Convolutional Neural Networks for Distant Speech Recognition

Pawel Swietojanski, *Student Member, IEEE,* Arnab Ghoshal, *Member, IEEE,* and Steve Renals, *Fellow, IEEE*

***Abstract*—We investigate convolutional neural networks (CNNs) for large vocabulary distant speech recognition, trained using speech recorded from a single distant microphone (SDM) and multiple distant microphones (MDM). In the MDM case we explore a beamformed signal input representation compared with the direct use of multiple acoustic channels as a parallel input to the CNN. We have explored different weight sharing approaches, and propose a channel-wise convolution with two-way pooling. Our experiments, using the AMI meeting corpus, found that CNNs improve the word error rate (WER) by 6.5% relative compared to conventional deep neural network (DNN) models and 15.7% over a discriminatively trained Gaussian mixture model (GMM) baseline. For cross-channel CNN training, the WER improves by 3.5% relative over the comparable DNN structure. Compared with the best beamformed GMM system, cross-channel convolution reduces the WER by 9.7% relative, and matches the accuracy of a beamformed DNN.**

***Index Terms*—distant speech recognition, deep neural networks, convolutional neural networks, meetings, AMI corpus**

I. Introduction

DISTANT speech recognition (DSR) [1] is a challenging task owing to reverberation and competing acoustic sources. DSR systems may be configured to record audio data using a single distant microphone (SDM), or multiple distant microphones (MDM). Current DSR systems for conversational speech are considerably less accurate than their close-talking equivalents, and usually require complex multi-pass decoding schemes and sophisticated front-end processing techniques [2]–[4]. SDM systems usually result in significantly higher word error rates (WERs) compared to MDM systems.

Deep neural network (DNN) acoustic models [5] have extended the state-of-the-art in acoustic modelling for automatic speech recognition (ASR), using both hybrid configurations [6]–[11] in which the neural network is used to estimate hidden Markov model (HMM) output probabilities and posteriorgram configurations [12]–[15] in which the neural network provides discriminative features for an HMM. It has also been demonstrated that hybrid neural network systems can significantly increase the accuracy of conversational DSR [16]. An advantage of the hybrid approach is the ability to use frequency domain feature vectors, which provide a small but consistent improvement over cepstral domain features [17].

Convolutional neural networks (CNNs) [18], which restrict the network architecture using local connectivity and weight sharing, have been applied successfully to document recognition [19]. When the weight sharing is confined to the time dimension, the network is called a time-delay neural network and has been applied to speech recognition [20]–[22]. CNNs have been used for speech detection [23], directly modelling the raw speech signal [24], and for acoustic modelling in speech recognition in which convolution and pooling are performed in the frequency domain [25]–[27]. Compared to DNN-based acoustic models, CNNs have been found to reduce the WER on broadcast news transcription by an average of 10% relative [26], [27].

Here we investigate weight sharing and pooling techniques for CNNs in the context of multi-channel DSR, in particular cross-channel pooling across hidden representations that correspond to multiple microphones. We evaluate these approaches through experiments on the AMI meeting corpus [28].

II. CNN Acoustic Models

Context-dependent DNN–HMM systems use DNNs to classify the input acoustics into classes corresponding to the HMM tied states. After training, the output of the DNN provides an estimate of the posterior probability $P(s \mid \mathbf{o}_t)$ of each HMM state s given the acoustic observations $\mathbf{o}_t$ at time t, which may be used to obtain the (scaled) log-likelihood of state s given observation $\mathbf{o}_t$: $\log p(\mathbf{o}_t \mid s) \propto \log P(s \mid \mathbf{o}_t) - \log P(s)$ [6], [8], [29], where $P(s)$ is the prior probability of state s calculated from the training data.

