SPEECH ACOUSTIC MODELING FROM RAW MULTICHANNEL WAVEFORMS

Yedid Hoshen[1]*, Ron J. Weiss*[2]*, and Kevin W. Wilson*[2]

[1]Hebrew University of Jerusalem, Jerusalem, Israel
[2]Google Inc, New York, NY, USA
ydidh@cs.huji.ac.il, {ronw,kwwilson}@google.com

ABSTRACT

Standard deep neural network-based acoustic models for automatic speech recognition (ASR) rely on hand-engineered input features, typically log-mel filterbank magnitudes. In this paper, we describe a convolutional neural network - deep neural network (CNN-DNN) acoustic model which takes raw multichannel waveforms as input, i.e. without any preceding feature extraction, and learns a similar feature representation through supervised training.

By operating directly in the time domain, the network is able to take advantage of the signal's fine time structure that is discarded when computing filterbank magnitude features. This structure is especially useful when analyzing multichannel inputs, where timing differences between input channels can be used to localize a signal in space. The first convolutional layer of the proposed model naturally learns a filterbank that is selective in both frequency and direction of arrival, i.e. a bank of bandpass beamformers with an auditory-like frequency scale. When trained on data corrupted with noise coming from different spatial locations, the network learns to filter them out by steering nulls in the directions corresponding to the noise sources. Experiments on a simulated multichannel dataset show that the proposed acoustic model outperforms a DNN that uses log-mel filterbank magnitude features under noisy and reverberant conditions.

Index Terms— Automatic speech recognition, acoustic modeling, convolutional neural networks, beamforming

1. INTRODUCTION

Recently, supervised training of deep classifiers has been shown to be effective at learning meaningful feature representations jointly with state-of-the art classifiers from raw, unprocessed data in both computer vision [1] and speech acoustic modeling [2, 3]. However, speech neural networks are usually trained on hand-designed features, e.g. PLP or log-mel filterbank magnitude coefficients, which are loosely inspired by the human auditory system and have been shown to work well for a variety of speech and audio processing tasks. [4] reviews recent work on neural network acoustic models.

In this work we present a CNN-DNN architecture able to learn acoustic models directly from waveforms. We follow a multi-condition training paradigm and train this network on a variety of noisy and reverberant conditions in both single- and multiple-microphone configurations. When trained on single-channel waveforms we show that this network is able to learn an auditory-like time domain filterbank. We compare the performance of this network to a DNN trained on log mel-frequency magnitude filterbank (mel-fb) features and find that performance is only slightly worse than this baseline. This is in line with previously reported results [2, 3, 5].

The first author performed this work as an intern at Google.

When trained on multichannel inputs, the network learns a filterbank with a similar frequency scale that also exhibits directional selectivity to filter out energy coming from different spatial directions, essentially a bank of bandpass beamformers. By learning to exploit the directional cues present in the multichannel waveform inputs, this network improves recognition performance when compared to a baseline trained on mel-fb features from each input channel.

2. PREVIOUS WORK

A large body of work exists on noise-robust ASR for distant microphones, often leveraging observations from multiple channels. Many techniques have been applied to this problem, including time-frequency masking, noise modeling, and beamforming. Results from the recent REVERB [6] and CHIME [7] challenges demonstrate how these techniques can be used in combination, e.g. [8] who combine a speech enhancement frontend comprised of linear prediction dereverberation, beamforming, and model-based noise reduction at the input to an ASR system. In this work we train a system end-to-end, jointly optimizing a noise-robust frontend along with the context-dependent phone state DNN classifier, rather than manually designing a noise-robust feature extraction stage prior to DNN training. Our work differs from the typical approaches used for time domain speech features, e.g. [9], by learning features rather than hand-designing them. In multi-microphone, i.e. microphone array, scenarios, much work has been done on tailoring the spatial response, or "beampattern", to emphasize the target while attenuating the noise [10].

There has been some previous work on moving more of the feature extraction into the neural network by using a lower level input representation, e.g. FFT or time domain waveform. Sainath et al. [11] improve performance by training a model on linear frequency FFT magnitude features to learn filterbank weights automatically rather than relying on the mel scale [12]. Their work demonstrates that deep neural networks can be used to learn a better representation using a lower level input representation than is traditionally used in ASR.

