise

2.1. Generative model for noise

In the state-of-the-art ASR systems, the feature representation of speech signal, such as Mel Frequency Cepstral Coefficients (MFCCs) or log-Mel Filterbank Energies (MFBEs) are used for acoustic modelling. Hence, the final decision regarding the spoken word is taken using the feature domain representation. Here, while designing the generative model, we take clues from the feature representation of the speech signal that is used to develop ASR system.

We consider the MFBEs to design our generative model. MFBEs are computed by filtering the magnitude of Short-time Fourier Transform (STFT) using an overlapping Triangular Filterbank (TFB). Filters of the TFB have linearly spaced centre frequencies and equal bandwidth on Mel scale. We decompose the magnitude of TFB filters in two parts to formulate our generative model. The b^{th} filter in TFB can be decomposed in left and right part where left part covers first half of the bandwidth in Mel scale and the right part covers the second half. Mathematically, these responses can be written as:

$$TFB_b^L(f) = \begin{cases} 0 & \text{if } f < F_b^L \text{ and } f > F_b^C \\ \frac{f - F_b^L}{F_b^C - F_b^L} & \text{if } F_b^L \leq f \leq F_b^C, \end{cases} \tag{1}$$

and

$$TFB_b^R(f) = \begin{cases} \frac{F_b^H - f}{F_b^H - F_b^C} & \text{if } F_b^C < f \leq F_b^H \\ 0 & \text{if } f \leq F_b^C \text{ and } f > F_b^H, \end{cases} \tag{2}$$

where $TFB_b^L(f), TFB_b^R(f)$ are the frequency response of left half and the right half of the b^{th} filter in TFB. F_b^L, F_b^C, and F_b^H are lower frequency, centre frequency, and higher frequency of b^{th} filter in TFB, respectively.

Now consider a band-limited STFT magnitude of a signal $n_b(t)$ defined as:

$$N_b(t, f) \begin{cases} = 0 & \text{if } f < F_b^L \text{ and } f > F_b^C \\ > 0 & \text{if } F_b^L \leq f \leq F_b^C, \end{cases} \tag{3}$$

where $N_b(t, f)$ is the STFT magnitude of a signal $n_b(t)$. Eq. 3 suggests that $N_b(t, f)$ is the band-limited with the lower frequency F_b^L and higher frequency F_b^C. Here, F_b^L and F_b^C are the lower frequency and centre frequency of the b^{th} filterbank in TFB. The signal $n_b(t)$ is strictly band-limited between F_b^L and F_b^C. If there are such signals for each band b in TFB, then the energy in each filter of TFB can we written as:

$$\begin{aligned} FBE(t, b) = \quad & \alpha_b^L(t) \textstyle\sum_f N_b(t, f) \times TFB_b^L(f) \\ & + \alpha_b^R(t) \textstyle\sum_f N_{b+1}(t, f) \times TFB_b^R(f) \end{aligned} \tag{4}$$

where $FBE(t, b)$ is the energy of b^{th} filterbank for t^{th} frame of the signal, and α_b^L(t) and α_b^R(t) are coefficients corresponding to left and right part of the filter magnitude response of the b^{th} filter in TFB and frame t. Here, the values of α_b^L(t) and α_b^R(t) are chosen such that $FBE(t, b) >= 0$. If we consider the property of overlapping TFB used for computing FBE where $F_b^C = F_{b+1}^L$ and $F_b^H = F_{b+1}^C$, then it can be said that Eq. 4 spans the entire space of FBEs. The energy in each band can be written as the linear combination of different $N_b(t, f)$ and its dot product with the TFB filter responses. Here each $n_b(t)$ is band-limited with non-overlapping bands. Hence, by varying the values of $\alpha_b^L(t)$ and $\alpha_b^R(t)$ for different $N_b(t, f)$s, we can generate the FBEs of any signal. We can use this generative model to sample FBEs of any signal and use it as the additive noise source.

2.2. Sampling noises from the generative model

Now, we consider how to sample the noises using the generative model described in Eq. 4. For that, we focus on designing the time-domain signals $n_b(t)$ that has the magnitude response as per Eq. 3. As per Eq. 3, $n_b(t)$ has to be band-limited according to lower and centre frequencies of b^{th} filter in the TFB. Theoretically, any band-limited signal can be used. We use band-passed white noise signals as $n_b(t)$. We take a white noise frame and multiply its DFT with the rectangular window with the frequencies in the range as per in Eq. 3. Then we use overlap-add method to convert frames into a continuous time-domain signal. We get total $B + 1$ such band-limited signals, where B is the total number of bands in the TFB. These signals $n_b(t)$ have the required magnitude response $N_b(t, f)$.

