INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Device playback augmentation with echo cancellation for keyword spotting

Kuba Łopatka, Katarzyna Kaszuba-Miotke, Piotr Klinke, Paweł Trella

Intel Corporation, Gdańsk, Poland

`kuba.lopatka, katarzyna.kaszuba, piotr.klinke, pawel.trella@intel.com`

Abstract

Keyword spotting (KWS) is required to operate in device playback conditions in which the device itself plays interfering signals. We propose a new method to augment the training set and adapt the acoustic model to the playback environment. It is based on acoustic simulation which models the coupling between the device's loudspeakers and microphones. The employed model involves frequency response of the device, as well as room impulse response and nonlinear distortions introduced in the playback path. Finally, we pass the simulated signals through Acoustic Echo Cancellation (AEC) to model the artifacts introduced by AEC algorithm. The proposed method reduces False Rejection Rate in device playback noise by 25-60% for a Time-Delay Neural Network-based KWS engine. It is shown that the introduction of device characteristics and nonlinear filtration is necessary to achieve improvement in playback conditions. The augmentation scheme is highly independent of the architecture of the KWS system.

Index Terms: keyword spotting, acoustic echo cancellation, data augmentation

1. Introduction

Keyword spotting (KWS) is an algorithm which detects a predefined word in the audio input stream, typically to trigger personal assistant software. The task is related to Automatic Speech Recognition (ASR) but has some unique challenges. KWS often runs locally on a battery-operated device, so power is a key factor [1]. The compute and memory footprint [2], as well false detection rate have to be kept to a minimum [3]. A practical system has two stages, where the first stage model works with limited resources and second stage verifies the detection with more resources available [4].

Several model architectures have been tried for KWS, including multi-layer perceptron [5], Convolutional Neural Networks (CNN) [3, 6], Time-Delay Neural Networks (TDNN) [1] or recurrent networks [7, 8]. Latest research on KWS includes attention pooling [9] or max pooling [7, 10] as well as spiking neural networks (SNN) [11]. Some researchers introduce end-to-end architectures with max-pooling loss [12, 13] or raw audio input [14].

Keyword spotting is required to work in challenging conditions, in the presence of room reflections and additive noise. For both KWS and ASR, the state-of-the-art approach to making the model robust against challenging environments consists of augmenting the training set. Most widely used augmentations include adding noise and reverberation to clean speech [15, 16, 17]. Some approaches involve transformations of source speech such as changing tempo, pitch or vocal tract characteristics [18]. Researchers also report improvement after adding synthesized speech samples to the training set [19]. Finally, methods like *SpecAugment* transform the features directly before they are fed into to the model [20].

Arguably, the most difficult condition is device playback. In this condition the user utters the keyword while the device is playing sounds (e.g. music) through built-in loudspeakers. As it is shown in Fig. 1, the playback sound leaks to the microphone through direct coupling, internal coupling and room reflections. Due to physical proximity of playback and capture transducers, the leakage is often louder than speech, thus making it difficult to recognize the keyword. Typically, Acoustic Echo Cancellation (AEC) is employed to filter out the leakage [21]. It is achieved by adaptive filtration with loopback signal used as reference. After AEC, playback is highly attenuated and speech is more intelligible. However, due to nonlinear distortions present in the audio path, considerable artifacts remain which significantly impair the accuracy of KWS [22]. There is little research focusing on improving KWS performance under playback conditions. Raju et al. study the impact of adding music signals to speech in the training set and report improvement compared to a model trained on clean speech [23].

Our solution is to include simulation of device playback in the process of training the acoustic model. We simulate the capture path of the device and process the training signals with AEC to incorporate specific echo cancellation artifacts. In order to achieve high accuracy the frequency response of loudspeakers and microphones, as well as nonlinear distortions are factored into the simulation. By augmenting the training set with playback simulation we achieve increased robustness of the acoustic model which leads to consistent improvement in KWS performance for signals after echo cancellation. To the best of our knowledge, simulation of device playback with AEC has not been included in training ASR or KWS models before.

