



Audio Engineering Society

Convention Express

Paper 387

Presented at the AES 159th Convention
2025 October 23–25, Long Beach, CA, USA

This Express Paper was selected on the basis of a submitted synopsis that has been peer-reviewed by at least two qualified anonymous reviewers. The complete manuscript was not peer reviewed. This Express Paper has been reproduced from the author's advance manuscript without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. This paper is available in the AES E-Library (<http://www.aes.org/e-lib>), all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Efficient Few-Shot Keyword Spotting for Edge Devices Using Self-Supervised Learning

Osman Erman Okman¹, Alican Gok¹, and Oguzhan Buyuksolak¹

¹Analog Devices, Istanbul

Correspondence should be addressed to Osman Erman Okman (erman.okman@analog.com)

ABSTRACT

Keyword spotting (KWS) is a foundational technology for enabling voice-driven user interfaces across a wide range of applications, including smart assistants, IoT devices, and wearables. Traditional KWS systems require large amounts of labeled data and are often limited to a fixed set of keywords, making them less adaptable to dynamic user needs and new commands. Few-shot keyword spotting (FS-KWS) systems address these limitations by enabling the detection and registration of new keywords from only a handful of examples, thus supporting real-time integration of new commands without the need for extensive retraining. Despite these advances, deploying accurate FS-KWS models on resource-constrained devices remains a significant challenge due to strict limitations on computational power, memory, and energy consumption. In this work, we adapt a suite of lightweight neural network architectures originally designed for KWS, for FS-KWS in resource-limited environments. Our models are trained using a knowledge distillation framework that leverages self-supervised learning (SSL) models to generate compact and highly discriminative speech embeddings. This approach enables the transformation of speech segments into lower-dimensional representations, facilitating efficient and robust keyword detection even with limited data. Experiments on benchmark KWS datasets show that by use of our training approach, the lightweight model architectures performs on par with larger and computationally more demanding architectures for FS-KWS applications. Our findings underscore the potential of combining knowledge distillation and SSL-based embeddings to advance FS-KWS, paving the way for practical, scalable, and adaptive voice interfaces in next-generation smart devices.

1 Introduction

Few-shot keyword spotting (FS-KWS) is an emerging paradigm within the broader field of speech recognition and audio processing. At its core, FS-KWS addresses the challenge of detecting specific spoken keywords or phrases in audio streams when only a handful of labeled

examples are available for each keyword. This stands in stark contrast to traditional KWS systems, which typically require extensive labeled datasets and significant retraining to accommodate new or user defined keywords [1].

The motivation for FS-KWS arises from the increasing demand for flexible, adaptive, and personalized voice

interfaces in modern technology. As voice activated systems become common across consumer electronics, smart devices, industrial automation, and assistive technologies, the ability to rapidly adapt to new keywords, languages, and user preferences without the burden of large scale data collection and retraining has become paramount. The practical impact of FS-KWS is evident in a wide array of real-world applications. In addition to personalized voice assistants and smart devices, FS-KWS is being deployed in industrial settings for hands-free machinery control, in healthcare for personalized assistive technologies, and in data mining to search rare keywords in large audio archives [2, 3]. Its ability to support multilingual and low-resource environments further underscores its importance in making voice interfaces more accessible and inclusive.

The technical foundation of FS-KWS is rooted in the few-shot learning paradigm, which seeks to generalize from very limited data by leveraging prior knowledge and advanced machine learning techniques. Early and seminal approaches in FS-KWS have drawn heavily from metric learning frameworks [4], such as prototypical networks, which map audio samples into an embedding space where samples of the same keyword cluster together and are well separated from others [5]. This enables the system to recognize new keywords by comparing the embeddings of test samples to those of a few labeled examples, facilitating rapid adaptation with minimal data [6]. However, there are limited number of studies in this new field especially for edge compatible use cases. Rusci et. al. presented performance of different model training procedures as well as self-learning with Triplet loss [7, 8]. They use ResNet15 and Depthwise Seperable Convolutiona Neural Network (DS-CNN) architectures, but there is still a gap to fulfill accuracy expectation at desired false acceptance rates.

