Few-Shot Open-Set Keyword Spotting with Multi-Stage Training

Lo-Ya Li*, Tien-Hong Lo†, Jeih-Weih Hung‡, Shih-Chieh Huang§ and Berlin Chen¶
* Institute of AI Interdisciplinary Applied Technology, National Taiwan Normal University, Taiwan
E-mail: 612k0010c@ntnu.edu.tw
†¶ Department of Computer Science and Information Engineering, National Taiwan Normal University, Taiwan
E-mail: {teinhonglo, berlin}@ntnu.edu.tw
‡ Department of Electrical Engineering, National Chi Nan University, Taiwan
E-mail: jwhung@ncnu.edu.tw
§ Realtek Semiconductor Corp., Taiwan

***Abstract*—As the advance of human-computer interaction technologies continued, keyword spotting (KWS) systems have gained prominence in everyday devices. This study is dedicated to exploring innovative approaches for few-shot keyword recognition under open-set conditions, a challenging yet crucial area in speech processing. To this end, we design and develop a multi-stage training method that synergistically combines the advantages of acoustic and phonetic features, thereby substantially enhancing the ability of a KWS model. By learning multi-type features with joint training from only one dataset, our KWS model is equipped with a more robustness feature extractor to deal with few-shot KWS. Experimental results demonstrate that our model outperforms strong baselines by achieving a 15% improvement in recognition accuracy on open-set tests in a 10shot-10way setting. This research confirms the effectiveness of our multi-stage strategy and suggests promising directions for future development in keyword recognition technologies.**

Index Terms: Keyword spotting, few-shot learning

I. Introduction

Keyword spotting (KWS) is a pivotal technology for human-machine interaction, enabling seamless control of personal and household devices during multitasking activities such as driving and exercising. This technology can be categorized into two primary types: fixed vocabulary KWS [1], [2] and flexible customized KWS [3]–[6]. While traditional fixed vocabulary KWS methods have performed robustly in data-rich scenarios, they have faced significant challenges in environments with limited data. Moreover, the demand for flexible KWS systems has surged recently, driven by an increasing need for more personalized voice-activated control devices and applications. To address the issue of data scarcity in user-defined keywords, numerous studies have proposed various approaches employing few-shot learning (FSL) techniques in KWS, aiming to achieve effective recognition of keywords with minimal speech input from users.

The advent of few-shot learning (FSL) provided a new direction for KWS research, particularly for customized applications. Studies have explored the use of few-shot learning techniques to train KWS models that could adapt to new keywords with minimal user input. These studies primarily used large pre-trained datasets to extract generalizable features that could be fine-tuned with a small number of examples [7]. Recent advancements have focused on enhancing the robustness of feature extraction in FSL settings by integrating acoustic features with linguistic data. For instance, some models have been developed to combine traditional acoustic features with phoneme-based information, achieving a balance between generalization and specificity. This approach facilitated more accurate predictions of user-defined keywords by aligning audio features with their corresponding phonetic sequences [8].

In addition to leveraging phoneme-based information, the concept of prototype networks (ProtoNets) [9] has been crucial in the evolution of FSL for KWS. ProtoNets compute class prototypes as the mean of feature vectors, significantly simplifying the adaptation process to new and rare keywords. However, recent work extended this concept by incorporating standard deviation measures into the ProtoNet calculations, capturing a wider range of intra-class feature variability and enhancing the model's ability to handle ambiguously represented classes [10], [11].

To further improve the performance of few-shot keyword spotting (FS-KWS), our research has focused on using a multi-stage strategy. Previous studies on FS-KWS predominantly leveraged prior knowledge from extensively annotated pre-training datasets to develop a robust feature extractor, often depending exclusively on audio sample features or on a combination of phoneme features and labeled text information for alignment purposes [12].

In our approach, we proposed a multi-stage framework [13]–[15] to explore the synergy between acoustic and phonetic features, constructing a more sophisticated and robust feature extractor to enhance FS-KWS performance. Specifically, we leveraged the dynamic properties of the acoustic spectrum—such as spectral energy variations—and the articulatory characteristics inherent in phoneme sequences. To achieve this integration, we employed two distinct models: a feature extractor model focused on deriving robust representations from acoustic features, and an additional Transformer-based model that predicted the phoneme sequence of a keyword. This phoneme sequence [16], [17] was then matched against a text-derived phoneme sequence to assist the main feature extractor model. Both models were trained jointly using a shared loss function, which encouraged them to learn complementary information and improved their generalization capabilities. By integrating these diverse feature sets, our method not only captured a broad spectrum of speech nuances but also adapted more effectively to new keywords and varied acoustic environments. This dual-model approach allowed for a richer and more adaptive learning process, resulting in improved performance across standard keyword spotting benchmarks.

