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

Multi-Span Acoustic Modelling using Raw Waveform Signals

P. von Platen[1,2], *C. Zhang*[1], *P. C. Woodland*[1]

[1] Cambridge University Engineering Dept., Trumpington St., Cambridge, CB2 1PZ U.K.
[2] Institute of Communication Systems (IKS), RWTH Aachen University, Germany
{pwv20,cz277,pcw}@eng.cam.ac.uk

Abstract

Traditional automatic speech recognition (ASR) systems often use an acoustic model (AM) built on handcrafted acoustic features, such as log Mel-filter bank (FBANK) values. Recent studies found that AMs with convolutional neural networks (CNNs) can directly use the raw waveform signal as input. Given sufficient training data, these AMs can yield a competitive word error rate (WER) to those built on FBANK features. This paper proposes a novel multi-span structure for acoustic modelling based on the raw waveform with multiple streams of CNN input layers, each processing a different span of the raw waveform signal. Evaluation on both the single channel CHiME4 and AMI data sets show that multi-span AMs give a lower WER than FBANK AMs by an average of about 5% (relative). Analysis of the trained multi-span model reveals that the CNNs can learn filters that are rather different to the log Mel-filters. Furthermore, the paper shows that a widely used single span raw waveform AM can be improved by using a smaller CNN kernel size and increased stride to yield improved WERs.

Index Terms: acoustic modelling, raw waveform, convolutional neural network, multi-span

1. Introduction

Automatic speech recognition (ASR) systems usually consist of an acoustic model (AM) that captures the acoustic and phonetic properties of the speech signal and a language model (LM) providing linguistic and syntactic context information at the word-level. Traditional AMs are normally built on handcrafted acoustic features, such as log Mel-filter bank values (FBANK) or their approximate linear decorrelations known as Mel frequency cepstral coefficients (MFCCs) [1]. These handcrafted acoustic features are broadly based on models from human speech production and perception [2, 3] so that they are not optimised toward the training criterion of the AM and might thus discard valuable information from the raw waveform signal.

For AMs based on hidden Markov models (HMMs) with diagonal Gaussian mixture output distributions, a compact feature representation such as MFCCs was required [4]. However with the resurgence of artificial neural networks (ANNs), along with increasing computational power, there are far fewer restrictions on the input features, and using the raw waveform signal now becomes an interesting alternative to handcrafted acoustic features [5, 6]. AMs built on the raw waveform signal input make no prior assumptions about the data, which allows the AM to learn the most suitable raw waveform feature representation given sufficient training data. Active research work has been carried out for the use of raw waveform features for acoustic modelling since 2014 [6–8], and has yielded competitive word error rates (WERs) to the standard approach using MFCC or FBANK features. In [6], a 35ms window of the raw waveform signal is fed into a convolutional neural network (CNN) layer with rectified linear unit (ReLU) [9] activation for time-frequency decomposition, followed by max-pooling and logarithm layers to imitate the logarithm compression of FBANK features. Analogous to a frame, it produces a feature vector which is fed into a second CNN layer [10], similar to the AMs applying a frequency convolution over FBANK features [11]. In [8], the first CNN layer also performs a temporal convolution while the second CNN layer extracts the spectral envelope followed by logarithm or root compression [2]. Seventeen consecutive output vectors from the second CNN layer are then stacked to have a total input span of 291ms, and the resulting output vector is fed into a deep neural network (DNN) with 12 fully connected layers. Non-linearities other than max-pooling with more discriminative kernels can be used to aggregate the output of the CNN input layer [7]. Zhu *et al.* [12] proposed another structure in which CNN layers with different kernel sizes are configured to learn features of different time-frequency resolutions within a 20ms window, similar to wavelets [13]. Several other studies have investigated the use of raw waveform signal input from multiple microphones in far-field ASR [14, 15].

