Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition

Yanmin Qian, *Member, IEEE*, Mengxiao Bi, *Student Member, IEEE*, Tian Tan, *Student Member, IEEE*, and Kai Yu, *Senior Member, IEEE*

***Abstract*—Although great progress has been made in automatic speech recognition, significant performance degradation still exists in noisy environments. Recently, very deep convolutional neural networks (CNNs) have been successfully applied to computer vision and speech recognition tasks. Based on our previous work on very deep CNNs, in this paper this architecture is further developed to improve recognition accuracy for noise robust speech recognition. In the proposed very deep CNN architecture, we study the best configuration for the sizes of filters, pooling, and input feature maps: the sizes of filters and poolings are reduced and dimensions of input features are extended to allow for adding more convolutional layers. Then the appropriate pooling, padding, and input feature map selection strategies are investigated and applied to the very deep CNN to make it more robust for speech recognition. In addition, an in-depth analysis of the architecture reveals key characteristics, such as compact model scale, fast convergence speed, and noise robustness. The proposed new model is evaluated on two tasks: Aurora4 task with multiple additive noise types and channel mismatch, and the AMI meeting transcription task with significant reverberation. Experiments on both tasks show that the proposed very deep CNNs can significantly reduce word error rate (WER) for noise robust speech recognition. The best architecture obtains a 10.0% relative reduction over the traditional CNN on AMI, competitive with the long short-term memory recurrent neural networks (LSTM-RNN) acoustic model. On Aurora4, even without feature enhancement, model adaptation, and sequence training, it achieves a WER of 8.81%, a 17.0% relative improvement over the LSTM-RNN. To our knowledge, this is the best published result on Aurora4.**

***Index Terms*—Convolutional neural networks, very deep CNNs, robust speech recognition, acoustic modeling.**

I. Introduction

We have witnessed significant progress in automatic speech recognition (ASR) in the last few years due to the introduction of deep neural network (DNN) based acoustic models [1]–[3]. These advancements have reduced word error rate (WER) to a level that has passed the threshold for adoption in many close-talk scenarios, such as voice search on a smart phone and dictation in an office room, where the signal-to-noise ratio (SNR) is relatively high. However, these systems still perform poorly in noisy environments, such as scenarios with additive noise or convolutional distortions [4], and noise robustness is still a critical issue for making ASR systems widely adopted in real scenarios. The performance degradation problem is magnified under the distant talking condition [5], where speech signal strength is lower, leading to low SNR and making the system susceptible to additive noise and reverberation.

Several technologies within the DNN framework [6]–[8] have been proposed to handle the difficult problem of mismatch between training and testing in the noisy speech recognition scenario, and some improvements have been obtained, however, there is still a large performance gap when compared to the close-talking scenario with high SNR. More recently, some other novel neural networks structures have been explored, of which convolutional neural networks (CNNs) is one of the most important types. CNNs have been shown to be better than a classic fully connected feed-forward DNN on several tasks [9], [10]. CNNs have several advantages: First, speech spectrogram have local correlations in both time and frequency, and CNNs are well suited to model those correlations explicitly through a local connectivity, whereas DNNs have relatively more difficulty to encode this information. Second, translational invariance, such as frequency shift due to speaking styles or speaker variations, can also be more easily captured by CNNs than by DNNs [11]–[14].

The effect of CNNs was first verified in image classification [11], [12], [15] and then applied to speech recognition [9], [11], [13], [14], [16], [17]. Most of the previous CNN approaches for speech recognition have only used up to 2 convolutional layers, and [9] tried to increase to 3 convolutional layers but obtained deteriorated performance. Based on such previous work, the configuration described in [14] is usually utilized for CNN modeling in speech recognition, with 2 convolutional layers plus 4 fully connected layers. Recently, the computer vision community [18]–[21] has found that the performance of image classification can be improved by using a substantially increased number of convolutional layers with carefully designed topology. Specifically, VGGNet [19] is constructed using very simple building blocks, such as convolutional layers with 3×3 filters and 2×2 pooling layers, and shows impressive performance.

These results on image processing suggest investigating very deep CNNs for speech recognition. Our previous work in [22]

Manuscript received June 5, 2016; revised August 22, 2016; accepted August 22, 2016. Date of publication August 25, 2016; date of current version September 19, 2016. This work was supported in part by the Shanghai Sailing Program 16YF1405300, in part by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, in part by the China NSFC projects (61573241 and 61603252), and in part by the Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sabato Marco Siniscalchi. *(Corresponding authors: Yanmin Qian and Kai Yu.)*

The authors are with the Computer Science and Engineering Department, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yanminqian@sjtu.edu.cn; sxkachilles@sjtu.edu.cn; tantian@sjtu.edu.cn; kai.yu@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2016.2602884

implemented a very deep CNN with many convolutional layers for speech recognition for the first time, and achieved a significant improvement in WER. The work in [23] has since applied a similar idea and verified the effectiveness of very deep CNNs.

This initial work [22], [23] has motivated further exploration of the very deep CNN structure, including application to noisy scenarios. As described above, noise robustness is still a critical issue for speech recognition in most real applications, where additive noise, channel distortions and reverberation exist [4], [5]. In this work, we develop a very deep CNN for acoustic modeling, and study various design aspects of the architecture in detail for the noisy scenario. Comprehensive investigation and in-depth analysis of the model are performed. The proposed very deep CNNs are evaluated on two tasks for robust speech recognition: Aurora4 noisy speech recognition with additive noise and channel distortion [24], and AMI distant speech recognition in a reverberant scenario [5], where the speech signals are captured by microphones located farther away from the speaker. The results show that very promising performance can be achieved on both tasks with the proposed new model, even without using front-end denoising [25], [26] or model adaptation [27]–[29].

The rest of this paper is organized as follows. In Section II the conventional CNN is revisited, and the basic structure and explanation are presented. In Section III a novel CNN architecture, named very deep CNNs, is presented and some fundamental principles are given. Section IV explores the design aspects of these very deep CNNs in detail, with comparisons to other approaches. The experimental results for the additive noisy task Aurora4 and reverberant task AMI are reported in Section V, with conclusions in Section VI.

II. Review of Convolutional Neural Networks

Traditional CNNs for speech recognition consist of several convolutional layers and pooling layers, followed by several fully-connected layers for acoustic modeling. The additional convolutional and pooling layers are the main differences of CNNs compared to DNNs.

A convolutional layer does the convolutions on feature maps of the previous layer using filters, and then adds a bias scalar to the corresponding feature map, followed by a non-linear operation.

Feature maps are the basic units of convolutional layers and pooling layers. The typical speech inputs, with static, delta and double delta features, can be represented as 3 feature maps and each of them can be viewed as an image-map with a size of $\#times \times \#freqs$, usually 11 $\times$ 40, where $\#times$ is the context window size of the input features and $\#freqs$ is the dimension of the frequency-based features. FBANK features have been shown to be more effective than MFCC and PLP for CNN usage in speech processing, due to two reasons: 1) there is correlation information at the frequency scale represented in the FBANK features which can be utilized by convolution operations, and 2) each pooling operates on ordered-frequency feature maps, which decreases the resolution in a meaningful way. In contrast, the DCT transform of the MFCC will break this property.

A convolution can be viewed as an operation applied to the feature map using a filter, where both the feature map and the filter can be represented as matrices. The process can be regarded as moving the filter from the left top corner of the feature map to the right bottom corner step by step (the step size is also a hyper-parameter, but in this work this is fixed to 1). The covered area of the feature map during this process is called a receptive field, and the size is equivalent to the size of the filter. At each step, the receptive field performs a dot product with the filter and gives one output, and all the outputs gathered during the process will form a new feature map. The left part of Fig. 1 gives one example of this convolution operation, and the complete process of a convolutional layer is expressed as Equation (1).

$$\mathbf{h}^{(l)} = \sigma\left(\mathbf{W}^{(l)} * \mathbf{h}^{(l-1)} + b^{(l)}\right) \tag{1}$$

where $\mathbf{h}^{(l-1)}$ and $\mathbf{h}^{(l)}$ are feature maps in two consecutive layers. The convolution operation (denoted as $*$) is performed within the filter $\mathbf{W}^{(l)}$ and the feature map $\mathbf{h}^{(l-1)}$. The bias $b^{(l)}$ is added and finally the activation function $\sigma(\cdot)$, typically sigmoid or ReLU, will be applied to generate the outputs of the convolutional layer. Equation (1) shows the simplest situation where only one feature map exists in the previous layer. When multiple feature maps are present in the previous layer, the results of convolution operations are accumulated first before adding the bias.

The number of parameters of convolutional layers is relatively small because the filter is shared among all receptive fields in one feature map. So the parameters number of one convolutional layer can be calculated as

$$\#params^{(l)} = filter\_size^{(l)} \times m^{(l)} \times m^{(l-1)} \tag{2}$$

where $m^{(l)}$ is the number of feature maps in the l^{th} layer.

