Multi-Scale Octave Convolutions for Robust Speech Recognition

MULTI-SCALE OCTAVE CONVOLUTIONS FOR ROBUST SPEECH RECOGNITION

Joanna Rownicka, Peter Bell, Steve Renals

Centre for Speech Technology Research, University of Edinburgh, UK

ABSTRACT

We propose a multi-scale octave convolution layer to learn robust speech representations efficiently. Octave convolutions were introduced by Chen et al [1] in the computer vision field to reduce the spatial redundancy of the feature maps by decomposing the output of a convolutional layer into feature maps at two different spatial resolutions, one octave apart. This approach improved the efficiency as well as the accuracy of the CNN models. The accuracy gain was attributed to the enlargement of the receptive field in the original input space. We argue that octave convolutions likewise improve the robustness of learned representations due to the use of average pooling in the lower resolution group, acting as a low-pass filter. We test this hypothesis by evaluating on two noisy speech corpora – Aurora-4 and AMI. We extend the octave convolution concept to multiple resolution groups and multiple octaves. To evaluate the robustness of the inferred representations, we report the similarity between clean and noisy encodings using an affine projection loss as a proxy robustness measure. The results show that proposed method reduces the WER by up to 6.6% relative for Aurora-4 and 3.6% for AMI, while improving the computational efficiency of the CNN acoustic models.

1. INTRODUCTION

Deep convolutional neural networks (CNNs) with 2D convolutions and small kernels [2], have achieved state-of-the-art results for several speech recognition tasks [3, 4, 5, 6, 7]. The accuracy of those models grows with their complexity, leading to redundant latent representations. Several approaches have been proposed in the literature to reduce this redundancy [8, 9, 10, 11, 12], and therefore to improve their efficiency.

Octave convolutional layers [1] address the problem of spatial redundancy in feature maps by learning feature representations at high and low resolutions. The low resolution processing path increases the size of the receptive field in the original input space, which is a plausible explanation of the improved performance for image classification. We extend the octave convolution concept to multi-scale octave convolutional layers, which include lower resolution feature maps with a higher compression rate (reduction by more than one octave), and the use of more than two feature map tensor groups in order to learn representations at multiple scales.

Multi-scale processing have been previously proposed for a variety of speech recognition tasks [13, 14, 15, 16, 17]. In deep CNN acoustic models, some of the feature maps may need to represent information which varies at a lower rate, such as the characteristics of the speaker or background noise, compared to the information necessary for phonetic discrimination. Spatial average pooling in a low resolution group of feature maps can be interpreted as a form of low-pass filtering, providing smoothed representations of the observed data, potentially leading to improved performance.

We investigate the use of multi-scale octave convolutional layers for robust speech recognition, and attempt to shed more light on the explainability of the models by evaluating the robustness of the learned representations using an affine transformation loss to measure the similarity between clean and noisy encodings.

2. MULTI-SCALE OCTAVE CONVOLUTIONS

An octave convolutional layer [1] factorizes the output feature maps of a convolutional layer into two groups. The resolution of the low-frequency feature maps is reduced by an octave – height and width dimensions are divided by 2. In this work, we explore spatial reduction by up to 3 octaves – dividing by 2^t, where $t = 1, 2, 3$ – and for up to 4 groups. We refer to such a layer as a multi-octave convolutional (MultiOctConv) layer, and an example with three groups and reductions of one and two octaves is depicted in Fig. 1.

In a vanilla CNN the convolutions have the same spatial resolution throughout the network. A multi-octave convolutional (MultiOctConv) layer has feature maps reduced by multiple octaves. Let the input feature tensor be $X \in \mathbb{R}^{c_{in} \times h \times w}$, where c_{in} denotes the number of input channels and h and w correspond to the spatial dimensions. In a MultiOctConv layer working at 3 resolutions, X is factorized along the channel dimension into $X = \{X^1, X^2, X^3\}$. The first group tensor, X^1, is a representation at the same spatial scale as X. The spatial dimensions of the second and third group tensors, X^2 and X^3, are reduced by one and two octaves respectively. The dimensions of the input tensors X^1, X^2 and X^3 are described in Fig. 1. The fraction of the channels for

This work was supported by a PhD studentship from the DataLab Innovation Centre, Ericsson Media Services, and Quorate Technology.