Analysis of the trained CNN layers with raw waveform input reveals a strong similarity between the learned kernels and audiological distributed narrow band pass filters such as log-Mel filter banks [6, 7, 16]. This finding has reaffirmed the effectiveness of using handcrafted acoustic feature inputs and has inspired joint training of only some of the feature extraction pipeline with the AM [17–19]. However, it also motivates trying to learn feature representations that are different to handcrafted acoustic features, *e.g.* [12].

In this paper, we propose a novel multi-span AM structure which combines multiple input streams to learn more diverse feature representations from different spans of the same raw waveform input. Each stream uses a stack of two consecutive CNN layers and each span is configured using the same kernel size but different stride numbers for temporal convolutions. Single channel experimental results on far-field CHiME4 data show that a 5 layer DNN with three streams outperformed the FBANK AM. It can be observed that the learned filters are rather different to the log-Mel ones. It may also noted that a set of small CNN kernels each having just 50 trainable parameters outperforms the set of larger CNN kernels each having 400 trainable parameters normally used for raw waveform input [5, 8, 14, 16]. These findings are validated by experiments with data from headset microphones from the AMI data set.

The paper is structured as follows. In Sec. 2, CNNs are revisited for raw waveform signal input. Section 3 explains in detail the proposed multi-span AM structure. The experimental setup and results on CHiME4 and AMI are given in Sec. 4 and Sec. 5, with discussion in Sec. 6, followed by conclusions.

http://dx.doi.org/10.21437/Interspeech.2019-2454

2. Revisiting CNNs with Waveform Input

CNNs [20] are powerful ANN models that can learn complex feature representations, as has been shown in image recognition with raw pixel input [20, 21]. Excluding the bias for simplicity, a CNN layer consists of K trainable kernels, $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K$. Each kernel $\mathbf{w}_k$ is convolved over T input samples of the raw waveform signal $\mathbf{x}_1^T$ with a stride S (denoted by $*^S$):

$$\widetilde{\mathbf{y}}_k = \mathbf{w}_k *^S \mathbf{x}_1^T \tag{1}$$

where $\widetilde{\mathbf{y}}_k$ denotes the k-th (one dimensional) output feature map. The output from a CNN layer at each time step comprises of K output feature maps, and the size of each map M can be determined by

$$M = \lfloor (T-L)/S \rfloor + 1, \tag{2}$$

where L is the kernel size and S the stride.

Splitting the raw waveform $\mathbf{x}_1^T$ into M overlapping windows $[x_1, \ldots, x_L], \ldots, [x_{1+SM}, \ldots, x_{L+SM}]$, with x_j representing the j-th sample of $\mathbf{x}_1^T$, then

$$\mathbf{y}_m = \left[x_{1+S(m-1)}, \ldots, x_{L+S(m-1)}\right] [\mathbf{w}_1, \ldots, \mathbf{w}_K], \tag{3}$$

results in a vector $\mathbf{y}_m$ based on a fixed window of raw waveform using K kernels. $\mathbf{y}_m$ can be viewed as a "frame" similar to the one used in traditional acoustic feature analysis and can be obtained by extracting the m-th elements from all K output feature maps.

Two examples of CNN kernels of the same size $L = 5$, but different strides $S_1 = 1, S_2 = 4$ and input spans $T_1 = 7, T_2 = 13$ are given in Fig. 1. From the figure and based on Eqn. (2), it is clear that the input span T can be viewed as a function of S, L, and M, *i.e.*

$$T = (M-1)S + L. \tag{4}$$

Therefore T is controlled by varying S while fixing L and M. For example in Fig. 1, both the orange and green kernels have the same size $L = 5$ and yield an output feature map sized $M = 3$, whereas the orange kernel considers a much larger input span of $T_2 = 13$ due to its bigger stride 4. In the rest of the paper, $\mathbf{y}_m$ will denote the m-th output feature vector. Throughout the paper, the notation

$$\mathbf{y} = \text{CNN}_S^L(\mathbf{x}_1^T, M) \tag{5}$$

is defined to denote a CNN layer, where $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M]$ is the concatenation of all M output feature vectors.