A. Convolutional and pooling layers

The structure of feed-forward neural networks may be enriched through the use of convolutional layers [19] which allows local feature receptors to be learned and reused across the whole input space. A max-pooling operator [30] can be applied to downsample the convolutional output bands, thus reducing variability in the hidden activations. Consider a neural network in which the acoustic feature vector $\mathbf{V}$ consists of filter-bank outputs within an acoustic context window of size Z. $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_b, \ldots, \mathbf{v}_B] \in \mathbb{R}^{B \cdot Z}$ is divided into B frequency bands with the b-th band $\mathbf{v}_b \in \mathbb{R}^Z$ comprising all the Z relevant coefficients (statics, Δ, Δ^2, …) across all frames of the context window. The k-th hidden convolution band $\mathbf{h}_k = [h_{1,k}, \ldots, h_{j,k}, \ldots, h_{J,k}] \in \mathbb{R}^J$ is then composed of a linear convolution of J weight vectors (filters) with F consecutive input bands $\mathbf{u}_k = [\mathbf{v}_{(k-1)L+1}, \ldots, \mathbf{v}_{(k-1)L+F}] \in \mathbb{R}^{F \cdot Z}$, where $L \in \{1, \ldots, F\}$ is the filter shift. Fig 1 gives

P Swietojanski and S Renals are with the Centre for Speech Technology Research, University of Edinburgh, email: {p.swietojanski,s.renals}@ed.ac.uk

A Ghoshal was with the University of Edinburgh when this work was done. He is now with Apple Inc., email: aghoshal@apple.com

This research was supported by EPSRC Programme Grant grant, no. EP/I031022/1 (Natural Speech Technology), and the European Union under FP7 project grant agreement 287872 (inEvent).

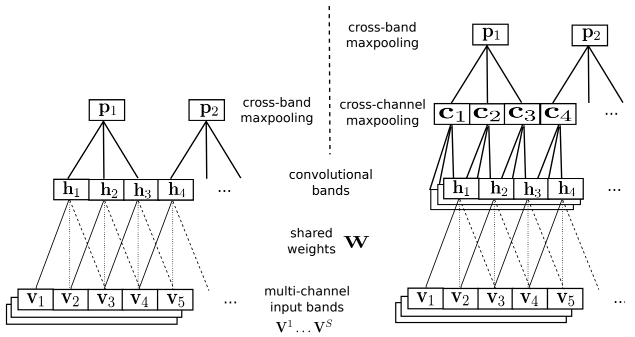

Fig. 1. Frequency domain max-pooling multi-channel CNN layer (left), and a similar layer with cross-channel max-pooling (right).

an example of such a convolution with a filter size and shift of $F = 3$ and $L = 1$ respectively. This may be extended to S acoustic channels $\mathbf{V}^1 ... \mathbf{V}^S$ (each corresponding to a microphone), in which the hidden activation $h_{j,k}$ can be computed by summing over the channels:

$$h_{j,k} = \sigma \left(b_{j,k} + \sum_{s=1}^{S} \mathbf{w}_j^s * \mathbf{u}_k^s \right), \tag{1}$$

where $\sigma(\cdot)$ is a sigmoid nonlinearity, $*$ denotes linear valid convolution[1], $\mathbf{w}_j^s \in \mathbb{R}^{F \cdot Z}$ is a weight vector of the j-th filter acting on the local input $\mathbf{u}_k^s$ of the s-th input channel, and $b_{j,k}$ is an additive bias for the j-th filter and k-th convolutional band. Since the channels contain similar information (acoustic features shifted in time) we conjecture that the filter weights may be shared across different channels. Nevertheless, the formulation and implementation allow for different filter weights in each channel. Similarly, it is possible for each convolutional band to have a separate learnable bias parameter instead of the biases only being shared across bands [25], [26].

The complete set of convolutional layer activations $\mathbf{h} = [\mathbf{h}_1, \ldots, \mathbf{h}_K] \in \mathbb{R}^{K \cdot J}$ is composed of $K = (B - F)/L + 1$ convolutional bands obtained by applying the (shared) set of J filters across the whole (multi-channel) input space $\mathbf{V}$ (as depicted in Fig 1). In this work the weights are tied across the input space (i.e. each $\mathbf{u}_k$ is convolved with the same filters); alternatively the weights may be partially shared, tying only those weights spanning neighbouring frequency bands [25]. Although limited weight sharing was reported to bring improvements for phone classification [25] and small LVSR tasks [32], a recent study on larger tasks [27] suggests that full weight sharing with a sufficient number of filters can work equally well, while being easier to implement.

