FACTORED SPATIAL AND SPECTRAL MULTICHANNEL RAW WAVEFORM CLDNNS

Tara N. Sainath, Ron J. Weiss, Kevin W. Wilson, Arun Narayanan, Michiel Bacchiani

Google, Inc., New York, NY, USA
{tsainath, ronw, kwwilson, arunnt, michiel}@google.com

ABSTRACT

Multichannel ASR systems commonly separate speech enhancement, including localization, beamforming and postfiltering, from acoustic modeling. Recently, we explored doing multichannel enhancement jointly with acoustic modeling, where beamforming and frequency decomposition was folded into one layer of the neural network [1, 2]. In this paper, we explore factoring these operations into separate layers in the network. Furthermore, we explore using multi-task learning (MTL) as a proxy for postfiltering, where we train the network to predict "clean" features as well as context-dependent states. We find that with the factored architecture, we can achieve a 10% relative improvement in WER over a single channel and a 5% relative improvement over the unfactored model from [1] on a 2,000-hour Voice Search task. In addition, by incorporating MTL, we can achieve 11% and 7% relative improvements over single channel and unfactored multichannel models, respectively.

1. INTRODUCTION

While state-of-the-art speech recognition systems perform reasonably well in close-talking microphone conditions, performance degrades in conditions when the microphone is far from the user due to degradation from reverberation and additive noise. To improve recognition, these systems often use multiple microphones to enhance the speech signal and reduce the impact of reverberation and noise [3, 4].

Multichannel ASR systems often use separate modules to perform recognition. First, microphone array speech enhancement is applied, typically via localization, beamforming and postfiltering. Then this enhanced signal is passed to an acoustic model [5, 6]. One widely used technique is delay-and-sum beamforming [4], in which signals from different microphones are first aligned in time to adjust for the propagation delay from the target speaker to each microphone. The time-aligned signals are then summed to enhance the signal from the target direction and to attenuate noise coming from other directions. This "spatial filtering" provides signal enhancement by directional selectivity, and additional signal enhancement can be obtained from "spectral filtering". Commonly used filter optimizations are based on Minimum Variance Distortionless Response (MVDR) [7, 8] and multichannel Wiener filtering (MWF) [3].

Instead of having separate modules for multichannel enhancement and acoustic modeling, performing both jointly has shown benefits, both for Gaussian Mixture Models [9] and more recently for neural networks [1]. In the latter paper, we trained neural networks to operate directly on raw multichannel waveforms using a single layer of multichannel "time convolution" filters [1], each of which independently filtered each channel of the input and then summed the outputs in a process analogous to filter-and-sum beamforming. The filters in this multichannel filterbank learned to do spatial and spectral filtering jointly.

In multichannel speech recognition systems, multichannel spatial filtering is often performed separately from single channel feature extraction. With this in mind, this paper investigates explicitly factorizing these two operations as separate layers in a neural network. The first layer in our proposed "factored" raw waveform CLDNN model consists of short-duration multichannel time convolution filters which map multichannel inputs down to a single channel, with the idea that the network might learn to do broadband spatial filtering in this layer. By learning several filters in this "spatial filtering layer", we hypothesize that the network can learn filters for multiple different look directions in space. The single channel waveform output of each filter in this spatial filtering layer is passed to a longer-duration time convolution "spectral filtering layer" intended to perform finer frequency resolution spectral decomposition analogous to a time-domain auditory filterbank as in [10]. The output of this spectral filtering layer is passed to a convolutional, long short-term memory, deep neural network (CLDNN) acoustic model [11].

It is common to apply a nonlinear postfilter to further enahance the linear beamformer output [12]. There have been numerous techniques studied to do speech enhancement with neural networks, including auto-encoders [13], time-frequency masking [14], and multi-task learning (MTL) [15]. We explore MTL as a form of regularization because it does not increase the number of operations performed during decoding. We modify the network architecture described above to contain two outputs, one which predicts context-dependent states and another which predicts clean log-mel features. Gradients from these layers are weighted appropriately. In this work, we explore whether MTL can improve performance on top of the improvements obtained from the factored multichannel processing in the network.

