An Experimental Study on Light Speech Features for Small-Footprint Keyword Spotting

Espejo, Ivan Lopez; Tan, Zheng-Hua; Jensen, Jesper

Published in:
IberSPEECH 2022

DOI (link to publication from Publisher):
10.21437/IberSPEECH.2022-27

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

IberSPEECH 2022
14-16 November 2022, Granada, Spain

An Experimental Study on Light Speech Features for Small-Footprint Keyword Spotting

Iván López-Espejo[1], *Zheng-Hua Tan*[1] *and Jesper Jensen*[1,2]

[1]Department of Electronic Systems, Aalborg University, Denmark
[2]Oticon A/S, Denmark

{ivl,zt,jje}@es.aau.dk, jesj@demant.com

Abstract

Keyword spotting (KWS) is, in many instances, intended to run on smart electronic devices characterized by limited computational resources. To meet computational constraints, a series of techniques —ranging from feature and acoustic model parameter quantization to the reduction of the number of model parameters and required multiplications— has been explored in the literature. With this same aim, in this paper, we study a straightforward alternative consisting of the reduction of the spectro/cepstro-temporal resolution of log-Mel and Mel-frequency cepstral coefficient feature matrices commonly employed in KWS. We show that the feature matrix size has a strong impact on the number of multiplications/energy consumption of a state-of-the-art KWS acoustic model based on convolutional neural network. Experimental results demonstrate that the number of elements in commonly used speech feature matrices can be reduced by a factor of 8 while essentially maintaining KWS performance. Even more interestingly, this size reduction leads to a 9.6× number of multiplications/energy consumption, 4.0× training time and 3.7× inference time reduction.

Index Terms: Keyword spotting, speech features, small footprint, energy consumption, deep learning.

1. Introduction

Keyword spotting (KWS) is a highly useful technology, which enables voice interaction with a plethora of smart electronic devices such as smartphones, tablets, smartwatches and the like. Since these devices are characterized by notable computational and energy constraints, embedding on them typical always-on KWS technology can pose a challenge.

The cornerstone of state-of-the-art KWS is the acoustic model producing word or subword posteriors from speech features, which is implemented by a neural network and can be computationally demanding [1]. To fit into computational and memory constraints as well as to limit the impact on battery lifetime, a series of methods has been explored in the literature. In the field of KWS, we can distinguish between two main categories of methods, which can be applied to essentially any neural network structure:

1. *Quantization*: This category refers to those techniques pursuing the reduction of the precision of the parameters of the acoustic model [2,3] and/or that of the speech features [4].
2. *Reduction of the number of parameters and/or multiplications*: This class of methods seeks decreasing the number of parameters and/or multiplications of the neural network acoustic model [1].

Henceforth, we focus on the second category above, which comprises a variety of approaches like weight pruning [5,6] and the use of depthwise separable convolutions [7] to decrease the memory footprint and number of multiplications of the model while essentially maintaining performance.

In this paper, we conduct an experimental study on speech feature matrix size reduction, which has an impact on the number of multiplications required by the acoustic model. Our results show that this impact is particularly remarkable when employing state-of-the-art acoustic modeling, which relies on convolutional neural networks (CNNs) with residual connections [1]. In particular, using a deep residual learning model [8], we show that the spectro/cepstro-temporal resolution of the log-Mel and Mel-frequency cepstral coefficient (MFCC) [9] feature matrices typically employed in KWS can be reduced by a factor of 8 without really hurting performance. More interestingly, this feature matrix size reduction yields 9.6× number of multiplications/energy consumption [10], 4.0× training time and 3.7× inference time reduction.

The remainder of this manuscript is structured as follows. In Section 2, we discuss related work motivating this study and outline the experimental methodology. Section 3 is devoted to present the experimental framework. Results are shown in Section 4. Finally, Section 5 wraps up this work.

2. Related Work and Methodology

In a previous work [11] in which we explored filterbank learning for KWS, we observed no statistically significant KWS accuracy differences between the use of learned filterbanks and handcrafted (log-Mel) speech features. After a number of experiments, we drew the conclusion that there is much redundant information contained in the speech features that are fed into modern neural network-based KWS acoustic models. Interestingly, the authors of [4], who showed that 8-bit log-Mel spectra can lead to the same KWS performance as full-precision MFCCs, independently reached to the same conclusion.