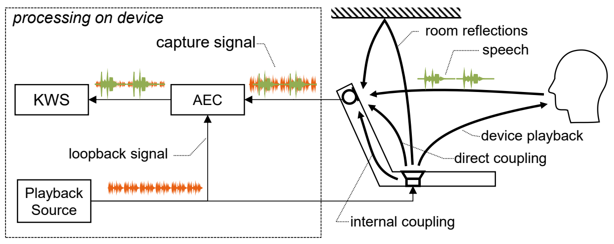

Figure 1: *Device playback scenario*

2. Acoustic simulation

2.1. Noise and reverberation

The most common approach to prepare a multi-condition training set is based on applying a room response filter h_{RIR} to clean speech $s(t)$ and adding a noise signal $n(t)$.

$$x(t) = s(t) * h_{RIR}(t) + G_N \cdot n(t) * h_{RIR}(t) \qquad (1)$$

Additional noise gain G_N is used to control the resulting Signal-to-Noise Ratio (SNR). Room impulse response h_{RIR}

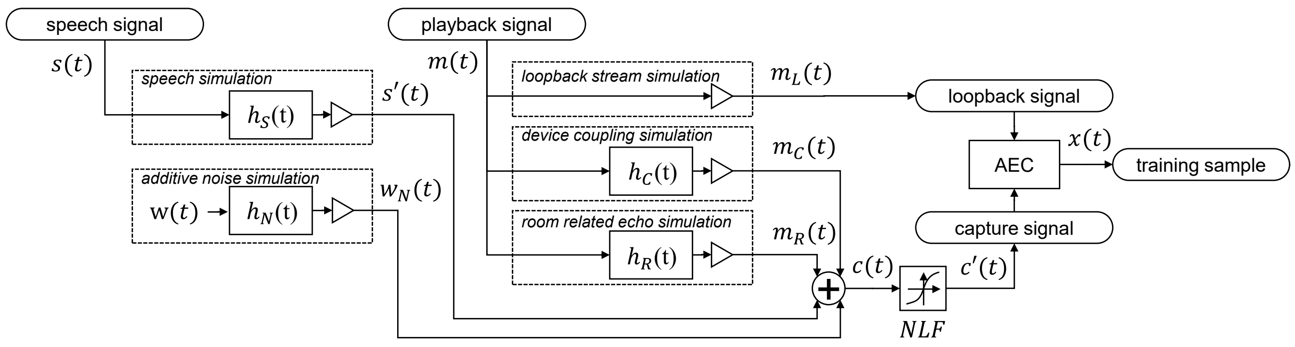

Figure 2: *Diagram of device playback simulation*

can originate from a database of impulse responses [24] or it can be generated using software such as *pyroomacoustics* [25] given the geometric setup of the room, sources and receivers.

This methodology is very widely adopted [23, 15, 16, 17]. It yields good results for noise and far field scenarios but falls short when dealing with device playback.

2.2. Device playback

To properly simulate device playback specific transformations have to be introduced in the capture path, which resemble distortions observed on real devices. The simulation diagram is featured in Fig. 2 In this subsection we introduce the formulas to simulate the respective components in the audio path.

The goal is to model the capture signal $c(t)$ which is registered by the built-in microphone. It is a sum of 4 components: speech signal $s'(t)$, self noise $w_N(t)$, playback leakage $m_C(t)$ and playback room-related echo $m_R(t)$.

$$c(t) = s'(t) + w_N(t) + m_C(t) + m_R(t) \tag{2}$$

These components are simulated by convolution of a source signal with an impulse response and scaling by gain.

$$\begin{aligned} c(t) =& G_S \cdot s(t) * h_S(t) + G_N \cdot w(t) * h_N(t) \\ &+ G_C \cdot m(t) * h_C(t) + G_R \cdot m(t) * h_R(t) \end{aligned} \tag{3}$$

Source signals for speech $s(t)$ and playback $m(t)$ are sampled from the training and augmentation set described in Sec. 4. For device self noise we use white noise $w(t)$. Gain is introduced to control the resulting SNR, or to model additional acoustic properties. For instance, speech gain G_S and room echo gain G_R are related to microphone sensitivity, whereas coupling gain G_C also depends on weighted terminal coupling loss TCL_w.