2. FACTORED NETWORK ARCHITECTURE

The proposed multichannel raw waveform network mimics filter-and-sum beamforming, a generalization of delay-and-sum beamforming which filters the signal from each microphone using a finite impulse response (FIR) filter and then sums them. Using similar notation to [9], filter-and-sum enhancement can be written as follows:

$$y[t] = \sum_{c=0}^{C-1} \sum_{n=0}^{N-1} h_c[n] x_c[t - n - \tau_c] \quad (1)$$

where $h_c[n]$ is the n^{th} tap of the filter associated with microphone c, $x_c[t]$, is the signal received by microphone c at time t, τ_c is the steering delay induced in the signal received by a microphone to align it to the other array channels, and $y[t]$ is the output signal generated by the processing. C is the number of microphones in the array and N is the length of the FIR filters.

2.1. Multichannel Raw Waveform Filterbank

Enhancement algorithms optimizing the model in Equation 1 require an estimate of the steering delay τ_c obtained from a separate localization model and will obtain filter parameters by optimizing an objective such as MVDR. In contrast, our aim is to exploit the spatial filtering capabilities of a multichannel filter without explicitly estimating a steering delay and to do so by directly optimizing acoustic modeling performance. Different steering delays can be modeled using a bank of P fixed filters. The output of filter $p \in P$ can be written as follows:

$$y^p[t] = \sum_{c=0}^{C-1} \sum_{n=0}^{N-1} h_c^p[n] x_c[t-n] \quad (2)$$

where the steering delay for each microphone is implicitly absorbed into the filter parameters $h_c^p[n]$.

[2, 1] learned such filters within a neural network, where the first layer modeled Equation 2 and performed multichannel time-convolution with a FIR filterbank $h_c = \{h_c^1, h_c^2, \dots h_c^P\}$ where $h_c \in \Re^{N \times P}$ for $c \in 1, \dots C$. The output after the convolution was then max-pooled across time to give a degree of short term shift invariance, and then passed through a compressive non-linearity. As shown in [1], the time convolution layer learned to do both spatial and spectral filtering. The output of the time convolution layer was passed to a CLDNN [11] and the whole network was trained jointly.

2.2. Factored Multichannel Filterbank

In our proposed network, shown in Figure 1, we factor out spatial and spectral filtering into separate layers. The motivation for this architecture is to design the first layer to be spatially selective, while implementing a frequency decomposition shared across all spatial filters in the second layer. Thus the combined output of the second layer will be the Cartesian product of all spatial and spectral filters.

The first layer, denoted by `tConv1` in the figure, again models Equation 2 and performs a multichannel time-convolution with a FIR spatial filterbank. First, we take a small window of the raw waveform of length M samples for each channel C, denoted as $\{x_1[t], x_2[t], \dots, x_C[t]\}$ for $t \in 1, \cdots, M$. Each channel c is convolved by a filter with N taps, and there are P such filters $h_c = \{h_c^1, h_c^2, \dots, h_c^P\}$. We stride the convolutional filter by 1 in time across M samples and perform a "same" convolution, such that the output for each convolutional filter remains length M. Finally, the outputs from each channel are summed to create an output feature of size $y^p[t] \in \Re^{M \times 1 \times P}$ where the dimensions correspond to time (sample index), frequency (spatial filter index), and look direction (feature map index), respectively.

The operation of each filter $p \in P$ can again be interpreted as a filter-and-sum beamformer, except it does not first shift the signal in each channel by an estimated time delay of arrival (TDOA). The main differences between the multichannel approach in [1] and the proposed factored approach are as follows. First, both the filter size N and number of filters P are much smaller in order to encourage the network to learn filters with a broadband response in frequency that span a small number of spatial look directions needed to cover all possible target speaker locations. We will show that the shorter filters in this layer will have worse frequency resolution than those in [1], but that will be dealt with in the next layer. In addition, instead of doing a convolution so the output of the convolution has fewer samples than the input, we perform a "same" convolution so the length of the output matches the input. In addition, this linear filter is not followed by any non-linear compression (i.e. ReLU, log). Finally, we do not perform any pooling on the output of this layer. We hope that this will encourage the network to use this layer to perform spatial filtering only with a limited spectral response.