Building on the aforementioned innovations, this paper presented a multi-stage framework for FSL architectures consisting of a feature encoder and a prototype-based open-set classifier. This classifier was initialized with few-shot open-set samples. We utilized the recent Multilingual Spoken Words Corpus (MSWC) dataset [18], from which we obtained both acoustic and textual information. Our methodology employed two distinct models for joint training to develop a robust feature extractor. Specifically, the feature extractor model used the ConvMixer architecture [19], [20], optimized with triplet loss, while the phoneme model utilized a Transformer architecture [21], [22] coupled with Connectionist Temporal Classification (CTC) loss. During each epoch, the losses from both models were aggregated and recursively applied in a joint training process. Upon completing the training of this powerful feature extractor model, we used the Google Speech Commands (GSC) v2 dataset to define prototypes for each category based on a few-shot, few-way setup. This approach not only simplified the adaptation to new or rare categories but also significantly enhanced the model's ability to generalize from a very limited number of examples. This made it particularly effective for applications such as keyword identification, where the ability to quickly adapt to new user-defined keywords was critical.

II. Method

In this section, we describe our approach to classifying speech commands using a multi-stage process. Our method consisted of three main stages: pretrain, prototype setting, and inference. Each stage played a crucial role in building and utilizing the model for accurate speech command classification. The complete architecture in Fig.1.

A. Stage 1: Pretrain

We utilized the MSWC dataset for pretraining. This dataset contained single-word audio files in multiple languages along with their corresponding text transcriptions.

First, we extracted features from the audio data. This step involved transforming raw audio signals into higher-dimensional feature vectors that better captured key information in the audio. We employed three different feature extractor models for this purpose:

a) Depthwise Separable Convolutional Neural Network (DSCNN): DSCNN [23] effectively extracted audio features using depthwise separable convolutions, reducing model parameters and computational complexity.

b) Broadband Convolutional Residual Network (BC_ResNet): This model [24], based on residual networks, was specifically designed to process audio features. It improved the training of deep networks through residual connections.

c) Convolutional Mixer (ConvMixer): ConvMixer [25] combined convolutional and mixing layers to extract multi-scale features, enhancing the model's ability to perceive different audio patterns.

The extracted audio features were used to compute triplet loss (1) [26]. A distance metric learning method that maximizes the distance between positive x_i^+ and negative x_i^- samples while minimizing the distance between positive samples and the anchor. This helped improve the discriminative ability of the feature extractor, ensuring that similar audio files were closer in the feature space.

$$\mathcal{L}_{TL} = -\frac{1}{B_t}\sum_{i=1}^{B_t}\max(\delta) \tag{1}$$

$$\delta = \big(0, d_{L2}(x_i, x_i^+) - d_{L2}(x_i, x_i^-) + \text{margin}\big) \tag{2}$$

Simultaneously, we utilized audio-to-phoneme and text-to-phoneme conversion models to convert the single-word audio from the MSWC dataset into corresponding phoneme sequences. For example, the word "Hello" was converted to ['HH', 'EH1', 'L', 'OW0']. To compare the phoneme sequences predicted by the model with the target phoneme sequences, we used CTC loss (3). It's suitable for aligning input and output sequences of variable lengths.

$$\mathcal{L}_{CTC} = -\log P(y|x) \tag{3}$$

where $P(y|x)$ is the probability of the target sequence y given the input sequence x. Then the losses from both models were aggregated and recursively applied in a joint training process. The total loss (4) as follows:

$$\mathcal{L}_{TOTAL} = \mathcal{L}_{TL} + \mathcal{L}_{CTC} \tag{4}$$

B. Stage 2: Set Prototype

In this stage, we used the GSC v2 dataset. This dataset contained a large number of speech samples, each belonging to a specific speech command category. The purpose of this stage was to extract feature prototypes for each category from the dataset for use in subsequent classification tasks.

We utilized the feature extractors pretrained in Stage 1 to process the GSC v2 dataset. The parameters of these extractors were frozen and not updated at this stage.

