# SMALL FOOTPRINT MULTI-CHANNEL KEYWORD SPOTTING

*Jilong Wu, Yiteng (Arden) Huang, Hyun-Jin Park, Niranjan Subrahmanya, Patrick Violette*

Google Inc., USA

{jlwu, ardenhuang, hjpark, sniranjan, pdv}@google.com

## ABSTRACT

Noise robustness remains a challenging problem in on-device keyword spotting. Using multiple-microphone algorithms like beamforming improves accuracy, but it inevitably pushes up computational complexity and tends to require more memory. In this paper, we propose a new neural-network based architecture which takes multiple microphone signals as inputs. It can achieve better accuracy and incurs just a minimum increase in model size. Compared with a single-channel baseline which runs in parallel on each channel, the proposed architecture reduces the false reject (FR) rate by 36.3% and 46.4% relative on dual-microphone clean and noisy test sets, respectively, at a fixed false accept rate.

***Index Terms—*** deep neural networks, keyword spotting, multi-microphone noise reduction, microphone array processing for machine learning, embedded speech recognition

## 1. INTRODUCTION

The usage of voice assistants is increasingly popular. Keywords, such as “Hey Google” are commonly used as the command to initiate the conversation with voice assistants. Keyword-spotting with low latency becomes the core technical challenge to achieve the task.

There has been much work trying to solve this problem, like using traditional Hidden Markov Model (HMM) [1] to make use of acoustic features from a deep neural networks (DNNs) [2, 3]. Furthermore, several approaches have been proposed to evaluate the outputs of the acoustic model and produce one score for keyword spotting system [4, 5, 6]. Besides the desire for high detection accuracy, small model size, low memory and computational consumption are all challenging requirements for this on-device application. For example, inspired by the causal convolution and gated activation from Wavenet [7], the keyword spotting system presented in this recent paper [8] has a small number of model parameters.

Other work, like temporal convolution [9], also shows model size reduction. Singular value decomposition (SVD) was also explored to compress DNNs for keyword spotting [10]. Last year an end-to-end keyword-spotting system using Singular Value Decomposition Filter (SVDF) op [10] was proposed which showed a large reduction in model size with improved accuracy [11]. However, most of the work is for single microphone use cases.

Beamforming is a widely adopted method to utilize multi-microphone input for speech enhancement [12]. For automatic speech recognition (ASR), DNNs or convolutional networks have been widely used [13]. Therefore, there have been many studies around mimicking beamforming behaviour using neural networks [14, 15, 16, 17]. For these papers [16, 17], direct use of features from acoustic signals fed into multi-dimensional convolutional neural networks (CNNs) helps with ASR. Besides beamforming, neural-net based denoising has also been widely explored [18, 19, 20] but those explorations do not show the effect of performance with smaller model parameters. This is something we consider exploring in the future.

For small footprint keyword spotting tasks, not much work has been done to explore the possible benefits from a microphone array. These papers [21, 22] explored an effective noise-cancellation algorithm using multi-microphone. But the cleaner algorithm is not integrated in the neural-net based end-to-end system and it adds extra noticeable latency in front-end preprocessing.

In this paper, we propose a new three-dimensional (3D) SVDF layer that processes multi-channel audio inputs in a neural-network based architecture for on-device keyword spotting. This 3D-SVDF input layer has time, frequency and channel as its three dimensions and will automatically learn the correlation among different channels. Compared against a known small-footprint end-to-end single channel keyword spotting architecture [11], the proposed topology has shown significant improvements on clean and noisy environments with a dual-microphone setup.