The second time-convolution layer, denoted by `tConv2` in the figure, consists of longer-duration single-channel filters. It therefore can learn a decomposition with better frequency resolution than the first layer but is incapable of doing any spatial filtering. Given the P feature maps from the first layer, we perform a time convolution on each of these signals, very similar to the single-channel time-convolution layer described in [10], except that the time convolution is shared across all P feature maps or "look directions". We denote this layer's filters as $g \in \Re^{L \times F \times 1}$, where 1 indicates sharing across the P input feature maps. The "valid" convolution produces an output $w[t] \in \Re^{M-L+1 \times F \times P}$. Next, we pool the filterbank output in time thereby discarding short-time (i.e. phase) information, over the entire time length of the output signal, to produce an output of dimension $1 \times F \times P$. Finally, we apply a rectified non-linearity, followed by a stabilized logarithm compression[1], to produce a frame-level feature vector at time t, i.e. $x_t \in \Re^{1 \times F \times P}$. We then shift the window of the raw waveform by a small (10ms) hop and repeat this time convolution to produce a set of time-frequency-direction frames at 10ms intervals.

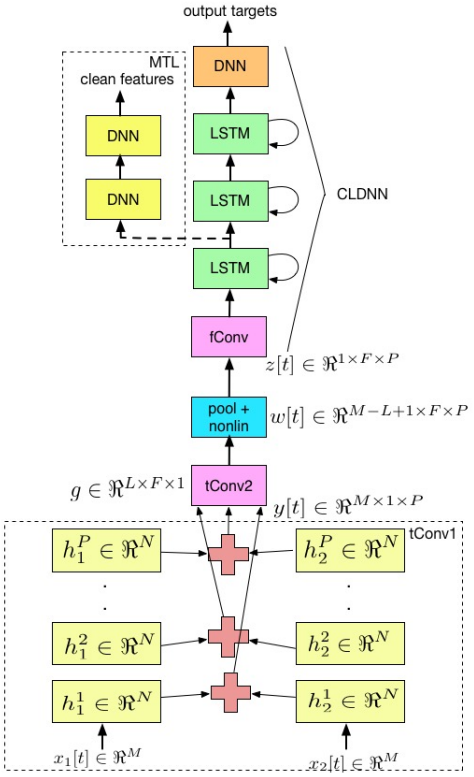

Fig. 1: Factored multichannel raw waveform CLDNN architecture for P look directions. The figure shows two channels for simplicity.

The output out of the time convolutional layer (`tConv2`) produces a frame-level feature, denoted as $z[t] \in \Re^{1 \times F \times P}$. This feature is then passed to the CLDNN block of Figure 1. First, the `fConv` layer applies frequency convolution to $z[t]$. This layer has 256 filters of size $1 \times 8 \times 1$ in time-frequency-direction. Our pooling strategy is to use non-overlapping max pooling along frequency, with a pooling size of 3 [16]. The output of the frequency convolution layer is then passed to a 256-dimensional linear low-rank layer. The output of this

[1]We use a small additive offset to truncate the output range and avoid numerical problems with very small inputs: $\log(\cdot + 0.01)$.

is passed to 3 LSTM layers with 832 cells and a 512 unit projection layer, and then one DNN layer with 1,024 hidden units. More details about the CLDNN architecture can be found in [10, 11].

The time convolution layers are trained jointly with the rest of the CLDNN. During training, the raw waveform CLDNN is unrolled for 20 time steps for training with truncated backpropagation through time. In addition, the output state label is delayed by 5 frames, as we have observed that information about future frames improves the prediction of the current frame [11].

2.3. Multi-task Learning

To further improve noise robustness, we apply MTL by configuring the network to have two outputs, one which predicts context-dependent (CD) states and the other "denoising" output which predicts clean features, similar to [15]. The latter output is only used during training to regularize the remaining network parameters; the denoising output and associated layers are not evaluated during inference. One novel aspect of our work is that we explore MTL in a multichannel setting.

The additional denoising layers are shown in the `MTL` block of Figure 1. In this example, we have shown the denoising subnetwork branching off of the first LSTM layer, though we also explore having this branch at different parts of the network to see if performance can be improved by performing denoising closer to the input. The MTL module is composed of two DNN layers followed by a linear low-rank layer to predict clean log-mel features. We do not predict the clean waveform since it contains extra fine time structure detail that is irrelevant to the recognition task and would therfore be more difficult to optimize. During training the gradients backpropagated from the CD and MTL outputs are weighted by α and $1-\alpha$ respectively.