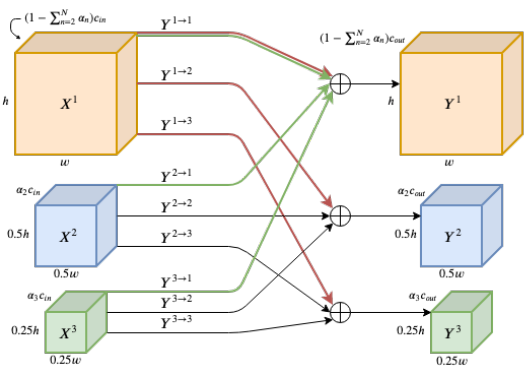

Fig. 1. Multi-octave convolution scheme for 3 resolution groups. Red and green arrows show the connections for the initial and final MultiOctConv layers, respectively. N corresponds to the total number of groups in the MultiOctConv layer ($N = 3$ in this example). α_n is a fraction of channels corresponding to group n. h and w are spatial dimensions.

each group is denoted with $\alpha_n \in [0, 1]$, where $\sum_{n=1}^{N} \alpha_n = 1$ for N resolution groups in the MultiOctConv layer. For simplicity, we use the same α_n for input and output representations within the same scale group. Similarly, the output tensors are also factorized into $Y = \{Y^1, Y^2, Y^3\}$. Their dimensions are also described in Fig. 1. To compute Y^1, Y^2 and Y^3 we operate directly on the factorized input tensors X^1, X^2 and X^3. Inter-frequency information update is implemented as a sum of feature maps from different resolution groups (we also experiment with disabling the inter-frequency exchange paths, keeping only the intra-frequency connections). To be able to sum those representations for a desired output scale, the spatial dimensions of the tensors must be the same. For this reason, two operations are employed: spatial average pooling $\texttt{pool}\,(X, p)$ and bilinear interpolation $\texttt{upsample}\,(X, u)$, where p is the kernel size and stride for the the 2D pooling layer and u is the upsampling factor. The output MultiOctConv representations are therefore computed as

$$
\begin{aligned}
Y_1 = f(X^1; W^{1\to1}) + \texttt{upsample}(f(X^2; W^{2\to1}), 2) + \\
+ \texttt{upsample}(f(X^3; W^{3\to1}), 4) \\
Y_2 = f(X^2; W^{2\to2}) + \texttt{upsample}(f(X^3; W^{3\to2}), 2) + \\
+ f(\texttt{pool}(X^1, 2); W^{1\to2}) \\
Y_3 = f(X^3; W^{3\to3}) + f(\texttt{pool}(X^1, 4); W^{1\to3}) + \\
+ f(\texttt{pool}(X^2, 2); W^{2\to3})
\end{aligned}
$$

where $f(.)$ is the convolution function and $W^{n_{in}\to n_{out}} \in \mathbb{R}^{c_{in}\times k\times k\times c_{out}}$ is the convolution filter for a $k \times k$ kernel. We call the information update "intra-frequency" when $n_{in} = n_{out}$, and "inter-frequency" when $n_{in} \neq n_{out}$. It is important to note that the convolution operates on the tensors compressed with average pooling and on the tensors before upsampling, making the design more efficient. The number of parameters in the MultiOctConv layer is the same as in a vanilla convolutional layer.

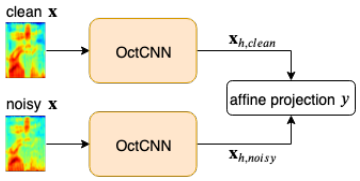

Fig. 2. Proposed method to measure the robustness of learned representations.