A pooling layer performs down-sampling on the feature maps of the previous layer and generates new ones with a reduced resolution. Several pooling strategies have been investigated [14], including max-pooling, stochastic-pooling [30] and L_p-pooling, with similar performance. In this work, max-pooling is used in all CNN models. A 1 $\times$ 2 max-pooling on one feature map is illustrated as the right part of Fig. 1.

Currently the most popular configuration for CNNs used in speech recognition is that of [14], which has 2 convolutional layers with 256 feature maps in each, using a 9 $\times$ 9 filter with 1 $\times$ 3 pooling in the 1st convolutional layer and a 3 $\times$ 4 filter in the 2nd convolutional layer without pooling. This setup is used as the baseline CNN in this paper.

III. Very Deep Convolutional Neural Networks

Some previous work shows that when stacking more fully-connected layers (more than 7 hidden layers) in normal DNNs for speech recognition, the accuracy is not further improved [31], [32]. Previous attempts increasing the number of CNN layers from 2 to 3 also gave a degradation [9]. However, the recent work in image shows that the accuracy of image classification can be improved by increasing the number of convolutional layers with carefully tuned architecture [18], [19].

Our previous work in [22] introduced very deep CNNs into speech recognition for conversational telephone speech,

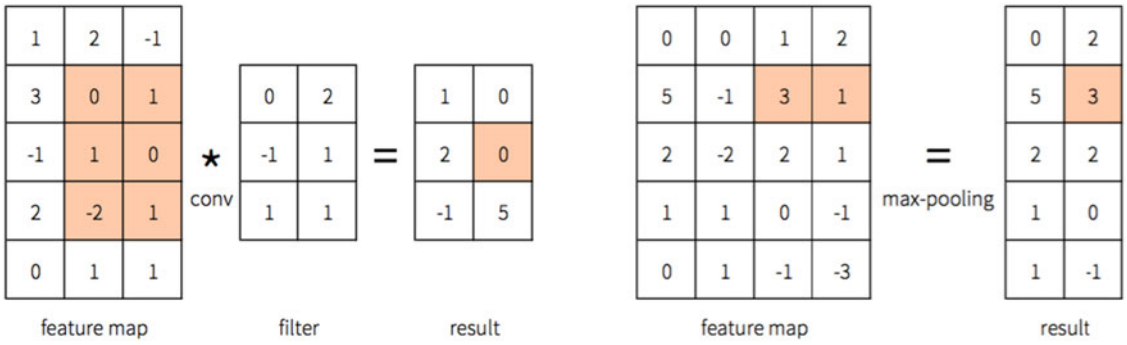

Fig. 1. Example of convolution and pooling (left: convolution; right: max-pooling).

showing the promising potential of this kind of deep model for speech recognition. Based on this initial work on the very deep CNNs, the models are further developed to improve the application in noisy scenarios.

Before a detailed architecture description, we first present some fundamental considerations regarding very deep CNNs for speech recognition:

1) As opposite to the traditionally used 9×9 or 3×4 filter and 1×3 pooling in speech recognition [9], the very deep CNNs for speech recognition proposed in this work use filters of 3×3 (sometimes 1×3 and 3×1), with pooling size constrained to 1×2 or 2×2. The stride of convolution is set to 1 and only non-overlapping pooling is used. These are the smallest reasonable sizes for filter and pooling as shown in [19], which make it possible to increase the number of convolutional layers.
2) Compared to computer vision tasks, the size of neural network inputs in speech recognition is relatively small. Both the context window size and the FBANK dimension are typically much smaller than image inputs[1]. Accordingly, in addition to the adjustment of the size of filters and pooling, the input size needs to be enlarged appropriately to allow more convolution and pooling operations. All the proposed very deep CNNs in this work only one input feature map, i.e. the static feature map, is used unless otherwise noted using three input feature maps with dynamic features.
3) For very deep CNNs, a pooling layer is added after at least two convolutional layers. The feature map size before the first fully-connected layer is set to a relatively small value in our proposed model, either 1×3 or 1×2. The number of feature maps is increased gradually and doubled after some pooling layers. For the model configurations in this paper, the number of feature maps is increased gradually from 64 to 128 to 256.
4) In addition to the convolutional layers and pooling layers, 4 fully-connected layers are added, followed by a softmax output layer.

Following these fundamental principles, the proposed very deep CNNs are designed and the detailed exploration will be described in the next section. Topologies are listed in Table I.

[1]Normally using 11×40 inputs in speech, 40-dim FBANK features with 11 consecutive frames.

IV. Architecture

In this section, the architectures of the proposed very deep CNNs will be described in detail. The `vd6` model configuration is shown in Table I. This model shares the same context window size and feature dimension, i.e. 11×40 as the traditional CNN introduced in [9]. Following the fundamental principle on the small size of filter and pooling described in Section 3, 5 convolutions are performed in time, and 6 convolutions and 2 poolings are performed in frequency, which results in a deep CNN with 6 convolutional layers and 2 pooling layers.

Based on the `vd6` model, further developments are explored. The structures of all models discussed in this work are illustrated in Fig. 2. In this figure, color indicates type of pooling layer, and border styles indicates padding strategies.

A. Experimental Setup on Aurora4

To investigate the behavior of the very deep CNNs for noise robust speech recognition, different structures are implemented and compared on the standard Aurora4 task. More details, including the data corpus, model configuration and training pipeline, are given in Section V-A and Section V-B.

Aurora4 [24] is a medium vocabulary task based on the Wall Street Journal (WSJ0) [33]. It contains 16 kHz speech data in the presence of additive noises and linear convolutional channel distortions, which were introduced synthetically to clean speech derived from the WSJ0 database. The multi-condition training set with 7138 utterances from 83 speakers includes a combination of clean speech and speech corrupted by one of six different noises at 10-20 dB SNR, some from the primary Sennheiser microphone and some from the secondary microphone. Similar to the training data, the same types of noise and microphones are used to generate the test set, grouped into 4 subsets: clean, noisy, clean with channel distortion, and noisy with channel distortion, which are referred to as A, B, C, and D, respectively. The GMM-HMM system is first built and then used to generate the senone alignments for later neural network training[2]. The task-standard WSJ0 bigram language model is used for decoding.

[2]In this task, the alignments are generated from the synchronized clean-condition training set of Aurora4, and more details could be found in Section V-B.

TABLE I
DETAILED CONFIGURATIONS OF VERY DEEP CNNS WITH DIFFERENT CONTEXT WINDOWS AND FEATURE DIMENSION SIZES. $M \times N$ REFERS TO A CONVOLUTIONAL LAYER, AND $[M \times N]$ REFERS TO A MAX-POOLING LAYER, WHERE M AND N INDICATE THE SIZES FOR TIME AND FREQUENCY RESPECTIVELY

Model	vd6	time-ext	freq-ext	vd10	full-ext	CNN	CNN2
Input map size	11×40	17×40	11×64	17×64	21×64	11×40	17×64
#(conv. layers)	6	8	10	10	10	2	2
64 feature maps	1×3	3×3	1×3	1×3	3×3	–	–
	3×3	3×3	1×3	1×3	3×3		
	$[1 \times 2]$	$[1 \times 2]$	$[1 \times 2]$	$[1 \times 2]$	$[1 \times 2]$		
128 feature maps	3×3	3×1	1×3	3×3	3×3	–	–
	3×3	3×3	1×3	3×3	3×3		
	$[1 \times 2]$	3×3	1×3	3×3	3×3		
		$[1 \times 2]$	3×3	3×3	3×3		
			$[1 \times 2]$	$[1 \times 2]$	$[1 \times 2]$		
256 feature maps	3×3	3×1	3×3	3×3	3×3	9×9	13×13
	3×3	3×3	3×3	3×3	3×3	$[1 \times 3]$	$[1 \times 4]$
		3×3	3×3	3×3	3×3	3×4	5×6
			3×3	3×3	3×3		
Output map size			1×3			1×8	