After getting $n_b(t)$ for different bands, we can add them in time domain with different values of α in Eq. 4. Here, while adding the signals in time-domain will put the constraint $\alpha_b^R(t) = \alpha_{(b+1)}^L(t)$ since one $n_b(t)$ is responsible for FBEs in two filters, and cannot be added with two magnitudes in the time domain. Here, while adding noises in the time domain, we consider one $\alpha_b(t)$ per time frame t for each $n_b(t)$. With this constraint the Eq. 4 no longer spans the entire FBE space, however still significant number of noise types can be generated. Moreover, we consider putting constraints on $\alpha_b(t)$ in time-domain also. We constrain $\alpha_b(t)$ to not vary in the time-domain for a defined number of consecutive frames K, i.e., $\alpha(t) = \alpha(t+k)$ for $k \in (1, 2, 3, ..., K)$. Here, the different values of K is chosen for different utterances. This is done to simulate the stationary behaviour of some noises that do not change randomly in time domain.

In addition to this, we put one more constraint on the values of $\alpha_b(t)$. For one frame at time t, we make few of the $\alpha_b(t)$ very small compared to other values. It is done to simulate noises that has more energy only in few sub-bands. With this constraints we can generate different noise signals with different values of $\alpha_b(t)$. The entire noise sampling process can be parameterized using two parameters. To simulate the stationary behaviour of noises, we divide an utterance with length L in N segments. And for each segment, we keep the values of $\alpha_b(t)$ constant. Moreover, to make some bands to have very small magnitude (order of 10^{-3}), we select maximum of M bands per segment randomly. We sample values of $\alpha_b(t)$ for these bands uniformly from $0.1 - 1$. For other $B - M$ bands, the values are sampled from $0 - 10^{-3}$. With different value of N for each utterance, and different value of M for each segment, different types of noise can be generated using the proposed model. We use these noises to develop multi-conditioned dataset, which is, in turn, used to develop ASR system. The optimal values of M and N are determined eperically on the developement data.

3. Channel simulation

Channel distortions are convolutive in nature. Hence, we process the signals in log STFT magnitude domain where they have additive effect. To simulate the channel effect, we multiply the magnitude of clean speech STFT by an estimate of the channel magnitude response by adding them in log domain. The proposed channel simulation technique shifts the mean of log STFT magnitude of clean signal to that of an unknown utterance with channel distortion. The channel simulation scheme can be describes as follows:

$$Y(t,f) = X(t,f) + \hat{C}(t,f), \tag{5}$$

where t is the frame index, f is the frequency index, $X(t,f)$ is the log STFT magnitude of clean signal, $\hat{C}(t,f)$ is the log STFT magnitude of an estimated channel distortion, and $Y(t,f)$ is the log STFT magnitude of channel simulated signal. $\hat{C}(t,f)$ is estimated using following formula:

$$\hat{C}(t,f) = \frac{1}{T}\sum_{t=0}^{T-1}\hat{Y}(t,f) - \frac{1}{T}\sum_{t=0}^{T-1}\hat{X}(t,f), \tag{6}$$

where T is the total number of frames, $\hat{Y}(t,f)$ is the log magnitude of channel distorted signal, $\hat{X}(t,f)$ is the log magnitude of corresponding clean signal. The signal is converted in time domain using inverse STFT with the phase of the clean signal. Individual frames are converted in a continuous time-domain signal with overlap-add method.

4. Experiments

4.1. Database description

All the experiments were performed on Aurora-4 database [19]. Aurora-4 is a medium vocabulary database used for noise robust continuous speech recognition task. It contains speech data in the presence of additive noises and linear convolutional (channel) distortions. It contains two training sets. One is clean training set consisting of 7138 utterances recorded by the primary Sennheiser microphone. The other one is time-synchronized multi-conditioned training set. One half of the utterances were recorded by the primary Sennheiser microphone while the other half were recorded using one of the secondary microphones. Both halves include a combination of clean speech (893 utterances) and speech corrupted by one of six different noises (street, train station, car, babble, restaurant, airport) at 10-20 dB SNR (2676 utterances). Two test sets consists of 330 utterances from 8 speakers, which was recorded by the primary microphone and a set of secondary microphones, respectively. Each set was then corrupted by the same six noises used in the training set at 5-15 dB SNR, creating a total of 14 test sets. These 14 test sets were grouped into 4 subsets: clean (Set 1, denoted by A), noisy (Set 2 to Set 7, denoted by B), clean with channel distortion (Set 8, denoted by C), noisy with channel distortion (Set 9 to Set 14, denoted by D). Moreover, 100 utterances are chosen from validation set available with Aurora 4 for tuning the parameters associated with proposed generative model. These utterances were similarly recorded and corrupted by conditions used in test set. Hence, giving 1400 total validation utterances.

