INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Few-shot Class-incremental Audio Classification Using Adaptively-refined Prototypes

Wei Xie[1]*,Yanxiong Li*[1]*, Qianhua He*[1]*, Wenchang Cao*[1]*, Tuomas Virtanen*[2]

[1]School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China
[2]Audio Research Group, Tampere University, Tampere, Finland

chester.w.xie@gmail.com, eeyxli@scut.edu.cn

Abstract

New classes of sounds constantly emerge with a few samples, making it challenging for models to adapt to dynamic acoustic environments. This challenge motivates us to address the new problem of few-shot class-incremental audio classification. This study aims to enable a model to continuously recognize new classes of sounds with a few training samples of new classes while remembering the learned ones. To this end, we propose a method to generate discriminative prototypes and use them to expand the model's classifier for recognizing sounds of new and learned classes. The model is first trained with a random episodic training strategy, and then its backbone is used to generate the prototypes. A dynamic relation projection module refines the prototypes to enhance their discriminability. Results on two datasets (derived from the corpora of Nsynth and FSD-MIX-CLIPS) show that the proposed method exceeds three state-of-the-art methods in average accuracy and performance dropping rate.

Index Terms: few-shot learning, class-incremental learning, meta-training, audio classification

1. Introduction

Audio classification is a basic task for many audio-related applications, such as acoustic scene classification [1–3], sound event detection [4–6], and bioacoustic monitoring [7–9]. Deep learning has been successfully applied to various audio processing tasks in recent years [10, 11]. However, these successes mainly rely on large amounts of labeled data, and tedious model tuning [12, 13]. Moreover, the models adopted in these works are generally trained in a supervised manner, which makes the models only recognize classes defined at the development stage [14]. If some classes (e.g., rare animal sounds) are not included in the development dataset, the deployed model cannot recognize those audio classes [15].

To improve the ability of the model to adapt to dynamic acoustic environments, some researchers have introduced the few-shot learning (FSL) paradigm to the audio domain [16–18]. However, these FSL-based studies mainly focus on improving the model's ability to recognize new audio classes but do not pay much attention to maintaining the model's ability to remember learned ones. Humans can recognize new audio classes with few samples or even one sample without forgetting the learned ones. Such human capacity is needed to improve the intelligence of the model. Therefore, we introduce the few-shot class-incremental learning (FSCIL) paradigm [19,20] to the audio domain and address a new problem of few-shot class-incremental audio classification (FCAC).

The problem setup of FCAC is different from the existing FSL-based audio processing tasks [21–23]. Specifically, the former required the model to continually recognize new classes of sounds without forgetting the learned ones. In contrast, the latter only required the model to recognize new classes of sounds. Recently, Wang et al. [24] proposed a few-shot continual learning framework for audio classification using the dynamic few-shot learning (DFSL) method [25]. They expand the classifier of the model with an attention-based weights generator (ABWG) to recognize both new and old sound classes. Their framework assumes a scenario where the model needs to recognize both old and new classes within a single session. However, the FCAC problem requires the model to perform recognition across multiple continuous sessions and distinguish between previously learned classes and newly encountered ones. Thus, the FCAC problem poses greater challenges and represents real-world scenarios of intelligent audio processing.

Most deep models can be decoupled into a backbone and a classifier [26]. Therefore, to expand a model for classifying new and learned audio classes, the classifier of the model needs to be expanded with new classification weights or prototypes [27]. To that end, one can expand the classifier with randomly initialized prototypes for new audio classes and then retrain the model using the methods such as the finetune method [28] and the method of incremental classifier and representation learning (ICaRL) [29]. Alternatively, we extend the solution proposed by Wang et al. [24] for the FCAC problem by reusing the ABWG to expand the classifier in each session. Therefore, we provide three baseline methods for the FCAC problem.