Motivated by the above, here we explore the impact of feature matrix size reduction on KWS performance. In this work, we focus on both log-Mel spectra and MFCCs [9], since these are the most commonly used speech features in KWS, e.g., see [8, 12–18]. A widespread feature extraction setting consists of the computation of 40 features per time frame while using a 30 ms analysis window with a hop size of 10 ms (e.g., again see [8, 12–18]). Therefore, with this setting, a one-second long audio segment is represented by a feature matrix of dimension 40 features×101 time frames[1].

Departing from the above standard feature extraction set-

[1]Notice that, due to the application of temporal zero-padding, the number of time frames is 101 and not 99.

10.21437/IberSPEECH.2022-27

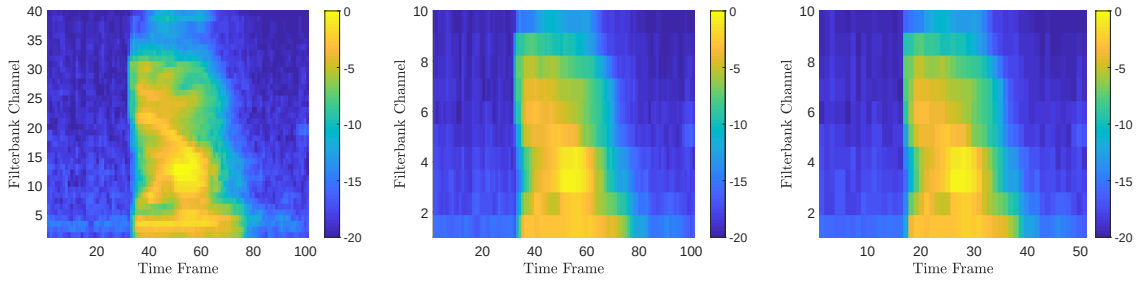

Figure 1: *Log-Mel spectra computed from the same one-second long utterance, comprising the word "down", by considering different spectro-temporal resolutions. In all cases, the spanned frequency range is [20, 8,000] Hz. See Section 2 for further details.*

ting, in Section 4, we explore the use of a smaller number of filterbank channels/cepstral coefficients (i.e., features) and also perform time resolution reduction by increasing the hop size of the analysis window. As an example, Figure 1 shows log-Mel spectra computed from the same one-second long utterance, comprising the word "down", by considering different spectro-temporal resolutions. While the spectrum of Figure 1a follows the standard feature extraction setting, the other two (see Figures 1b and 1c) consider 10 filterbank channels spanning the same frequency range of [20, 8,000] Hz. Moreover, the spectrum of Figure 1c, differently from the other two, is computed from a hop size of 20 ms instead of 10 ms. In Section 4, we will show that these three feature extraction settings essentially yield equivalent KWS performance while leading to quite distinct computational load.

3. Experimental Framework

This section is devoted to present the experimental framework. First, Subsection 3.1 outlines the speech dataset employed in this work. Second, Subsections 3.2 and 3.3 describe the deep residual learning acoustic model and the way it is trained, respectively.

3.1. The Google Speech Commands Dataset

For experimental purposes, we utilize the most popular publicly available benchmark for KWS research: the Google Speech Commands Dataset (GSCD) *v2* [19]. The GSCD is composed of a little more than 100k one-second long speech utterances comprising one word each from a set of 35 different words. At a 16 kHz sampling rate, these utterances were recorded from 2,618 different speakers by phone and laptop microphones. Recordings contain some background noise and, within them, words can be located anywhere.

As is standard [1, 19], we consider the 10 keywords "yes", "no", "up", "down", "left", "right", "on", "off", "stop" and "go". The remaining 25 words are employed to shape the non-keyword/filler class. In addition, the GSCD is split into training, validation and test sets according to the ratio 80:10:10, and speakers do not overlap across sets. Word classes are also quite balanced across sets.

3.2. Deep Residual Acoustic Modeling

From a recent overview paper on KWS [1], it is clear that state-of-the-art KWS acoustic modeling relies on CNNs integrating a mechanism to exploit long time-frequency dependencies and residual connections [20]. Accordingly, we make use of the deep residual learning model depicted in Figure 2, which also has dilated convolutions increasing the network's receptive field [8].