For each speech command category, we computed the mean of its feature vectors $x_{j,i}^S$ as the prototype for that category i. This mean feature vector c_i represented the centroid of the category (5). We also computed the standard deviation σ_i of

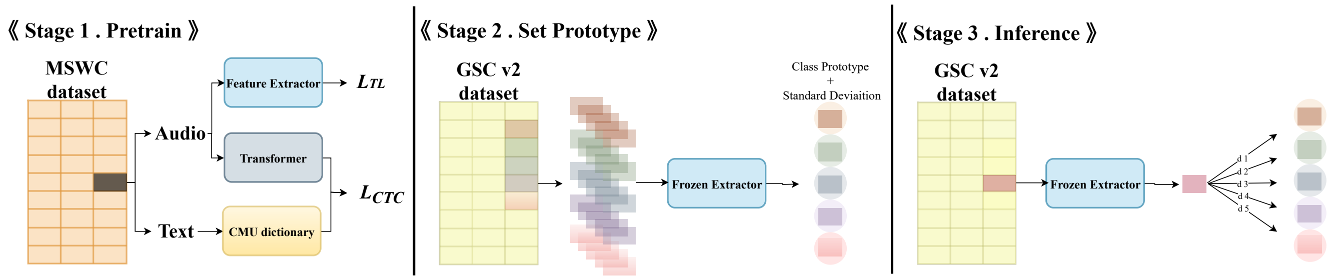

Fig. 1. Overall architecture of the multi-stage training process for few-shot keyword spotting. Used large dataset to pretrain a feature extractor, then GSC dataset to do few shot few way's keyword spotting.

the feature vectors for each category to capture the variability in the feature distribution of that category (6). These measures were used as baselines in subsequent classification tasks.

$$c_i = \frac{1}{S}\sum_{j=1}^{S} f(x_{j,i}^{S}) \tag{5}$$

$$\sigma_i = \sqrt{\frac{1}{S}\sum_{j=1}^{S}\left(f(x_{j,i}^{S}) - c_i\right)^2} \tag{6}$$

C. Stage 3: Inference

The purpose of this stage was to classify speech commands using the class prototypes set in Stage 2. We utilized the feature extractors pretrained in Stage 1 and frozen in Stage 2 to process the GSC v2 dataset. Each audio file was passed through the frozen feature extractors, generating corresponding high-dimensional feature vectors. These feature vectors captured key information in the audio file.

For each extracted feature vector, we calculated its distance to all class prototypes using two distance metrics: Euclidean distance, which refers to the length of the line segment in space connecting these two points. Mahalanobis distance (7) is an effective method to calculate the similarity of two unknown sample sets. Unlike Euclidean distance, it takes into account the connection between various characteristics. Then based on the calculated distances, we assigned the audio file to the class prototype with the smallest distance. This process ensured that each audio file was classified into the most matching category.

$$d_{MA}(\vec{X},\vec{Y}) = \sqrt{(\vec{x}-\vec{y})^T\Sigma^{-1}(\vec{x}-\vec{y})} \tag{7}$$

The objective of the inference stage was to achieve accurate speech command classification using pretrained models and the set class prototypes. This method leveraged the discriminative power of the pretrained feature extractors and the representational power of the class prototypes, making the classification process more reliable and precise.

The multi-stage approach described in this method section aimed to leverage the strengths of deep learning for feature extraction and distance-based classification for speech command recognition. By pretraining robust feature extractors, setting accurate class prototypes, and utilizing efficient distance metrics during inference, our method provided a reliable and precise classification framework.

III. EXPERIMENTS

A. Experimental Setup

a) Datasets: During the pretraining stage, we utilized the MSWC dataset. For each keyword, we selected a subset of samples based on the smallest sample size among all keywords. Specifically, we randomly selected 500 samples for each keyword, except for those from the GSC v2 dataset, resulting in an average of 5,470 samples per class. Additionally, we incorporated background noise from the DEMAND dataset, with Signal-to-Noise Ratio (SNR) values ranging from 0 to 5 dB. For fine-tuning, we employed the GSC v2 dataset, which comprises 35 keywords. From these, we selected 10 keywords (yes, no, up, down, left, right, on, off, stop, go) for our experiments. We conducted a 10-shot 10-way classification, utilizing 10 samples per keyword for training. Moreover, we implemented the openNCM method to incorporate an ”unknown” class for words not included in the selected 10 keywords. The ”unknown” class was comprised of samples from the words ”backward,” ”forward,” ”visual,” ”follow,” and ”learn,” averaging these five words to form the ”unknown” class.

b) models: The DSCNN model had 407k parameters, the BC_ResNet model had 817k parameters, and the ConvMixer model had 119k parameters. The models utilized Mel Frequency Cepstral Coefficients (MFCC) for feature extraction. Audio signals were divided into short frames, each lasting 40 milliseconds, with a window stride of 50%. This configuration resulted in each window overlapping the previous one by 20 milliseconds. The resulting feature map had a size of 49x10, where each time step consisted of 49 frames, and each frame contained 10 MFCC features. The models were trained over 40 epochs, with each epoch consisting of 400 episodes. The Adam optimizer was employed with an initial learning rate of 0.001, and the models were evaluated every 20 episodes using a learning rate decay of 0.1. Furthermore, the models were also evaluated using an online learning method to adapt to unseen classes.