Impulse responses $h_S(t)$, $h_C(t)$ and $h_R(t)$ are decomposed into the following factors.

$$h_S(t) = h_{RIR_S}(t) * h_{mic_S}(t) \tag{4}$$

$$h_C(t) = h_{C_i}(t) * h_{C_d}(t) \tag{5}$$

$$h_R(t) = h_{RIR_R}(t) * h_{spk_R}(t) * h_{mic_R}(t) \tag{6}$$

where h_{RIR_S} and h_{RIR_S} are room impulse responses applied to speech and room echo respectively, h_{C_i} and h_{C_d} model internal and direct coupling, h_{spk_R} is the diffuse field loudspeaker impulse response, h_{mic_R} is the diffuse field microphone impulse response and h_{mic_S} is the free field microphone impulse response.

So far, the model incorporates only linear distortions of the audio path. In reality, there are non-linear distortions present as well, which are the main cause of artifacts observed after echo cancellation [26]. To factor in the nonlinearities, the resulting capture signal $c'(t)$ is processed by a non-linear filter NLF. The employed non-linear filter is discussed in Sec. 2.4.

2.3. Parametric equalization

The impulse responses (h_N, h_{C_i}, h_{C_d}, h_{spk_R} h_{mic_R}, h_{mic_S}) are modeled by parametric equalization. Parametric filters are designed to resemble the frequency responses of real devices, based on multiple measurements. We provide an example of how the microphone impulse response is modeled.

The two most important factors which shape the frequency response of a digital microphone are low frequency ventilation hole effect and Helmloltz resonance in high frequencies [27, 28]. Low frequency filtration $H_{hpf}(f)$ is characterized by the cutoff frequency f_c and attenuation in decibels per octave k, while Helmholtz resonance $H_{peak}(f)$ depends on resonance frequency f_{peak}, resonance gain g_{peak} and resonance Q-factor Q_{peak}. Both factors are modeled in frequency domain $H_{mic_R}(f)$ and approximating filter $h_{mic_R}(t)$ is calculated using frequency sampling FIR filter design method [29].

$$H_{hpf}(f) = -20 * \log \sqrt{1 + 10^{\frac{-k*\log(f)/\log(2)+\log(f_c)}{10}}} \tag{7}$$

$$H_{peak}(f) = g_{peak} \cdot e^{-\frac{1}{2}*(f-f_{peak})\cdot Q_{peak}\cdot \frac{e}{f_{peak}}} \tag{8}$$

$$H_{mic_R}(f) = H_{hpf}(f) + H_{peak}(f) \tag{9}$$

By altering the parameters of these filters, an arbitrary number of device characteristics can be generated. This mechanism prevents overfitting to specific devices which is an advantage over using frequency responses originating from measurement. An analogous process is employed to model the remaining impulse responses featured in the simulation.

2.4. Nonlinear filtering

Nonlinear behavior of the acoustic path is mostly related to physical properties of loudspeakers, such as suspension stiffness and voice coil inductance [30], as well as internal coupling through non-ideal common chassis supporting both speakers and microphones [31]. We model the nonlinearities with waveshaping method using N^{th} order Chebyshev Polynomials with $N = 5$. Model produces both intermodulation and harmonic distortion components. Each polynomial has a single scaling parameter α_n. First order linear component uses

$\alpha_1 = 1$. Scales for higher order polynomials are smaller. DC component is compensated afterwards.

$$\begin{cases} T_0(x) = 1 \\ T_1(x) = 2x - 1 \\ T_{n+1}(x) = 2xT_n - T_{n-1} \end{cases} \tag{10}$$

$$NLF(x) = \sum_{n=1}^{N} T_n(x) \cdot \frac{\alpha_n}{N} \tag{11}$$

2.5. Acoustic echo cancellation

Finally, the capture and loopback signals are processed with AEC. The employed algorithm is Frequency-domain adaptive Kalman Filter (FDKF) [32]. The filter requires input (capture) and reference (loopback). The loopback signal $m'_L(t)$ is the raw device playback signal $m(t)$ scaled with gain relative to the device loopback sensitivity. Due to the distortions introduced in the simulation path the resulting sample $x(t)$ incorporates artifacts very similar to the ones observed in real signals passed through AEC. Hence, the keyword spotting engine can be adapted to classify such signals with higher accuracy.