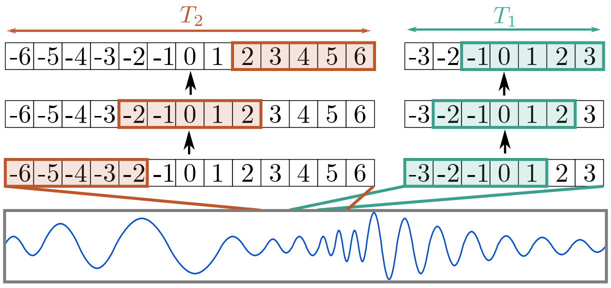

Figure 1: *Examples for temporal convolution with a output feature map size $M = 3$ and kernel length $L = 5$. The strides $S_1 = 1$ (green) and $S_2 = 4$ (orange) define the spans $T_1 = 7$ (green) and $T_2 = 13$ (orange) respectively.*

3. Multi-Span Acoustic Model

Frames of traditional acoustic features, such as MFCC and FBANK, are usually derived using the short-time Fourier transform (STFT) based on a 25ms window, within which the speech signal is assumed to be stationary, and a window shift of 10ms. Conventional cross-entropy (CE) trained feed-forward DNN AMs have been found to yield the lowest WERs when 11 concatenated frames (or alternatively 9 concatenated frames if first order differentials are included) are used as the AM input [22–24], which results in an input span of 125ms of the raw waveform signal. Actually, it has been found that more powerful ANN AMs, such as recurrent or time-delayed neural networks, can effectively use a much longer span than DNNs [25, 26]. This shows the importance of input span for acoustic modelling.

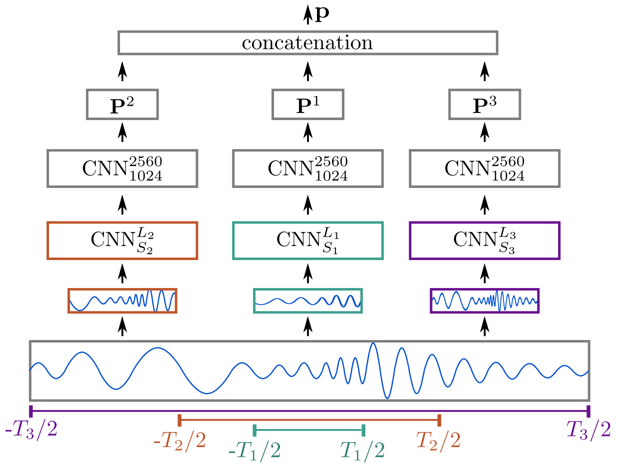

Figure 2: *A sketch map of using three CNN input streams to convolve over different raw waveform spans based on the ranges of* [$-T_1/2, T_1/2$], [$-T_2/2, T_2/2$], [$-T_3/2, T_3/2$] *respectively.*

The multi-span AM is proposed in this paper, which improves FBANK based AMs by using multiple input streams to extract a more diverse set of complementary features from the raw waveform. As an example, three input streams of the multi-span AM are shown in Fig. 2, which produce the outputs $\mathbf{o}^1$, $\mathbf{o}^2$ and $\mathbf{o}^3$ from different spans T_1, T_2 and T_3 respectively by using two consecutive CNN layers. More specifically, for each input stream i, CNN input layers are convolved over a unique span of the raw waveform signal $\mathbf{x}$ yielding

$$\mathbf{y}^i = \text{CNN}_{S_i}^{L_i}(\mathbf{x}_{-T_i/2}^{T_i/2}, M_i), \tag{6}$$