A convolutional layer is usually followed by a pooling layer which downsamples the activations $\mathbf{h}$. The *max-pooling* operator [33] passes forward the maximum value within a group of R activations. The m-th max-pooled band is composed of J related filters $\mathbf{p}_m = [p_{1,m}, \ldots, p_{j,m}, \ldots, p_{J,m}] \in \mathbb{R}^J$:

$$p_{j,m} = \max_{r=1}^{R} \left(h_{j,(m-1)N+r} \right), \tag{2}$$

where $N \in \{1, \ldots, R\}$ is a pooling shift allowing for overlap between pooling regions when $N < R$ (in Fig 1, $R = N = 3$). The pooling layer decreases the output dimensionality from K convolutional bands to $M = (K - R)/N + 1$ pooled bands and the resulting layer is $\mathbf{p} = [\mathbf{p}_1, ..., \mathbf{p}_M] \in \mathbb{R}^{M \cdot J}$.

B. Channel-wise convolution

Multi-channel convolution (1) builds feature maps similarly to the LeNet-5 model [19] where each convolutional band is composed of filter activations spanning all input channels. We also constructed feature maps using max-pooling across channels, in which the activations $h_{j,k}^s$ are generated in channel-wise fashion and then max-pooled (4) to form a single cross-channel convolutional band $\mathbf{c}_k = [c_{1,k}, \ldots, c_{j,k}, \ldots, c_{J,k}] \in \mathbb{R}^J$ (Fig 1 (right)):

$$h_{j,k}^s = \sigma \left(b_{j,k} + \mathbf{w}_j * \mathbf{u}_k^s \right) \tag{3}$$

$$c_{j,k} = \max_{s=1}^{S} \left(h_{j,k}^s \right). \tag{4}$$

Note that here the filter weights $\mathbf{w}_j$ need to be tied across the channels such that the cross-channel max-pooling (4) operates on activations for the same feature receptor. The resulting cross-channel activations $\mathbf{c} = [\mathbf{c}_1, \ldots, \mathbf{c}_K] \in \mathbb{R}^{K \cdot J}$ can be further max pooled along frequency using (2). Channel-wise convolution may also be viewed as a special case of 2-dimensional convolution, where the effective pooling region is determined in frequency but varies in time depending on the actual time delays between the microphones.

C. Fully-connected layers

The complete acoustic model is composed of one or more CNN layers, followed by a number of fully-connected layers, with a softmax output layer. With a single CNN layer, the computation performed by the network is as follows:

$$\mathbf{h}^l = \boldsymbol{\sigma}(\mathbf{W}^l \mathbf{h}^{l-1} + \mathbf{b}^l), \quad \text{for } 2 \leq l < L \tag{5}$$

$$\mathbf{a}^L = \mathbf{W}^L \mathbf{h}^{L-1} + \mathbf{b}^L,$$

$$P(s|\mathbf{o}_t) = \frac{\exp\{a^L(s)\}}{\sum_{s'} \exp\{a^L(s')\}}, \tag{6}$$

where $\mathbf{h}^l$ is the input to the $(l+1)$-th layer, with $\mathbf{h}^1 = \mathbf{p}$; $\mathbf{W}^l$ is the matrix of connection weights and $\mathbf{b}^l$ is the additive bias vector for the l-th layer; $\boldsymbol{\sigma}(\cdot)$ is a sigmoid nonlinearity that operates element-wise on its input vector; $\mathbf{a}^L$ is the activation at the output layer.