Other work has attempted to do feature learning directly from time domain waveforms. Jaitly and Hinton [5] separated feature learning from acoustic modeling, using a generative restricted Boltzmann machine model to learn features from single-channel waveforms, then used these features for DNN acoustic model training, demonstrating good results on the TIMIT dataset. More recently, Palaz el al. [2] trained a CNN directly on raw TIMIT waveforms, and show that the network tends to learn bandpass filters when trained to do phone classification. Tüske et al [3] similarly train a DNN acoustic model on waveforms and show that auditory-like features can be learned from random initialization. They pass the waveform into a fully connected layer, which likely requires additional hidden units in order for the network to learn multiple phase shifts of the same filter. This con-

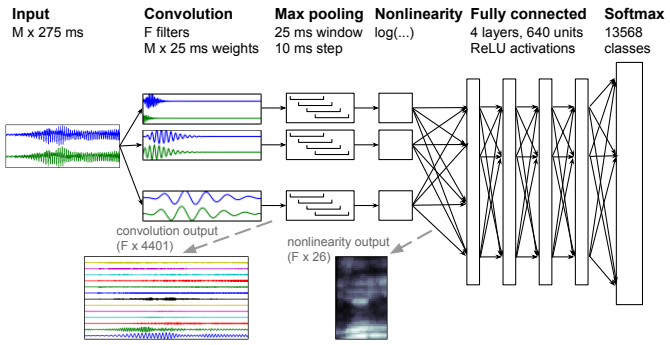

Fig. 1. Schematic of the proposed neural network architecture.

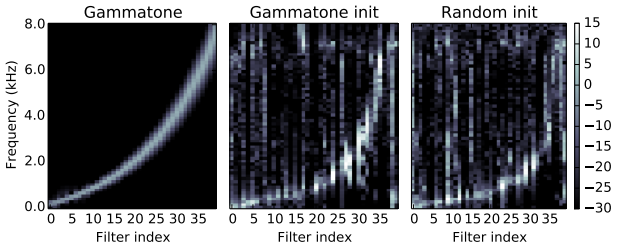

Fig. 2. Magnitude responses of the CNN filterbank when initialized from a gammatone filterbank whose center frequencies match the mel scale (left), gammatone-initialized filters at the end of training (middle), and random-initialized filters at the end of training (right). The filters are sorted by the frequency bin containing the peak response. At convergence, both the gammatone and randomly initialized weights end up with qualitatively similar responses.

trasts with [2] and the current work which use a convolutional time domain filtering layer which can share weights across all time shifts. [2] and [3] both found that their waveform-based models performed slightly worse than baselines trained on mel-fb or MFCC features.

To our knowledge, no previous work has been done on end-to-end feature learning and acoustic model training from multichannel waveforms. Swietojanski, et al. [13, 14] trained deep learning-based acoustic models on multichannel inputs, however they used mel filterbank magnitude features for each input channel. The network architecture in [14] is composed of a single convolutional layer followed by several fully-connected DNN layers. Their best results are obtained by processing each channel independently in the initial layer and then max pooling across channels i.e. choosing the channel with the largest response in each node. Since their approach only utilizes magnitude-based features, it is unable to make use of the spatial information found in the fine time structure (which lies primarily in the previously discarded FFT phase values) of the multichannel signals.

3. ARCHITECTURE

By constructing a neural network architecture to mimic the steps involved in computing mel-fb features, we can train an acoustic model that performs well without requiring manual design of features. Given a single-channel input, our architecture learns a representation qualitatively quite similar to mel-fb features. The choice of operating in the time domain is further justified when using multichannel inputs which allows the network to learn spatial filters that filter out noise.

A baseline deep neural network acoustic model can be described in three stages: (i) the signal is windowed into 25ms frames hopped by 10ms and passed through a mel-scale filterbank, (ii) filterbank outputs from several adjacent frames are stacked to create a feature vector containing additional temporal context, spanning a few tens of frames (hundreds of ms), (iii) each stacked feature vector is then classified into tied context-dependent (CD) units using a deep classifier.