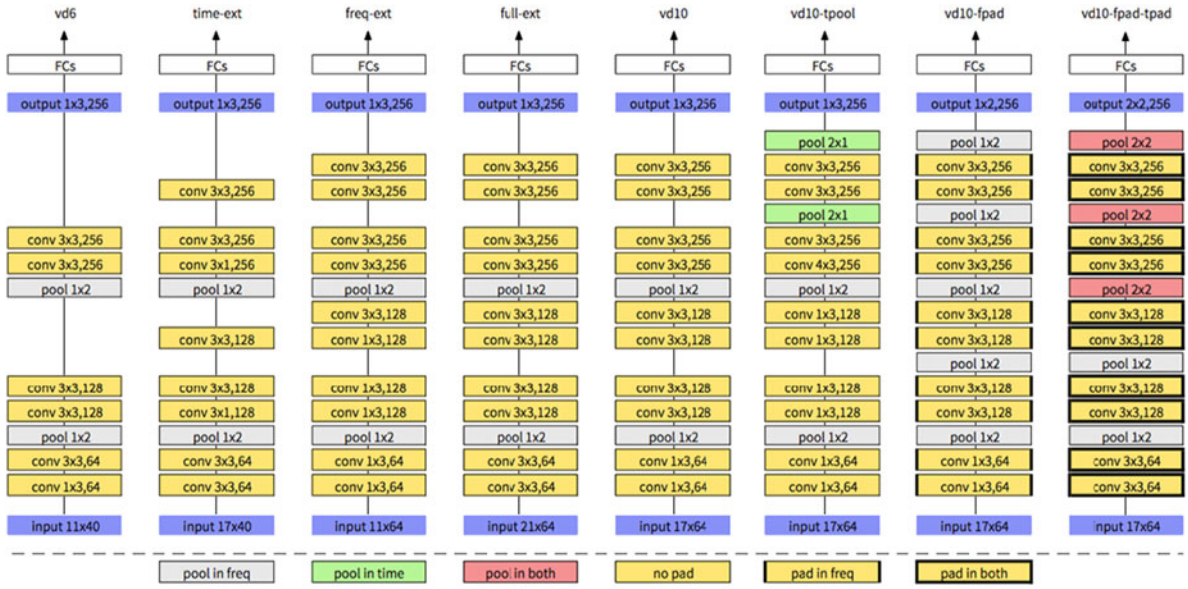

Fig. 2. Architecture of very deep CNNs.

B. Context Window Extension

Compared to computer vision, where the input features are large enough to allow stacking many building blocks in the very deep CNNs, a typical size of input features in speech recognition is 11×40, which is too small to stack with more convolutional layers and pooling layers. Accordingly, context window extension and feature dimension extension are explored in this and the next section respectively. Related models are listed in Table I.

In traditional DNN or CNN configurations, the context window size is typically 11. Using this context window size, convolutions can be performed in time 5 times with a filter size of 3, as in vd6. In this work, the context window size is extended to 17 and further to 21, which allows 8 and 10 convolutions to be performed in time respectively.

For model time-ext, the context window size is extended to 17, and 8 convolutional layers are stacked. Shown in the first three lines of Table II, the first line is the traditional baseline CNN with 2 convolutional layers and the other two lines are the proposed very deep CNNs. Increasing the number of convolutional layers can obtain a large improvement, and time-ext with context window extension, which stacks more convolutional layers, gets another significant improvement compared to vd6. Moreover most of the gain is obtained from subsets B, C and D, which indicates that more convolutional layers with appropriate context window extension is reasonable and effective for noise robust speech recognition.

C. Feature Dimension Extension

Based on the 40-dim FBANK features for CNNs, at most 6 convolutions and 2 poolings can be performed in frequency, leading to the vd6 model. In this work, the FBANK features are extended to 64-dim, so that 4 more convolutions can be performed in frequency. For model freq-ext, the feature

TABLE II
WER (%) COMPARISONS OF THE MODELS HAVING VARIOUS CONTEXT WINDOW AND FEATURE DIMENSION EXTENSIONS. F INDICATES THE SIZE ON FREQUENCY AXIS AND T INDICATES THE SIZE ON TIME AXIS. L INDICATES THE NUMBER OF CONVOLUTIONAL LAYERS IN THE MODEL

Model	T × F	L	A	B	C	D	AVG
CNN	11 × 40	2	4.11	7.00	6.33	16.09	10.64
vd6	11 × 40	6	3.94	6.86	6.33	15.56	10.34
time-ext	17 × 40	8	3.72	6.57	5.83	14.79	9.84
freq-ext	11 × 64	10	3.79	6.51	6.26	15.19	10.02
vd10	17 × 64	10	4.13	6.62	5.92	14.53	9.78
full-ext	21 × 64	10	4.04	6.23	5.40	13.86	**9.28**
CNN2	17 × 64	2	4.20	7.36	6.84	16.36	10.96

dimension is extended, and 10 convolutional layers are stacked. Shown as the fourth line of Table II, this larger feature dimension increases the CNN depth, giving a small but clear performance gain that is smaller than that from the context window extension.

Finally the input extension is performed in both time and frequency leading to a 17 × 64 input. The resulting model is named vd10 to indicate it has 10 convolutional layers. vd10 in Table II shows that the benefits from time-ext and freq-ext can be combined. In addition, the full-ext model further extends this from 17 × 64 to 21 × 64, so that 2 more convolution operations can be performed in time giving 10 convolution operations in both time and frequency. The results in Table II show that the extension in time is still helpful within the proposed very deep CNN architecture.

To confirm that the performance gain is mainly from the increased convolutional depth and not from the extended input features, a model with the same wider input features (17 × 64) but shallow convolutional layers (2 convolutional layers) is developed. This configuration is shown in the last column of Table I, with the results given in the last line of Table II. The results confirm that simply enlarging the size of input features is not helpful in the traditional shallow CNN structure, whereas very deep CNNs do benefit from the increased convolutional depth.

The results in Table II show that a large improvement can be obtained by increasing the convolutional layer from 2 (baseline CNN) to 6 (vd6), and further to 10 (vd10 or full-ext). Most of the contributions are in the three noisy subsets B, C and D, demonstrating the effect of the proposed very deep CNNs for noise robust speech recognition.

Considering the computational load and the latency in real-time applications, the very deep CNN model vd10 with 17 × 64 inputs shows a good trade-off between accuracy and real-time calculation, and is selected for the further development in the remainder of this work.

D. Pooling in Time

As shown in Table I, these very deep CNN models all use pooling in frequency and do no pooling in time. As shown in [9], pooling in time may result a degradation in the system performance. It is believed that pooling in time without overlap can be seen as sub-sampling the signal in time, and with overlapping can be seen as a way to smooth out the signal, which is another form of regularization [9].

TABLE III
WER (%) COMPARISON OF THE MODELS WITH OR WITHOUT POOLING IN TIME

Model	A	B	C	D	AVG
vd10	4.13	6.62	5.92	14.53	9.78
vd10-tpool	3.68	6.46	6.13	15.03	9.91

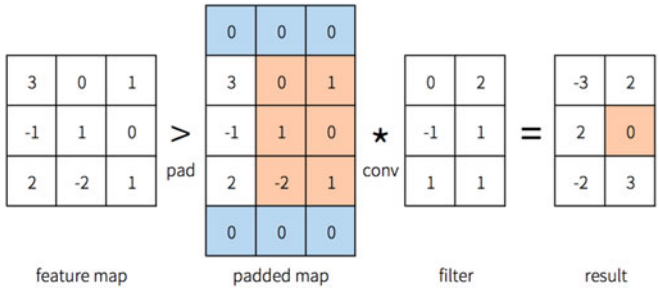

Fig. 3. Example of convolution with padding in one dimension.

To investigate whether pooling in time for very deep CNNs with a wider input and a deeper structure is still unhelpful, more experiments are conducted. As shown in Fig. 2, the vd10-tpool is a model with 2 additional temporal poolings applied (temporal convolutions are adapted to keep the sizes of input and output feature maps unchanged). The results in Table III show that pooling in time has no additional contribution beyond the very deep CNN vd10, a similar conclusion to that for the traditional CNNs [9].

E. Padding in Feature Maps

In most work on CNNs for speech recognition, including our previous work [22] on very deep CNNs and the work on traditional CNNs [9], [14], [34], the convolutions are performed without padding. However for computer vision [18], [19], convolutions are usually performed after zero-padding in feature maps. It can save the size of feature maps so that it is useful to increase the depth. In [19], the size of feature maps is reduced only after pooling layers. Padding of feature maps can better utilize the border information of feature maps by the neural network, which is beneficial for the final performance. Recently, [35] had some investigations on this topic. Fig. 3 gives an example of convolution with padding in one dimension. The size of the padding dimension can be preserved after the convolution operation.

In this work, we tried to use padding in very deep CNNs for speech recognition. Based on the model vd10, various padding strategies are implemented. Model vd10-fpad pads only in frequency, allowing more pooling operations in frequency, as shown in Fig. 2. In addition, padding in both dimensions is also applied, indicated as vd10-fpad-tpad. In this model, considering that pooling is a necessary approach to reduce the feature map size to a reasonable value when doing padding in time, so pooling in time is also applied. The model vd10-fpad

TABLE IV
WER (%) COMPARISON OF THE MODELS WITH DIFFERENT PADDING STRATEGIES

Model	A	B	C	D	AVG
CNN	4.11	7.00	6.33	16.09	10.64
vd10	4.13	6.62	5.92	14.53	9.78
vd10-fpad	3.57	6.17	5.31	14.24	9.38
vd10-fpad-tpad	3.27	5.61	5.32	13.52	**8.81**

TABLE V
WER (%) COMPARISON OF THE MODELS USING ONE CHANNEL OR THREE CHANNELS AS INPUTS

Model	#	A	B	C	D	AVG
vd6	1	3.94	6.86	6.33	15.56	10.34
	3	4.13	7.14	6.16	15.60	10.48
vd10	1	4.13	6.62	5.92	14.53	9.78
	3	3.90	6.93	6.26	14.75	10.01
vd10-fpad-tpad	1	3.27	5.61	5.32	13.52	8.81
	3	3.79	6.11	5.60	13.62	9.13

Indicates the Number of Input Feature Channels

and vd10-fpad-tpad are illustrated as the last two columns of Fig. 2.