III. Experiments

We have performed experiments using the AMI meeting corpus [28] (http://corpus.amiproject.org/) using an identical training and test configuration to [16]. The AMI corpus comprises around 100 hours of meetings recorded in instrumented meeting rooms at three sites in the UK, the Netherlands, and Switzerland. Each meeting usually has four participants and the language is English, albeit with a large proportion of non-native speakers. Multiple microphones were used, including individual headset microphones (IHM), lapel microphones, and one or more microphone arrays. Every recording used a

[1] The convolution of two vectors of size X and Y may result either in the vector of size $X + Y - 1$ for a full convolution with zero-padding of non-overlapping regions, or the vector of size $X - Y + 1$ for a valid convolution where only the points which overlap completely are considered [31].

primary 8-microphone uniform circular array (10 cm radius), as well as a secondary array whose geometry varied between sites. In this work we use the primary array for our MDM experiments, and the first microphone of the primary array for our SDM experiments. Our systems are trained and evaluated using the split recommended in the corpus release: an 80 hour training set, and development and test sets each of 9 hours. We use the segmentation provided with the AMI corpus annotations (v1.6). For training purposes we consider all segments (including those with overlapped speech), and the WERs of the speech recognition outputs are scored by the `asclite` tool [34] following the NIST RT recommendations for scoring simultaneous speech (http://nist.gov/speech/tests/rt/2009). WERs for non-overlapped segments only may also be produced by `asclite`, using the `-overlap-limit 1` option. Here, we report results using the development set only: both development and test sets are relatively large, and we previously found that the best parameters selected for the development set were also optimal for the evaluation set [16].

All CNN/DNN models were trained on 40-dimensional log Mel filterbank (FBANK) features appended with the first and the second time derivatives [17] which were presented in $Z = 11$ frames long symmetric context windows. Our distant microphone systems within this work remain unadapted to both speakers and sessions. Ascribing speakers to segments without diarisation is unrealistic while a small mismatch between training and evaluation acoustic environments makes feature-space maximum likelihood linear regression only moderately effective (less than 1% absolute reduction in WER) for session adaptation. Our experiments were performed using the Kaldi speech recognition toolkit [35], and the `pylearn2` machine learning library [36].

Our experiments used a 50,000 word pronunciation dictionary [4]. An in-domain trigram language model (LM) was estimated using the AMI training transcripts (801k words). This was interpolated with two further trigram LMs – one estimated from the Switchboard training transcripts (3M words), and the other from the Fisher English transcripts (22M words) [37]. The LMs are estimated using modified Kneser-Ney smoothing [38]. The LM interpolation weights were as follows: AMI transcripts (0.73); Switchboard (0.05); Fisher (0.22). The final interpolated LM had 1.6M trigrams and 1.5M bigrams, resulting in a perplexity of 78 on the development set.

IV. Results

We have tested the CNNs with both SDM and MDM inputs. In each case we compare the CNN to two baseline systems: (1) a Gaussian mixture model (GMM) system, discriminatively trained using boosted maximum mutual information (BMMI) [39], with mel-frequency cepstral coefficient (MFCC) features post-processed with linear discriminative analysis (LDA) and decorrelated using a semi-tied covariance (STC) transform [40]; and (2) a deep neural network (DNN) with 6 hidden layers, with 2048 units in each layer [16] trained using the same FBANK features as used for the CNNs. We used restricted Boltzmann machine (RBM) pretraining [41] for the baseline DNN systems, but not for the CNN systems. The CNN results are reported for the networks composed with a single CNN layer followed by 5 fully-connected layers. The CNN hyperparameters are as follows: number of filters $J = 128$, filter size $F = 9$, and filter shift $L = 1$.

TABLE I
Word Error Rates (%) on AMI development set – SDM.

System	with overlap	no overlap
BMMI GMM-HMM (LDA+STC)	63.2	55.8
DNN +RBM (FBANK)	53.1	42.4
CNN ($R = 3$)	53.3	42.8
CNN ($R = 2$)	51.3	40.4
CNN ($R = 1$)	52.5	40.9

A. Single Distant Microphone

We applied two CNN approaches to the SDM case, in which acoustics from a single channel only is used. In the first approach the same bias terms were used for each band [26] (section II-A), and the results of the single channel CNN can be found in Table I. The first two rows are the SDM baselines (reported in [16])[2]. The following three lines are results for the CNN using max-pool sizes (PS) of $R = N = 1, 2, 3$. By using CNNs we were able to obtain 3.4% relative reduction in WER with respect to the best DNN model and a 19% relative reduction in WER compared with a discriminatively trained GMM-HMM. Note, the total number of parameters of the CNN models vary here as $R = N$ while J is kept constant across the experiments. However, the best performing model had neither the highest nor the lowest number of parameters, which suggests it is due to the optimal pooling setting.