2.6. Simulation parameters

Selected parameters which can be adjusted in simulation along with the assumed range of values are listed in Tab. 1. The values can be sampled at random but some supervision is beneficial to make sure that the resulting characteristics are realistic. Thus, we define a set of virtual devices which we employ to generate the training samples.

Table 1: *Selected parameter values assumed in simulation*

parametric model	**parameter**	**value**
Gain	SNR	$(-7dB, +7dB)$
	sensitivity	$-26dB_{FS}/Pa$
Frequency response	f_c	$(100Hz, 250Hz)$
	k	$(3dB/oct, 6dB/oct)$
	f_{peak}	$(6kHz, 20kHz)$
	Q_{peak}	$(1, 15)$
	g_{peak}	$(10dB, 40dB)$
Device coupling G_C	TCLw	$(-10dB, +10dB)$
Self noise h_N	SNR	$(50dB, 70dB)$
Nonlinear filter	α_n	$(0, 0.1)$

3. Keyword spotting engine

Our engine, which is presented in Fig. 3, is composed of three main blocks: frontend, acoustic model and keyword/rejection model. The details of our algorithms are discussed in previous work [33]. The effect of augmenting the training set with device playback simulation is largely independent from the employed KWS algorithm.

3.1. Acoustic model

The input of the acoustic model are 40 log-filterbank features extracted from 25 ms overlapping frames with 10 ms shift. The network is prepended with a scaling layer which standarizes the input features. We use 3 TDNN layers with 128 units and bottleneck layers in between, all with rectified linear unit (ReLU) activation. The TDNN layers connect to activations from previous layers from 3 frames back $(t - 3)$ and 3 frames ahead $(t + 3)$. This enables skipping every 2nd and 3rd frame during inference to save computations, as proposed in [34]. The final layer has 3897 outputs for all Large Vocabulary Continuous Speech Recognition (LVCSR) posteriors. Only 86 of these posteriors are used in the keyword/rejection model (36 for keyword, 50 for rejection). In total, the neural network has 135K parameters which makes it suitable for deployment as a first pass model in embedded environment.

3.2. Keyword model

The phrase used in experiments is *Hello computer*. It comprises 12 triphones, which yields $N = 36$ states (3 per triphone).

For each time step t the acoustic model estimates a vector of framewise posteriors $[p_1, p_2, ..., p_N]$ which relate to triphone states of the keyword S_1 to S_N. The first state S_0 is a *rejection* state. It is updated based on a set R of 50 *rejection* posteriors chosen with a heuristics based on *a priori* phonetic knowledge and triphone statistics of the English language.

$$\begin{cases} S_0(0) = 0 \\ S_0(t) = S_0(t-1) + \max_{m \in R}(p_m(t)) \end{cases} \tag{12}$$

For $n = 1..N$ the state scores are updated in each time step:

$$\begin{cases} S_n(0) = -\infty \\ S_n(t) = \max\{S_{n-1}(t-1) + p_n(t), S_n(t-1) + p_n(t)\} \end{cases} \tag{13}$$

Finally, the final score S_{final} is computed.

$$S_{final} = \max_{t \in \{0,1,...,T-1\}} (S_N(t) - S_0(t)) \tag{14}$$

3.3. Model training

The model is trained with state sequence pooling method [33]. It combines standard framewise cross entropy with a dedicated loss function tied to the keyword score. We first pretrain the model for 50 iterations employing just cross entropy loss with LVSCR targets and then we train for 20 more iterations with state sequence pooling loss. We use Adam optimizer with learning rate of 0.001.

