3-D CNN MODELS FOR FAR-FIELD MULTI-CHANNEL SPEECH RECOGNITION

Sriram Ganapathy
Indian Institute of Science, Bangalore.
sriramg@iisc.ac.in

Vijayaditya Peddinti†
Google, Inc., U.S.A.
vpeddinti@google.com

ABSTRACT
Automatic speech recognition (ASR) in far-field reverberant environments, especially when involving natural conversational multi-party speech conditions, is challenging even with the state-of-the-art recognition methodologies. The two main issues are artifacts in the signal due to reverberation and the presence of multiple speakers. In this paper, we propose a three dimensional (3-D) convolutional neural network (CNN) architecture for multi-channel far-field ASR. This architecture processes time, frequency & channel dimensions of the input spectrogram to learn representations using convolutional layers. Experiments are performed on the REVERB challenge LVCSR task and the augmented multi-party (AMI) LVCSR task using the array microphone recordings. The proposed method shows improvements over the baseline system that uses beamforming of the multi-channel audio along with a 2-D conventional CNN framework (absolute improvements of 1.1 % over the beamformed baseline system on AMI dataset).

Index Terms— Far-field speech recognition, 3D CNN modeling, Multi-party conversational speech.

1. INTRODUCTION
Multi-speaker conversations in far-field environments pose a significant challenge to automatic speech recognition systems even when employing state-of-the-art speech recognition systems [1]. e.g. Peddinti et al., [2] report a 75% rel. increase in word error rate (WER) when signals from a far-field array microphone are used in place of those from headset microphones in the ASR systems, both during training and testing. This degradation could be primarily attributed to reverberation and multi-speaker overlaps [3, 4]. The availability of multi-channel signals can be leveraged for alleviating these issues.

Beamforming is a popular approach for multi-channel signal based enhancement of speech [5, 6]. In this paper, we develop a neural network architecture consisting of 3-D convolutional models as a front-end for processing multi-channel speech. The spectrogram representation of speech is extracted from each channel independently. The 3-D CNN is fed with three dimensional input representation consisting of time, frequency and channel dimensions. The feature maps from the initial CNN layers are then fed to time delay neural network layers (TDNN) [7] followed by a final layers of sequence modeling with LSTM networks. The entire model is trained using a sequence cost function with lattice free minimum mutual information (MMI) criterion [8].

Experiments are primarily performed on the augmented multi-party speech database [9]. In these experiments, the proposed method improves the performance over a baseline system using a 2-D CNN-TDNN-LSTM architecture which uses either single channel speech or beamformed output of multi-channel speech as inputs. This model was shown to provide the best reported results on the single channel AMI LVCSR task [10]. We also contrast the performance of the proposed approach in the case of a single speaker setting (using the REVERB Challenge corpus [4]) with multi-party conversations in AMI dataset.

The rest of the paper is organized as follows. Prior work is discussed in § 2, proposed architecture is discussed in § 3, experimental setup is described in § 4 and the results for multi-channel speech recognition are reported in § 5. This is followed by a discussion § 6 and conclusion in § 7.

2. PRIOR WORK
Beamforming fundamentally relies on estimating the time delay between speech signals recorded from multiple channels, and designing a spatial filter to perform a delay and sum operation for generating an enhanced single channel signal [11]. This can be used in improving the downstream speech systems, including ASR. These methods can also be modified for maximizing the likelihood [12].

With the advent of neural network based acoustic models multi-channel acoustic models have also been explored. Recently Świetojanski et al [13] proposed the use of features from each channel of the multi-channel speech directly as input to a convolutional neural network based acoustic model. Here the neural network is seen as a replacement for conventional beamformer. Joint training of a more explicit beamformer with the neural network acoustic model has been proposed by Xiao et al., [14]. Training of neural networks, which operate on the raw signals and are optimized for the discriminative cost function of the acoustic model, has also been recently explored. These approaches are termed Neural Beamforming approaches as the neural network acoustic model subsumes the functionality of the beamformer [15, 16, 17].

In this paper, we develop a neural network architecture consisting of 3-D convolutional models as a front-end for processing multi-channel speech. 3-D CNN architectures have shown promising results in video signal based human action segmentation [18] and more recently in the bio-medical imaging applications for lesion segmentation [19].