B. Multiple Distant Microphones

For the MDM case we compared a delay-sum beamformer with the direct use of multiple microphone channels as input to the network. For beamforming experiments, we follow noise cancellation using a Wiener filter with a delay-sum beamforming on 8 uniformly-spaced array channels using BeamformIt [42]. The results are summarised in Table II. The first block of Table II presents the results for the case in which the models were trained on a beamformed signal from 8 microphones. The first two rows show the WER for the baseline GMM and DNN acoustic models as reported in [16]. The following three rows contain the comparable CNN structures with different pooling sizes (PS) $R = N = 1, 2, 3$. The best model (pool size $R = 1$, equivalent to no max-pooling) scored 46.3% WER which is 6.4% relative WER better than the best DNN network and a relative improvement in WER of 16% compared with a discriminatively trained GMM-HMM system.

The second part of Table II shows WERs for the models directly utilising multi-channel features. The first row is a baseline DNN variant trained on 4 concatenated channels [16]. Then we present the CNN models with MDM input convolution performed as in equation (1) and pooling size of 2, which was optimal for the SDM experiments. This

[2] DNN baseline WERs are lower than [16] due to the intial values chosen for the hyper-parameters.

TABLE II
WORD ERROR RATES (%) ON AMI DEVELOPMENT SET – MDM.

System	with overlap	no overlap
MDM with beamforming (8 microphones)		
BMMI GMM-HMM	54.8	46.1
DNN +RBM	49.5	37.4
CNN ($R = 3$)	46.5	34.2
CNN ($R = 2$)	46.8	34.4
CNN ($R = 1$)	**46.3**	**34.3**
MDM without beamformer		
DNN +RBM 4ch concatenated	51.2	40.3
CNN ($R = 2$) 2ch conventional	50.5	39.5
CNN ($R = 2$) 4ch conventional	50.4	38.7
CNN ($R = 2$) 2ch channel-wise	50.0	38.5
CNN ($R = 2$) 4ch channel-wise	**49.4**	**37.5**

TABLE III
WORD ERROR RATES (%) ON AMI DEVELOPMENT SET – IHM

System	WER(%)
BMMI GMM-HMM (SAT)	29.4
DNN +RBM	26.6
CNN ($R = 1$)	25.6

TABLE IV
WORD ERROR RATES (%) ON AMI DEVELOPMENT SET.
DIFFERENT WEIGHT SHARING AND POOLING TECHNIQUES.

System	with overlap	no overlap
MDM without beamformer		
CNN ($R = 3$) 2ch not tied $\mathbf{w}_j^s$	51.2	-
CNN ($R = 2$) 2ch not tied $\mathbf{w}_j^s$	51.3	-
SDM		
CNN ($R = 3$) bias b_j	53.3	42.8
CNN ($R = 3$) bias $b_{j,k}$	52.5	40.9
CNN ($R = 2$) bias $b_{j,k}$	51.9	40.5

scenario decreases WER by 1.6% relative when compared to a DNN structure with concatenated channels. Applying channel-wise convolution with two-way pooling (outlined in section II-B) brings further gains of 3.5% WER relative. Furthermore, channel-wise pooling works better for more input channels: conventional convolution on 4 channels achieves 50.4% WER, practically the same as the 2 channel network, while channel-wise convolution with 4 channels achieves 49.5% WER, compared to 50.0% for the 2-channel case. These results indicate that picking the best information (selecting the feature receptors with maximum activations) within the channels is crucial when doing model-based combination of multiple microphones.

C. *Individual Headset Microphones*

We observe similar relative WER improvements between DNN and CNN for close talking speech experiments (Table III) as were observed for the DSR experiments (Tables I and II). The CNN achieves 3.6% WER reduction relative to the DNN model. Both DNN and CNN systems outperform a BMMI-GMM system trained in a speaker adaptive (SAT) fashion by 9.4% and 12.9% relative WER respectively. We did not see any improvements by increasing pooling size. [26] has previously suggested that pooling may be task dependent.