Robustness of learned representations

To evaluate the robustness of learned representations, we compare clean and noisy Aurora-4 samples. The similarity between them is measured using the Mean Squared Error (MSE) loss of an affine projection y of N clean to noisy samples (Eq. 1), to take into account permutations of hidden representations and to ensure invariance of the metric to affine transformations of the encodings. The number of units in layer y and the dimensionality D of $\mathbf{x}_h$ is 1024.

$$
\theta^* = \arg\min_{\theta} \frac{1}{ND} \sum_{i=1}^{N} \left\| y(\mathbf{x}^{(i)}_{h,clean}, \theta) - \mathbf{x}^{(i)}_{h,noisy} \right\|^2 \quad (1)
$$

We use the Aurora-4 test sets and compare clean encodings $\mathbf{x}_{h,clean}$ with noisy encodings $\mathbf{x}_{h,noise}$, obtained as the activations from the last convolutional layer with a forward pass through a trained model. Both hidden representations were obtained for CNN and octave CNN (OctCNN) models. Also, for intra-model comparison, we evaluate the loss with the encodings from high and low-resolution groups (paths $Y^{1\to1}$ and $Y^{2\to1}$). This analysis aims to evaluate if the low-resolution groups for noisy samples are indeed more similar to the clean ones than the high-resolution encodings, suggesting more robust representations. We optimize the parameters of y with back-propagation using a fixed number of 3 epochs and we report the loss for Aurora-4 test sets.

3. EXPERIMENTAL SETUP

Aurora-4 [18]: We evaluate our models on the simulated multi-condition Aurora-4 dataset, consisting of ∼15h of audio for training and ∼9h for testing. The test set is divided into 4 subsets: A, B, C, and D. Subset A contains clean-condition recordings, subset B has 6 noise types added to the recordings (car, babble, restaurant, street, airport, train), subset C is recorded with a mismatched microphone, and subset D is recorded with a mismatched microphone and with noise

added. In our experiments, we use multi-condition GMM-HMM forced alignments as targets for CNN training. The number of CD states for Aurora-4 is 3422.

AMI [19]: AMI contains ∼100h of meeting recordings, captured by an independent headset microphone (IHM), single distant microphone (SDM), and multiple distant microphones (MDM), where the mics are combined using the BeamformIt [20] toolkit. We train our models using the MDM data and evaluate the models for all 3 types of recordings to analyze the effect of mismatched training/testing conditions. We use the suggested train/dev/eval data split [21], and we evaluate the models on both dev and eval sets. The number of CD states for AMI is 3984.

Features: In our experiments, we use 40-dimension mel-scaled filterbank (FBANK) features with {-5,..,5} context for splicing, resulting in a 40×11 input feature map.

Models: Our baseline CNN model [22] consists of 15 convolutional and one fully-connected layer. We use 3×3 kernels throughout the network. We start with 64 output channels in the first layer and double them after 3 and 9 layers. We use batch normalization in every convolutional layer, and ReLU afterwards (unless a reverse order is noted). The initial learning rate is 0.001. We use early stopping for training.

4. RESULTS

We present our results in terms of accuracy and robustness on Aurora-4 and AMI, as well as in terms of the computational cost, which is calculated as the number of multiply-accumulate operations (MACCs) performed for a single input feature map. The cost reduction when using octave convolutions stems from reduced dimensions c_{in}, c_{out}, h, and w compared to a vanilla convolutional layer.

Aurora-4: Results for Aurora-4 are presented in Table 1. We replace vanilla convolutional layers of our baseline model (CNN) with OctConv and MultiOctConv layers. We first evaluate which layers can be replaced and find that all but the first layer, operating directly on the input representation, should be replaced for the best performance. This approach (L2-L15) is also the least costly. Reducing the ratio of low-resolution representations to 0.125 improves the WER for the mismatched microphone scenario C, but not for all test conditions. Applying batch normalization after ReLU is beneficial for test set C and D. For OctCNN models, the WER for test set D dropped by $\sim 0.4\%$ with a compression by one octave, and by another $\sim 0.4\%$ with a reversed batch normalization and ReLU order.