TABLE I
SUMMARY OF MODEL PERFORMANCE ON THE GSC TESTSET UNDER 5,10-SHOT 10-WAY OPEN-SET CLASSIFICATION SETTINGS.

Feature Extractor	Parameters	Transformer	5 shot		10 shot	
			ACC↑	AUROC↑	ACC↑	AUROC↑
TC_ResNet[7]	61k	X	0.52	0.63	0.58	0.67
Ours_TC_ResNet	61k	O	0.61	0.71	0.66	0.73
DSCNN[7]	407k	X	0.71	0.93	0.76	0.94
Ours_DSCNN	407k	O	0.80	0.87	0.91	0.93
Ours_BC_ResNet	817k	O	0.84	0.90	0.92	0.94
Ours_ConvMixer	119k	O	0.83	0.87	0.93	0.92

TABLE II
DIFFERENCES IN ACCURACY PERFORMANCE BETWEEN DISTANCES CALCULATED.

Feature Extractor	Distance Metrics		10 shot
	Euclidean	Mahalanobis	ACC↑ (%)
Ours_BC_ResNet	O		0.86
Ours_BC_ResNet		O	0.92
Ours_ConvMixer	O		0.84
Ours_ConvMixer		O	0.93

B. Experimental Results

Table 1 included the results for all the experiments. We initially utilized the TC_ResNet and DSCNN model's results on few-shot experiments with acoustic features of audio files as a reference baseline. Subsequently, we implemented our proposed method, employing the same dataset and simultaneously extracting both acoustic and phonetic features for joint training to develop a robust feature extractor. Additionally, we introduced the concept of standard deviation in the computation of prototypes and used Mahalanobis distance formulas. By incorporating a multi-stage training approach, the original accuracy increased from 76% to 86%. Further modifications involved changing the definition of few-shot, few-way prototypes and switching to the Mahalanobis distance, which elevated the experimental accuracy up to 91%, thereby achieving superior classification performance. Furthermore, considering the requirement for on-device operation, we opted for the BC_ResNet and ConvMixer feature encoder, which achieved a lower parameter count and also get a higher accuracy.

Table 2 illustrates the various methods of defining the open-set's few-shot prototype concept, as well as the different distances calculated between the test samples and the prototypes. We observed that the same pretrained model, as shown in rows 1 and 2, achieved better accuracy with the Mahalanobis distance when different distance metrics were applied. The table also indicates that averaging few-shot samples into a single vector prototype might result in some imbalanced outcomes when testing with 400 samples per class, which significantly contributes to the final evaluation phase.

IV. CONCLUSIONS

This study has systematically explored the implementation and optimization of keyword spotting systems under open-set conditions using few-shot learning frameworks. Our experimental results get an accuracy to 93%,and also affirm the critical influence of both multi-stage's feature extraction training and the choice of distance metrics on the performance of keyword spotting models. Specifically, the use of MFCC alongside sophisticated models has demonstrated considerable promise in enhancing the robustness and accuracy of keyword recognition.

A significant finding from our research is the superiority of the Mahalanobis distance over conventional Euclidean distance in computing the similarity between test samples and prototypes. This approach not only improved accuracy but also ensured more stable and reliable performance across various testing scenarios. Furthermore, our exploration into prototype averaging methods revealed that single vector representation might introduce imbalances, particularly when a large number of samples per class are used in testing phases. This insight underscores the necessity for adaptive and dynamic prototype calculation methods in few-shot learning environments.

In conclusion, our research contributes valuable insights into the development of adaptable, efficient, and accurate keyword spotting systems, paving the way for further innovations in the field of speech processing technology. Future studies may focus on refining the integration of phonetic features with acoustic signals and exploring the potential of neural network architectures in enhancing the generalizability of keyword spotting systems under diverse operational conditions.

REFERENCES

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

[2] R. Tang and J. J. Lin, "Deep residual learning for small footprint keyword spotting," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5484–5488.

[3] J. Jung, Y. Kim, J. Park, Y. Lim, B. -Y. Kim, Y. Jang and J. -S. Chung, "Metric learning for user-defined keyword spotting," in 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1-5.

[4] L. Lei, G. Yuan, H. Yu, D. Kong and Y. He, ”Multilingual customized keyword spotting using similar-pair contrastive learning,” in 2023 IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 31, pp. 2437-244.