We perform two sets of experiments. In the first set, we use multi-conditioned data by only adding additive noises. To compare the performance of different noise types, we created different training sets by adding different types of noise in the clean utterances of Aurora 4. We added the noises designed using the proposed generative model to generate multi-conditioned dataset. We generated noises with the different values of number of segment per utterance (5, 10, 15, 20) and maximum bands per segment (5, 10, 15, 20, 24). The noise signals from the proposed generative model were added with the SNR of 0-15 dB with 5 dB of increment, following [13]. We also derived two additional multi-conditioned training set by adding 100 types of environmental noises [20], and 11 types of noises from Noisex noise database [21] following the same SNR scheme. We removed the babel noise from Noisex since it is present in the test set and our goal was to test the system performance in unseen noise conditions. We retain the clean utterances recorded by both set of microphones available with Aurora-4 multi-conditioned data. The resulting training set had 893 clean utterances recorded using primary microphone, 893 clean utterances recorded using secondary microphones, and 5352 utterances with additive noise recorded using primary microphone.

In second set of experiments we consider channel plus additive noise distortions. Aurora-4 dataset does not provide clean utterances recorded using secondary microphones corresponding to the utterances corrupted with channel plus additive noise. Hence, to simulate channel degradations, we first apply the proposed channel distortion technique and then add the noise signals. To estimate the channel distortions, we use 893 clean utterances recorded by secondary microphones and their counterparts recorded using primary microphone. We use Eq. 6 to estimate channel responses corresponding to 893 utterances. Note that all the channel responses are computed from the training set only. Then we randomly select a channel response and apply it using Eq. 5 to clean utterances. Then we add aforementioned noises to channel distorted signals. We get 2676 utterances corrupted by channel and additive noise using this method. We evaluated the performance of all these ASR systems on the test set of Aurora 4. All the noises and channel distortions in test dataset can be considered unseen noise conditions.

4.2. ASR system description

For ASR system building we use DNN-HMM acoustic model. First, we develop GMM-HMM system using 13 dimensional MFCCs features in Kaldi [22] using the WSJ recipe. The GMM-HMM system is trained on clean data. Then the alignments of clean data is used to develop DNN-HMM system on multi-conditioned data. DNNs were implemented in Tensorflow. The DNN had 7 hidden layers with 2048 hidden units and ReLU activation. The input layer had 11-frame context of 24 dimensional log-MFBEs with their delta and delta-delta features. Hence, the input layer has ($11 \times 72 = 792$) units. The output layer had 3088 softmax units, corresponding to 3088 tied states of HMMs. Input features were normalized to have zero mean and unit variance. Moreover, the utterance-level mean normalization was also used as suggested in [23]. The network was trained with random initialization for 30 epochs. The batch size of 256 was and the learning rate was scheduled to decrease linearly as per training epochs. For first 20 epochs, the learning rate was decreased from initial 0.001 to final 0.0001. The final learning rate of 0.0001 was kept constant for remaining epochs. The network was trained using Stochastic Gradient Descent (SGD) with Adam optimizer.

4.3. Results

The system trained on the Aurora 4 multi-conditioned data gives **11.48%** Word Error Rate (WER) on the test set. This result represents the performance of multi-conditioned data in seen

Table 1: *WER (%) on validation set using noises generated from proposed model with different number of segments (N) and maximum bands per segment (M) without any channel distortions.*