Recent advances in FS-KWS have been propelled by the adoption of self-supervised learning (SSL) models, such as Wav2Vec 2.0 [9] and HuBERT [10], which are pre-trained on large, unlabeled speech corpora and provide robust, generalizable feature representations. [11] proposes to use transfer learning for KWS where the backbone network based on Wav2Vec 2.0 and the transfer learning network includes an encoder to convert encoded speech representations to keyword attributes. In one of our recent studies, we proposed using pre-trained SSL models to train lightweight, efficient models suitable for deployment on resource-

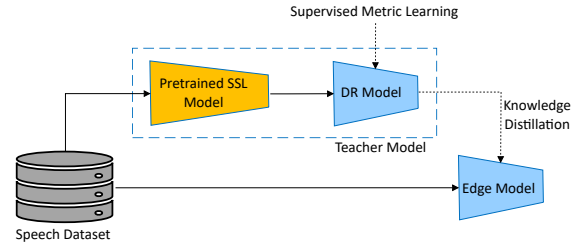


Fig. 1: Proposed approach for edge FS-KWS model training

constrained edge devices without sacrificing accuracy using knowledge distillation techniques [12].

In this study, we extend our research to analyze the performance of different edge friendly model architectures to prove the extensibility of the knowledge distillation approach and to compare performances of different lightweight model architectures for FS-KWS applications.

2 Methodology

This section describes the details of the model architectures used and the training procedure we follow throughout this study.

In this paper, we adopted the same FS-KWS model training scheme from [12] that combines a pretrained SSL model with an alternative metric learning objective. Specifically, our training scheme given in Fig. 1, leverages knowledge distillation with the SSL-based teacher network.

2.1 Edge-Friendly Model Architectures

2.1.1 ResNet15

Residual Network (ResNet)[13] is a type of convolutional neural network (CNN) architecture, which introduces the use of residual blocks, which include “skip connections” that allow gradients to bypass layers during training. Despite the model is widely used for vision applications, it is also common to solve different audio problems. ResNet15, given in Fig.2(a), is a lightweight version of this architecture and widely used as a baseline for KWS on edge devices [8, 14].