where S_i, L_i, and M_i are parameters defining the first CNN layer. Next, $\mathbf{y}^i$ is fed into a separate second CNN layer which can be two dimensional including both temporal and frequency convolutions with stride, kernel size, and output feature map size set to S_{i2}, L_{i2}, and M_{i2}, respectively, *i.e.*

$$\mathbf{o}^i = \text{CNN}_{S_{i2}}^{L_{i2}}(\mathbf{y}^i, M_{i2}). \tag{7}$$

Multiple CNN layers could be stacked in each stream which can result in the use of smaller kernel sizes [21]. The size of the resulting output $\mathbf{o}^i$ from each stream i can be reduced by using a linear projection $\mathbf{P}^i$, and the final multi-span feature vector $\mathbf{p}$ can be formed by concatenating $\mathbf{P}^i\mathbf{o}^i$ from all streams.

In this paper, only input streams with two CNN layers are investigated. For the CNN input layers given in Eqn. (6), $M_i = 200$ and $K = 64$ kernels are fixed throughout the paper, while for the second CNN layers, $M_{i2} = 11$, $S_{i2} = 1024$, and $L_{i2} = 2560$ are used in this paper. The ReLU activation function is applied to the output of both CNN layers in each stream. By fixing the kernel number of the second CNN layers to be 128, the size of each output $\mathbf{o}^i$ is $128 \times 11 = 1408$, which is reduced to 150-d by $\mathbf{P}^i$.

It is to be emphasised that the only parameters that differ in each stream i are the stride S_i and the kernel size L_i of the input CNN layers. If the vectors $\mathbf{o}^i$ from all streams are of equal size, then the input span T_i of the raw waveform signal for each stream is given by Eqn. (4).

It is worth noting that in contrast to other models [7, 8, 14, 27], there is no log-compression, root-compression, max-pooling or other special non-linearity used in our current setup in order to constrain the model as little as possible to learn the best possible feature representations from multiple input spans. It may be possible to further improve the multi-span model by *e.g.* using different non-linearities for different input streams.

4. Experimental Setup

The proposed multi-span AM was evaluated by training systems on CHiME4 [28] and AMI [29] using HTK 3.5.1 and PyHTK [30, 31]. In the results reported here, the multi-span feature vector $\mathbf{p}$ of the concatenated input streams is fed into a simple feed forward DNN with 4 hidden layers each having 512 output nodes and ReLU activation function. The DNN output layer dimension corresponds to the number of clustered triphone-states and applies the softmax activation function. This structure is abbreviated as *4L-512d-DNN*. We used rather small AMs without many parameters compared to other AMs using the same data sets [32, 33], to ensure a quick turn around.

The training data is aligned at 10ms frame intervals to the clustered triphone-states. For both corpora, 10% of the aligned training data was held back for cross-validation. All models were trained by the CE criterion, using stochastic gradient descent optimization with momentum, weight decay and the NewBob$^+$ learning rate scheduler [18]. To match the number of alignment frames, the raw waveform input is shifted by 10ms or 160 samples after every forward pass of the model.

4.1. CHiME4

Initial DNN AMs were trained on 18h of the training corpus recorded by a close talking microphone (tr05-org + channel 0 on tr05-real) and the alignments obtained were used for all subsequent experiments. The data was aligned at a 10ms frame interval level to one of 3006 clustered triphone-states. The 18h training set for DNN AMs consisted of real and simulated data from channel 5. The raw waveform signal input was globally normalised for both zero mean and unit variance. Because of the known microphone failures [28], for every utterance, the channel used for decoding the 5.6h development (dev) set was chosen according to a microphone failure detection algorithm presented in [32]. Speech recognition experiments were conducted using Viterbi decoding based on a 5k vocabulary 3-gram (tg) LM trained on the official CHiME4 LM training data.