This model makes independent keyword predictions from one-second long input speech segments. Input speech features[2] are processed by six residual blocks each comprising two convolutional and two batch normalization layers in addition to one skip connection. Skip connections make the number of multiplications of the model heavily depend on the feature matrix size, since the successive feature maps have to preserve it for addition throughout the residual blocks. Thus, this CNN has around 238k parameters and requires around 895M multiplications per second of input speech when considering the standard feature extraction setting described in Section 2. To carry out classification, the final fully-connected layer has 11 nodes corresponding to the 10 different keywords plus the non-keyword/filler class. The reader is referred to [8] for further details on this deep residual learning model.

3.3. Model Training

The acoustic model of Figure 2 is trained by using cross-entropy as the loss function. Unlike [8] (where stochastic gradient descent is considered), Adam [22] with default parameters (learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$) is the optimizer employed in this paper. The size of the mini-batch is 64 training samples. For regularization purposes, early-stopping [23] monitoring the validation loss with a patience of four epochs is used. Notice that, differently from [8], training data augmentation is not considered in this work. The implementation was done by means of Keras [24] performing on top of TensorFlow [25].

4. Results

To assess both KWS and computational performance when reducing the spectro/cepstro-temporal resolution of the speech feature matrices, the following metrics are used:

1. *Accuracy*: This is simply the ratio of the number of correct predictions over the total number of them [26]. Notice that accuracy is a very popular primary KWS performance metric when employing the GSCD [1, 8, 18, 27, 28], where word classes are quite balanced.

[2]For speech feature extraction, we use Librosa [21]. Furthermore, prior to be fed to the acoustic model, features are normalized in such a manner that they have zero mean and unit standard deviation.

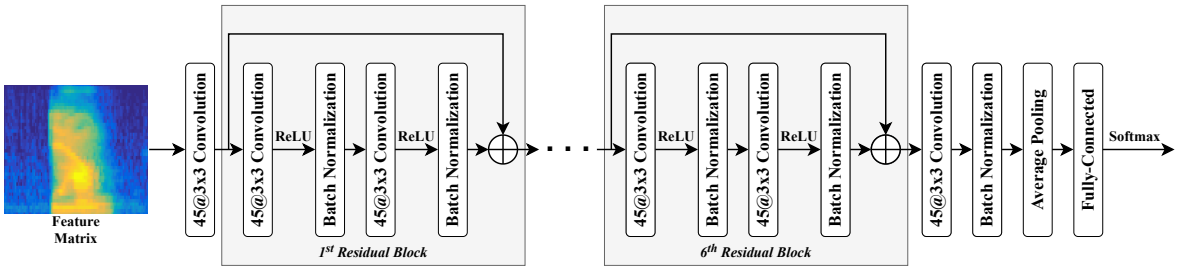

Figure 2: *Deep residual neural network employed for keyword spotting acoustic modeling in this study.*

Table 1: *Keyword spotting accuracy results (%), per-epoch training times (s) and inference times (μs), from using log-Mel and MFCC features, as a function of the number of features. Accuracy values and times are shown along with 95% confidence intervals. The number of multiplications of the acoustic model, which directly depends on the number of features, is also shown. The hop size/number of time frames is 10 ms/101.*

No. of Features	**No. of Mult.**	**Log-Mel**			**MFCC**		
		Accuracy (%)	Training Time per Epoch (s)	Inference Time (μs)	Accuracy (%)	Training Time per Epoch (s)	Inference Time (μs)
40	895M	95.33 ± 0.28	98.7 ± 0.4	980 ± 7	95.24 ± 0.96	96.5 ± 0.4	980 ± 12
20	424M	95.70 ± 0.58	58.3 ± 0.7	590 ± 11	95.55 ± 0.65	55.2 ± 0.4	595 ± 7
10	188M	95.34 ± 0.76	36.2 ± 0.3	390 ± 21	95.24 ± 0.64	36.3 ± 0.3	385 ± 12
5	71M	93.00 ± 0.36	27.3 ± 0.5	292 ± 16	92.60 ± 0.87	26.3 ± 0.5	289 ± 10