Table IV shows the results of very deep CNNs with different padding strategies, compared to baseline shallow CNN. Model vd10-fpad is significantly better than vd10, and vd10-fpad-tpad padding in both time and frequency shows a further improvement. These results demonstrate that padding in feature maps for very deep CNNs is important, possibly benefiting from better encoding on the border information. The additional time pooling may have some joint impact with padding, however, it is challenging to completely separate the influences from padding and pooling, which may need more research in future.

Model vd10-fpad-tpad is the best proposed architecture using very deep CNNs for noise robust speech recognition. Compared to the traditional CNN, the very deep CNN containing up to 10 convolutional layers with appropriate padding and pooling strategies achieves a very large improvement.

F. One Channel vs. Three Channels Based Input Feature Maps

All the proposed very deep CNNs are using one channel feature map as input, i.e. the static FBANK features. Considering that most published work uses three channels for speech recognition (including the dynamic features, Δ and $\Delta\Delta$), the number of input channels are compared for very deep CNNs in the noisy scenario, and the results are shown in Table V.

It is interesting to find that the one channel based very deep CNNs are consistently better than the models using three channels under all architecture types in this noisy scenario. This conclusion is different from that of prior work using shallow CNN models [9]. One possible explanation would be that the dynamic features may have less information than the static features, and that this knowledge can be better extracted from the raw static features directly by the very deep CNNs. Some information, previously captured by dynamic features, can also be well captured using very deep CNNs with a wider context window.

Moreover, the analysis is made on the feature maps for the noisy data. One same noisy speech frame (more accurately it is the same context window of frames) with three channels (static, Δ and $\Delta\Delta$ respectively, and with 17×64 in each channel) is selected from Aurora4 as illustrated in Fig. 4(a). Selected feature maps of the first convolutional layer are illustrated in Fig. 4(b) and (c), where the same one noisy speech frame with one channel or three channels is propagated through the model vd10-fpad-tpad. We can see that the energy is more concentrated in the first layer feature maps when only using one channel inputs, whereas the energy distribution is more distributed using dynamic features. This observation may give another explanation on the results of Table V in noisy scenarios.

TABLE VI
COMPARISON OF MODEL PARAMETERS SIZE

Model	#Params	#Conv.	#Neck	#MLP
DNN	23.67M	–	–	23.67M
CNN	17.62M	0.85M	4.19M	12.58M
vd6	15.29M	1.14M	1.57M	12.58M
vd10	16.74M	2.59M	1.57M	12.58M
vd10-fpad	16.22M	2.59M	1.05M	12.58M
vd10-fpad-tpad	17.30M	2.62M	2.10M	12.58M

G. Model Parameters Size

The number of parameters is compared across models in Table VI. The DNN (6-layer) and CNN (2-cnn-layer + 4-mlp-layer) lines are also shown as baselines. The parameters are grouped into 3 types: #Conv indicates the parameters belonging to convolutional layers, #Neck indicates the parameters between the narrowed outputs of the full CNN block to the first fully-connected layer, and #MLP indicates the remaining parameters within the fully-connected layers. Softmax layer parameters are not included as they are the same for all models. 2048 hidden nodes are used for all the fully-connected layers in both DNN and CNN. The DNN and traditional shallow CNN use 11×120 FBANK static features with Δ and $\Delta\Delta$, and the new very deep CNNs use 11×40 static FBANK features for vd6 and 17×64 static FBANK features for vd10. The statistics show that convolution-related parameters (#Conv and #Neck) occupy only a rather small fraction of the total parameters.

It is observed that although the number of convolutional layers is increased significantly in the proposed very deep CNNs, the total parameter size is even smaller than the baseline CNN and DNN. The #Conv parameter increases slowly acompanied with the more convolutional layers, and the #Neck parameter has almost the same size due to the small neck value designed in our proposed very deep CNNs. The compact model size demonstrates another characteristic of the proposed very deep CNNs.

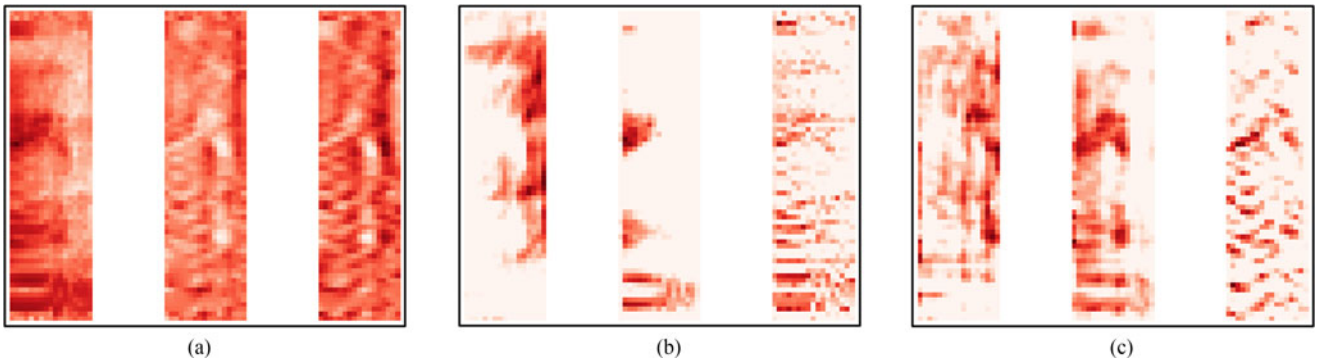

Fig. 4. One speech frame is selected to show the influence of one or three input channels. (a) Illustration of all three channels (i.e. static, Δ and $\Delta\Delta$ features) of the selected frame. (b) Illustration of some feature maps of the first convolutional layer, using one channel as input. (c) Illustration of some feature maps of the first convolutional layer, using three channels as inputs.

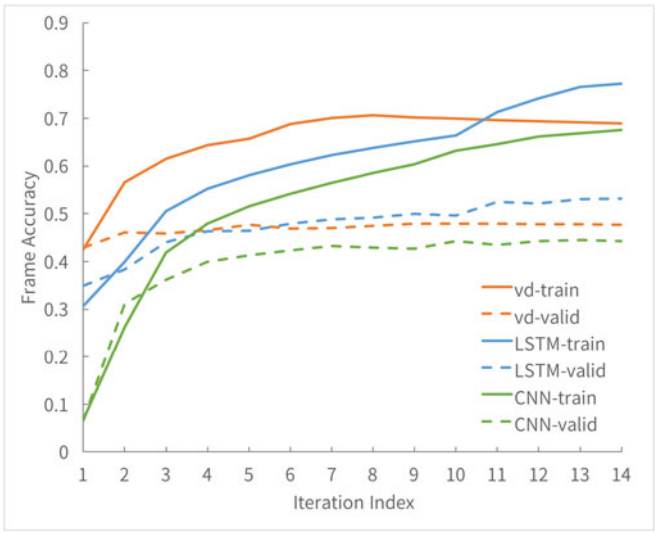

Fig. 5. Convergence curve comparison of different models: LSTM, traditional shallow CNN and the proposed very deep CNN.

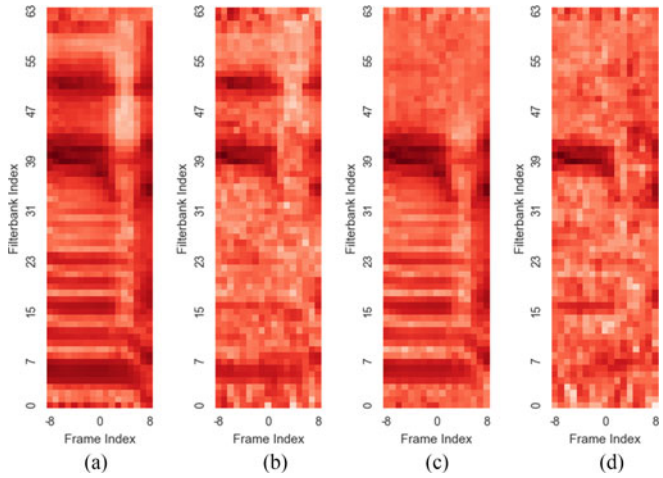

Fig. 6. One speech frame is selected to show the influence of additive noise and channel distortion. Illustration of static features of the same frame in Aurora4 from four conditions A, B, C and D in order.