N	M	A	B	C	D	Avg
5	5	17.17	24.03	19.92	27.44	24.71
10	5	16.73	24.73	19.74	28.17	25.28
15	5	16.98	24.79	19.42	28.2	25.31
20	5	16.17	24.9	19.11	28.83	25.58
5	**10**	18.11	24.01	19.86	27.31	**24.7**
10	10	17.48	24.98	20.11	27.64	25.24
15	10	17.23	24.64	19.86	28.05	25.23
20	10	16.6	25.1	18.98	28.21	25.39
5	15	17.48	24.91	19.24	27.15	24.93
10	15	17.54	25.3	19.55	28.21	25.58
15	15	16.48	25.28	18.98	27.84	25.3
20	15	17.17	25.03	20.24	27.84	25.33
5	20	17.23	25.25	19.42	27.87	25.38
10	20	17.29	25.08	20.05	28	25.42
15	20	16.85	25.64	19.17	28.19	25.64
20	20	16.98	25.39	19.17	28.22	25.38

Table 2: *WER (%) on validation set using additive noises generated from proposed model and channel distortions simulated using the proposed method.*

N	M	A	B	C	D	Avg
5	5	17.42	24.99	19.49	27.78	25.25
10	5	16.6	24.51	18.36	27.61	24.83
15	5	17.42	24.66	19.99	28.45	25.43
20	5	16.79	24.68	19.11	28.28	25.26
5	**10**	17.61	23.93	19.36	26.58	**24.29**
10	10	17.17	24.52	19.11	27.58	24.92
15	10	16.98	24.39	19.74	28.29	25.2
20	10	17.36	24.91	19.36	27.58	25.12
5	15	17.04	24.49	19.61	27.22	24.78
10	15	17.36	24.5	19.17	27.21	24.77
15	15	17.11	25.1	19.99	27.77	25.31
20	15	17.42	25.39	19.36	28.14	25.57
5	20	17.67	25.06	19.99	27.84	25.36
10	20	17.36	25.07	19.8	27.94	25.38
15	20	16.6	25.45	19.11	27.87	25.4
20	20	17.61	25.48	19.67	28.1	25.63

conditions. It is better than the ones reported in [23] for the same feature-set and network architecture, which shows that the training scheme employed in this paper works well.

Table 1 shows the performance of ASR systems trained using noises generated from the proposed generative model with different number of segments per utterance (N) and number of maximum bands per segment (M) on development set. The results show that the performance of ASR system changes with the change in the above two parameters. By changing the number of segments per utterance, we can control the possible number of different noise conditions. With more number of segments per utterance, we can increase the number of noise conditions in the database. And by changing the number of maximum bands per segment we can control the frequency regions in which noise is added. Table 2 shows the results with channel simulations introduced using the method proposed in Section 3. Although we have conducted experiments with $M > 20$ and $N > 20$, we could not report those due to the space constraint. However, they follow the trend reported in here. In both cases the best performance is obtained with $M = 10$ and $N = 5$. Moreover, the proposed channel simulation technique improved the performance on test set D, as expected.

Table 3: *WER (%) on Aurora-4 test set in unseen conditions. Results are shown for multi-conditioning using noises generated by proposed generative model, 100 environmental noises [20] and Noisex [21] noises. Results with channel simulation are denoted by (condition + C).*

Noise source	A	B	C	D	Average
Proposed	4.58	9.7	11.48	23.38	**14.80**
Environment	4.02	8.79	9.76	25.4	15.38
Noisex	4.08	10.19	9.98	24.29	15.34
Proposed + C	4.51	9.73	10.81	21.62	**14.00**
Environment + C	4.25	9.26	10.57	24.2	14.88
Noisex + C	4.43	10.12	10.29	23.1	14.66

Table 3 shows the results for ASR systems trained using various multi-conditioned training methods on Aurora-4 test set. The results are shown for multi-conditioned data generated by proposed generative model, 100 types of environmental noises and noises from Noisex database. The results are shown for systems trained with and without channel simulation. The results of proposed method is shown for $M = 10$ and $N = 5$. In the case of only additive noises without channel distortions, the system trained environmental noises gives 15.38% WER. The system trained on Noisex database gives 15.34% WER. While the system trained on only white noise (not shown in the table) gave 20.64% WER. The proposed generative model gives 14.80% WER, which is an improvement over both. This result is all the more remarkable, since it was achieved with only 24 band limited white noise signals.

Introducing the proposed channel simulations further improved the performance of all the systems by a substantial margin. The environmental noises provide 14.88% WER, while Noisex provides 14.66% WER. The noises from the proposed generative model outperfom both with 14% WER. The substantial improvement in test set D can be observed due to the proposed channel simulation technique. This shows the effectiveness of the proposed generative noise model and the channel simulation technique in unseen conditions.