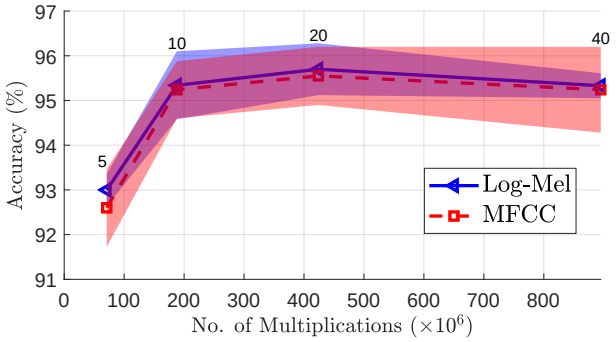

Figure 3: *Keyword spotting accuracy (%) with 95% confidence intervals, from using log-Mel and MFCC features, as a function of the number of multiplications of the acoustic model. The corresponding number of features is indicated above the confidence area. The hop size/number of time frames is 10 ms/101.*

2. *Number of multiplications*: The authors of [10] found a strong positive linear relationship ($R^2 = 0.9641$, $p = 0.0001$) between the number of multiplications of the KWS acoustic model and its energy consumption. Therefore, reducing the number of multiplications can be expected to reduce energy consumption by the same amount.

3. *Training time per epoch*: Per-epoch training time of the KWS acoustic model is measured when model training is run on a GPU NVIDIA GeForce GTX 1080 Ti.

4. *Inference time*: Again using a GPU NVIDIA GeForce GTX 1080 Ti, inference time is defined in this work as the time it takes a forward pass of a one-second long input speech segment.

Except for the number of multiplications, we report results along with 95% confidence intervals calculated from the Student's t-distribution [29]. For accuracy, they are calculated from 5 different CNN models trained with different random weight initialization. For per-epoch training times and inference times, confidence intervals are obtained from the total of training epochs and one-second long test segments, respectively.

4.1. Reducing the Number of Features

Table 1 shows KWS accuracy results, number of model multiplications, per-epoch training times and inference times as a function of the number of features when the hop size/number of time frames[3] is fixed to 10 ms/101. These results were obtained by means of a Mel filterbank spanning the frequency range [20, 8,000] Hz. Preliminary experiments (not reported here) revealed no statistically significant differences between the results in Table 1 and those from employing a Mel filterbank constrained to be in the frequency range [20, 4,000] Hz. Hence, all the experiments reported in this paper consider a Mel filterbank spanning 20-8,000 Hz.

From Table 1, we can see a strong positive linear relationship between the number of multiplications of the model, the training and inference times, and the number of features. Furthermore, similar performance is provided by using either log-Mel or MFCC features. Interestingly, reducing the number of features from 40 to only 10 produces no statistically significant KWS accuracy differences while a 4.8× number of multiplications/energy consumption, 2.7× training time and 2.5× inference time reduction is achieved. To better assess the extent of the improvement yielded by feature size reduction, Figure 3 depicts KWS accuracy as a function of the number of multiplications of the acoustic models in Table 1.

[3]Recall that the acoustic model is designed to process one-second long speech segments (see Subsection 3.2).

Table 2: *Keyword spotting accuracy results (%), per-epoch training times (s) and inference times (μs), from using log-Mel and MFCC features, as a function of the hop size (ms)/number of time frames. Accuracy values and times are shown along with 95% confidence intervals. The number of multiplications of the acoustic model, which directly depends on the hop size/number of time frames, is also shown. The number of features is 10.*

Hop Size (ms) / No. of Frames	**No. of Mult.**	**Log-Mel**			**MFCC**		
		Accuracy (%)	Training Time per Epoch (s)	Inference Time (μs)	Accuracy (%)	Training Time per Epoch (s)	Inference Time (μs)
10 / 101	188M	95.34 ± 0.76	36.2 ± 0.3	390 ± 21	95.24 ± 0.64	36.3 ± 0.3	385 ± 12
20 / 51	93M	94.63 ± 0.65	24.2 ± 0.5	265 ± 11	94.61 ± 0.89	24.1 ± 0.3	265 ± 9
30 / 34	61M	94.53 ± 0.47	19.2 ± 0.4	216 ± 10	93.50 ± 0.83	19.0 ± 0.4	216 ± 14
40 / 26	46M	93.24 ± 0.50	15.3 ± 0.4	197 ± 9	92.36 ± 0.55	15.2 ± 0.3	201 ± 8