3. EXPERIMENTAL DETAILS

3.1. Data

Our experiments are conducted on about 2,000 hours of noisy training data consisting of 3 million English utterances. This data set is created by artificially corrupting clean utterances using a room simulator, adding varying degrees of noise and reverberation. The clean utterances are anonymized and hand-transcribed voice search queries, and are representative of Google's voice search traffic. Noise signals, which include music and ambient noise sampled from YouTube and recordings of "daily life" environments, are added to the clean utterances at SNRs ranging from 0 to 20 dB, with an average of about 12 dB. Reverberation is simulated using the image model [17] – room dimensions and microphone array positions are randomly sampled from 100 possible room configurations with RT_{60}s ranging from 400 to 900 ms, with an average of about 600 ms. The simulation uses an 8-channel linear microphone array, with inter-microphone spacing of 2 cm. Both noise and target speaker locations change between utterances; the distance between the sound source and the microphone array is chosen between 1 to 4 meters. The speech and noise azimuths were uniformly sampled from the range of ± 45 degrees and ± 90 degrees, respectively, for each noisy utterance.

Our evaluation set consists of a separate set of about 30,000 utterances (over 20 hours). The simulated set is created similarly to the training set under similar SNR and reverberation settings. Care was taken to ensure that the room configurations, SNR values, T60 times, and target speaker and noise positions in the evaluation set are not identical to those in the training set, although the microphone array geometry between the training and simulated test sets is identical.

3.2. Acoustic model details

The CLDNN architecture and training setup follow a similar recipe to [1, 10]. The input window size for the raw waveform is 35ms ($M = 560$) at a sampling rate of 16kHz. The first layer `tConv1` filters are 5ms in length ($N = 80$), and we have P such filters, which we vary. The filters are initialized to be an impulse centered at a delay of zero for channel 0, and offset from zero in channel 1 by different delays for each filter. This amounts to performing delay-and-sum filtering across a set of fixed look directions. The second layer `tConv2` follows [1], with $F = 128$ filters each 25ms ($L = 400$) long. This layer is initialized using the Glorot-Bengio strategy [18].

Single channel models are trained using signals from channel 1, while $C = 2$ channel models use channels 1 and 8 (14 cm spacing). All neural networks are trained with the cross-entropy (CE) criterion, using asynchronous stochastic gradient descent (ASGD) optimization [19]. Sequence training experiments also use ASGD [20]. All networks have 13,522 CD output targets. The weights for the `fConv` and DNN layers are initialized with Glorot-Bengio, while the LSTM layers are randomly sampled from a uniform distribution between ± 0.02. We use an exponentially decaying learning rate, which starts at a value of 0.004 and decays by 0.1 over 15 billion frames.

4. RESULTS

In sections 4.1 and 4.2 we explore the impact of the spatial/spectral filter factoring. Specifically, we explore the impact of the numer of look directions and the spatial/spectral response of the filters learned. Subsequently, in 4.3 we explore how multi-task learning, akin to post filtering, further enhances the learned filter structure.

4.1. Number of Spatial Filters

We begin by exploring the behavior of the proposed factored multichannel architecture as the number of spatial filters P varies. Table 1 shows that we get good improvements up to 10 spatial filters. We did not explore above 10 filters due to the computational complexities of passing 10 feature maps to the `tConv2` layer. The factored network, with 10 spatial filters, achieves a WER of 20.4%, a 6% relative improvement over the baseline 2 channel multichannel raw-waveform CLDNN from [1]. It is important to note that since the `tConv2` layer is shared across all look directions P, the total number of parameters is actually less than the architecture in [1].

# Spatial Filters P	WER
baseline 2 ch, raw [1]	21.8
1	23.6
3	21.6
5	20.9
10	**20.4**

Table 1: WER when varying the size of `tConv1`, 2 channel input.

4.2. Filter Analysis

To better understand what the `tConv1` layer learns, Figure 2 plots two-channel filter coefficients and the corresponding spatial responses, or beampatterns, after training. The beampatterns show the magnitude response in dB as a function of frequency and direction of arrival, i.e. each horizontal slice of the beampattern corresponds to the filter's magnitude response for a signal coming from a particular

direction. In each frequency band (vertical slice), lighter shades indicate sounds from those angles are passed through, while darker shades indicate directions whose energy is attenuated.

Despite the intution described in Section 2.2, the first layer filters appear to perform both spatial and spectral filtering. However, the beampatterns can nevertheless be categorized into a few broad classes. For example, filters 2, 3, 5, 7, and 9 in Figure 2 only pass through some low frequency subbands below about 1.5 kHz, where most vowel energy occurs, but steered to have nulls in different directions. Very little spatial filtering is done in high-frequency regions, where many fricatives and stops occur. The low frequencies are most useful for localization because they are not subject to spatial aliasing and because they contain much of the energy in the speech signal; perhaps that is why the network exhibits this structure.