The biggest differences between the MultiOctCNN models can be observed for test set D. The models with the lowest WERs are the ones with a spatial reduction by 2 or 3 octaves, and with 2 or 3 groups. This indicates that multi-scale octave convolutions seem to be an effective, as well as an efficient design for processing speech with background noise and channel mismatch. For MultiOctCNNs, batch normalization after ReLU also gives a performance boost for test set D, with a drop to 13.57%.

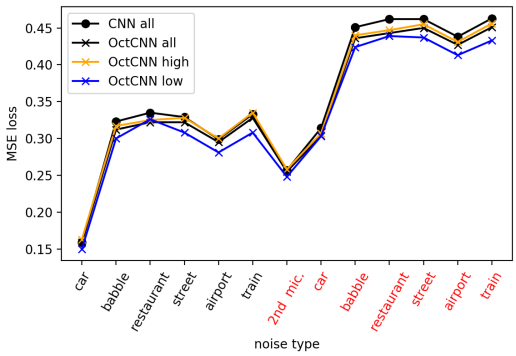

Fig. 3. MSE affine transformation loss to measure the similarity of "clean" and "noisy" encodings ($\mathbf{x}_{h,clean}$ and $\mathbf{x}_{h,noisy}$). "all" corresponds to the output of the last convolutional layer (Conv15), "high" and "low" correspond to its $Y^{1\to1}$ and $Y^{2\to1}$ branch, respectively.

To further evaluate the robustness of the latent representations we measured the MSE between the (projected) representations (Fig. 3). The loss for the activations at the output of Conv15 ("all") is similar for CNN and OctCNN models for test sets B and C, but lower for test set D for OctCNN, indicating that the learned representations are more robust, contributing to lower WERs. As expected, within-model comparison of the loss show that the representations at low resolution are more similar to the clean encodings from test set A than the ones at high resolution. We believe that this effect improves the robustness of latent representations and results in a decreased WER.

AMI: Results for AMI are presented in Table 2. In contrast to the Aurora-4 findings, better performance was achieved with an all OctCNN model (L1-L15). This is an interesting finding, and we believe that the multi-scale processing of the input feature space is beneficial for AMI MDM because of the reverberation in the data. The reverbarated input time×freq representation can be viewed as a spatially redundant one, therefore the OctConv layer applied to the input representation is effective. Unfortunately, the only MultiOctConv model superior to the baseline CNN is the one with 3 groups with a spatial reduction by 1 and 2 octaves. This result indicates that the spatial redundancy for this architecture for AMI MDM is not degrading the performance. However, in terms of the computational cost, we can reduce the #MACCs by a factor of 1.8 with only a small WER increase for a model with 4 resolution groups.

5. CONCLUSIONS

We have presented multi-scale octave CNN models for robust and efficient speech recognition. We build on Chen et al [1],

Model	OctConv	α (low $\rightarrow$ high)	2^1	2^2	2^3	#MACCs (M)	A	B	C	D	Avg.
CNN	-	[0, 1]	-	-	-	174.7	2.19	4.68	4.22	14.53	8.69
OctCNN	L1-L3	[0.2, 0.8]	✓	-	-	167.6	2.19	4.74	4.32	14.83	8.85
OctCNN	L1-L15	[0.2, 0.8]	✓	-	-	126.9	2.22	4.61	4.30	14.40	8.61
OctCNN	L2-L15	[0.2, 0.8]	✓	-	-	126.2	2.02	4.65	4.35	14.16	8.52
OctCNN †	L2-L15	[0.2, 0.8]	✓	-	-	126.2	2.22	4.82	4.22	13.72	8.41
OctCNN	L2-L15	[0.125, 0.875]	✓	-	-	143.1	2.11	4.56	4.07	14.55	8.63
MultiOctCNN	L2-L15	[0.1, 0.1, 0.8]	✓	✓	-	120.6	**1.98**	4.51	4.11	14.00	8.37
MultiOctCNN	L2-L15	[0.1, 0.1, 0.8]	✓	-	✓	119.5	2.02	4.59	**3.92**	13.82	**8.31**
MultiOctCNN †	L2-L15	[0.1, 0.1, 0.8]	✓	-	✓	119.5	2.28	4.81	4.04	13.76	8.41
MultiOctCNN	L2-L15	[0.1, 0.1, 0.1, 0.7]	✓	✓	✓	**94.3**	2.30	4.88	4.18	14.06	8.58
MultiOctCNN ‡	L2-L15	[0.1, 0.1, 0.1, 0.7]	✓	✓	✓	**94.3**	2.24	4.89	4.07	14.22	8.64
MultiOctCNN	L2-L15	[0.2, 0.8]	-	✓	-	115.7	2.15	4.77	4.07	13.77	8.39
MultiOctCNN	L2-L15	[0.125, 0.875]	-	✓	-	136.3	2.09	4.56	4.22	14.32	8.54
MultiOctCNN	L2-L15	[0.2, 0.8]	-	-	✓	113.5	2.09	4.54	3.94	14.03	8.39
MultiOctCNN	L2-L15	[0.125, 0.875]	-	-	✓	134.9	2.02	**4.50**	4.17	13.87	8.32
MultiOctCNN †	L2-L15	[0.125, 0.875]	-	-	✓	134.9	2.32	4.73	4.24	**13.57**	**8.31**