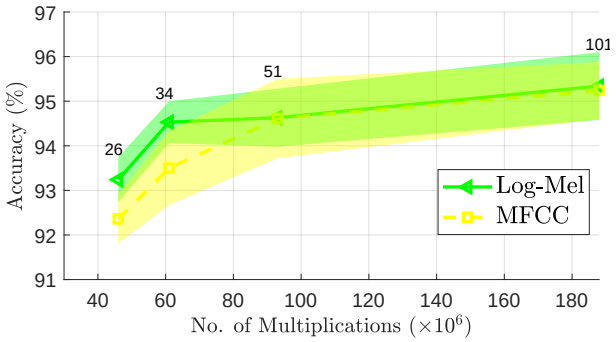

Figure 4: *Keyword spotting accuracy (%) with 95% confidence intervals, from using log-Mel and MFCC features, as a function of the number of multiplications of the acoustic model. The corresponding number of time frames is indicated above the confidence area. The number of features is 10.*

4.2. Increasing the Hop Size

Departing from fixing the number of features to be 10, in this subsection, we study the impact of decreasing the time resolution of the feature matrix by increasing the hop size of the analysis window. Table 2 reports KWS accuracy results, number of model multiplications, per-epoch training times and inference times as a function of the hop size/number of time frames. In addition, to visually inspect the impact of time resolution reduction on KWS and computational performance, Figure 4 plots KWS accuracy as a function of the number of multiplications of the acoustic models in Table 2.

Similarly to Subsection 4.1, from Table 2, we can observe a strong positive linear relationship between the number of model multiplications, the training and inference times, and the number of time frames. Despite using a hop size equal to or larger than 30 ms significantly deteriorates spotting performance, utilizing a 10×51 feature matrix (hop size of 20 ms) does not statistically significantly worsen the KWS accuracy given by any of the evaluated feature matrices with a higher spectro/cepstro-temporal resolution (mind overlapping confidence intervals in Tables 1 and 2). In other words, we can reduce the standard size of both log-Mel and MFCC matrices employed in KWS by a factor of 8 while essentially maintaining spotting performance. In turn, this leads to a notable $9.6\times$ number of multiplications/energy consumption, $4.0\times$ training time and $3.7\times$ inference time reduction.

5. Conclusions

In this paper, we have experimentally studied an indirect way of decreasing the computational complexity of a state-of-the-art CNN acoustic model for KWS (which typically comprise residual connections): the reduction of the spectro/cepstro-temporal resolution of the speech feature matrix. Our experimental results have shown that we can notably reduce the size of standard feature matrices without really hurting KWS performance while achieving a remarkable computational load reduction. We believe that these results very much endorse our previous hypothesis that modern neural network-based KWS acoustic models are fed with much redundant information. And, more importantly, this is an interesting finding to bear in mind when designing light and compact KWS systems that are intended to be embedded on low-resource devices.

6. Acknowledgements

This work was supported, in part, by the Demant Foundation.

7. References

[1] I. López-Espejo, Z.-H. Tan, J. H. L. Hansen, and J. Jensen, "Deep spoken keyword spotting: An overview," *IEEE Access*, vol. 10, pp. 4169–4199, 2021.

[2] Y. Mishchenko, Y. Goren, M. Sun, C. Beauchene, S. Matsoukas, O. Rybakov, and S. N. P. Vitaladevuni, "Low-bit quantization and quantization-aware training for small-footprint keyword spotting," in *Proceedings of ICMLA 2019 – 18th IEEE International Conference on Machine Learning and Applications, December 16-19, Boca Raton, USA*, 2019, pp. 706–711.

[3] D. Peter, W. Roth, and F. Pernkopf, "Resource-efficient DNNs for keyword spotting using neural architecture search and quantization," in *Proceedings of ICPR 2020 – 25th International Conference on Pattern Recognition, January 10-15, Milano, Italy*, 2020.

[4] A. Riviello and J.-P. David, "Binary speech features for keyword spotting tasks," in *Proceedings of INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association, September 15-19, Graz, Austria*, 2019, pp. 3460–3464.

[5] J. Kim, S. Chang, and N. Kwak, "PQK: Model compression via pruning, quantization, and knowledge distillation," in *Proceedings of INTERSPEECH 2021 – 22nd Annual Conference of the International Speech Communication Association, August 30-September 3, Brno, Czechia*, 2021, pp. 4568–4572.