Our CNN-DNN architecture is general enough to be able to describe filterbank extraction and classification jointly. We have chosen the network meta-parameters to closely resemble the pipeline described in steps (i) and (ii) above, but all weights and filter coefficients are learned. A sketch of the architecture can be seen in Fig. 1, consisting of the following layers:

1. Input: This layer extracts 275 ms waveform segments from each of M input microphones. Successive inputs are hopped by 10 ms. At the 16kHz sampling rate used in our experiments each segment contains $M \times 4401$ dimensions.
2. Convolution: F space-time filters of support ($M \times 25$ ms) are learned. The coefficients are trained jointly with the classification task without any constraints or regularization.
3. Rectification and pooling: The output of each filter is passed through a ReLU unit (a half-wave rectifier, [15]) and max-pooled (across time, separately for each filter) over a window of 25 ms hopped by 10 ms. The window duration and hop size match those of mel-fb features.
4. Compressive non-linearity: Each pooled feature is passed through a pointwise log function $\log(\cdot + 0.01)$, where the offset of 0.01 truncates the output range and avoids giving too much weight to small values, and potential numerical issues that would occur if the input were zero. This compresses the dynamic range of the features, which has consistently been found to improve speech recognition performance.
5. Fully connected layers: 4 fully connected layers of 640 units with ReLU activations.
6. Softmax multiclass classifier: 13568 tied CD state units.

The output of the compressive non-linearity layer is analogous to several stacked frames of mel-fb features, and the following layers are identical to our baseline fully connected DNN acoustic model.

3.1. Relationship to a mel-scale filterbank

Mel-fb features are normally computed in the frequency domain by warping the frequency axis of the short-time Fourier transform (STFT) magnitude to match the mel frequency scale. While the network described in this section computes features in the time domain, it can be configured to compute nearly the same representation. The convolution layer learns a bank of finite impulse response (FIR) filters, and the pooling layer decimates each filter output in time, analogous to the windowing operation in the STFT. By setting the weights of each filter to the impulse response of an appropriate bandpass filter, e.g. a gammatone, a bank of filters can be constructed so that the filter center frequencies and bandwidths match an auditory scale, e.g. the mel scale, in which case the combined operation of layers 1-4 will compute a representation qualitatively very similar to mel-fb. Such an auditory-scale gammatone filterbank has been shown to yield speech recognition performance comparable to mel-fb [16]. As shown in Figures 2 and 3, the network indeed learns an auditory-like time domain filterbank, even when the weights are initialized randomly.

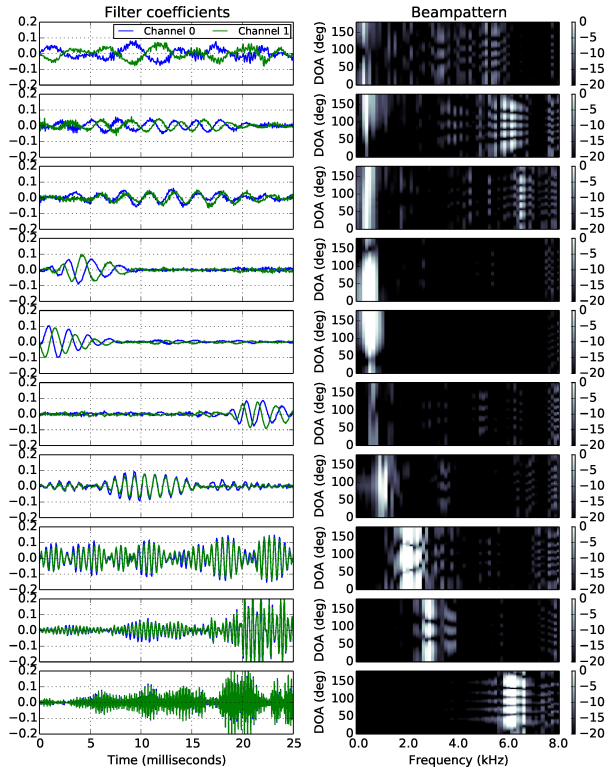

Fig. 3. Examples of the trained filters and their spatial responses. These come from a network trained on audio in which the target speaker was fixed near 90 degrees, and interfering noise sources came from a wide range of directions.

3.2. Spatial filtering

Sampling the same signal at different points in space using a microphone array makes it possible to extract information about the spatial location of components of the signal. For example, a delay-and-sum beamformer uses the expected time delay of the signal arriving at each microphone to emphasize sound coming from a particular direction.