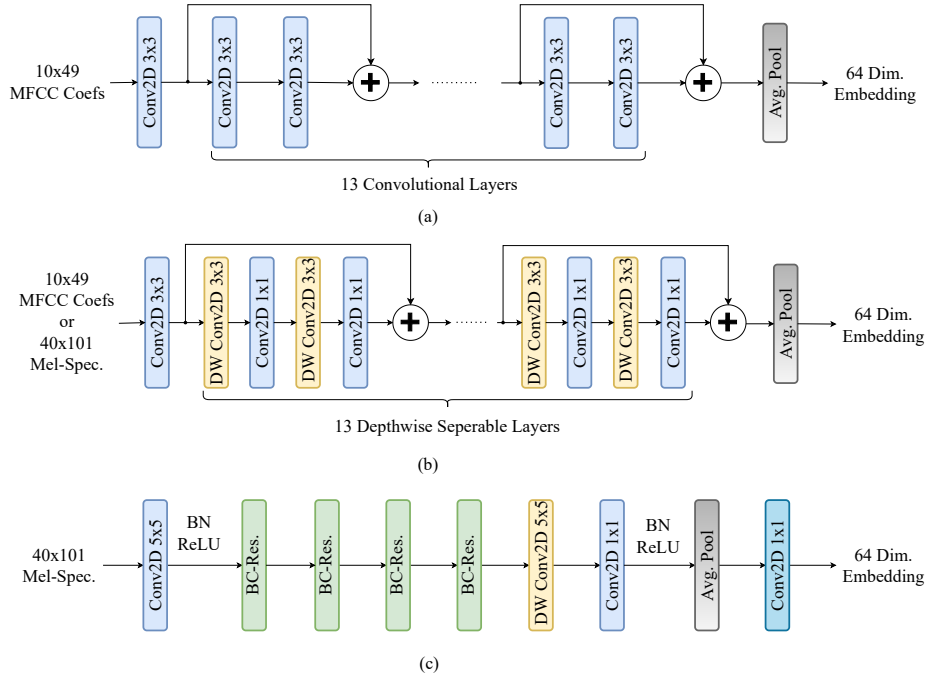


Fig. 2: Lightweight model architectures: (a) ResNet-15 , (b) depthwise seperable ResNet-15 (ResNet15-DW), (c) broadcasted residual connection network (BC-ResNet)

2.1.2 Depthwise Seperable(DW)-ResNet15

This architecture is a variant of ResNet15 where standard convolutions are replaced with depthwise separable convolutions (see Fig.2(b)); a depthwise convolution followed by a pointwise convolution. The motivation of this modification is significantly reduce the parameter count and MACs while aiming to preserve the accuracy. Note that, our initial experiments show that the input characteristics affect the overall accuracy of the model, so the experiments cover both for MFCC and Mel Filterbank inputs.

2.1.3 BC-ResNet

As there are few proposed efficient model architectures in FS-KWS literature, we also adopt an architecture designed for fixed vocabulary KWS. Broadcasted residual network (BC-ResNet) is an efficient state-of-the-art model that combines 1D temporal convolutions with 2D convolutions via broadcasted residual connections, defined as BC-ResBlock, to capture both temporal and frequency features [15]. In order to adapt the model to

the FS-KWS setting, we replaced the final classification layer of the original architecture with a Conv2D layer with an output size of 64, to be consistent with the embedding size of other tested models. The resulting architecture is shown in Fig. 2(c). We evaluate BC-ResNet at multiple scaling factors ($\tau = \{1, 2, 3, 4\}$) to explore the trade-off between accuracy and computational efficiency. Note that, the scaling factor of the model is selected aiming the number of parameters to be less than 100k.

2.2 Model Training

The student models (ResNet15, Depthwise Seperable ResNet15, and BC-ResNet) are trained to mimic the teacher's embeddings using a combination of mean squared error (MSE) distillation loss and Sub-center ArcFace (SCAF) loss [16] loss. This dual-objective training encourages the student to both approximate the teacher's representation and maintain discriminative power for FS-KWS.

The teacher model is based on Wav2Vec 2.0, a state-of-the-art SSL model that has achieved superior performance in various audio processing tasks. As the

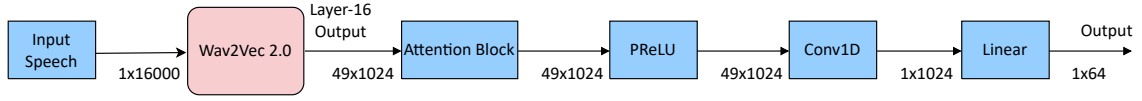


Fig. 3: Teacher model architecture based on Wav2Vec 2.0 and dimension reduction layers

model includes different level of information at different layers, the ultimate output of the model does not include the purest information for keyword spotting task. The feature extraction process involves obtaining high-dimensional embeddings from the 16th transformer layer of a pre-trained Wav2Vec 2.0 model. This specific layer selection is based on comprehensive empirical analysis conducted by Sanabria et al.[17], who demonstrated optimal representational characteristics at this depth across multiple SSL architectures.

At this stage, the model processes the audio with a stride of 20ms; therefore, it produces a 49×1024 feature matrix for each 1-second audio signal. However, processing such amounts of data at edge processors is not practical. Following the method in [12], we use an attention-based dimension reduction layer, to transform the feature matrix to a single 64-dimensional embedding for each 1-second sample. The resultant teacher model architecture is given in Fig.3 and it is trained via knowledge distillation using a combination of MSE and SCAF losses [12].