5. Conclusions

We presented a generative model for noise signals and a channel simulation technique to design multi-conditioned training dataset to make ASR systems robust. The generative noise model is designed in the Mel filterbank domain and spans the entire space of MFBEs. We achieve this by representing energy in each filter of the TFB as the linear combination of dot product of TFB filter response of the left half and the right half with the STFT magnitude of appropriate band-limited signals. We use band-limited white noise as our basis to span the entire MFBE space. For channel simulation we use means of channel response estimated using a pair of clean and distorted utterances. We sample noises from our generative model by constraining the model parameters and use this noises to develop multi-conditioned datasets for ASR system training with channel distortion added using proposed method. We compare the results obtained using proposed designed dataset with those from datasets containing real life noises. The proposed method is a better alternative to randomly adding large number of noises in the training set for unseen conditions. Moreover, the proposed channel simulation technique further improves the performance of the systems in the case of channel distortion with additive noise.

6. References

[1] V. Mitra, H. Franco, R. M. Stern, J. Van Hout, L. Ferrer, M. Graciarena, W. Wang, D. Vergyri, A. Alwan, and J. H. Hansen, "Robust features in deep-learning-based speech recognition," in *New Era for Robust Speech Recognition*. Springer, 2017, pp. 187–217.

[2] K. Janod, M. Morchid, R. Dufour, G. Linares, and R. De Mori, "Denoised bottleneck features from deep autoencoders for telephone conversation analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 9, pp. 1809–1820, 2017.

[3] V. Mitra, H. Franco, C. Bartels, J. van Hout, M. Graciarena, and D. Vergyri, "Speech recognition in unseen and noisy channel conditions," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5215–5219.

[4] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust asr," in *Proc. Interspeech*, 2012.

[5] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1996–2000.

[6] C.-T. Do and Y. Stylianou, "Improved automatic speech recognition using subband temporal envelope features and time-delay neural network denoising autoencoder," *Proc. Interspeech 2017*, pp. 3832–3836, 2017.

[7] B. Das and A. Panda, "Robust front-end processing for speech recognition in noisy conditions," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5235–5239.

[8] V. Mitra and H. Franco, "Leveraging deep neural network activation entropy to cope with unseen data in speech recognition," *arXiv preprint arXiv:1708.09516*, 2017.

[9] V. Mitra, G. Sivaraman, H. Nam, C. Espy-Wilson, E. Saltzman, and M. Tiede, "Hybrid convolutional neural networks for articulatory and acoustic information based speech recognition," *Speech Communication*, vol. 89, pp. 103–112, 2017.

[10] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

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

[12] J. Du, Q. Wang, T. Gao, Y. Xu, L.-R. Dai, and C.-H. Lee, "Robust speech recognition with speech enhanced deep neural networks," in *Proc. Interspeech*, 2014.

[13] Y. Qian, M. Yin, Y. You, and K. Yu, "Multi-task joint-learning of deep neural networks for robust speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 310–316.

[14] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks-studies on speech recognition tasks," *arXiv preprint arXiv:1301.3605*, 2013.

[15] L. Couvreur, C. Couvreur, and C. Ris, "A corpus-based approach for robust asr in reverberant environments," in *Sixth International Conference on Spoken Language Processing*, 2000.

[16] R. Hsiao, J. Ma, W. Hartmann, M. Karafiát, F. Grézl, L. Burget, I. Szöke, J. H. Černockỳ, S. Watanabe, Z. Chen *et al.*, "Robust speech recognition in unknown reverberant and noisy conditions," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 533–538.

[17] S. Yin, C. Liu, Z. Zhang, Y. Lin, D. Wang, J. Tejedor, T. F. Zheng, and Y. Li, "Noisy training for deep neural networks in speech recognition," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, p. 2, 2015.

[18] Y. Wang, A. Misra, and K. K. Chin, "Time-frequency masking for large scale robust speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[19] D. Pearce and J. Picone, "Aurora working group: Dsr front end lvcsr evaluation au/384/02," *Inst. for Signal & Inform. Process., Mississippi State Univ., Tech. Rep*, 2002.

[20] G. Hu, "100 nonspeech environmental sounds, 2004." [Online]. Available: http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html

[21] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech communication*, vol. 12, no. 3, pp. 247–251, 1993.

[22] 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," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[23] A. Narayanan and D. Wang, "Joint noise adaptive training for robust automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2504–2508.