Table 1. WERs [%] for Aurora-4 test sets A, B, C and D for octave and multi-octave CNNs. Notation explanation for both tables: "OctConv" column indicates where a Conv layer was replaced with an OctConv or MultiOctConv. 2^1, 2^2 and 2^3 correspond to the factor of spatial dimension reduction. Models with batch normalization after ReLU are denoted by †. Models without the inter-frequency exchange paths are denoted by ‡.

							IHM		SDM		MDM	
Model	OctConv	α (low $\rightarrow$ high)	2^1	2^2	2^3	#MACCs (M)	dev	eval	dev	eval	dev	eval
CNN	-	[0, 1]	-	-	-	175.2	33.4	38.3	49.1	54.0	43.9	48.0
OctCNN	L1-L3	[0.2, 0.8]	✓	-	-	168.2	33.0	38.1	49.0	54.1	43.8	47.9
OctCNN	L2-L15	[0.2, 0.8]	✓	-	-	126.7	33.0	37.7	48.9	54.0	43.7	47.7
OctCNN	L1-L15	[0.2, 0.8]	✓	-	-	127.5	**32.2**	**37.2**	48.3	53.5	43.1	47.3
OctCNN	L1-L15	[0.125, 0.875]	✓	-	-	144.1	32.5	37.4	**48.2**	**53.3**	**42.9**	**47.2**
OctCNN†	L1-L15	[0.125, 0.875]	✓	-	-	144.1	33.2	38.3	48.8	54.3	43.7	48.0
MultiOctCNN	L1-L15	[0.1, 0.1, 0.8]	✓	✓	-	121.6	32.8	38.1	48.9	53.9	43.7	47.9
MultiOctCNN	L1-L15	[0.1, 0.1, 0.8]	✓	-	✓	120.4	33.3	38.5	49.2	54.5	44.1	48.4
MultiOctCNN	L1-L15	[0.1, 0.1, 0.1, 0.7]	✓	✓	✓	**95.2**	33.7	38.7	49.5	54.6	44.1	48.4
MultiOctCNN ‡	L1-L15	[0.1, 0.1, 0.1, 0.7]	✓	✓	✓	**95.2**	33.2	38.3	49.3	54.5	44.0	48.5
MultiOctCNN	L1-L15	[0.125, 0.875]	-	✓	-	136.9	33.6	38.6	49.7	54.6	44.3	48.4
MultiOctCNN	L1-L15	[0.125, 0.875]	-	-	✓	135.4	32.9	38.1	49.1	54.3	43.8	48.0

Table 2. WERs [%] for models trained on AMI MDM and evaluated on IHM, SDM and MDM conditions.

applying the method to robust ASR and extending it to multiple resolution groups with a spatial reduction of more than one octave. Our experiments confirm that multi-scale processing of the hidden representations is not only more computationally efficient, but also improves the recognition. Similarity measures between clean and noisy encodings indicate that multi-scale processing in a deep CNN acoustic model improves the robustness of learned representations, especially in the additive noise and mismatched microphone scenario. The gain of the octave convolutions was also observed for AMI MDM data with significant reverberation, when applied to the input feature space. However, the model performance for AMI MDM was not improved with multi-octave convolutions. More careful tuning of the α hyperparameter could improve the results, as it controls the ratio of multi-scale feature maps in the model, enabling both learning of fine-grained representations preserving the details necessary for phonetic discrimination, as well as smoothed, more invariant representations improving the robustness of the model. It would also be possible to set α layer-by-layer to enable the fractions of channels at different resolutions to vary according to the depth of the representation.