Inter-microphone delays for an array spanning 14 cm (as in our experiments) are less than 1ms, much smaller than the window used in short-time representations like mel-fb. Such fine time structure is preserved in the phase of the STFT, but this is discarded when computing magnitude-based features.

By learning filters spanning all microphones in the time domain, the network is able to use the fine inter-microphone time structure and steer filters in different directions. In practice, the network learns a bank of filters with a primarily bandpass response in frequency, but at different time delays in each microphone to steer nulls in different directions. Examples of this behavior can be seen in Fig. 3.

4. EXPERIMENTS

We train and evaluate the proposed network on several large-vocabulary Voice Search [17] datasets. The Clean training set contains about 400 hours of audio from thousands of different speakers, containing roughly 330000 utterances. There is a matched test set of 36 hours of audio containing 30000 utterances.

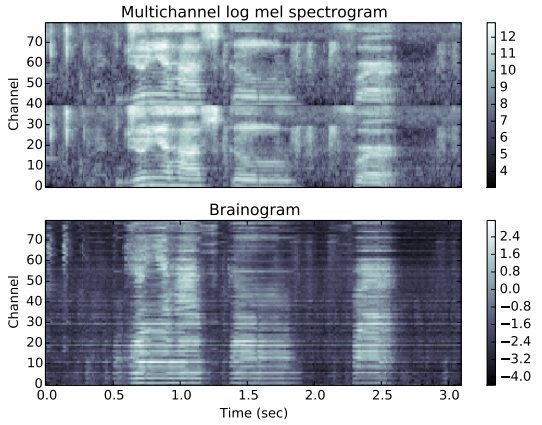

Fig. 4. A two-channel stacked mel-fb spectrogram (top) and brainogram (bottom). Note how the additive noise around 2.0s has been attenuated by the spatial selectivity of the filters formed by the convolutional layer. The brainogram appears stretched with respect to the spectrogram because the spectrogram image consists of two microphones (two channels) stacked on top of one another, while the brainogram is the outputs of the trained convolutional units (sorted by peak response frequency).

The Fixed set was generated by passing each utterance of Clean through a room simulator (based on the image method of simulating reverberation [18]) that simulates reverberation and additive noise. For training, the simulated room dimensions were 4.8m×4.3m×2.9m, and for testing, they were 5m×4m×3m. For both testing and training, the absorption coefficients of the simulated walls were set to vary the RT_{60} from 0 to 0.4 seconds, and the noise signal gain was varied to achieve signal-to-noise ratios between 5dB and 25dB. The target speaker was simulated as coming from broadside of a two-element (stereo) array with inter-microphone spacing of 14cm. The noise signals were obtained from YouTube videos and from recordings in a cafe, and were modeled to come from 30 degrees off of broadside.

In realistic settings the location of the different sources are unlikely to be fixed. We therefore created a Varied dataset, similar to Fixed in most ways, except that for each utterance the target speaker is situated randomly in the range of ± 5 degrees of broadside (to model target direction estimation error), while noise may come from a range of ± 90 degrees of broadside.

For training, each utterance was first forced-aligned to its component CD units using a baseline mel-fb DNN acoustic model trained on clean data. Each training example consisted of 275 ms input samples and a CD unit label as output. Each training dataset was randomly split into train (95%) and validation (5%) sets. The training data was normalized to have zero mean and unit variance.

All weights and biases were randomly initialized with the unit normal distribution unless specified otherwise. The network was trained with Asynchronous Stochastic Gradient Descent [19] using 1200 CPUs in a compute cluster. Weights were adjusted by Adagrad [20], with a learning rate of 0.01. The batch size was 100 examples, each containing a single frame, and the network was trained for 800 million examples (13 epochs).

4.1. Features learned

The magnitude responses of the convolution filters trained on a single channel of Varied are plotted in the right two panels of Fig. 2. The

Model	Clean	Fixed	Varied
Mel-fb DNN	25.6%	39.8%	39.5%
Waveform CNN	27.2%	41.6%	41.5%
Waveform CNN mel gammatone fixed	28.8%	43.6%	43.5%
Waveform CNN mel gammatone init	27.1%	41.7%	41.5%
Waveform CNN no log	28.5%	43.0%	42.9%

Table 1. Single-channel word error rate (WER). The DNN has 10.59M parameters. CNNs have 10.61M params. All models have 40 dimensions/frame, 26 stacked frames, and are trained on Varied.