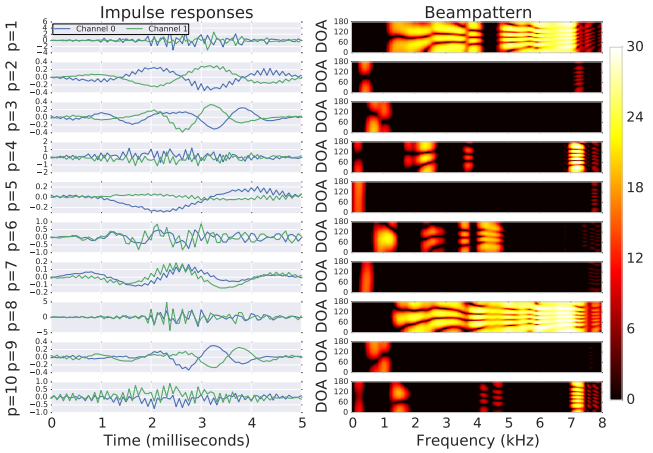

Fig. 2: Trained filters and spatial responses for 10 spatial directions.

To further understand the benefit of the spatial and spectral filtering in `tConv1`, we enforce this layer to only perform spatial filtering by initializing the filters as described in Section 3.2 and not training the layer. Table 2 compares performance when fixing vs. training the `tConv1` layer. The results demonstrate that learning the filter parameters, and therefore performing some spectral decomposition, improves performance over keeping this layer fixed.

# Spatial Filters P	`tConv1` Layer	WER
5	fixed	21.9
5	trained	20.9

Table 2: WER for training vs. fixing the `tConv1` layer, 2 channel.

4.3. Multi-task Learning

Our first MTL experiment is to analyze where to branch the denoising `MTL` layers. For these experiments, we found the optimal weight (in terms of minimizing WER) on the CE term, α, to be 0.9, similar to [15]. Since the unfactored model in [1] is faster to train than the factored model (since it does not contain a second time convolution layer over P feature maps), we do our initial analysis using the unfactored model. The results are shown in Table 3. We find that it is best to put the `MTL` layers at the upper parts of the network, either after the first LSTM or the DNN layers in the CLDNN. The jump in performance in moving the MTL from `fConv` to `1LSTM` indicates that the lower layers of the network, which we know are good at reducing variations due to noise [21], benefit more from MTL compared to the upper layers of the network. Furthermore, notice that for 2 channels, MTL provides a 3-5% relative improvement over the "no MTL" baseline, demonstrating the benefit of this enhancement scheme even when processing multiple channels. Based on these results we place the MTL branch after the first LSTM layer in the remaining factored multichannel experiments.

Denoising task branching layer	1 channel	2 channel
no MTL [1]	23.5	21.8
tConv	23.2	21.7
fConv	23.2	21.8
1LSTM	22.6	20.7
DNN	22.6	20.7

Table 3: WER when multi-task training the unfactored model [1].

4.4. Sequence Training

Table 4 shows the WER of the factored model with MTL for $P = 10$ spatial filters after both CE and sequence training. We compare the proposed factored model to a set of baselines, including (1) single channel log-mel, (2) single channel raw waveform, (3) delay-and-sum beamforming on 8 channel input given oracle knowledge of the true TDOA, (4) MVDR beamforming on 8 channel input where both the true TDOA and noise/speech covariance matrices are known, and (5) the 2 channel unfactored raw waveform approach from [1].

After sequence training, the factored model shows a 5% relative improvement over the model from [1], while incorporating MTL gives a 7% relative improvement. Finally, factored raw + MTL offers a 9% relative improvement over commonly used beamforming techniques.

Method	CE	Seq
log-mel, 1 channel	25.2	20.7
raw, 1 channel	23.5	19.2
delay-and-sum, 8 channel	22.4	18.8
MVDR, 8 channel	22.4	18.7
unfactored raw, 2 channel [1]	21.8	18.2
factored raw, 2 channel	20.4	17.3
factored raw, 2 channel, MTL	**20.0**	**17.0**

Table 4: WER after Sequence Training.