D. *Different weight-sharing techniques*

When using multiple distant microphones directly as input to a CNN, we posit that the same filters should be used across the different channels even when cross-channel pooling is not used. Each channel contains the same information, albeit shifted in time, hence using the same feature detectors for each channel is a prudent constraint to learning. The first two rows of Table IV show the results when a separate set of filters are learned for each channel. Sharing the filter weights across channels improves the WER by 0.7% absolute (comparing with the 2 channel CNN, Table II).

The second block of Table IV shows the effect of training a separate bias parameter for each of the K convolutional bands for the SDM system of Table I. These results are generated for non-overlapping pools of size 3 and 2. If the pooling size is too large, we observe that the WER increases. This increase in WER is mitigated by using a band-specific bias. We hypothesise that, under noisy conditions, the max-pooling operator, which may be interpreted as a local hard-decision heuristic, selects non-optimal band activations, while the not-tied bias can actually "boost" the meaningful frequency regions (on average). A band-specific bias does not lead to further improvements: e.g., when $R = 2$, the overlapped speech CNN with tied biases had a WER of 51.3% compared to 51.9% for the not-tied version.

V. DISCUSSION

We have investigated using CNNs for DSR with single and multiple microphones. A CNN trained on a single distant microphone is found to produce a WER approaching these of a DNN trained using beamforming across 8 microphones. In experiments with multiple microphones, we compared CNNs trained on the output of a delay-sum beamformer with those trained directly on the outputs of multiple microphones. In the latter configuration, channel-wise convolution followed by a cross-channel max-pooling was found to perform better than multi-channel convolution.

A beamformer uses time-delays between microphone pairs whose computation requires knowledge of the microphone array geometry, while these convolutional approaches need no such knowledge. CNNs are able to compensate better for the confounding factors in distant speech than DNNs. However, the compensation learned by CNNs is complementary to that provided by a beamformer. In fact, when using CNNs with cross-channel pooling, similar WERs were obtained by changing the order of the channels at test time from the order in which they were presented at training time, suggesting that the model is able to pick the most informative channel.

Early work on CNNs for ASR focussed on learning shift-invariance in time [20], [43], while more recent work [25], [26] have indicated that shift-invariance in frequency is more important for ASR. The results presented here suggest that recognition of distant multichannel speech is a scenario where shift-invariance in time between channels is also important, thus benefitting from pooling in both time and frequency.

References

[1] M Wölfel and J McDonough, *Distant Speech Recognition*, Wiley, 2009.

[2] A Stolcke, "Making the most from multiple microphones in meeting recognition," in *Proc IEEE ICASSP*, 2011.

[3] K Kumatani, J McDonough, and B Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 127–140, 2012.

[4] T Hain, L Burget, J Dines, PN Garner, F Grezl, AE Hannani, M Huijbregts, M Karafiat, M Lincoln, and V Wan, "Transcribing meetings with the AMIDA systems," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 2, pp. 486–498, 2012.

[5] G Hinton, L Deng, D Yu, GE Dahl, A-R Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, TN Sainath, and B Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[6] H Bourlard and N Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[7] S Renals, N Morgan, H Bourlard, M Cohen, and H Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 1, pp. 161–174, 1994.

[8] N Morgan and H Bourlard, "Neural networks for statistical recognition of continuous speech," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 742–772, 1995.

[9] AJ Robinson, GD Cook, DPW Ellis, E Fosler-Lussier, SJ Renals, and DAG Williams, "Connectionist speech recognition of broadcast news," *Speech Communication*, vol. 37, no. 1–2, pp. 27–45, 2002.

[10] TN Sainath, B Kingsbury, B Ramabhadran, P Fousek, P Novak, and A Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc IEEE ASRU*, 2011.

[11] GE Dahl, D Yu, L Deng, and A Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[12] H Hermansky, DPW Ellis, and S Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc IEEE ICASSP*, 2000, pp. 1635–1638.

[13] Q Zhu, A Stolcke, BY Chen, and N Morgan, "Using MLP features in SRI's conversational speech recognition system," in *Proc. Eurospeech*, 2005.