4.2. AMI

The training data for AMI includes 78.2h of speech from individual headset microphones (AMI-IHM). The alignments were generated based on 10ms frames and the decision trees with 3996 clustered triphone-states. Both FBANK and raw waveform data was normalised at the utterance level for zero mean and at the meeting level for unit variance. The systems were evaluated with the official dev and evaluation (eval) sets, which contain 9.0h and 8.7h speech, using the official testing dictionary with an 49.4k word vocabulary [29], a 4-gram (fg) LM, and Viterbi decoding.

5. Experiments

Initially all systems were evaluated on the CHiME4 dataset. At a later stage, key results were validated on the AMI dataset.

5.1. CHiME4 Channel 5

The 4L-512d-DNN baseline based on the FBANK features is denoted as F^{400}_{160} with 160 and 400 defining the filter shift and filter size in number of samples used in the STFT respectively[1]. For the single-span AM using raw waveform signal input, the output $\mathbf{o}$ of a single input stream

$$\mathbf{o} = \mathrm{CNN}^{2560}_{1024}(\mathrm{CNN}^{L}_{S}(\mathbf{x}^{T/2}_{\text{-}T/2})) \qquad (8)$$

is directly fed into a 4L-512d-DNN without dimension reduction. We denote the single-span AM as I^L_S with L and S corresponding to the kernel size and stride of the CNN input layer. All weights were randomly initialised without any pretraining.

Table 1: *%WERs with a tg LM and AMs with single input stream on CHiME4 dev set. Stride S and kernel size L are varied, and L and span length T are counted in waveform samples and ms.*

ID	S	L	T	dev
F^{400}_{160}	160	400	125	18.1
I^{400}_{10}	10	400	149	20.2
I^{100}_{10}	10	100	131	19.4
I^{50}_{10}	10	50	128	19.3
I^{25}_{10}	10	25	125	20.7
I^{50}_{4}	4	50	53	23.2
I^{50}_{9}	9	50	115	19.7
I^{50}_{15}	15	50	190	18.3
I^{50}_{20}	20	50	252	20.7

The single-span AM is an extension of the model proposed in [16]. In the first experiment, different kernel sizes L and strides S for I^L_S were tested giving the WERs in Table 1. The single-span AM gives lower WERs when using smaller kernel sizes, with I^{50}_{10} giving a 4.5 % relative improvement over using the standard kernel size of 400 [8, 14, 16]. The input span T makes a noticeable difference to the WERs. Using I^{50}_{10} as our reference point, a span of 190ms (I^{50}_{15}) relatively improves the WER by 5.3 %. Furthermore, our best performing single-span AM I^{50}_{15} only gives a slightly worse WER than the baseline F^{400}_{160}, and yields a relative 18.4% improvement over the comparable raw waveform system on CHiME4 in [33].

In the next experiment, the proposed multi-span structure was investigated for different constraints on stride and kernel size. After concatenation, the output vector $\mathbf{p}$ of 450-d was fed into the 4L-512d-DNN. All systems in this section use layer-by-layer pre-training by first training one epoch on a sub-network where $\mathbf{p}$ is directly fed into the output layer and then training another epoch extending the sub-network with two 512-d hidden

[1] In comparison with [28] where the AM is much larger, or [34] where the AM uses recurrent layers and discriminative sequence training, the baseline F^{400}_{160} WER in Table 1 is reasonably good.

DNN layers before the output layer. We denote the multi-span AM as $M^{L_1,L_2,L_3}_{S_1,S_2,S_3}$ with L_i and S_i giving the stride and kernel size of the CNN input layer in stream $i \in \{1, 2, 3\}$. Table 2 shows the results.