H. Convergence of Very Deep CNNs

In the experiments, another interesting observation is that the proposed very deep CNNs converge faster than other model types. Fig. 5 gives the convergence curve comparison within the proposed very deep CNNs, the traditional shallow CNN and the LSTM-RNN. Stochastic gradient descent (SGD) based back-propagation (BP) algorithm is used for all training. Results show that the very deep CNNs converge the fastest, at about 6 iterations. In contrast, it usually takes about 12–14 iterations for the traditional shallow CNN and LSTM-RNN to converge. Accordingly although very deep CNNs need more computations in each iteration (9.5 times more computations compared to the baseline CNN), they need fewer training epochs (approaching half of other deep models). Based on this observation and our experiments, the very deep CNNs take comparable time for model training.

I. Noise Robustness of Very Deep CNNs

Finally we do an in-depth analysis on the effect of very deep CNNs in noisy scenarios. The individual input static feature maps (17×64) of the same single frame across four conditions in Aurora4 are illustrated in Fig. 6, denoted as A, B, C and D from left to right.

Compared to the clean data of A, the additive noise in B blurs the spectrum. Convolutional noise with channel distortion in C filters out some frequency bands but keeps the remainder almost unchanged. The mixture of both additive noise and channel distortion, shown as D, combines these impacts on the spectrum.

To better understand how the CNN processes noisy speech, each condition of this selected frame is propagated through the best performing model `vd10-fpad-tpad`. The outputs of the 1st convolutional layer and the 6th convolutional layer for A, B, C and D are plotted in Figs. 7 and 8, respectively. From Fig. 7 we can see that the feature maps after the 1st convolutional layer still inherit most of the properties from the inputs, and there are similar differences comparing the noisy data B, C, D to clean data A in both Figs. 6 and 7. If the feature maps from A are considered as the clean templates, the differences to the corrupted feature maps (B, C and D) are still obvious after only one convolutional layer processing. However, after the 6th convolutional layer illustrated in Fig. 8, we observe that the

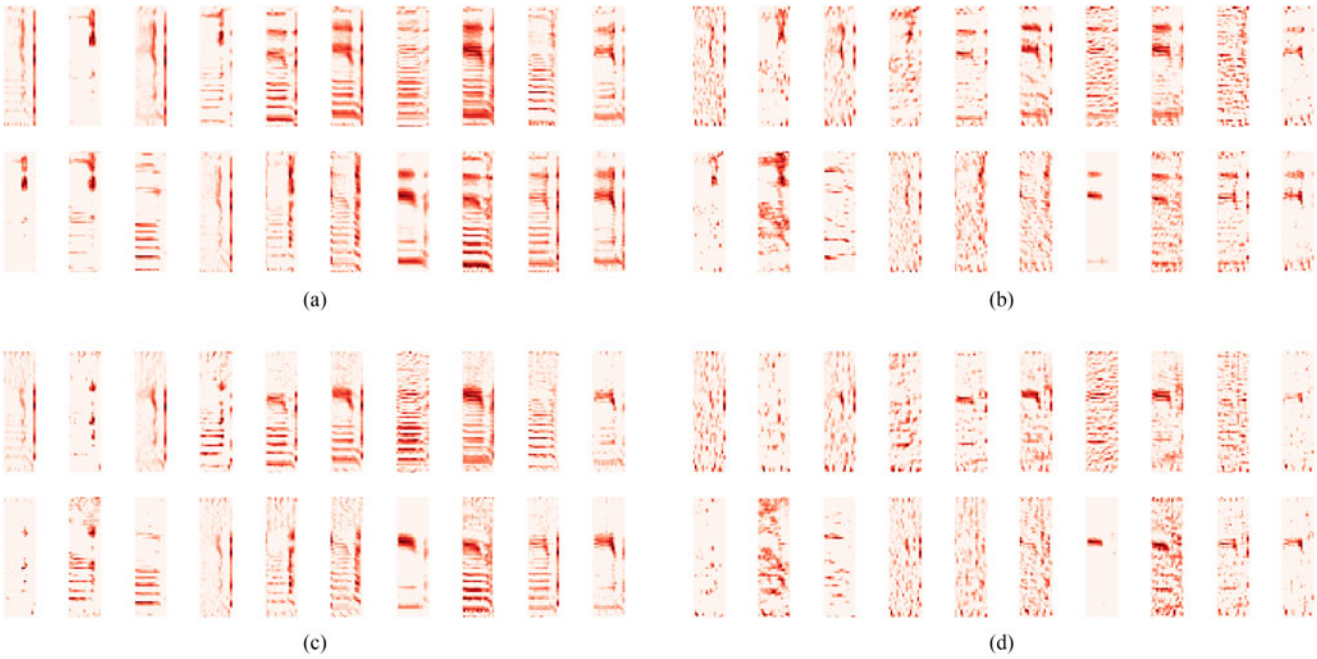

Fig. 7. Selected feature maps of the first convolutional layer using the same single frame from 4 different conditions in Aurora4.

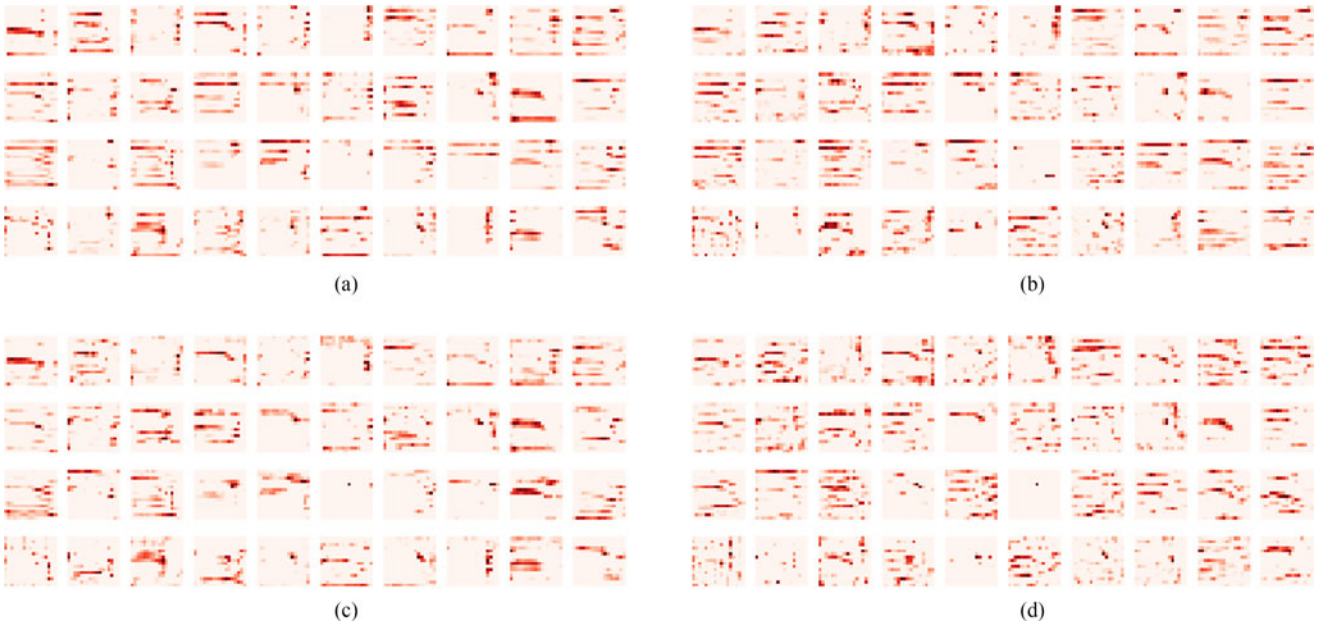

Fig. 8. Selected feature maps of the sixth convolutional layer using the same single frame from 4 different conditions in Aurora4.

feature maps from the three noisy conditions now look much like those in A. The differences between the corrupted noisy data and clean data are reduced significantly. Based on this observation, very deep CNNs seem to remove the noise embedding and denoise gradually across the stacked convolutional layers, which makes it especially useful for noise robust speech recognition.

To further verify this observation, the differences between noisy feature maps and clean feature maps are measured for all convolutional layers. Using data in the test, we compute the averaged mean square error (MSE) to evaluate the differences between the three noisy conditions and the clean condition. The MSE values after all convolution operations and pooling operations are shown in Fig. 9. The convolutional layers in these very deep CNNs seem to have distinct functions. The first four convolutional layers, which gradually increase the MSE, seem to construct the speech representations, while the higher six convolutional layers, which rapidly reduce the MSE, seem to perform denoising. (Note that there is an MSE increment after each max-pooling, and it is reasonable. Most of the outputs discarded by max-pooling are very small values (near zero), so the accumulated squared error decreases slightly from the preceding convolutional layer. However the total outputs number in

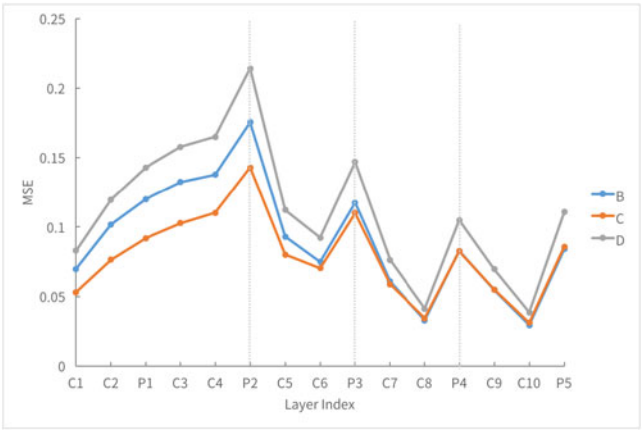

Fig. 9. The mean square error (MSE) variation of different layers of the very deep CNN (`vd10-fpad-tpad` is used) is illustrated. The MSE is calculated between layer outputs using noisy inputs (B, C and D, respectively) and clean inputs (A).