[14] F Grézl, M Karafiát, S Kontár, and J Černocký, "Probabilistic and bottleneck features for LVCSR of meetings," in *Proc IEEE ICASSP*, 2007, vol. 4, pp. IV–757–IV–760.

[15] TN Sainath, B Kingsbury, and B Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc IEEE ICASSP*, 2012.

[16] P Swietojanski, A Ghoshal, and S Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *Proc. IEEE ASRU*, Dec. 2013.

[17] J Li, D Yu, J-T Huang, and Y Gong, "Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM," in *Proc IEEE SLT*, 2012, pp. 131–136.

[18] Y LeCun and Y Bengio, "Convolutional networks for images, speech and time series," in *The Handbook of Brain Theory and Neural Networks*, pp. 255–258. The MIT Press, 1995.

[19] Y LeCun, L Bottou, Y Bengio, and P Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] A Waibel, T Hanazawa, G Hinton, K Shikano, and KJ Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[21] KJ Lang, AH Waibel, and GE Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, vol. 3, no. 1, pp. 23–43, 1990.

[22] T Zeppenfeld, R Houghton, and A Waibel, "Improving the MS-TDNN for word spotting," in *Proc IEEE ICASSP*, 1993, vol. 2, pp. 475–478.

[23] S Sukittanon, AC Surendran, JC Platt, and CJC Burges, "Convolutional networks for speech detection," in *Proc. ICSLP*, 2004.

[24] D Palaz, R Collobert, and M Magimai-Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," in *Proc Interspeech*, 2013.

[25] O Abdel-Hamid, A-R Mohamed, J Hui, and G Penn, "Applying convolutional neural networks concepts to hybrid NN–HMM model for speech recognition.," in *Proc IEEE ICASSP*, 2012, pp. 4277–4280.

[26] TN Sainath, A Mohamed, B Kingsbury, and B Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc IEEE ICASSP*, 2013.

[27] TN Sainath, B Kingsbury, A Mohamed, GE Dahl, G Saon, H Soltau, T Beran, AY Aravkin, and B Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," in *Proc IEEE ASRU*, 2013.

[28] J Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus," *Language Resources & Evaluation Journal*, vol. 41, no. 2, pp. 181–190, 2007.

[29] MD Richard and RP Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation*, vol. 3, no. 4, pp. 461–483, 1991.

[30] MA Ranzato, FJ Huang, Y-L Boureau, and Y LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *IEEE CVPR*, 2007.

[31] "NumPy Reference," March 2014, http://docs.scipy.org/doc/numpy/numpy-ref-1.8.1.pdf [Online; accessed 27-March-2014].

[32] O Abdel-Hamid, L Deng, and D Yu, "Exploring convolutional neural network structures and optimisation techniques for speech recognition," in *In Proc. Interspeech*. 2013, ICSA.

[33] M Riesenhuber and T Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, pp. 1019–1025, 1999.

[34] JG Fiscus, J Ajot, N Radde, and C Laprun, "Multiple dimension Levenshtein edit distance calculations for evaluating ASR systems during simultaneous speech," in *Proc. LREC*, 2006.

[35] D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlíček, Y Qian, P Schwarz, J Silovský, G Stemmer, and K Veselý, "The Kaldi speech recognition toolkit," in *Proc. IEEE ASRU*, December 2011.

[36] IJ Goodfellow, D Warde-Farley, P Lamblin, V Dumoulin, M Mirza, R Pascanu, J Bergstra, F Bastien, and Y Bengio, "Pylearn2: a machine learning research library," *arXiv preprint arXiv:1308.4214*, 2013.

[37] C Cieri, D Miller, and K Walker, "From Switchboard to Fisher: Telephone collection protocols, their uses and yields," in *Proc Eurospeech*, 2003.

[38] SF Chen and J Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–393, 1999.

[39] D Povey, D Kanevsky, B Kingsbury, B Ramabhadran, G Saon, and K Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc IEEE ICASSP*, 2008, pp. 4057–4060.

[40] MJF Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[41] G Hinton, S Osindero, and Y Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[42] X Anguera, C Wooters, and J Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 7, pp. 2011–2021, 2007.

[43] H Lee, P Pham, Y Largman, and A Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 1096–1104.