## 2. MODEL ARCHITECTURE

### 2.1. Singular Value Decomposition Filter

This layer topology is proposed in [10]. The idea has its origins in doing singular value decomposition of a fully connected weight matrix. For a rank-1 SVDF, the weight matrix is decomposed into two vectors. As shown in Fig. 1, those two vectors can be interpreted as filters in frequency ($\alpha$) and time ($\beta$) domains. In inference, feature maps from input will first convolve with 1-D feature filters ($\alpha$). Then the output of that current state is pushed into a memory buffer. For a given memory size $M$, the buffer will store states in the past $M$ states. If SVDF is used in the input layer, the past $M$ states correspond to the past $M$ frames. Then states in the memory buffer will convolve with time-domain filters ($\beta$) to produce a final output $O$, as shown in Fig. 1.

### 2.2. 3D-SVDF Based Multi-Channel Architecture

SVDF can work well with single-channel input features [11]. To support multi-channel features, in this paper, we propose 3D-SVDF topology, as shown in Fig. 1. It extends existing two dimensions which cover time × frequency to a 3rd dimension - channel. Filterbank energies from each channel are fed into this 3D-SVDF. Fig. 1 shows how a 2-channel 3D-SVDF is created. Each channel learns its weights of its own time- and frequency-domain filters. The outputs of all channels are concatenated after the layer. Following the first 3D-SVDF, the encoder has other layers as SVDF and there are some fully-connected layers as the bottleneck layers to further cut the total number of parameters. The decoder consists of three SVDF layers

10.21437/Odyssey.2020-55

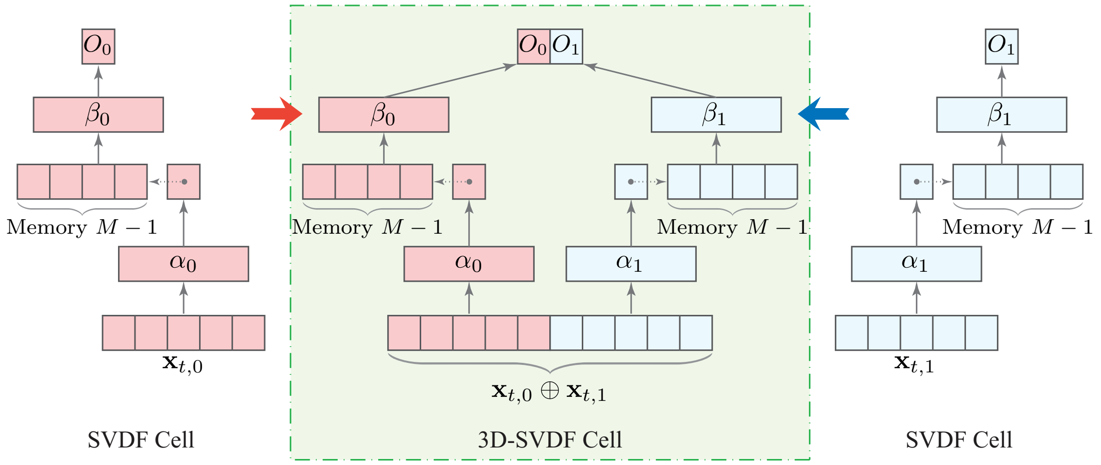


**Fig. 1**: Illustration of SVDF and 3D-SVDF cells.

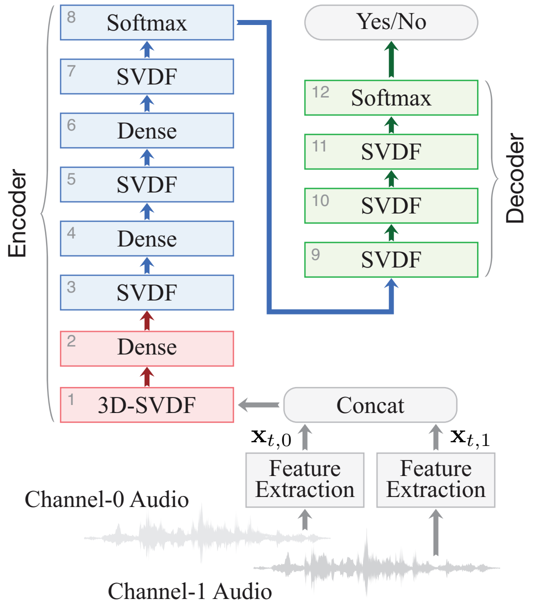


**Fig. 2**: Structure of 3D-SVDF based neural network layers for multichannel hotword detection.

with softmax as the final activation. The whole topology is shown in Fig. 2.