[5] H.-K. Shin, H. Han, D. Kim, S.-W. Chung, and H.-G. Kang, ”Learning audio-text agreement for open-vocabulary keyword spotting,” in 2022 Proc. Interspeech, pp. 1871–1875.

[6] Z. Yang, S. Sun, J. Li, X. Zhang, X. Wang, L. Ma and L. Xie, ”CaTT-KWS: A multi-stage customized keyword spotting framework based on cascaded transducer-transformer,” in 2022 Interspeech, arXiv: 2207.01267.

[7] M. Rusci and T. Tuytelaars, ”Few-shot open-set learning for on-device customization of keyWord spotting systems,” in 2023 Interspeech, pp. 1-5.

[8] Y. Li, C. Yu, G. Sun, H. Jiang, F. Sun, W. Zu, Y. Wen, Y. Yang and J. Wang, ”Cross-utterance conditioned VAE for non-autoregressive text-to-speech,” in 2022 Association for Computational Linguistics (ACL), pp. 1-10.

[9] J. Snell, K. Swersky and R. S. Zemel, ”Prototypical networks for few-shot learning,” in 2017 Advances in neural information processing systems 30, arXiv: 1703.05175.

[10] S. Yang, B. Kim, K. Shim and S. Chang, ”Improving small footprint few-shot keyword spotting with supervision on auxiliary data,” 2023, arXiv: 2309.00647.

[11] W. Lin, X. Tang, X. Han, J. Ma, X. Zhang and L. Jiao, ”NQ-Protonet: Noisy query prototypical network for few-shot remote sensing scene classification,” in 2022 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Kuala Lumpur, Malaysia, pp. 3620-3623.

[12] Y.-H. Lee and N. Cho, ”iPhonMatchNet: Zero-shot user-defined keyword spotting using implicit acoustic echo cancellation,” in 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1-5.

[13] Z. Liu, T. Li, and P. Zhang, ”Rnn-t based open-vocabulary keyword spotting in mandarin with multi-level detection,” in 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5649–5653.

[14] R. Yang, G. Cheng, H. Miao, T. Li, P. Zhang, and Y. Yan, ”Keyword search using attention-based end-to-end asr and frame-synchronous phoneme alignments,” in 2021 IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 3202–3215.

[15] S. Sigtia, J. Bridle, H. Richards, P. Clark, E. Marchi, and V. Garg, ”Progressive voice trigger detection: Accuracy vs latency,” in 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6843–6847.

[16] K. Nishu, M. Cho and D. Naik, ”Matching latent encoding for audio-text based keyword spotting,” 2023, arXiv: 2306.05245.

[17] K. Nishu, M. Cho, P. Dixon and D. Naik, ”Flexible keyword spotting based on homogeneous audio-text embedding,” in 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5050-5054.

[18] M. Mazumder, S. Chitlangia, C. Banbury, Y. Kang, J. M. Ciro, K. Achorn, D. Galvez, M. Sabini, P. Mattson, D. Kanter, et al, ”Multilingual spoken words corpus,” in Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021.

[19] D. Ng, R. Zhang, J.Q. Yip, C. Zhang, Y. Ma, T.H. Nguyen, C. Ni, E.S. Chng and B. Ma, ”Contrastive speech mixup for low-resource keyword spotting,” in 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1-5.

[20] ND. Ng, Y. Chen, B. Tian, Q. Fu and E. S. Chng, ”Convmixer: Feature interactive convolution with curriculum learning for small footprint and noisy far-field keyword spotting,” in 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3603-3607.

[21] B. Axel, O'C. Mark and C. M. Tairum, ”Keyword Transformer: A self-attention model for keyword spotting,” in 2021 Proc. Interspeech, pp. 1-5.

[22] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu et al, “Conformer: Convolution-augmented transformer for speech recognition,” in 2020 Proc. Interspeech, pp. 1-5.

[23] M. Li, C. Ma, W. Dang, R. Wang, Y. Liu and Z. Gao, ”DSCNN: Dilated shuffle CNN model for SSVEP signal classification,” in 2022 IEEE Sensors Journal, vol. 22, no. 12, pp. 12036-12043.

[24] R. Tang and J. Lin, ”Deep residual learning for small-footprint keyword spotting,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, pp. 5484-5488.

[25] D. Ng, Y. Chen, B. Tian, Q. Fu and E. S. Chng, ”Convmixer: Feature interactive convolution with curriculum learning for small footprint and noisy far-field keyword spotting,” in 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3603-3607.

[26] C. Zhang and K. Koishida, ”End-to-end text-independent speaker verification with triplet loss on short utterances,” in 2017 Interspeech, pp. 1487–1491.