TABLE VII
THE MEAN SQUARE ERROR (MSE) OF OUTPUTS BEFORE THE FINAL SOFTMAX OPERATION OF DIFFERENT MODELS IS CALCULATED

Model	B	C	D
CNN	3.0282	2.3778	5.1597
`vd6`	2.9182	2.3515	4.3358
`vd10`	2.5510	2.0103	4.0397
`vd10-fpad`	2.2857	2.0037	3.7599
`vd10-fpad-tpad`	1.7611	1.4873	2.9115

The MSE is Calculated Between the Outputs Using Noisy Inputs (B, C and D, Respectively) and Clean Inputs (A)

pooling layer is reduced to $1/2$ or $1/4$ by 1×2 pooling or 2×2 pooling, so the final Mean Squared Error will increase rapidly in each pooling operation, which is calculated as the accumulated squared error divided by the number of nodes in that layer.) In contrast, in the CNN block the gap between the noisy and clean features is reduced gradually. This further demonstrates the embedded denoising process in the very deep CNNs, and that additional convolutional layers are helpful. Moreover comparing within the curves B, C and D, the MSE in D is higher than the other two, which means that D condition with both additive noise and channel mismatch is more difficult to remove the noise by the model. This observation is also consistent with the final WER on the different testing conditions, as shown in Table IV.

Finally, the MSE comparison between the noisy features and the clean features is evaluated for different CNN models, with results shown in Table VII. We use the final linear transformed outputs before the softmax operation to calculate the MSE between the noisy conditions (B, C and D) and the clean condition A for each different model. Results show that the MSE is gradually reduced as the number of convolutional layers increased, which supports the denoising ability of very deep CNNs. The MSE results are also consistent with the final WER, with smaller MSE values corresponding to an improved performance in noisy scenarios.

V. EXPERIMENTS

The proposed very deep CNNs are evaluated on two most interesting tasks for noise robust speech recognition: the Aurora4 task in which multiple noise types and channel mismatch exist, and the AMI meeting transcription task in which reverberation is the main concern.

A. Experimental Setup

The GMM-HMM systems are first built to generate the senone alignments for later neural network training. In both tasks, the GMM-HMM system is built with Kaldi [36] using the standard recipes. All neural network models, including DNN/CNN/LSTM, are trained using CNTK [37], running on one K20 GPU card. The first iteration is trained with a relatively small learning rate and zero momentum. On the subsequent iterations, the learning rate starts at 0.01 for DNN/CNN and 1.0 for LSTM, with a momentum of 0.9. The learning rate halves when validation loss stops decreasing. The standard testing pipelines in the Kaldi recipes are used for decoding and scoring. For better comparison, several baselines are constructed for both tasks, including DNN, CNN and LSTM.

1) The baseline DNN and CNN systems use the 40-dim FBANK features with $\Delta/\Delta\Delta$ and an 11-frame context window. The baseline DNN consists of 6 hidden layers of 2048 nodes. The baseline CNN utilizes the classical CNN configuration as in [14], with 2 convolutional layers each having 256 feature maps. The first layer has 9×9 filters and 1×3 pooling, and the second has 3×4 filters without pooling. Four fully-connected hidden layers with 2048 nodes in each layer are added after the CNN block. The final softmax layers of both DNN and CNN models have the same number of senones as the HMM model, and the minibatch is set to 256 in training.
2) The LSTM-RNN acoustic model has achieved great success recently [34], [38], [39], with good performance on several tasks. Accordingly, the LSTM-RNN is also used for comparison in noisy scenarios. The baseline LSTM system uses a single frame of 40-dim FBANK features with a 5 frame shift as input, and the model has 3 LSTMP [38] layers, where each LSTMP layer has 1024 memory cells and 512 hidden nodes in the projection. The truncated BPTT is utilized to train the LSTM model with the chunk size set to 20 frames, and 40 utterances are processed in parallel to form a mini-batch. To ensure the stability of training, the gradient is clipped to the range of $[-1, 1]$ during the parameter update.

Most of the pipeline for very deep CNNs training is the same as the baseline, and the detailed neural network configuration and structural design are described in the previous sections. Recently researchers from IBM and NYU have also developed a very deep CNN structure for speech recognition [23], with good results on telephone transcription similar to our previous work [22]. For better comparison, the best model architecture recently

TABLE VIII
WER (%) COMPARISON OF DIFFERENT MODELS ON AURORA4

Model	A	B	C	D	AVG
DNN	4.17	7.46	7.19	16.57	11.11
CNN	4.11	7.00	6.33	16.09	10.64
LSTM	3.92	7.21	6.63	15.94	10.68
vd6	3.94	6.86	6.33	15.56	10.34
vd10	4.13	6.62	5.92	14.53	9.78
vd10-fpad	3.57	6.17	5.31	14.24	9.38
vd10-fpad-tpad	3.27	5.61	5.32	13.52	**8.81**
IBM-VGG	3.92	6.15	5.34	14.20	9.38

proposed in [23] is also constructed, with 10 convolutional layers and 3 input channels, 17×40 in each channel (Full details can be found in [23]). This model is indicated as IBM-VGG in this paper.

All neural networks developed in this work were trained using cross-entropy criterion (CE) with stochastic gradient descent (SGD) based backpropagation (BP) algorithm.

B. Evaluation on Aurora4

Aurora4 [24] is a medium vocabulary task based on the Wall Street Journal (WSJ0) [33]. It contains 16 kHz speech data in the presence of additive noise and linear convolutional distortion, introduced synthetically to clean speech derived from the WSJ0 database. Two training sets were designed for this task: one is a clean-condition training set consisting of 7138 utterances from 83 speakers recorded by the primary Sennheiser microphone, the other is the multi-condition training set also comprising 7138 utterances, which is time-synchronized with the clean-condition training set. Half of the multi-condition training set is recorded on the primary Sennheiser microphone and the other half is recorded on several different secondary microphones. Both halves in the multi-condition training set include a combination of clean speech and speech corrupted by one of six different noises (street traffic, train station, car, babble, restaurant and airport) at 10–20 dB SNR.

The evaluation set is derived from the WSJ0 5K-word closed vocabulary test set, which consists of 330 utterances from 8 speakers. This test set was recorded by the primary microphone and one secondary microphone. Each of these two parts is then corrupted by the same six noises used in the training set at 5-15 dB SNR, creating a total of 14 test sets. Notice that the types of noise are common across the training and test sets but SNRs of the data are not. These 14 test sets can then be grouped into 4 subsets: clean, noisy, clean with channel distortion, and noisy with channel distortion, which will be referred to as A, B, C, and D, respectively.

The GMM-HMM system consists of 3K states trained using maximum likelihood estimation with the MFCC-LDA-MLLT-FMLLR features. After building the GMM-HMM, forced-alignment is performed to get the senone labels using the synchronized clean-condition training set. The task-standard WSJ0 bigram language model is used for decoding.

Results are summarized in Table VIII. The CNN baseline is consistent with the previous published work in [40]. Ours is $\sim 0.3\%$ better, perhaps due to using more tied-states, i.e. 3K vs. 2K.

Both CNN and LSTM models get a large improvement over the normal feed-forward DNN, and the shallow CNN is competitive with the LSTM-RNN on this noisy task with additive noise and channel mismatch, which shows the advantage of CNN in the noisy scenario. Compared to the shallow CNN baseline, the performance can be improved substantially by increasing the number of convolutional layers, and appropriate padding and pooling strategies are helpful in very deep CNNs. Our best configuration achieves significant improvements on all subsets, and also gets a better performance than IBM-VGG. Some hypotheses may explain the reasons that the proposed model vd10-fpad-tpad is better than IBM-VGG. First, extended feature dimension with 64 FBanks in our architecture enables the model to better encode frequency knowledge compared to the 40 FBanks in IBM-VGG. Second, only a single input feature map is used in our very deep CNNs, which is demonstrated more useful for noise robust speech recognition based on our experiments, and in contrast IBM-VGG utilizes 3 input channels including dynamic features. Third, in our architecture three 2×2 pooling operations are applied at the top part of the model following every two convolution operations. This design makes the sizes in time and frequency of the feature maps more balanced, and it is better for modeling, e.g. the map sizes are respectively 17×32, 17×16, 8×8, 4×4, 2×2 after each pooling operation in the proposed vd10-fpad-tpad. In addition, this design also results in a smaller neck size of the map size on the top (2×2, the narrowed layer within the whole CNN block and the first fully-connected layer), which can contribute to a more compact model scale.