We proposed a single projection layer MSE loss to measure the robustness as an affine relationship of clean and noisy hidden representations, and to improve the explainability of our models. We confirmed that the noisy lower resolution representations are more similar to the clean counterparts than high resolution representations, and thus are more robust. However, we did not investigate the reason for the increased similarity, leaving future work to ascertain if the lower resolution group corresponds to better speaker or noise characteristics, or more invariant phonetic representations.

6. REFERENCES

[1] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng, "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3435–3444.

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[3] Tom Sercu, Christian Puhrsch, Brian Kingsbury, and Yann LeCun, "Very deep multilingual convolutional neural networks for LVCSR," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[4] Tom Sercu and Vaibhava Goel, "Advances in very deep convolutional neural networks for LVCSR," in *Proc. Interspeech*, 2016.

[5] Dong Yu, Wayne Xiong, Jasha Droppo, Andreas Stolcke, Guoli Ye, Jinyu Li, and Geoffrey Zweig, "Deep convolutional neural networks with layer-wise context expansion and attention," in *Proc. Interspeech*, 2016.

[6] Yanmin Qian and Philip C Woodland, "Very deep convolutional neural networks for robust speech recognition," *IEEE Spoken Language Technology Workshop (SLT)*, 2016.

[7] T. Tan, Y. Qian, H. Hu, Y. Zhou, W. Ding, and K. Yu, "Adaptive very deep convolutional residual network for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1393–1405, Aug 2018.

[8] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf, "Pruning filters for efficient ConvNets," in *ICLR*, 2017.

[9] J. Luo, H. Zhang, H. Zhou, C. Xie, J. Wu, and W. Lin, "ThiNet: Pruning CNN Filters for a Thinner Net," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2525–2538, 2019.

[10] Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q. Weinberger, "CondenseNet: An Efficient DenseNet Using Learned Group Convolutions," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2752–2761, 2017.

[11] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun, "ShuffleNet v2: Practical guidelines for efficient CNN architecture design," in *ECCV*, 2018, pp. 116–131.

[12] Mingxing Tan and Quoc V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *ICML*, 2019.

[13] Mohamed Hesham Farouk, *Application of Wavelets in Speech Processing*, Springer Publishing Company, Incorporated, 2013.

[14] M. Gupta and A. Gilbert, "Robust speech recognition using wavelet coefficient features," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2001.

[15] L. Tóth, "Multi-resolution spectral input for convolutional neural network-based speech recognition," in *International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, 2017.

[16] Chun-Fu Chen, Quanfu Fan, Neil Mallinar, Tom Sercu, and Rogério Schmidt Feris, "Big-Little Net: An Efficient Multi-Scale Feature Representation for Visual and Speech Recognition," in *ICLR*, 2019.

[17] P. von Platen, Chao Zhang, and P.C. Woodland, "Multi-Span Acoustic Modelling Using Raw Waveform Signals," in *Proc. Interspeech*, 2019.

[18] Naveen Parihar and Joseph Picone, "Aurora working group: DSR front end LVCSR evaluation AU/384/02," *Tech. Rep.*, 2002, Institute for Signal and Information Processing, Mississippi State Univ.

[19] S. Renals, T. Hain, and H. Bourlard, "Recognition and understanding of meetings: The AMI and AMIDA projects," in *IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, 2007.

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

[21] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013.

[22] J. Rownicka, S. Renals, and P. Bell, "Simplifying very deep convolutional neural network architectures for robust speech recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017.