3. PROPOSED 3-D CNN ARCHITECTURE
The block schematic of the proposed architecture is shown in Fig. 1. The multi-channel recordings are converted into spectrogram representation containing 40 bands of log-mel filtered filter bank energies...
sampled at every 25 ms of the audio file with a shift of 10 ms. The multi-channel audio segments are stacked in a 3-D fashion and fed as input to the neural network model. The neural network architecture consists of convolutional layers which have 3-D kernel. The CNN layers perform the following convolutional operation,

\[ Y(i, j, k) = \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} \sum_{z=1}^{N_z} X(i + x, j + y, k + z) K(x, y, z) \]  

(1)

where \( K \) is the 3-D kernel, \( X \) is the input multi-channel spectrogram, \( Y \) is the output of the feature map and \( (N_x, N_y, N_z) \) represents the kernel size. The operation is performed without any padding on the input spectrogram so that output dimensions are reduced in each convolutional operation. In our case, we perform two layers of 3-D convolutions.

The feature maps generated by the CNN layers are flattened along the frequency and channel dimensions, every time step, and are fed to successive layers of time delay neural networks [7]. While standard feed forward networks process the entire input contexts, TDNN architecture captures narrow temporal context in the initial layers and then approximates the wider temporal context in the final layers. This has shown to improve the ASR performance for far-field reverberant speech [20].

The higher TDNN layers are interleaved with the LSTM layers as a combination of these layers has been shown to be optimal for low latency temporal context modeling [2]. The final output of the TDNN-LSTM stack is mapped to the senone (context dependent HMM state) targets using a fully connected layer.

### 4. EXPERIMENTAL SETUP

#### 4.1. AMI LVCSR task

The AMI meeting corpus [21] contains conversational speech with the training and test configuration chosen similar to [22]. The training data corpus consist of about 100 hours of meetings recorded in instrumented meeting rooms at three sites in the UK, the Netherlands, and Switzerland. Each meeting usually has four participants and the language is English, albeit with a large proportion of non-native speakers. The recording involves multiple parallel microphones, including individual headset microphones (IHM), lapel microphones, and one or more microphone arrays. Every recording used a primary 8-microphone uniform circular array (10 cm radius), as well as a secondary array whose geometry varied between sites. In this work we use the first three microphone recordings of the primary array for our single distant microphone (SDM) experiments.

A portion of the speech signal from the AMI corpus and the corresponding spectrogram is illustrated in Fig. 2. As seen here, the spectrogram contains reverberation artifacts and there is substantial amount of overlap speech (about 12% of the recording duration contains multi-talker speech). Also, the dataset consists of substantial amounts short segments with large number of speaker turns.

We use the lattice-free maximum mutual information (LF-MMI) cost function to train in a purely sequence discriminative training approach [8]. The cost function is similar to connectionist temporal classification (CTC) [23], however the lattice free MMI involves a global normalization.

We use the Kaldi toolkit [24] to conduct our experiments. This recipe follows the corpus release for the training and evaluation splits. For training purposes we consider all segments (including those with overlapped speech), and the WERs of the speech recognition outputs are scored by the tool following the NIST RT recommendations for scoring simultaneous speech\(^1\).

A HMM-GMM system is used to generate numerator lattices for LF-MMI training and also for cross-entropy regularization. The training and decoding closely follows the one described in [25]. We use the speed-perturbation technique [26] for data augmentation with 3-way speed perturbation. The log-mel features are also appended with iVectors to perform instantaneous adaptation of the neural network [27]. The WER results are reported after 4-gram LM re-scoring of lattices generated using a trigram LM. All the neural network models in AMI corpus use 40 dimensional log-mel filter bank en-
Table 1. Word error rate (%) on REVERB Challenge corpus for simulated (S) and naturally reverberant conditions (R) on development (dt) and evaluation (et) datasets. Here BF corresponds to beamforming.