3 Results

3.1 Datasets

To train the student networks optimized for edge devices, we utilize the training split of the English section from the Multilingual Spoken Words Corpus (MSWC) [18]. The dataset includes 5.5 million one-second audio samples spanning 39,000 unique words. For evaluation, we employ the test split of the MSWC, comprising 700,000 samples from 8,900 distinct words, as well as the Google Speech Commands (GSC) dataset [19], consisting of 100,000 one-second clips representing 35 different short commands.

3.2 Testing Protocol

To measure the effectiveness of our few-shot keyword spotting (FS-KWS) models, we implement an evaluation methodology consistent with the protocol described in [7]. In the enrollment stage, K utterances

per target keyword are processed by the frozen feature encoder, and the resulting embeddings are averaged to establish a prototype for each class. During inference, the system extracts features from a test utterance and computes the cosine similarity between this embedding and all class prototypes. If the minimum distance falls below a predefined threshold T , the sample is classified as the corresponding keyword; otherwise, it is assigned to the “others” category. By systematically varying T , we can explore the balance between recognition accuracy and the false acceptance rate (FAR) for out-of-vocabulary inputs, across multiple K -shot configurations.

For the MSWC dataset, we emulate a wake-word detection scenario by isolating individual keywords. Each experimental iteration involves selecting K examples from a randomly chosen keyword in the test partition to form its prototype, followed by evaluation against one positive (same keyword) and one negative (different keyword) sample. An experiment batch is obtained by repeating this step for 100,000 times, independently. To ensure statistical robustness, we report the results by averaging 3 experiment batches.

For the Google Speech Commands (GSC) dataset, we adapt the KWS-12 evaluation framework [19] to the few-shot context. The classification task involves distinguishing among 11 predefined keywords (*on, off, left, right, up, down, go, stop, yes, no* and silence), while the remaining 25 classes are treated as “others”. The silence class includes recordings with different types of background noise. Please note that the inclusion of the silence class makes the tests more challenging compared to the work in [7]. For prototype creation during enrollment, K samples per target class are drawn from the training set. The entire test set then is evaluated against these prototypes. To account for variability in few-shot sample selection, each experiment is conducted 100 times, with final results reported as the average across all runs.

Table 1: Model performance on MSWC (1-/10- shot) and GSC (1-/10- shot). Each cell reports $\text{mean}_{1\text{-shot}}/\text{mean}_{10\text{-shot}}$ with std as a tiny superscript.