[6] M. Ø. Nielsen, J. Østergaard, J. Jensen, and Z.-H. Tan, "Compression of DNNs using magnitude pruning and nonlinear information bottleneck training," in *Proceedings of MLSP 2021 – 31st IEEE International Workshop on Machine Learning for Signal Processing, October 25-28, Gold Coast, Australia*, 2021, pp. 1–6.

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861v1*, 2017.

[8] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *Proceedings of ICASSP 2018 – 43rd IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Canada*, 2018, pp. 5484–5488.

[9] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, pp. 357–366, 1980.

[10] R. Tang, W. Wang, Z. Tu, and J. Lin, "An experimental analysis of the power consumption of convolutional neural networks for keyword spotting," in *Proceedings of ICASSP 2018 – 43rd IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Canada*, 2018, pp. 5479–5483.

[11] I. López-Espejo, Z.-H. Tan, and J. Jensen, "Exploring filterbank learning for keyword spotting," in *Proceedings of EUSIPCO 2020 – 28th European Signal Processing Conference, January 18-21, Amsterdam, Netherlands*, 2021, pp. 331–335.

[12] C. Shan, J. Zhang, Y. Wang, and L. Xie, "Attention-based end-to-end models for small-footprint keyword spotting," in *Proceedings of INTERSPEECH 2018 – 19th Annual Conference of the International Speech Communication Association, September 2-6, Hyderabad, India*, 2018, pp. 2037–2041.

[13] Y. Bai, J. Yi, J. Tao, Z. Wen, Z. Tian, C. Zhao, and C. Fan, "A time delay neural network with shared weight self-attention for small-footprint keyword spotting," in *Proceedings of INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association, September 15-19, Graz, Austria*, 2019, pp. 2190–2194.

[14] R. Alvarez and H.-J. Park, "End-to-end streaming keyword spotting," in *Proceedings of ICASSP 2019 – 44th IEEE International Conference on Acoustics, Speech and Signal Processing, May 12-17, Brighton, UK*, 2019, pp. 6336–6340.

[15] M. Xu and X.-L. Zhang, "Depthwise separable convolutional ResNet with squeeze-and-excitation blocks for small-footprint keyword spotting," in *Proceedings of INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association, October 25-29, Shanghai, China*, 2020, pp. 2547–2551.

[16] H.-J. Park, P. Violette, and N. Subrahmanya, "Learning to detect keyword parts and whole by smoothed max pooling," in *Proceedings of ICASSP 2020 – 45th IEEE International Conference on Acoustics, Speech and Signal Processing, May 4-8, Barcelona, Spain*, 2020, pp. 7899–7903.

[17] L. Wang, R. Gu, N. Chen, and Y. Zou, "Text anchor based metric learning for small-footprint keyword spotting," in *Proceedings of INTERSPEECH 2021 – 22nd Annual Conference of the International Speech Communication Association, August 30-September 3, Brno, Czechia*, 2021, pp. 4219–4223.

[18] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," in *Proceedings of INTERSPEECH 2021 – 22nd Annual Conference of the International Speech Communication Association, August 30-September 3, Brno, Czechia*, 2021, pp. 4538–4542.

[19] P. Warden, "Speech Commands: A dataset for limited-vocabulary speech recognition," *arXiv:1804.03209v1*, 2018.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR 2016 – Conference on Computer Vision and Pattern Recognition, June 26-July 1, Las Vegas, USA*, 2016, pp. 770–778.

[21] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in Python," 01 2015, pp. 18–24.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of ICLR 2015 – 3rd International Conference on Learning Representations, May 7-9, San Diego, USA*, 2015.

[23] N. Gershenfeld, "An experimentalist's introduction to the observation of dynamical systems," in *Directions in Chaos — Volume 2*. World Scientific, 1988, pp. 310–353.

[24] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[25] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[26] Google Developers, "Machine Learning Crash Course - Classification: Accuracy," https://developers.google.com/machine-learning/crash-course/classification/accuracy.

[27] I. López-Espejo, Z.-H. Tan, and J. Jensen, "Improved external speaker-robust keyword spotting for hearing assistive devices," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1233–1247, 2020.

[28] ——, "A novel loss function and training strategy for noise-robust keyword spotting," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2254–2266, 2021.

[29] N. Blachman and R. Machol, "Confidence intervals based on one or more observations," *IEEE Transactions on Information Theory*, vol. 33, pp. 373–382, 1987.