4. Experiment

4.1. Training set

Table 2: *Training sets employed for specific models*

	model	**A**	**B**	**C**	**D**	**E**
non keywords	clean	477k	477k			
	noise/reverb	206k	165k			
	playback	0	52k			
keywords	clean	1.5k	1.5k			
	noise/reverb	10k	7k			
	playback	0	3k			
	TOTAL	706k	720k			
playback simulation factors	device	–	✓	✓	✗	✗
	nonlinearity	–	✓	✗	✓	✗
	AEC	–	✓	✓	✓	✓

The structure of the training set is outlined in Tab. 2. For non-keywords, we use US English Speecon database [35]. As keyword samples we use 1500 internal and crowd-sourced

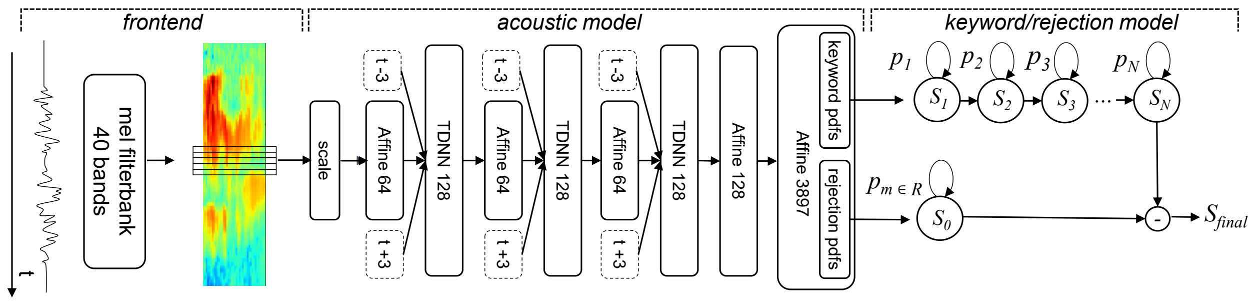

Figure 3: *Keyword spotting algorithm*

recordings of *Hello computer*. We apply noise/reverb augmentations as per (1) and device playback augmentations as per the apparatus defined in Sec. 2. The ratio of augmented vs. original utterances is kept at a similar level across all models.

In order to study the importance of respective simulation factors, we train 5 models. To avoid overfitting to playback conditions we keep the ratio of device playback augmentations relatively low compared to noise/reverb augmentations. The factor *device* relates to the impulse responses h_N, h_{C_i}, h_{C_d}, h_{spk_R} h_{mic_R}, h_{mic_S}. When it is disabled, all impulse responses become identity ($\delta(t)$). The factor *nonlinearity* pertains to processing with non-linear filter as per (11). Finally, *AEC* factor indicates that echo cancellation is applied to the signal. We do not include a model trained just on clean speech as it has already been shown in related works that a model trained on signals augmented with music performs better in playback conditions [23].

4.2. Augmentation signals

For noise signal $n(t)$ we utilize Speecon [35] and MUSAN [36] open corpora. We select 2k samples for noise and 700 samples for music. For room simulation ($h_{RIR}(t)$) we employ impulse responses published in [24]. Device playback simulation utilizes the same 700 samples of music signals for $m(t)$. Target SNR for the noise/reverb case is set in the range (-10,10) dB.

4.3. Evaluation set

For evaluation, we use 389 internal recordings of keyword samples. Instead of simulated characteristic, real impulse responses and real music leakage signals recorded on the target device are used to generate signals with high fidelity. The generated signal was shown to be very close to real-life signals through extensive acoustic tests. The signals are passed through AEC with FDKF algorithm and then fed into the keyword recognizer. For non-keywords, we use snippets from US English podcasts mined from sources with open license. We extract 17280 random 5-second snippets which totals of 24 hours.