<table>
<thead>
<tr>
<th>Model</th>
<th>S-dt</th>
<th>S-et</th>
<th>R-dt</th>
<th>R-et</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN-Single-Chn.</td>
<td>12.7</td>
<td>13.6</td>
<td>31.8</td>
<td>37.5</td>
</tr>
<tr>
<td>CNN2D-Single-Chn</td>
<td>11.3</td>
<td>11.4</td>
<td>26.8</td>
<td>29.6</td>
</tr>
<tr>
<td>CNN2D-Multi-BF</td>
<td>9.7</td>
<td>10.0</td>
<td>24.8</td>
<td>26.4</td>
</tr>
<tr>
<td>CNN2D-Multi-BF + Dropout</td>
<td>10.7</td>
<td>11.5</td>
<td>26.7</td>
<td>27.5</td>
</tr>
<tr>
<td>CNN3D-Multi</td>
<td>9.8</td>
<td>10.3</td>
<td>26.7</td>
<td>28.4</td>
</tr>
<tr>
<td>CNN3D-Multi + Dropout</td>
<td>9.1</td>
<td>9.8</td>
<td>24.6</td>
<td>25.8</td>
</tr>
</tbody>
</table>

The multi-channel CNN architecture (CNN3D) improves significantly over the single channel set up. The best performance obtained for the multi-channel CNN3D experiments are reported here. We have used two layers of convolutional 3D filters of kernel size $(3 \times 3 \times 2)$ with 256 filters followed by max-pooling and another 2 layers of 128 filters along with max-pooling. The convolutional layers are followed with 2 dense layers of 1024 dimensions. While training the models, we have observed an overfitting trend where the training and validation accuracies for frame level senone targets are drastically different. In order to circumvent this issue, we have used a regularization approach using dropout [31]. The dropout scheme provides significant improvements to the CNN3D architecture and the CNN3D outperforms the best beamforming system (average relative improvements of 2.2% over the CNN2D-Beamform baseline).

The results for baseline experiments on the AMI corpus using the SDM recordings (single channel) are reported in Table 3 where the results are reported separately for development and evaluation meetings. The results of the baseline AMI system using HMM-GMM is reported in the first row. The HMM-GMM system is trained with linear discriminant analysis (LDA) - maximum likelihood linear transform (MLLT) approach [24] followed by a speaker adaptive training. A similar ASR system was built on the IHM recordings and alignments obtained from IHM setup (using force alignment) are used as labels for neural network models on the SDM data. The next two results reported in Table 3 are the TDNN based ASR system with four hidden layers trained with cross-entropy cost function and the sequence cost function respectively. As seen here, the performance is significantly improved using a TDNN acoustic model over the HMM-GMM system. The sequence cost function further improves the WER. All further experiments use the sequence training cost function.

A similar pipeline with CNN 2D (5 layers) followed by 2 layers of TDNN is reported next. All the CNN layers have 64 filters and the TDNN layers have 512 dimensions. The use of CNN front-end results in decrease in performance compared to the TDNN architecture. The last result in Table 3 reports the performance of the CNN2D-TDNN-LSTM setup. In this setup, two CNN layers with 256 and 128 filters respectively are used for generating the front-end features maps. This is followed by 2 layers of TDNN architecture with 1024 dimensions and 3 layers of LSTM architecture with 1024 LSTM cells in each layer. As seen in this table, the addition of the LSTM layers provides significant improvements to the CNN2D-TDNN model as well as the baseline TDNN model (average relative improvements of about 11% over the TDNN model and about 15% over the CNN2D-TDNN model). The single channel SDM results from two recent efforts using attention LSTM [30] as well as bidirectional LSTM with TDNN front-end [2] are also added to this table for reference.

The multi-channel experiments using the CNN3D-TDNN-LSTM model are reported in Table 2. Here, we use the first three recordings of the array microphone as input representation to the CNN3D model. Various choices of input filter sizes, weight sharing and pooling are experimented for the 3-D CNN architecture front-end. The rest of model architecture containing 2 layers of 1024 dimensions of TDNN layers and 3 LSTM layers each with 1024 cells is used as before. As seen in Table 2, increasing the number of input filters (each filter has a fixed kernel size of $(3 \times 3 \times 1)$)
Table 2. Word error rate (%) on AMI corpus for multi-channel SDM experiments using CNN3D-TDNN-LSTM architecture with various choices of model parameters.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev.</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer1-(64 Filt.) Layer2-(32 Filt.)</td>
<td>34.8</td>
<td>37.4</td>
</tr>
<tr>
<td>Layer1-(96 Filt.) Layer2-(32 Filt.)</td>
<td>34.5</td>
<td>37.2</td>
</tr>
<tr>
<td>Layer1-(128 Filt.) Layer2-(64 Filt.)</td>
<td>34.4</td>
<td>37.4</td>
</tr>
<tr>
<td>Layer1-(256 Filt.) Layer2-(128 Filt.)</td>
<td>34.9</td>
<td>37.9</td>
</tr>
<tr>
<td>Layer1-(256 Filt.) Layer2-(128 Filt.) + Reg.</td>
<td>32.7</td>
<td>35.7</td>
</tr>
<tr>
<td>Layer1-(256 Filt.) Layer2-(128 Filt.) + Reg. and Sharing</td>
<td><strong>32.6</strong></td>
<td><strong>35.4</strong></td>
</tr>
<tr>
<td>Layer1-(384 Filt.) Layer2-(128 Filt.) + Reg. and Sharing</td>
<td>32.6</td>
<td>35.7</td>
</tr>
<tr>
<td>Layer1-(384 Filt.) Layer2-(192 Filt.) + Reg. and Sharing</td>
<td>32.7</td>
<td>35.7</td>
</tr>
</tbody>
</table>