	MSWC			GSC				
	$DET_1\%$	$DET_5\%$	AUROC	$ACC_1\%$	$ACC_5\%$	AUC		
Model	1-/10- shot			1-/10- shot			#MACs	#Params
Teacher [12]	91.4 $\pm$ 0.03/97.0 $\pm$ 0.09	97.4 $\pm$ 0.02/99.3 $\pm$ 0.05	99.3 $\pm$ 0.01/99.8 $\pm$ 0.01	65.1 $\pm$ 7.6/83.0 $\pm$ 0.5	75.1 $\pm$ 6.9/85.2 $\pm$ 0.5	82.7 $\pm$ 5.7/90.9 $\pm$ 0.7	63.3* G	217.8* M
ResNet15 [12]	86.3 $\pm$ 0.13/ 96.1 $\pm$ 0.05	93.9 $\pm$ 0.03/98.4 $\pm$ 0.02	98.3 $\pm$ 0.01/99.5 $\pm$ 0.01	44.3 $\pm$ 7.4/ 75.4 $\pm$ 1.4	60.6 $\pm$ 8.0/ 85.2 $\pm$ 0.8	73.7 $\pm$ 6.9/ 91.0 $\pm$ 0.5	235 M	480 k
ResNet15-DW-MFCC	74.5 $\pm$ 0.25/87.4 $\pm$ 0.07	87.4 $\pm$ 0.07/96.3 $\pm$ 0.06	96.7 $\pm$ 0.02/99.1 $\pm$ 0.01	24.1 $\pm$ 5.2/55.3 $\pm$ 2.5	40.5 $\pm$ 6.4/69.1 $\pm$ 2.2	58.9 $\pm$ 6.1/81.2 $\pm$ 0.2	30 M	62 k
ResNet15-DW-MelFB	80.4 $\pm$ 0.27/94.6 $\pm$ 0.01	91.7 $\pm$ 0.01/98.1 $\pm$ 0.04	98.0 $\pm$ 0.01/99.5 $\pm$ 0.01	39.5 $\pm$ 6.1/73.0 $\pm$ 0.2	57.4 $\pm$ 6.7/84.5 $\pm$ 2.2	72.5 $\pm$ 6.2/90.4 $\pm$ 0.1	254 M	62 k
BC-ResNet-1	61.2 $\pm$ 0.08/81.2 $\pm$ 0.05	85.3 $\pm$ 0.08/95.5 $\pm$ 0.13	97.0 $\pm$ 0.01/98.9 $\pm$ 0.01	14.1 $\pm$ 3.9/35.6 $\pm$ 2.7	27.3 $\pm$ 5.6/52.9 $\pm$ 2.5	48.5 $\pm$ 6.6/71.7 $\pm$ 1.7	2.5 M	10.9 k
BC-ResNet-2	77.3 $\pm$ 0.17/91.1 $\pm$ 0.09	93.3 $\pm$ 0.14/98.2 $\pm$ 0.03	98.5 $\pm$ 0.01/99.5 $\pm$ 0.01	25.8 $\pm$ 5.4/53.4 $\pm$ 2.5	42.0 $\pm$ 6.8/69.0 $\pm$ 1.6	62.9 $\pm$ 6.7/83.1 $\pm$ 0.9	7.3 M	30.6 k
BC-ResNet-3	83.5 $\pm$ 0.03/94.0 $\pm$ 0.18	95.5 $\pm$ 0.05/98.8 $\pm$ 0.04	98.9 $\pm$ 0.01/99.6 $\pm$ 0.01	36.8 $\pm$ 7.0/68.4 $\pm$ 1.4	56.6 $\pm$ 8.0/79.7 $\pm$ 0.8	73.7 $\pm$ 7.1/89.2 $\pm$ 0.5	14.5 M	59.2 k
BC-ResNet-4	87.1 $\pm$ 0.26/95.5 $\pm$ 0.08	96.4 $\pm$ 0.05/ 99.1 $\pm$ 0.03	99.1 $\pm$ 0.01/ 99.7 $\pm$ 0.01	44.5 $\pm$ 7.6/73.7 $\pm$ 1.2	60.5 $\pm$ 7.8/82.1 $\pm$ 0.6	74.3 $\pm$ 7.1/90.6 $\pm$ 0.4	24.1 M	96.6 k

* The teacher model consists of a Wav2vec2 encoder up to the 16th transformer layer and the dimensionality reduction module.

3.3 Results

Table 1 presents the outcomes of the 1-shot and 10-shot experiments conducted on the MSWC and GSC datasets, following the evaluation protocol described in the previous section. The table reports multiple performance metrics for each student model evaluated, as well as for the teacher model, which serves as a reference. Additionally, model sizes and the number of multiply-accumulate operations (MACs) required per inference are included, enabling a direct comparison of the computational efficiency and resource requirements among the various lightweight architectures. This comprehensive summary facilitates an assessment of both the accuracy and practical deployability of the proposed models in resource-constrained environments.

The results indicate that ResNet15 and BC-ResNet-4 achieve the highest accuracies on the MSWC and GSC datasets. Notably, BC-ResNet-4 either closely matches or exceeds the performance of ResNet15 across all evaluated metrics, particularly outperforming it in cross-domain evaluations on the GSC dataset. This is achieved while requiring only one-tenth of the computational operations and one-fifth of the model size compared to ResNet15, highlighting BC-ResNet-4 as the most effective model in terms of the performance-to-complexity trade-off.