4.4. Results

The plot in Fig. 4 presents the tradeoff between false rejection rate (FRR) and false accept rate (FAR). The models defined in Tab. 2 are compared. The models are quantized in 8-bit fixed point format. We present average results for two music recordings. The model trained on just noise/reverb augmentations (A) serves as baseline. With the full simulation path, including device characteristics and non-linear filtration (model B), there is visible improvement over the baseline. Model C, including device characteristics but no nonlinearities, is better than baseline for some threshold values, but the improvement is inconsistent. We also observe that the model trained without device characteristics (D) and the model incorporating only the AEC algorithm (E) yield no improvement over the baseline. Comparing the best model (B) to the baseline (A) we observe an FRR reduction ranging from 25.2% at 1 false wakes per 8 hours (from 27.98% to 20.92%) to 60.4% at 1 false wake in 10 minutes (from 6.16% to 2.44%) . On average the FRR is reduced by 45.6%.

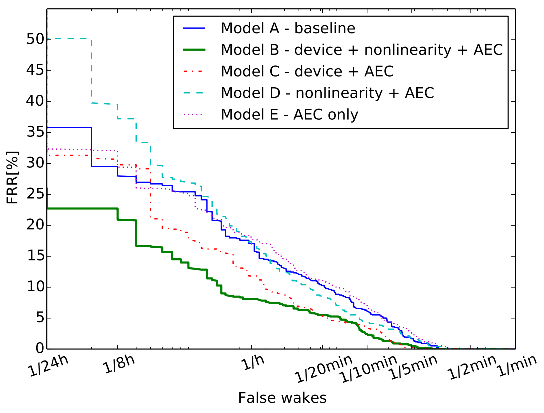

Figure 4: *Results of evaluation in playback conditions*

5. Conclusions

We introduce a simulation framework to augment the training set for keyword spotting under device playback conditions. We achieve visible improvement in keyword detection accuracy. Interestingly, consistent improvement is observed only when all simulation factors are enabled, which enables close reproduction of the acoustic path of the device.

This work can be extended in the following ways. Firstly, we aim to examine the relation between the number of simulated devices and the ranges of simulation parameters to the accuracy of the trained model. Secondly, instead of fixed impulse responses, room geometry can be defined and the impulse responses can be generated on the fly. Thirdly, it is interesting to examine the performance of the trained model in various conditions, i.e. recorded on different devices, with different playback signals or noises. Finally, the proposed augmentation method can be applied to other tasks than keyword spotting, including automatic speech recognition or acoustic event detection.

6. References

[1] M. Sun, D. Snyder, Y. Gao, V. Nagaraja, M. Rodehorst, S. Panchapagesan, N. Strom, S. Matsoukas, and S. Vitaladevuni, "Compressed time delay neural network for small-footprint keyword spotting," in *Interspeech 2017*, 08 2017, pp. 3607–3611.

[2] O. Rybakov, N. Kononenko, N. Subrahmanya, M. Visontai, and S. Laurenzo, "Streaming keyword spotting on mobile devices," *Interspeech 2020*, Oct 2020.

[3] 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," in *INTERSPEECH*, 2019.

[4] M. Wu, S. Panchapagesan, M. Sun, J. Gu, R. Thomas, S. N. Prasad Vitaladevuni, B. Hoffmeister, and A. Mandal, "Monophone-based background modeling for two-stage on-device wake word detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5494–5498.

[5] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4087–4091.

[6] X. Li, X. Wei, and X. Qin, "Small-footprint keyword spotting with multi-scale temporal convolution," in *INTERSPEECH*, 2020.

[7] M. Sun, A. Raju, G. Tucker, S. Panchapagesan, G. Fu, A. Mandal, S. Matsoukas, N. Strom, and S. Vitaladevuni, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," *2016 IEEE Spoken Language Technology Workshop (SLT)*, pp. 474–480, 2016.

[8] S. Sigtia, E. Marchi, S. Kajarekar, D. Naik, and J. Bridle, "Multi-task learning for voice trigger detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7449–7453.