Table 3. Word error rate (%) on AMI corpus for single channel SDM experiments.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev.</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM-GMM (LDA-MLLT-SAT)</td>
<td>59.5</td>
<td>64.0</td>
</tr>
<tr>
<td>TDNN (CE)</td>
<td>41.7</td>
<td>46.7</td>
</tr>
<tr>
<td>TDNN (Seq.)</td>
<td>40.2</td>
<td>44.1</td>
</tr>
<tr>
<td>CNN2D-TDNN (Seq.)</td>
<td>41.8</td>
<td>46.7</td>
</tr>
<tr>
<td>CNN2D-TDNN-LSTM (Seq.)</td>
<td><strong>36.0</strong></td>
<td><strong>39.0</strong></td>
</tr>
<tr>
<td>Attention-LSTM [30]</td>
<td>41.3</td>
<td>45.8</td>
</tr>
<tr>
<td>TDNN-LSTM [2]</td>
<td>37.4</td>
<td>40.4</td>
</tr>
</tbody>
</table>

Table 4. Word error rate (%) in AMI corpus for multi-channel SDM experiments using proposed approach and baseline beamformed [11] approach

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev.</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN2D-TDNN-LSTM (single)</td>
<td>36.0</td>
<td>39.0</td>
</tr>
<tr>
<td>CNN2D-TDNN-LSTM (multi beamformed)</td>
<td>33.9</td>
<td>36.2</td>
</tr>
<tr>
<td>CNN3D-TDNN-LSTM (multi)</td>
<td><strong>32.6</strong></td>
<td><strong>35.4</strong></td>
</tr>
</tbody>
</table>

The previous efforts on CNN with multi-channel framework [13] use separate kernels on each channel which get merged in the network. The proposed approach uses 3D framework to capture the time-frequency-channel correlations. In comparison with the raw-waveform based beamforming efforts [15], the proposed approach is more robust to speaker and source location changes. While raw waveforms based approaches have the advantage of using the signal phase information for multi-channel combination, the regions of speaker changes causes degradation which could potentially impede the performance. Also, the model proposed in [15] attempts to summarize the spatial dimension in the first layer of the network where as the proposed approach preserves the space dimension deep in CNN architecture (until the feed-forward/TDNN layers). Hence, we hypothesize that proposed model may be more suitable for ASR applications on natural multi-speaker conversations.

7. SUMMARY AND FUTURE WORK

In this paper, we have proposed a three dimensional neural network consisting of convolutional layers followed by LSTM layers. The 3D CNN architecture receives input from time-frequency-channel dimensions of the input multi-channel speech. Various speech recognition experiments were performed in the REVERB challenge dataset as well as the AMI speech recognition database. The main finding is that while the beamforming approach of speech enhancement is effective in simple settings involving single speaker recordings (as seen in REVERB challenge corpus), the proposed approach of 3D CNN architecture improves noticeably over the beamforming methods on REVERB challenge corpus and the multi-party conversational settings. The promising results motivate us to further pursue novel modeling methods for multi-channel speech recognition involving spatial recurrence in LSTM models. The current way of combining the channels uses a time resolution of 10 ms in the spectrogram representations which can be improved by having a higher temporal granularity (frames taken below 10 ms). The higher temporal sampling rate could allow the model to have more spatial resolution and improve the 3D structure of the data. In addition, the AMI database contains 8 parallel microphones which could potentially allow greater flexibility in the 3D modeling.
8. REFERENCES


[29] François Chollet et al., “Keras: Deep learning library for theano and tensorflow,” URL: https://keras.io/k