3D-SVDF can be considered as applying SVDF on each channel and then fusing the results. This enables the neural network to not only take advantage of the redundancy in frequency features from each channel but also exploit the temporal variation across channels to enhance noise robustness. After the signal of each channel is processed with filters in 3D-SVDF layer, it is fed into a fully connected layer which works as a weighted summing of the filtered signal. The output from that can be treated as the enhanced signal from the multichannel audio inputs. Then we can use a similar end-to-end architecture in [11] for the rest of the layers of the model.

## 3. MODEL TRAINING

For the multi-channel training setup, we use the same label generation and similar loss functions from [11]. We concatenate the log-mel features from all the channels.

### 3.1. Keyword Label Generation

Suppose that there are $C$ microphones. Preprocessing audio in the $c$th ($c = 0, 1, \cdots, C-1$) channel yields a vector of 40 log-mel filter-bank energies at the frame in time indexed by $t$

$$\mathbf{x}_{t,c} \triangleq \left[ X_{t,c,0},\ X_{t,c,1},\ \cdots,\ X_{t,c,39} \right]^T, \tag{1}$$

where $(\cdot)^T$ denotes the transpose of a vector or matrix.

The LVCSR system [23] is then used to force-align the class label of keyword events $U_t \in \{0, 1\}$ with the feature vector of frame $t$ in the first channel. For example, the keyword of “ok google” is decomposed into a sequence of phonemes in order as: “ou”, “k”, “eI”, “⟨silence⟩”, “g”, “u”, “g”, “@”, and “l”. We then assign 1 only to the last phoneme “l”, but 0 to all the other phonemes. Since the microphones are close to each other, the time delays among their signals are much smaller than the frame size and we can safely use the aligned labels of channel 0 for all the other channels. As a result, the input feature map at frame $t$ is generated as follows:

$$\mathbf{x}_t \triangleq \langle \mathbf{x}_{t,0} \oplus \mathbf{x}_{t,1} \oplus \cdots \oplus \mathbf{x}_{t,C-1}, U_t \rangle \tag{2}$$

where $\mathbf{a} \oplus \mathbf{b}$ denotes the concatenation of two vectors **a** and **b**.

### 3.2. Training Loss Function

The neural network is broken down into two parts: encoder and decoder. For loss function, we use a frame-level cross-entropy (CE). Based on the results from [11], we are using a one-stage unified loss for both encoder and decoder.

### 3.3. Training Data

The training data we use is 2.1-million single-channel anonymous audio with “Ok Google” or “Hey Google” in it. Instead of using the data directly in training, we use a multi-style room simulation with a dual-microphone setup to simulate 2-channel outputs. The room simulation includes different room dimensions and microphone spacings (3.3, 5.5, 6.6, and 7.1 cm). To improve the robustness, it also applies different types of noise sources and different levels of reverberations. We train our proposed model with these 2-channel outputs and train the single-channel baseline using channel 0 of the 2-channel outputs.

**Table 1**: Summary of Re-recorded Data for Testing

| Dataset | Number of Utterances | Length (hours) |
|---|---|---|
| Far-Field Clean | 13,344 | 61.9 |
| TV Noise | 14,205 | 68.7 |
| Negative | 55,469 | 1,175.1 |

## 4. EXPERIMENTS

To showcase the improvement of proposed architecture, we do the comparison between our architecture with a small-footprint end-to-end single-channel keyword spotting topology [11]. In this experiment, we evaluated our architecture with a dual-microphone setup. Both architectures are evaluated on 2-channel test audios with different runtime strategies.