5. CONCLUSIONS

We have presented a factored multichannel raw waveform CLDNN architecture, which explicitly factors "spatial" and "spectral" filtering as separate layers in the network. Analysis of our learned filters show that the "spatial" filter layer learns filters which are selective in frequency as well as space. Furthermore, we incorporated MTL as a postfilter. Overall, the proposed factored model + MTL yields between a 7-9% relative improvement in WER over the unfactored model [1] and commonly used beamforming techniques. One of the limitations of the algorithm is that the look directions are factored explicitly and thus fixed for test, and future work will look at learning a filter adaptively for each input.

6. ACKNOWLEDGEMENTS

Thank you to Andrew Senior and Arden Huang for discussions related to neural networks and beamforming. Also, thank you to Izhak Shafran for discussions related to MVDR.

7. REFERENCES

[1] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and A. Senior, "Speaker Localization and Microphone Spacing Invariant Acoustic Modeling from Raw Multichannel Waveforms," in *Proc. ASRU*, 2015.

[2] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech Acoustic Modeling from Raw Multichannel Waveforms," in *Proc. ICASSP*, 2015.

[3] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*, Springer, 2001.

[4] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*, Springer, 2009.

[5] T. Hain, L. Burget, J. Dines, P.N. Garner, F. Grezl, A.E. Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "Transcribing Meetings with the AMIDA Systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 486–498, 2012.

[6] A. Stolcke, X. Anguera, K. Boakye, O. Çetin, A. Janin, M. Magimai-Doss, C. Wooters, and J. Zheng, "The SRI-ICSI Spring 2007 Meeting and Lecture Recognition System," *Multimodal Technologies for Perception of Humans*, vol. Lecture Notes in Computer Science, no. 2, pp. 450–463, 2008.

[7] B. D. Veen and K. M. Buckley, "Beamforming: A Versatile Approach to Spatial Filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[8] M. Delcroix, T. Yoshioka, A. Ogawa, Y. Kubo, M. Fujimoto, N. Ito, K. Kinoshita, M. Espi, T. Hori, T. Nakatani, and A. Nakamura, "Linear Prediction-based Dereverberation with Advanced Speech Enhancement and Recognition Technologies for the REVERB Challenge," in *REVERB Workshop*, 2014.

[9] M. Seltzer, B. Raj, and R. M. Stern, "Likelihood-maximizing Beamforming for Robust Handsfree Speech Recognition," *IEEE Trascations on Audio, Speech and Language Processing*, vol. 12, no. 5, pp. 489–498, 2004.

[10] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Senior, and O. Vinyals, "Learning the Speech Front-end with Raw Waveform CLDNNs," in *Proc. Interspeech*, 2015.

[11] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks," in *Proc. ICASSP*, 2015.

[12] R. Zelinski, "A Microphone Array with Adaptive Post-filtering for Noise Reduction in Reverberant Rooms," in *Proc. ICASSP*, 1988.

[13] A. 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.

[14] A. Narayanan and D. L. Wang, "Ideal Ratio Mask Estimation Using Deep Neural Networks for Robust Speech Recognition," in *Proc. ICASSP*, 2013.

[15] R. Giri, M. Seltzer, D. Yu, and J. M. Droppo, "Improving Speech Recognition in Reverberation using a Room-aware Deep Neural Network and Multi-Task Learning," in *Proc. ICASSP*, 2015.

[16] T. N. Sainath, B. Kingsbury, A. Mohamed, G. Dahl, G. Saon, H. Soltau, T. Beran, A. Aravkin, and B. Ramabhadran, "Improvements to Deep Convolutional Neural Networks for LVCSR," in *Proc. ASRU*, 2013.

[17] J. B. Allen and D. A. Berkley, "Image Method for Efficiently Simulation Room-Small Acoustics," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943 – 950, April 1979.

[18] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proc. AISTATS*, 2014.

[19] J. Dean, G.S. Corrado, R. Monga, K. Chen, M. Devin, Q.V. Le, M.Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A.Y. Ng, "Large Scale Distributed Deep Networks," in *Proc. NIPS*, 2012.

[20] G. Heigold, E. McDermott, V. Vanhoucke, A. Senior, and M. Bacchiani, "Asynchronous Stochastic Optimization for Sequence Training of Deep Neural Networks," in *Proc. ICASSP*, 2014.

[21] D. Yu, M. Seltzer, J. Li, J.T. Huang, and F. Seide, " Feature Learning in Deep Neural Networks - Studies on Speech Recognition," in *Proc. ICLR*, 2013.