The depthwise separable variant of ResNet15 was implemented to reduce its model size. However, this approach resulted in substantial performance degradation when using MFCC features as input, which is ResNet15's original frontend. Instead, employing a Mel filterbank (MelFB) frontend with the same architecture improved performance, and bridged most of

the performance gap to the original ResNet15, but this came at the cost of significantly increased computational requirements due to the larger input dimensions (from 10×49 to 40×101). In contrast, BC-ResNet-3, which has a parameter count comparable to the ResNet15-DW, delivers similar results on the GSC dataset and surpasses it on MSWC. This is accomplished with a much lower computational cost (15.4M vs. 254M MACs), demonstrating that the efficiency advantages reported for BC-ResNet in fixed-vocabulary settings [15] also extend to few-shot scenarios.

Figure 4 illustrates the relationship between accuracy and computational cost (MACs) for the evaluated edge models for the 10-shot setting on the GSC dataset. The BC-ResNet models, across different scales, consistently form the Pareto front, offering superior accuracy with reduced parameter and compute requirements, or achieving comparable accuracy at lower computational cost relative to the ResNet architectures.

These set of results can guide the practitioners in selecting the model architecture, depending on the constraints of the target hardware. For applications where minimizing parameter count is paramount while maintaining high performance at tight false alarm rates, using ResNet15 with depthwise convolutions and a mel-filterbank frontend could be beneficial, as it is slightly ahead of BC-ResNet-3 with a similar parameter count, as evidenced by the GSC results. Conversely, if computational efficiency is the primary concern, selecting an appropriately scaled BC-ResNet model would be highly recommended.

Consistent with prior studies on fixed-vocabulary keyword spotting [14, 7], ResNet15 demonstrates strong

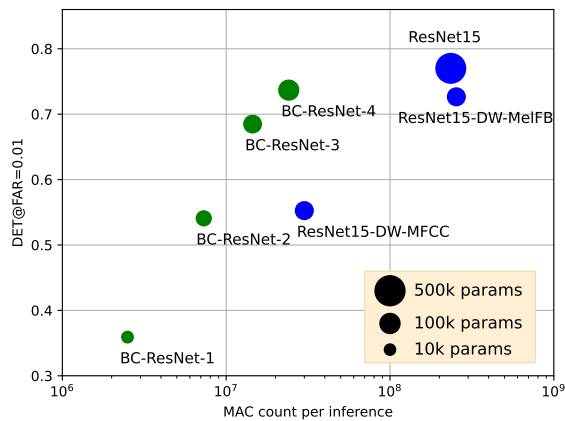


Fig. 4: Performance of edge compatible models on GSC for the 10-shot setting vs model size and required computations per inference. The dot sizes denote parameter count.

performance in open-set few-shot keyword spotting. However, its memory and computational demands are substantially higher than those of alternative models. Despite being about three orders of magnitude smaller than the teacher model in both parameter count and computational requirements, the best-performing student models closely approach the teacher’s results on both datasets, particularly at more relaxed false alarm rates on GSC. This outcome underscores the effectiveness of the training methodology described in [12].

4 Conclusion

We have presented a comprehensive comparison of three leading edge-friendly neural network architectures trained using a unified teacher-student framework for few-shot keyword spotting. Our results demonstrate that BC-ResNet, particularly at moderate scaling factors, achieves the best trade-off between accuracy and computational efficiency, making it ideally suited for real-world edge deployment. Despite a higher computational cost, the depthwise version of ResNet15 is also a strong contender if parameter count is the primary concern. These results may help practitioners looking to deploy FS-KWS systems on different resource-constrained edge hardware.

Future work will focus on optimizing the edge model architecture to minimize the performance gap between the teacher model. Further reducing the model size

and computational cost through quantization and pruning will enable deployment on even more constrained devices.