Table 2: *%WERs with a tg LM and AMs with multiple input streams on CHiME4 dev set. Stride combinations S_1, S_2, S_3 and kernel size combinations L_1, L_2, L_3 are varied.*

ID	S	L	T	dev
$M^{50,100,400}_{15,15,15}$	15	50,100,400	190-212	18.4
$M^{50,100,400}_{4,9,15}$	4,9,15	50,100,400	53-212	17.9
$M^{50,50,50}_{4,9,15}$	4,9,15	50	53-190	17.1

For the first system $M^{50,100,400}_{15,15,15}$, every input CNN layer convolves over the raw waveform signal with the same stride leading to a small range of input spans 190–212ms. Similar to [12], it was observed that the small kernels mainly act as a filter for high frequencies and that the larger kernels filter principally lower frequencies, which strongly resembles wavelet filters. However, this did not improve the WER over the single-span. Additionally using different strides in each CNN input layer and therefore increasing the range of different spans to $53 - 212$ms, the system $M^{50,100,400}_{4,9,15}$ yields an improvement over the single-span AM. Finally, all kernels were set to size 50 and it can be seen that the system $M^{50,50,50}_{4,9,15}$ reduces the WER to 17.1 % absolute. Also, we found that even for a fixed kernel size of 50, the multi-span AM learns wavelet-like filters by setting the weights at the beginning or the end of a kernel to close to zero to effectively shorten the kernel size.

5.2. AMI-IHM

The key results were validated using AMI to see how well the model architectures generalize to different datasets. A baseline based on 40-d FBANK input features was evaluated for comparison[2]. Table 3 summarizes the results of the key systems I^{400}_{10}, I^{50}_{10} and $M^{50,50,50}_{4,9,15}$ on AMI-IHM.

Table 3: *%WERs with a fg LM and AMs with single and multiple input streams compared to baseline AM based on FBANK on AMI-IHM dev and eval set.*

ID	System	dev	eval
F^{400}_{160}	FBANK-DNN	28.3	31.1
I^{400}_{10}	Single-Span-DNN	29.1	31.9
I^{50}_{15}	Single-Span-DNN	28.1	30.8
$M^{50,50,50}_{4,9,15}$	Multi-Span-DNN	27.2	29.3

Table 3 shows that the single-span AM using raw waveform signal input gives lower WERs with a smaller kernel size and larger input span also on AMI. I^{50}_{15} gives a similar WER to the FBANK-DNN AM, while the multi-span AM $M^{50,50,50}_{4,9,15}$ outperforms the FBANK-DNN AM by a relative WER reduction of 4.8%. Comparing $M^{50,50,50}_{4,9,15}$ to F^{400}_{160} on both AMI and CHiME4 data sets, a similar relative WER reduction of 5.5% is obtained on the CHiME4 dev set.

6. Discussion

Plotting the input CNN layer kernel weights of the single-span AMs I^{400}_{10} and I^{50}_{15} in the frequency domain reveals the typical audiological distributed narrow band pass filters as in [6, 7, 16]. When plotting the 64 kernels of size 400 in the time domain, it can be seen that some filter responses are learned only for a small part of the kernel, while the other part is set to zero (*cf.* Fig. 3 right). While this filter length shortening also happens when a kernel size of 50 is used, only a much smaller part of the kernel is set close to zero (*cf.* Fig. 3 left). This shows that the model automatically learns wavelet-like filters of different time-frequency resolution even for a small fixed kernel size.

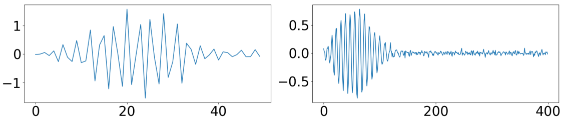

Figure 3: *Left: CNN input layer kernel in time domain of size 50 of trained system I^{50}_{15}. Right: CNN input layer kernel in time domain of size 400 of trained system I^{400}_{10}.*

In Fig. 4, the learned filters of the three CNN input layers of $M^{50,50,50}_{4,9,15}$ are smoothed by zero-padding, transformed to the Fourier domain and sorted by frequency. It can be seen that the learned filters of the three CNN input layers more or less cover the whole frequency spectrum with each filter focusing on a certain area, and that they are rather different compared to the log Mel curve used for handcrafted acoustic features.