### 4.1. Audio Preprocessing

The baseline and the proposed architecture use the same front-end prepossessing method. Input audio per channel generates 40-dimensional log-mel filter-bank energies from a 30ms window with a 10ms overlap.

### 4.2. Models Setup

The baseline (1ch model) is the architecture proposed in [11]. It is composed of SVDF layers with fully connected bottleneck layers in between. It is trained end-to-end with the input of a sequence with 1 frame of left context and 1 frame of right context. The stride is $\sigma$ = 2. The baseline topology consists of the front-end described in Section 4.1, followed by the encoder and decoder. The encoder has 4 rank-1 SVDF layers [10] of 576 nodes and memory $M = 8$. In between it has fully-connected bottleneck layers with 64 nodes. The encoder ends with a softmax function. The decoder has 3 rank-1 SVDF layers with no bottleneck layers. Each SVDF layer has 32 nodes with memory $M = 32$.

For proposed multi-channel topology (2ch model), we use a dual-microphone setup ($c = 2$). It has the front-end setup described in Section 4.1. Before being fed into the neural-net, $\mathbf{x}_{t,0}$ and $\mathbf{x}_{t,1}$ are concatenated. The proposed architecture has the 3D-SVDF input layer with 576 nodes and has memory $M = 8$. The encoder also has three SVDF layers each with 576 nodes and memory $M = 8$ and three bottleneck layers each with 64 nodes. Also, a softmax is used as the activation of the encoder. The proposed architecture has the same decoder as the baseline which is previously described above. The whole architecture is illustrated in Fig. 2.

To show the effect of number of model parameters on the model performance, we also include another model in our comparisons. This model has the same topology as the baseline but with more nodes in the SVDF layer. This baseline has the same total number of parameters as our proposed 3D-SVDF based architecture.

### 4.3. Runtime Strategies

To evaluate a single channel model on 2-channel audio, there are two strategies (Fig. 3):

1) Run keyword detection with either channel 0 or channel 1 of the audio.
2) Run keyword detection with the same model on each channel given a fixed threshold. Use logical OR to produce a final result based on the binary outcome of each channel.

**Table 2**: Model Comparisons

| Models | (#Params, | FR Rate | |
|---|---|---|---|
| | MAC/10ms) | Clean | TV noise |
| 1ch model on Ch0\|Ch1 | (318K, 0.32M) | 2.2% | 28.6% |
| 1ch model on Ch0\|Ch1 | (429K, 0.43M) | 1.96% | 24.8% |
| 1ch model on Ch0 | (429K, 0.21M) | 1.98% | 25.8% |
| 1ch model on Ch1 | (429K, 0.21M) | 2.0% | 26.0% |
| 1ch model on Broadside-BF | (429K, 0.21M) | 1.93% | 24.1% |
| 2ch model on Ch0+Ch1 | (429K, 0.21M) | **1.6%** | **19.5%** |

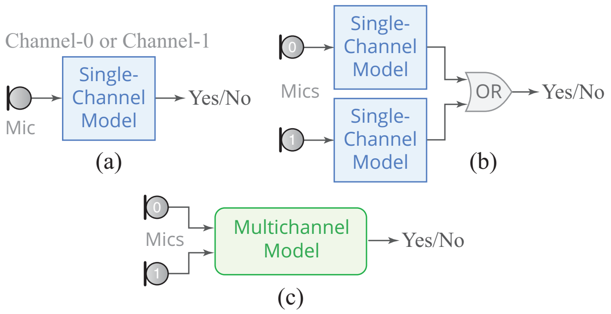


**Fig. 3**: Runtime strategies with 2-channel audio: (a) 1ch model on either Ch0 or Ch1 alone, (b) 1ch model on Ch0 | Ch1, and (c) 2ch model on Ch0 + Ch1.

The 3rd strategy in Fig. 3 is for our proposed multi-channel model.