[9] C. Shan, J. Zhang, Y. Wang, and L. Xie, "Attention-based end-to-end models for small-footprint keyword spotting," in *Interspeech 2018*, 09 2018, pp. 2037–2041.

[10] H. Park, P. Violette, and N. Subrahmanya, "Learning to detect keyword parts and whole by smoothed max pooling," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7899–7903, 2020.

[11] E. Yilmaz, Ö. B. Gevrek, J. Wu, Y. Chen, X. Meng, and H. Li, "Deep convolutional spiking neural networks for keyword spotting," in *INTERSPEECH*, 2020.

[12] R. Alvarez and H. Park, "End-to-end streaming keyword spotting," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6336–6340.

[13] H. Park, P. Violette, and N. Subrahmanya, "Learning to detect keyword parts and whole by smoothed max pooling," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7899–7903.

[14] S. Mittermaier, L. Kurzinger, B. Waschneck, and G. Rigoll, "Small-footprint keyword spotting on raw audio data with sinc-convolutions," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7454–7458, 2020.

[15] I. Rebai, Y. BenAyed, W. Mahdi, and J.-P. Lorré, "Improving speech recognition using data augmentation and acoustic model fusion," *Procedia Computer Science*, vol. 112, pp. 316–322, 2017.

[16] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[17] E. Bezzam, R. Scheibler, C. Cadoux, and T. Gisselbrecht, "A study on more realistic room simulation for far-field keyword spotting," 2020.

[18] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH*, 2015.

[19] J. Lin, K. Kilgour, D. Roblek, and M. Sharifi, "Training keyword spotters with limited and synthesized speech data," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7474–7478, 2020.

[20] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *INTERSPEECH 2019*, 2019, pp. 2613–2617.

[21] A. Deb, D. A. Kar, and M. Chandra, "A technical review on adaptive algorithms for acoustic echo cancellation," 04 2014.

[22] H. Song and J. W. Shin, "Residual echo suppression considering harmonic distortion and temporal correlation," *Applied Sciences*, vol. 10, no. 15, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/15/5291

[23] A. Raju, S. Panchapagesan, X. Liu, A. Mandal, and N. Strom, "Data augmentation for robust keyword spotting under playback interference," *ArXiv*, vol. abs/1808.00563, 2018.

[24] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[25] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2018.

[26] M. I. Mossi, C. Yemdji, N. Evans, and C. Beaugeant, in *IEEE 10th Int. Conf. on Signal Processing, title = A comparative assessment of noise and non-linear echo effects in acoustic echo cancellation, year = 2010, volume = , number = , pages = 223-226, doi = 10.1109/ICOSP.2010.5655142*.

[27] *Tutorial for MEMS microphones*, STMicroelectronics NV, 2017.

[28] *Frequency response and latency of MEMS microphones: theory and practice*, Knowles Corporation, 2017.

[29] L. B. Jackson, *Digital Filters and Signal Processing: With MATLAB Exercises*, 3rd ed. USA: Kluwer Academic Publishers, 1996.

[30] W. Klippel, "Modeling the large signal behavior of micro-speakers," in *133rd AES Convention*, 2012.

[31] Kun Shi, Xiaoli Ma, and G. Tong Zhou, "Adaptive acoustic echo cancellation in the presence of multiple nonlinearities," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 3601–3604.

[32] G. Enzner and P. Vary, "Frequency-domain adaptive kalman filter for acoustic echo control in hands-free telephones," *Signal Processing*, vol. 86, no. 6, pp. 1140–1156, 2006, applied Speech and Audio Processing.

[33] K. Lopatka and T. Bocklet, "State sequence pooling training of acoustic models for keyword spotting," in *INTERSPEECH*, 2020.

[34] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *INTERSPEECH*, 09 2016, pp. 2751–2755.

[35] D. Iskra, B. Grosskopf, K. Marasek, H. van den Heuvel, F. Diehl, and A. Kiessling, "SPEECON - speech databases for consumer devices: Database specification and validation," *Proceedings of LREC*, 06 2002.

[36] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1510.08484