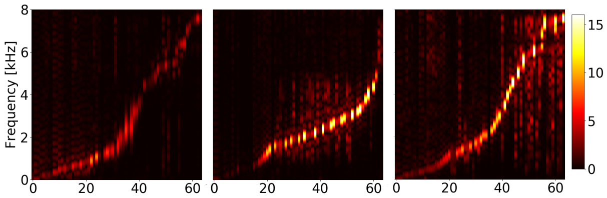

Figure 4: *Learned filters of the CNN input layers from our CHiME multi-span AM $M^{50,50,50}_{4,9,15}$ in frequency domain sorted by frequency, which are rather different to the log Mel-filters. Left: Stride 4, Middle: Stride 9, Right: Stride 15 .*

7. Conclusions

We have presented a novel achitecture for acoustic modelling using raw waveform input. Our model outperforms a conventional DNN-HMM system based on FBANK features on the CHiME4 dev set and on the AMI dev and eval sets. By reducing the kernel size from 400 to 50, leaving out any kind of compression layers in the model and tuning the input span, we achieved a significant reduction in WER, which questions the usefulness of imitating feature extraction pipelines when designing AMs based on raw waveform signal input. Analysis of the best-performing multi-span AM $M^{50,50,50}_{4,9,15}$ showed that the learned filters are different from log-Mel filters in that they do not seem to follow an audiological distribution (*cf.* Fig. 4).

8. Acknowledgements

Thanks to G. Sun for providing the language models for both the AMI-IHM and CHiME4 speech corpus. P. von Platen is funded by Studienstiftung des Deutschen Volkes.

[2] Considering F^{400}_{160} is a small DNN with four 512-d hidden layers and 4k node output layer, and fg LM is used for decoding, its WER is reasonable compared to those in [35].

9. References

[1] S. Davis, P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 357–366, 1980.

[2] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech", *Journal of the Acoustical Society of America*, vol. 87, pp. 1738–1752, 1990.

[3] G. Von Békésy, E.G. Wever, "Experiments in hearing", McGraw-Hill New York, 1960.

[4] v. Mitra, F. Horacio, R.M. Stern, "Robust features in deep-learning-based speech recognition", *New Era for Robust Speech Recognition*, pp. 187–217, 2017.

[5] Z. Tüske, P. Golik, R. Schlüter, H. Ney, "Acoustic modeling with deep neural networks using raw time signal for LVCSR", *Proc. Interspeech*, Singapore, 2014.

[6] T.N. Sainath, R.J. Weiss, A. Senior, K.W. Wilson, O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs", *Proc. Interspeech*, Dresden, 2015.

[7] P. Ghahremani, V. Manohar, D. Povey, S. Khudanpur, "Acoustic modelling from the signal domain using CNNs", *Interspeech*, San Francisco, 2016.

[8] Z. Tüske, R. Schlüter H. Ney, "Acoustic modeling of speech waveform based on multi-resolution, neural network signal processing", *Proc. ICASSP*, Calgary, 2018.

[9] V. Nair, G.E. Hinton, "Rectified linear units improve restricted Boltzmann machines", *Proc. ICML*, Haifa, 2010.

[10] T.N. Sainath, O. Vinyals, A. Senior, H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks", *Proc. ICASSP*, Brisbane, 2015.

[11] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, "Convolutional neural networks for speech recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1533–1545, 2014.

[12] Z. Zhu, J.H. Engel, A.Y. Hannun, "Learning multiscale features directly from waveforms", *Proc. Interspeech*, San Francisco, 2016.

[13] A. Haar, "Zur theorie der orthogonalen funktionensysteme", *Mathematische Annalen*, pp. 331–371, 1910.

[14] T.N. Sainath, R.J. Weiss, K.W. Wilson, A. Narayanan, M. Bacchiani, "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms", *Proc. ASRU*, Scottsdale, 2015.