References

- [1] Yuan, J., Shi, Y., Li, L., Wang, D., and Hamdulla, A., “Few-Shot Keyword Spotting from Mixed Speech,” in *Proc. Interspeech 2024*, pp. 5063–5067, 2024.
- [2] Saeed, A. and Tsouvalas, V., “Plug-and-Play Multilingual Few-shot Spoken Words Recognition,” *arXiv preprint arXiv:2305.03058*, 2023.
- [3] Garai, S. and Samui, S., “Advances in Small-Footprint Keyword Spotting: A Comprehensive Review of Efficient Models and Algorithms,” *arXiv preprint arXiv:2506.11169*, 2025.
- [4] Mazumder, M., Banbury, C., Meyer, J., Warden, P., and Reddi, V. J., “Few-Shot Keyword Spotting in Any Language,” in *Interspeech 2021*, pp. 4214–4218, 2021.
- [5] Kim, B., Yang, S., Chung, I., and Chang, S., “Dummy Prototypical Networks for Few-Shot Open-Set Keyword Spotting,” in *Interspeech 2022*, pp. 4621–4625, 2022.
- [6] Lee, D., Kim, M., Mun, S. H., Han, M. H., and Kim, N. S., “Fully Unsupervised Training of Few-Shot Keyword Spotting,” in *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 266–272, 2023.
- [7] Rusci, M. and Tuytelaars, T., “On-Device Customization of Tiny Deep Learning Models for Keyword Spotting With Few Examples,” *IEEE Micro*, 43(6), pp. 50–57, 2023.
- [8] Rusci, M. and Tuytelaars, T., “Few-Shot Open-Set Learning for On-Device Customization of KeyWord Spotting Systems,” in *Proc. Interspeech 2023*, pp. 2768–2772, 2023.
- [9] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M., “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” in *Advances in Neural Information Processing Systems*, volume 33, pp. 12449–12460, 2020.

- [10] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A., “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM transactions on audio, speech, and language processing*, 29, pp. 3451–3460, 2021.
- [11] Seo, D., Oh, H.-S., and Jung, Y., “Wav2KWS: Transfer Learning From Speech Representations for Keyword Spotting,” *IEEE Access*, 9, pp. 80682–80691, 2021.
- [12] Gok, A., Buyuksolak, O., Okman, O. E., and Saraclar, M., “Enhancing Few-shot Keyword Spotting Performance through Pre-Trained Self-supervised Speech Models,” *arXiv preprint arXiv:2506.17686*, 2025.
- [13] He, K., Zhang, X., Ren, S., and Sun, J., “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [14] Vygon, R. and Mikhaylovskiy, N., “Learning Efficient Representations for Keyword Spotting with Triplet Loss,” in *Speech and Computer: 23rd International Conference, SPECOM 2021 Proceedings*, p. 773–785, 2021.
- [15] Kim, B., Chang, S., Lee, J., and Sung, D., “Broadcasted Residual Learning for Efficient Keyword Spotting,” in *Proc. Interspeech 2021*, pp. 4538–4542, 2021.
- [16] Deng, J., Guo, J., Liu, T., Gong, M., and Zafeiriou, S., “Sub-center ArcFace: Boosting Face Recognition by Large-Scale Noisy Web Faces,” in *European Conference on Computer Vision (ECCV)*, pp. 741–757, 2020.
- [17] Sanabria, R., Tang, H., and Goldwater, S., “Analyzing Acoustic Word Embeddings from Pre-Trained Self-Supervised Speech Models,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.
- [18] Mazumder, M., Chitlangia, S., Banbury, C., Kang, Y., Ciro, J. M., Achorn, K., Galvez, D., Sabini, M., Mattson, P., Kanter, D., et al., “Multilingual spoken words corpus,” in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- [19] Warden, P., “Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition,” 2019, arXiv:1804.03209.