### 4.4. Testing Data

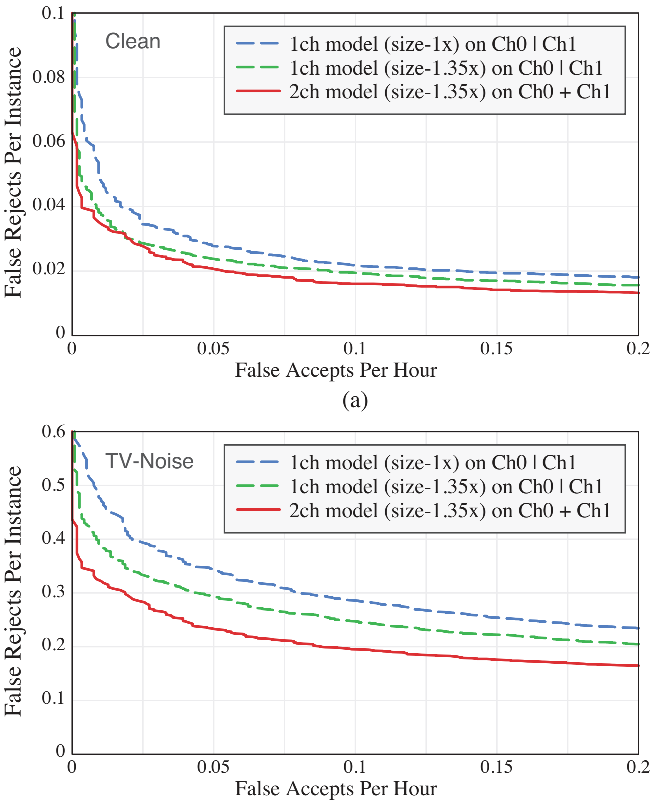
For testing data, we use prompts from anonymized and aggregated search queries, randomly prepended or appended with the keyword. These prompts were spoken by mechanical turk contributors and volunteers, and re-recorded with a dual-microphone setup. We collect the re-recorded data for our testing. The summary of the data is shown in Table 1. For positive dataset which contains either “Ok Google” or “Hey Google” in the audio, we cover two conditions: far-field clean and TV background noise. Data with TV noise has 10dB SNR. The negative set is collected from internal far-field Google Home devices with only random TV audio without any keyword.

### 4.5. Evaluation Metrics

For keyword spotting task, it is common to use FR (false reject) and FA (false accept) to measure the performance of a system. We draw Receiver Operating Characteristic Curve (ROC curve) whose axes are FR per instance and FA per hour.

## 5. RESULTS

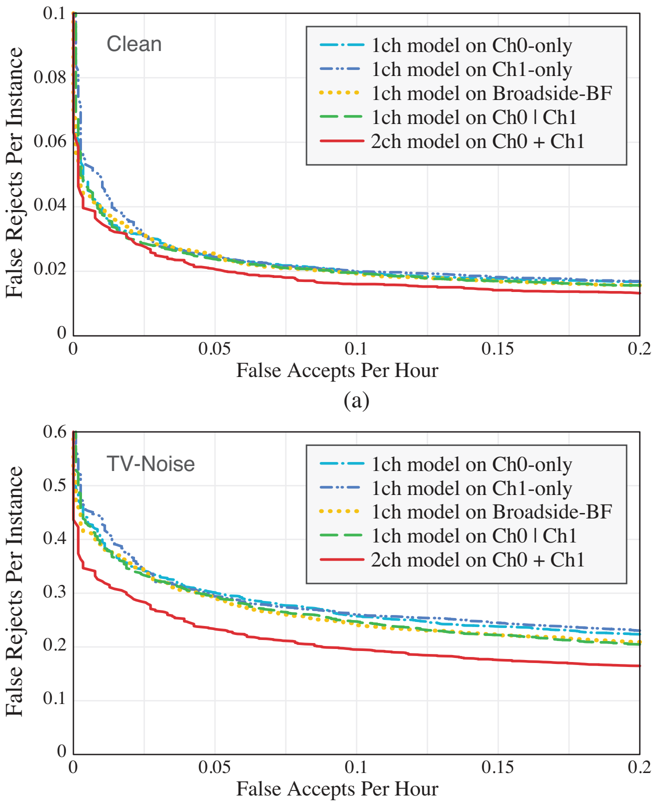
Our experiments aim to compare the quality improvement using the proposed 3D-SVDF based multi-channel keyword spotting system against a small size but high accuracy baseline [11]. Our evaluation is on a dual-microphone setup. We first compare the proposed system with the baseline and its size-matching alternative. We use the 2nd strategy from Section 4.3 to run these single-channel models. It shows increasing number of parameters does help a little given the same topology. However, the proposed architecture has a clear improvement over the two single-channel models in both clean and noisy conditions. The ROC curves in Fig. 4. show the consistent