[15] S. Kim, I. Lane, "End-to-end speech recognition with auditory attention for multi-microphone distance speech recognition", *Proc. Interspeech*, Hyderabad, 2017.

[16] P. Golik, Z. Tüske, R. Schlüter, H. Ney, "Convolutional neural networks for acoustic modeling of raw time signal in LVCSR", *Proc. Interspeech*, Dresden, 2015.

[17] E. Variani, T.N. Sainath, I. Shafran, M. Bacchiani, "Complex linear projection (CLP): A discriminative approach to joint feature extraction and acoustic modeling", *Proc. Interspeech*, San Francisco, 2016.

[18] C. Zhang, *Joint Training Methods for Tandem and Hybrid Speech Recognition Systems using Deep Neural Networks*, Ph.D. thesis, University of Cambridge, UK, 2017.

[19] P. Ghahremani, H. Hadian, H. Lv, D. Povey, S. Khudanpur, "Acoustic modeling from frequency-domain representations of speech", *Proc. Interspeech*, 2018.

[20] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[21] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *Proc. ICLR*, San Diego, 2015.

[22] A.-R. Mohamed, G.E. Hinton, G. Penn, "Understanding how deep belief networks perform acoustic modelling", *Neural Networks*, pp. 6–9, 2012.

[23] S.M. Siniscalchi, D. Yu, L. Deng, C.-H. Lee, "Speech recognition using long-span temporal patterns in a deep network model", *IEEE Signal Processing Letters*, vol. 20, pp. 201–204, 2013.

[24] P.C. Woodland, X. Liu, Y. Qian, C. Zhang, M.J.F. Gales, P. Karanasou, P. Lanchantin, & L. Wang, "Cambridge university transcription systems for the Multi-Genre Broadcast Challenge", *Proc. ASRU*, Scottsdale, 2015.

[25] T. Robinson, M. Hochberg and S. Renals. The use of Recurrent Neural Networks in Continuous Speech Recognition. In *Automatic speech and speaker recognition*, pp. 233–258, Springer, 1996.

[26] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K.J. Lang, "Phoneme recognition using time-delay neural networks", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 328–339, 1989.

[27] Y. Hoshen, R.J. Weiss, K.W. Wilson, "Speech acoustic modeling from raw multichannel waveforms", *Proc. ICASSP*, Brisbane, 2015.

[28] E. Vincent, S. Watanabe, A.A. Nugraha, J. Barker, R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition", *Computer Speech & Language*, pp. 535–557, 2017.

[29] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, "The AMI meeting corpus: A pre-announcement", *Proc. MLMI*, Edinburgh, 2005.

[30] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, A. Ragni, V. Valtchev, P. Woodland, & C. Zhang, *The HTK Book (for HTK version 3.5)*, Cambridge University Engineering Department, 2015.

[31] C. Zhang, F.L. Kreyssig, Q. Li, & P.C. Woodland, "PyHTK: Python library and ASR pipelines for HTK", *Proc. ICASSP*, Brighton, 2019.

[32] T. Menne, J. Heymann, A. Alexandridis, K. Irie, A. Zeyer, M. Kitza, R. Schlüter, "The RWTH/UPB/FORTH system combination for the 4th CHiME challenge evaluation", *Proc. Interspeech*, San Francisco, 2016.

[33] T. Menne, Z. Tüske, R. Schlüter, H. Ney, "Learning acoustic features from the raw waveform for automatic speech recognition", *44. Jahrestagung fr Akustik der Deutschen Gesellschaft für Akustik*, Munich, 2018.

[34] C. Weng, D. Yu, S. Watanabe, B.-H.F. Juang, "Recurrent deep neural networks for robust speech recognition", *Proc. ICASSP*, Florence, 2014.

[35] S. Renals, P. Swietojanski, "Neural networks for distant speech recognition", *Proc. HSCMA*, Nancy, 2014.