Model	Train set	Fixed	Varied
Stacked mel-fb DNN	Fixed	39.2%	39.3%
Stacked mel-fb DNN	Varied	39.0%	38.9%
Waveform CNN	Fixed	37.5%	52.0%
Waveform CNN	Varied	38.4%	38.1%
Beamformer mel-fb DNN	Fixed	35.9%	36.6%
Beamformer mel-fb DNN	Varied	36.0%	36.3%

Table 2. Two-channel WER. The DNNs have 11.26M parameters. CNNs have 11.32M params. All models have 80 dimensions/frame and 26 stacked frames.

filterbanks converge to an auditory filter-like representation, consisting of primarily bandpass filters with nonuniformly spaced center frequencies (although not exactly matching the mel scale) and bandwidths that increase with center frequency. These properties have been observed in the human auditory system and are typically manually encoded in the hand engineered features. Similar filters were also observed by [3, 5].

To test the effects of initialization we initialized the weights of the convolutional layer to gammatone impulse responses with center frequencies on the same scale as mel-fb. Fig. 2 shows the magnitude response of the initial filters, the converged filters after gammatone initialization and the filters learned using random initialization. In all cases, the filters have similar bandpass structures, and the converged filters look similar regardless of the initialization.

In the multichannel scenario, we train our network on 2 channels from Varied using 80 filters in the first layer to match the 80-dimensional stacked mel-fb features (40 features from each microphone) in our baseline DNN. Fig. 3 shows several example filters learned by the network. The right column shows the time domain filter weights and the left column shows the beampattern [10, Ch. 3], plotting the magnitude response of each filter as a function of direction of arrival relative to the microphone array. As in the single channel case, the filters primarily have bandpass responses. For each filter, the impulse responses tend to have similar shape but different time delays across channels. The delay corresponds to a beam steered in a particular direction which can suppress interference from elsewhere. The network sometimes learns filters with the same support in frequency, but different inter-channel delays and therefore different spatial responses, e.g. as in the fourth and fifth rows of the figure.

Fig. 4 shows an utterance passed through a two-microphone mel-scale spectrogram without frame stacking (top) and the corresponding "brainogram" output of the log layer in our 2 channel network (bottom). The response of the pooling layer is similar to the response of the mel-scale spectrogram, but note that our features have been able to suppress some of the noise that the standard spectrogram retained (see for example at time 2.0s). This illustrates that our network has effectively learned mel-spectrogram-like multichannel features that are able to deal with noisy conditions.

4.2. Quantitative results

In Table 1 (single channel), our waveform CNN is compared to a standard front-end using the stacked mel-fb features described in Sec. 3 fed into the top layers of the network described in Sec. 3 (the fully connected layers and the softmax layer). We find that the network is able to learn good quality acoustic models with only slightly lower performance than the standard features (1.5-2%) for clean and noisy data. This result extends the results in [2, 3] to multicondition training with noise and reverberation. We also found that the network performs equally well with random initialization as with gammatone initialization. In both cases learning filter weights performs better (2%) than fixing (not training) the convolutional layer to compute gammatone filters. Finally, removing the log operation from the frontend hurts waveform CNN performance by 1.5%, highlighting the importance of dynamic range compression.

In the multichannel case (Table 2) we compare the waveform CNN against stacked mel-fb features similar to [13, 14] (but without pooling across channels) and against mel-fb features computed on the output of a delay-and-sum beamformer. By learning spatial filters the waveform CNN is able to outperform the mel DNN baseline in both scenarios when the train and test sets are matched. In contrast, the stacked mel-fb baseline only observes the magnitude from each channel, so it cannot make use of the timing differences between them. Delay-and-sum beamforming, which exploits the fine time structure of the waveforms to reduce noise coming from directions other than the target direction, performs best overall, consistent with our single-channel findings regarding waveform CNN vs. mel-fb.

The waveform CNN trained on Fixed performs poorly on the Varied test set because it is overfit to the (fixed) geometry of the training data. In contrast, the waveform CNN trained on Varied outperforms stacked mel-fb on both Varied and Fixed.