(a)

(b)

**Fig. 4**: ROCs comparing performance of two 1-channel models using the logical-or runtime strategy and a 2-channel model on (a) clean and (b) TV-noise data sets. These three models are different in size: 1x and 1.35x have 318k and 429k parameters, respectively.



(a)

(b)

**Fig. 5**: ROCs comparing performance of the three different runtime strategies on (a) clean and (b) TV-noise data sets. The involved 1-channel and 2-channel models are all in the same size of 429k parameters.

improvement at different operating points. Besides that, as shown in Table 2, proposed architecture also shows the advantage in computation. Its MAC/10ms is only 65.6% of the baseline model and 50% of the size-matching alternative. This translates into the runtime latency improvement of the proposed model.

We also compare the proposed architecture with the baseline alternative running with different strategies. As shown in Fig. 5, we run the detection on either channel 0 or channel 1. Also, we run the model on each channel and use logical OR for the final detection results. We can see the 2nd strategy has the better ROC curves compared with running on either channel 0 or channel 1. The proposed architecture outperforms all of the single-channel models shown from Fig. 5.

Finally, we experimented with a simple, fixed broadside delay-and-sum beamformer [24] and the result is shown in Table 2 and Fig. 5. When the test data was recorded, the position of the speech source with respect to the microphone array was unfortunately not saved. But we believe that the speech sources were in the broadside in most of the recordings. This is also in consistent with the finding that the broadside beamformer yields better results than the endfire beamformer in our research. As shown by Table 2, the relative improvements of the proposed architecture against this beamformer are 20.6% and 23.4% on the clean and noisy datasets, respectively. This implies that the proposed approach is more adaptable to variations in signal directions.

In summary, the proposed 3D-SVDF based multi-channel keyword spotting has the best performance on both the clean and noisy datasets with a dual-microphone setup, given the same number of model parameters. This can translate into an implementation without any increase in memory or computation.

## 6. CONCLUSIONS

In this paper we have presented a 3D-SVDF based architecture for keyword spotting which not only enables detection improvement from a multi-microphone setup, but also keeps the model size small. We compared with a known memory and computationally efficient end-to-end single-channel keyword spotting architecture [11]. We also compared with a simple broadside delay-and-sum beamformer. With two microphones, the proposed architecture has a significant improvement on both clean and noisy data. Our method provides the possibility by utilizing multi-channel information directly in a neural-net architecture with a small footprint in memory. Future work includes integrating the architecture with the known adaptive noise-cancellation methods from these papers [21, 22].

## 7. ACKNOWLEDGMENTS

The authors would like to thank Ian McGraw, Alex Gruenstein, Joe Caroselli and Arun Narayanan for the discussion on early ideas of this research.

## 8. REFERENCES

[1] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system," in *Proc. IEEE ICASSP*, 1990, pp. 129–132.

[2] M. Wu, S. Panchapagesan, M. Sun, J. Gu, R. Thomas, S. N. P. Vitaladevuni, B. Hoffmeister, and A. Mandal, "Monophone-based background modeling for two-stage on-device wake word detection," in *Proc. IEEE ICASSP*, 2019, pp. 5494–5498.