On Aurora4, the proposed best very deep CNN architecture achieves a WER of 8.81%, which is a 17.0% relative improvement over the LSTM-RNN. To our best knowledge this is also the best published result on Aurora4 without model adaptation, and even achieves similar performance to approaches using adaptation [29], [41].

C. Evaluation on AMI

The AMI corpus contains around 100 hours of meetings recorded in meeting rooms with specifically equipped instruments at three locations in Europe (Edinburgh, IDIAP, TNO) [5]. The acoustic signal was captured and synchronized with multiple microphones including individual head microphones (IHM, close-talk), lapel microphones, and one or more microphone arrays. For the data recorded by microphone arrays, there are two data sets, single distant microphone (SDM) and multiple distant microphones (MDM), where SDM used data recorded by one of the microphone array, and MDM was processed by a standard beamforming algorithm in the Kaldi recipe [42] to generate a single channel data set. Considering that the beamforming algorithm has been the standard front-end signal enhancement for distant speech recognition using microphone array, in this work, the beamformed MDM dataset is used for very deep CNNs evaluation. The AMI corpus is partitioned into about 80 hours as training set, 9 hours as development set and 9 hours as

TABLE IX
WER (%) COMPARISON OF THE PROPOSED VERY DEEP CNNs ON AMI MDM CONDITION WITH DIFFERENT INPUT SIZES. F INDICATES THE SIZE OF THE FREQUENCY AXIS AND T INDICATES THE SIZE OF THE TIME AXIS

Model	T × F	L	dev	eval
vd6	11 × 40	6	46.5	51.1
time-ext	17 × 40	8	45.5	50.1
freq-ext	11 × 64	10	45.7	50.7
vd10	17 × 64	10	44.8	49.3
full-ext	21 × 64	10	**44.5**	**49.0**

L Indicates the Number of Convolutional Layers in the Model

TABLE X
WER (%) COMPARISON OF THE PROPOSED VERY DEEP CNNs ON AMI MDM CONDITION WITH DIFFERENT POOLING AND PADDING STRATEGIES

Model	#	Pooling	Padding	dev	eval
vd10	3	F	–	45.7	50.5
vd10	1	F	–	44.8	49.3
vd10-tpool	1	F & T	–	45.0	49.6
vd10-fpad	1	F	F	43.7	48.2
vd10-fpad-tpad	1	F & T	F & T	**42.5**	**46.9**

F Indicates the **F**requency Axis and **T** indicates the **T**ime Axis. **#** Indicates the Number of Input Channels

evaluation set (the development set and evaluation set for MDM are referred as dev and eval respectively) as suggested in [17]. For decoding, the 50K-word AMI dictionary and a trigram language model are used, which is interpolated between the one created using the AMI training transcripts and the one using the Fisher English corpus.

The officially released Kaldi recipe was followed to build an MFCC-LDA-MLLT-SAT GMM-HMM model. This model uses 39-dim MFCC features and has roughly 4K tiedstates and 80K Gaussians. We then use this acoustic model to generate the senone alignment for neural network training.

The size of input features is also investigated for the very deep CNNs in the reverberant scenario, with results shown in Table IX. It is observed that extensions on both time and frequency are helpful for accuracy, and the time-extension is slightly more effective than that on frequency, as with in Aurora4. In addition, the larger context can better encode temporal knowledge which is especially useful for reverberation. The extended input features enable us to construct the model with more convolutional layers, giving a significant improvement.

As with Aurora4, the `vd10` is used as the suitable trade-off, and the effect of other designs, including pooling, padding, and input channel are investigated and compared for distant speech recognition. The results in Table X show that appropriate pooling and padding strategies are also very critical for the very deep CNNs in distant speech recognition: time pooling without padding is not useful, and padding on features maps is helpful for very deep CNNs. The best configuration with pooling and padding on both time and frequency achieves another large improvement based on `vd10`. Moreover, consistent with the results for Aurora4, the dynamic input features add no more contributions on the very deep CNNs in the reverberant scenario.

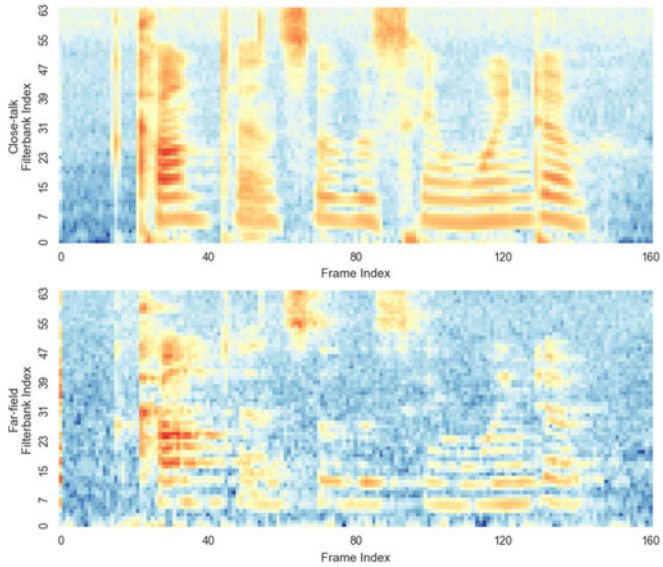

Fig. 10. Spectrogram comparison of the synchronized close-talk and distant speech in AMI to show the influence of distant condition.

TABLE XI
WER (%) COMPARISON OF DIFFERENT MODELS ON AMI MDM CONDITION

Model	dev	eval
DNN	47.5	52.3
CNN	46.3	51.3
LSTM	42.9	46.6
vd10-fpad-tpad	**42.5**	**46.9**
IBM-VGG	42.9	47.4

To better explain the superiority of very deep CNNs on distant speech recognition, the related feature maps are illustrated in Figs. 10–12. Fig. 10 shows the original input spectral feature comparison of the synchronized close-talk feature and distant corrupted feature. The spectral contamination from the reverberation is obvious, giving low SNR. Then one same single synchronized frame is selected from the close-talk and distant speech (one frame in the utterance shown in Fig. 10), and is propagated through the proposed very deep CNN `vd10-fpad-tpad`. The feature map outputs of the 1st convolutional layer and 6th convolutional layer are visualized in Figs. 11 and 12. Taking the feature maps from the close-talk speech as the clean templates and doing the comparison from distant to close-talk, the differences are still obvious after the 1st convolutional layer processing, however, they are reduced significantly after the 6th convolutional layer. The feature map outputs of the 6th convolutional layer from the distant speech look similar to those from the close-talk speech. This observation is consistent with that in Aurora4 with additive noise and channel mismatch, with the very deep CNNs restraining the reverberation and de-reverberation gradually across multiple convolutional layers, which makes it more effective for distant speech recognition.

Finally the baseline systems and the new proposed very deep CNN are summarized in Table XI on AMI distant speech recognition, including `IBM-VGG`. Both CNN and LSTM are

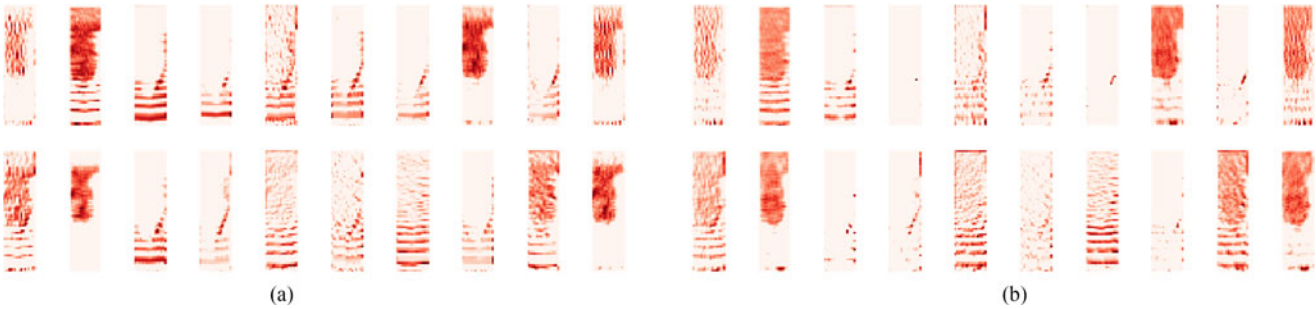

Fig. 11. Selected feature maps of the first convolutional layer using the synchronized frame from close-talk and distant conditions in AMI. (a) Close-Talk, (b) Distant.