5. CONCLUSION

We presented a DNN architecture for speech acoustic modeling from multichannel waveforms. With network filter length, pooling window and hop chosen to match a mel-fb baseline, we find that the network learns a bank of bandpass beamformers which qualitatively follow an auditory filterbank-like scale and which have spatial selectivity that exploits the structure of the data. When noise always arrives from a consistent direction, the network learns to steer nulls in that direction, reducing the noise level and improving recognition performance compared to a stacked mel-fb magnitude-based baseline. However this is not robust to more realistic situations where interfering noise can arrive from any direction. We show that expanding the training data to include noise from a variety of directions leads to more robust performance, and to spatial filters steered in many different directions.

Although the network learns a reasonable feature representation, performance remains slightly worse than the baselines, mirroring previous results in the literature. We believe that there is room for improvement by moving away from the parameterization of the mel-fb baseline to see if performance can be improved, e.g. by exploring smaller pooling window sizes, potentially retaining filterbank output at multiple time scales, and by giving the frontend layers additional filtering capacity, e.g. by incorporating recurrent state in the early layers which might be able to compensate for reverberation.

6. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[2] Dimitri Palaz, Ronan Collobert, and Mathew Magimai Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," *Interspeech*, 2014.

[3] Zoltán Tüske, Pavel Golik, Ralf Schlüter, and Hermann Ney, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in *Interspeech*, Singapore, Sept. 2014.

[4] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[5] Navdeep Jaitly and Geoffrey Hinton, "Learning a better representation of speech soundwaves using restricted Boltzmann machines," in *ICASSP*. IEEE, 2011, pp. 5884–5887.

[6] Keisuke Kinoshita, Marc Delcroix, Takuya Yoshioka, Tomohiro Nakatani, Armin Sehr, Walter Kellermann, and Roland Maas, "The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *WASPAA*. IEEE, 2013, pp. 1–4.

[7] Emmanuel Vincent, Jon Barker, Shinji Watanabe, Jonathan Le Roux, Francesco Nesta, and Marco Matassoni, "The second CHiME speech separation and recognition challenge: Datasets, tasks and baselines," in *ICASSP*. IEEE, 2013, pp. 126–130.

[8] Marc Delcroix, Takuya Yoshioka, Atsunori Ogawa, Yotaro Kubo, Masakiyo Fujimoto, Nobutaka Ito, Keisuke Kinoshita, Miquel Espi, Takaaki Hori, Tomohiro Nakatani, and Atsushi Nakamura, "Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the reverb challenge," in *REVERB Workshop*, 2014.

[9] Zhaozhang Jin and DeLiang Wang, "Reverberant speech segregation based on multipitch tracking and classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2328–2337, 2011.

[10] Jacob Benesty, Jingdong Chen, and Yiteng Huang, *Microphone Array Signal Processing*, Springer, 2008.

[11] Tara N. Sainath, Brian Kingsbury, Abdel-rahman Mohamed, and Bhuvana Ramabhadran, "Learning filter banks within a deep neural network framework," in *ASRU*. IEEE, 2013, pp. 297–302.

[12] Satoshi Imai, "Cepstral analysis synthesis on the mel frequency scale," in *ICASSP*. IEEE, 1983, vol. 8, pp. 93–96.

[13] Pawel Swietojanski, Arnab Ghoshal, and Steve Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *ASRU*. IEEE, 2013, pp. 285–290.

[14] Pawel Swietojanski, Arnab Ghoshal, and Steve Renals, "Convolutional neural networks for distant speech recognition," *Signal Processing Letters*, 2014.

[15] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[16] Ralf Schlüter, Ilja Bezrukov, Hermann Wagner, and Hermann Ney, "Gammatone features and feature combination for large vocabulary speech recognition," in *ICASSP*. IEEE, 2007, vol. 4, pp. IV–649.

[17] Johan Schalkwyk, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope, "Your Word is my Command: Google search by voice: A case study," in *Advances in Speech Recognition*, pp. 61–90. Springer, 2010.

[18] Stephen G McGovern, "A model for room acoustics," `http://www.sgm-audio.com/research/rir/rir.html`, 2003.

[19] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al., "Large scale distributed deep networks," in *NIPS*, 2012, pp. 1223–1231.

[20] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.