[3] S. Panchapagesan, M. Sun, A. Khare, S. Matsoukas, A. Mandal, B Hoffmeister, and S. N. P. Vitaladevuni, "Multi-task learning and weighted cross-entropy for DNN-based keyword spotting," in *Proc. Interspeech*, 2016, pp. 760–764.

[4] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. IEEE ICASSP*, 2014, pp. 4087–4091.

[5] R. Prabhavalkar, R. Alvarez, C. Parada, P. Nakkiran, and T. N. Sainath, "Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks," in *Proc. IEEE ICASSP*, 2015, pp. 4704–4708.

[6] A. Gruenstein, R. Alvarez, C. Thornton, and M. Ghodrat, "A cascade architecture for keyword spotting on mobile devices," arXiv:1712.03603, 2017.

[7] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, Senior A. W., and K. Kavukcuoglu, "WaveNet: a generative model for raw audio," arXiv:1609.03499, 2016.

[8] A. Coucke, M. Chlieh, T. Gisselbrecht, D. Leroy, M. Poumeyrol, and T. Lavril, "Efficient keyword spotting using dilated convolutions and gating," in *Proc. IEEE ICASSP*, 2018, pp. 6351–6355.

[9] S. Choi, S. Seo, B. Shin, H. Byun, M. Kersner, B. Kim, D. Kim, and S. Ha, "Temporal convolution for real-time keyword spotting on mobile devices," arXiv:1904.03814, 2019.

[10] P. Nakkiran, R. Alvarez, R. Prabhavalkar, and C. Parada, "Compressing deep neural networks using a rank-constrained topology," in *Proc. Interspeech*, 2015, pp. 1473–1477.

[11] R. Alvarez and H.-J. Park, "End-to-end streaming keyword spotting," in *Proc. IEEE ICASSP*, 2019, pp. 6336–6340.

[12] M. Wölfel and J. W. McDonough, *Distant Speech Recognition*, John Wiley & Sons, West Sussex, UK, 2009.

[13] S. Sukittanon, A. C. Surendran, J. C. Platt, and C. J. C. Burges, "Convolutional networks for speech detection," in *Proc. Interspeech*, 2004, pp. 1077–1080.

[14] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. W. Senior, K. K. Chin, A. Misra, and C. Kim, "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, and Language Process.*, vol. 25, pp. 965–979, 2017.

[15] T. Ochiai, S. Watanabe, T. Hori, J. R. Hershey, and X. Xiao, "Unified architecture for multichannel end-to-end speech recognition with neural beamforming," *IEEE J. Sel. Topics Signal Process.*, vol. 11, pp. 1274–1288, 2017.

[16] S. Ganapathy and V. Peddinti, "3-D CNN models for far-field multi-channel speech recognition," in *Proc. IEEE ICASSP*, 2018, pp. 5499–5503.

[17] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *IEEE Signal Process. Lett.*, vol. 21, pp. 1120–1124, 2014.

[18] N. Tawara, T. Kobayashi, and T. Ogawa, "Multi-channel speech enhancement using time-domain convolutional denoising autoencoder," in *Proc. Interspeech*, 2019, pp. 86–90.

[19] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. Interspeech*, 2013, pp. 25–29.

[20] A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Trans. Audio, Speech, and Language Process.*, vol. 24, pp. 1652–1664, 2016.

[21] Y. Huang, T. Z. Shabestary, and A. Gruenstein, "Hotword cleaner: dual-microphone adaptive noise cancellation with deferred filter coefficients for robust keyword spotting," in *Proc. ICASSP*, 2019, pp. 6346–6350.

[22] Y. Huang, T. Z. Shabestary, A. Gruenstein, and L. Wan, "Multi-microphone adaptive noise cancellation for robust hotword detection," in *Proc. Interspeech*, 2019, pp. 1233–1237.

[23] N. Jaitly, P. Nguyen, A. W. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. Interspeech*, 2012.

[24] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*, Springer, Berlin, Germany, 2008.