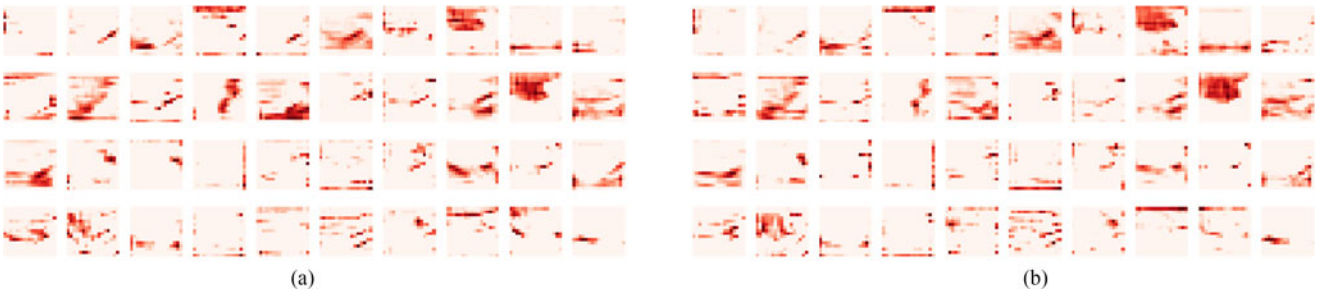

Fig. 12. Selected feature maps of the sixth convolutional layer using the synchronized frame from close-talk and distant conditions in AMI. (a) Close-Talk, (b) Distant.

significantly better than DNN, and LSTM performs particularly superior in this task, since long-time information encoded in LSTM-RNN is especially useful for modelling reverberation. Compared to the shallow CNN, the performance is improved significantly by the proposed very deep CNN, achieving competitive performance with the LSTM-RNN and better than the IBM-VGG. On AMI distant speech recognition, the proposed very deep CNN obtains a 10.0% relative WER reduction over a traditional CNN.

VI. Conclusion

Inspired by recent work in image processing, we propose very deep CNNs, which have more convolutional layers than traditional CNNs, for noise robust speech recognition. Compared to the traditional CNN strcuture in speech recognition, the sizes of filters and poolings are constrained to be small and the input feature maps are made larger. This adjustment enables us to increase the number of convolutional layers up to 10. Detailed analysis on pooling, padding and input feature maps selection are performed. Results show that time pooling without padding is not useful, but that padding on both dimensions of feature maps is effective for very deep CNNs. Compared to the traditional input feature channel usage with dynamic features, very deep CNNs only using the static features are more effective for noise robust speech recognition. The in-depth analysis of deep CNNs reveals that they have the ability to perform de-noising or de-reverberation gradually across multiple convolutional layers, which is particularly useful for noise robust speech recognition. In addition to superior performance, the proposed model has other advantages, such as compact model size and faster training convergence speed.

The proposed new model is evaluated on two noisy tasks: Aurora4 with additive noise and channel mismatch, and AMI meeting transcription mainly with reverberation. The very deep CNN obtains a 10.0% relative reduction over a traditional CNN on AMI, with accuracy competitive to an LSTM-RNN acoustic model. On Aurora4, it achieves a WER of 8.81%, which is a 17.0% relative improvement over the LSTM-RNN. That is a very promising system performance for noise robust speech recognition.

Acknowledgment

The authors would like to thank for the help from Prof. M. T. Johnson in University of Kentucky on the writing improvement.

References

[1] G. Hinton *et al.*,"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, Nov. 2012.
[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011, pp. 437–440.
[3] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.
[4] Y. Wang and M. J. Gales, "Speaker and noise factorization for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 7, pp. 2149–2158, Sep. 2012.
[5] T. Hain *et al.*,"Transcribing meetings with the amida systems," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 486–498, Feb. 2012.

[6] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 4, pp. 745–777, Feb. 2014.
[7] L. Shilin and K. C. Sim, "Joint adaptation and adaptive training of TVWR for robust automatic speech recognition," in *Proc. Interspeech*, 2014, pp. 636–640.
[8] T. Yoshioka *et al.*,"Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 114–126, Nov. 2012.
[9] T. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*., May 2013, pp. 8614–8618.
[10] T. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. Interspeech*, 2015, pp. 1478–1482.
[11] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *Handbook of Brain Theory and Neural Networks*, vol. 3361. Cambridge, U.K.: MIT Press, 1995, pp. 255–258.
[12] 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, Nov. 1998.
[13] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*., Mar. 2012, pp. 4277–4280.
[14] T. N. Sainath *et al.*, "Deep convolutional neural networks for large-scale speech tasks," *Neural Netw.*, vol. 64, pp. 39–48, 2015, special issue on Deep Learning of Representations.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[16] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.
[17] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1120–1124, Sep. 2014.
[18] C. Szegedy *et al.*,"Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit*., 2015, pp. 1–9.
[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.
[20] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1026–1034.
[21] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conf. Comput. Vision Pattern Recognit*., 2016, pp. 770–778.
[22] M. Bi, Y. Qian, and K. Yu, "Very deep convolutional neural networks for LVCSR," in *Proc. Interspeech*, Sep. 2015, pp. 3259–3263.
[23] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*., Mar. 2016, pp. 4955–4959.
[24] D. Pearce, "Aurora working group: DSR front-end LVCSR evaluation au/384/02," Ph.D. dissertation, Mississippi State Univ., Starkville, MS, USA, 2002.
[25] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "A minimum-mean-square-error noise reduction algorithm on mel-frequency cepstra for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*., 2008, pp. 4041–4044.
[26] T. Yoshioka and M. J. Gales, "Environmentally robust asr front-end for deep neural network acoustic models," *Comput. Speech Lang.*, vol. 31, no. 1, pp. 65–86, 2015.
[27] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7398–7402.
[28] Y. Qian, T. Tan, D. Yu, and Y. Zhang, "Integrated adaptation with multi-factor joint-learning for far-field speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5770–5775.
[29] P. Swietojanski, J. Li, and S. Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 8, pp. 1450–1463, Apr. 2016.
[30] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2013.
[31] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2011, pp. 24–29.
[32] D. Yu and L. Deng, *Automatic Speech Recognition—A Deep Learning Approach*. New York, NY, USA: Springer, 2015.
[33] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proc. Workshop Speech Natural Lang.*, 1992, pp. 357–362.
[34] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4580–4584.
[35] T. Yoshioka, K. Ohnishi, F. Fang, and T. Nakatani, "Noise robust speech recognition using recent developments in neural networks for computer vision," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2016, pp. 5730–5734.
[36] D. Povey *et al.*, "The kaldi speech recognition toolkit," presented at *IEEE 2011 Workshop Automatic Speech Recognition Understanding*, Big Island, HI, USA, Dec. 2011, IEEE Catalog no.: CFP11SRW-USB.
[37] A. Agarwal *et al.*, "An introduction to computational networks and the computational network toolkit," *Tech. Rep. MSR-TR-2014–112*, Aug. 2014. [Online]. Available: http://research. microsoft.com/apps/pubs/default.aspx
[38] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014, pp. 338–342.
[39] H. Sak *et al.*, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," *Entropy*, vol. 15, no. 16, pp. 17–18, 2014.
[40] S. J. Rennie, V. Goel, and S. Thomas, "Deep order statistic networks," in *Proc. 2014 IEEE Spoken Lang. Technol. Workshop*, 2014, pp. 124–128.
[41] S. Kundu, G. Mantena, Y. Qian, T. Tan, M. Delcroix, and K. C. Sim, "Joint acoustic factor learning for robust deep neural network based automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5025–5029.
[42] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 7, pp. 2011–2022, Sep. 2007.

Yanmin Qian (S'09–M'13) received the B.S. degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2012. Since 2013, he has been with the Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, China, where he is currently an Assistant Professor. From 2015 to 2016, he also worked as an Associate Research in the Speech Group, Cambridge University Engineering Department, Cambridge, U.K. His current research interests include the acoustic and language modeling in speech recognition, speaker and language recognition, key word spotting, and multimedia signal processing.

Mengxiao Bi (S'16) received the B.S. degree from the School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan, China, in 2013. He is currently working toward the graduate degree in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, working on speech recognition. His current research interests include speech recognition, natural language understanding, and deep learning.

Tian Tan (S'15) received the B.S. degree, in 2013, from the Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, China, where he is currently working toward the Ph.D. degree on speech recognition. His current research interests include speech recognition and deep learning.

Kai Yu (SM'10) received the B.Eng. degree in automation and the M.Sc. degree from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree from the Machine Intelligence Laboratory, Engineering Department, Cambridge University, Cambridge, U.K., in 2006. He is currently a Research Professor at Shanghai Jiao Tong University, Shanghai, China. His main research interests include the area of speech-based human machine interaction including speech recognition, synthesis, language understanding, and dialogue management. He was selected into the 1000 Overseas Talent Plan (Young Talent) by Chinese government and the Excellent Young Scientists Project of National Natural Science Foundation of China, China. He is a member of the Technical Committee of the Speech, Language, Music and Auditory Perception Branch of the Acoustic Society of China.