To tackle the FCAC problem, we propose a method to generate discriminative prototypes and use them to expand the model's classifier. The proposed method consists of the random episodic training strategy (RETS) and the dynamic relation projection module (DRPM). The RETS is designed to train the initial model with strong generalization, and the DRPM is designed to refine the prototypes to make the model more discriminative. Two audio datasets, namely Neural synthesis of 100 classes of musical instrument sound (NSynth-100) and Freesound clips of 89 classes (FSC-89), are developed for performance evaluation of all methods. Experimental results on the two datasets show that the proposed method outperforms the baseline methods in average accuracy (AA) and performance dropping rate (PD). The contributions of this work are mainly two-fold. First, we address the new problem of FCAC, which aims to makes the model incrementally recognize new classes with few training samples without forgetting learned classes. Second, we provide baseline solutions and propose an alterna-

Corresponding author : Yanxiong Li. This work was supported by the national natural science foundation of China (62111530145, 61771200), Guangdong basic and applied basic research foundation (2022A1515011687, 2021A1515011454), and international scientific research collaboration project of Guangdong (2021A0505030003).

10.21437/Interspeech.2023-1380

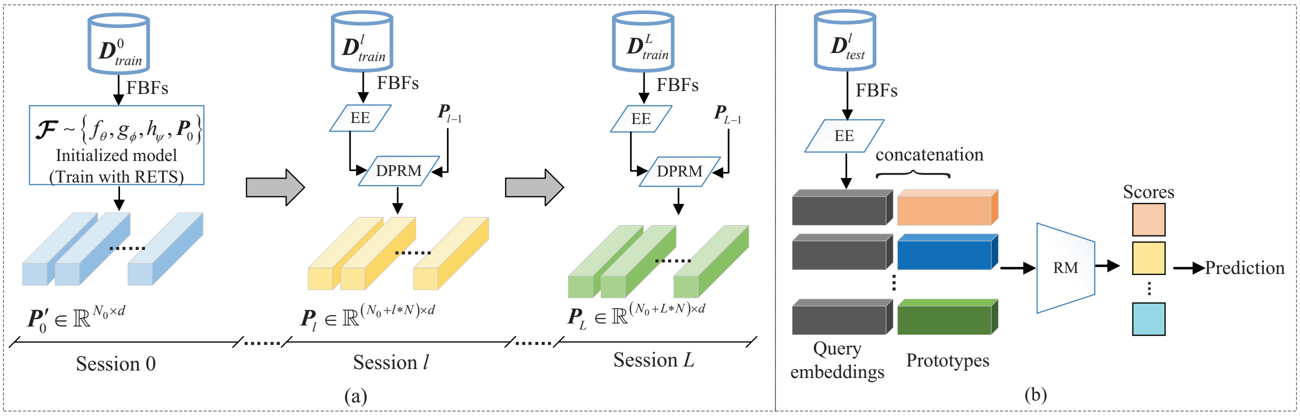

Figure 1: *Schematic diagram of the proposed method for FCAC. (a) The process of prototype generation and refinement. (b) Classification of the test samples using the RM and the prototypes.*

tive method for tackling the FCAC problem.

2. Problem Description

The FCAC problem involves L learning sessions that come in sequence. The first session is called the base session, and the remaining ones are called incremental sessions. A model for FCAC is developed in the base session and evaluated in all incremental sessions. Let $\boldsymbol{D}^l_{train} = \left\{\left(x^l_i, y^l_i\right)\right\}^{n^l_{train}}_{i=1}$ and $\boldsymbol{D}^l_{test} = \left\{\left(\hat{x}^l_j, \hat{y}^l_j\right)\right\}^{n^l_{test}}_{j=1}$ denote the training set and the test set in session $l\,(0 \leq l \leq L)$, respectively. x^l_i and $\hat{x}^l_j$ denote the training sample and the test sample of session l, respectively. y^l_i and $\hat{y}^l_j$ are the label of x^l_i and $\hat{x}^l_j$, respectively. n^l_{train} and n^l_{test} are the number of samples in $\boldsymbol{D}^l_{train}$ and $\boldsymbol{D}^l_{test}$, respectively. $\boldsymbol{D}^l_{train}$ and $\boldsymbol{D}^l_{test}$ have the same label set $\boldsymbol{Y}_l$, i.e., $y^l_i, \hat{y}^l_j \in \boldsymbol{Y}_l$. The label sets of any two training sets are disjoint, i.e., $\forall l \neq k, \boldsymbol{Y}_l \cap \boldsymbol{Y}_k = \varnothing$. Among all L training sets, only $\boldsymbol{D}^0_{train}$ contains a substantial amount of training samples. In contrast, each of the remaining training sets contains only a few training samples, which are organized in the format of N-way K-shot, i.e., there are N classes of data in the training set, and each class has K training samples. Due to data privacy and the shortage of memory, the training data in prior sessions (sessions 0 to l-1) are generally discarded after usage and thus only $\boldsymbol{D}^l_{train}$ can be used to update the model $\boldsymbol{\mathcal{F}}$ in session l. Meanwhile, the updated model needs to make predictions on test sets of both the current session and all prior sessions, i.e., $\boldsymbol{D}^0_{test} \cup \boldsymbol{D}^1_{test} \cup \cdots \cup \boldsymbol{D}^l_{test}$. As a result, the objective for solving the FCAC problem is to minimize the empirical risk [30] of the model over all test sets:

$$\min \left\{ \sum\nolimits_{(\hat{x}_j, \hat{y}_j) \in D^0_{test} \cup \cdots \cup D^L_{test}} \mathcal{L}\left(\boldsymbol{\mathcal{F}}\left(\hat{x}_j\right), \hat{y}_j\right) \right\} \tag{1}$$

where $\mathcal{L}(\cdot, \cdot)$ denotes a loss function.

3. Method

The motivation of the proposed method is to expand the classifier of the model with discriminative prototypes. Each prototype [27] is initialized as the mean of all the embeddings of training samples from that class:

$$\boldsymbol{p}_k = \frac{1}{|\boldsymbol{S}_k|} \sum_{(x_i, y_i) \in \boldsymbol{S}_k} f_\theta\left(x_i\right) \tag{2}$$

where $\boldsymbol{p}_k$ and $\boldsymbol{S}_k$ are the prototype and the training set of class k, respectively. f_θ denotes an embedding extractor (EE).

An EE with strong generalization ability can be more robust across different audio classes, which results in more semantically meaningful embeddings and prototypes that capture the essence of different audio classes. In addition, the generalization ability of the EE is critical to avoid the impact of noise and redundant information on the prototypes. Therefore, one way to obtain discriminative prototypes is to improve the generalization ability of the EE. To this end, we propose using RETS to train the EE by simulating the test scenario, where the training data is organized to mimic the evaluation process. Besides, due to the limited number of training samples used to derive prototypes of new classes, their discriminability may be weaker than that of the audio classes in the base session. This can result in an unbalanced discriminability between the new and learned prototypes, leading to degraded classification performance of the model. To address this challenge, we propose using DRPM to refine the prototypes of new and learned classes to minimize their confusion. The refined prototypes can be more well-separated, thus enhancing the model's classification performance.

As shown in Fig. 1 (a), the number of prototypes expands incrementally session by session. In session 0, a model consisting of four modules is trained using the RETS with $\boldsymbol{D}^0_{train}$. The first module is an EE $f_\theta : \mathbb{R}^D \longrightarrow \mathbb{R}^d$, which is used to convert the filter-bank features (FBFs) of audio samples into embeddings. D and d denote the dimension of FBFs and embeddings, respectively. The second module is a DRPM g_ϕ, which is used to refine the prototypes. The third module is a relation module (RM) h_ψ for implementing the classification. The fourth module is the initial prototypes of base classes, which is represented by a learnable parameter matrix $\boldsymbol{P}_0 \in \mathbb{R}^{N_0 \times d}$. N_0 is the number of base classes.

In each incremental session, the EE takes the training set of the new classes and generates prototypes for those classes. Then, the DPRM is utilized to refine the new prototypes with the prototypes of the previous session. After that, predictions of the test samples can be made based on the similarity between the embedding of each test sample and the prototypes of all classes in that session. As is shown in Fig. 1 (b), a relation module (RM) is used as a learnable metric [31] to calculate the similarity between the prototypes and the test embedding. Besides, to prevent the learned knowledge from being forgotten in incremental sessions, the parameters of EE, DRPM, and

RM obtained in the base session are frozen in all incremental sessions. Instead, the prototypes are continually expanded and refined to recognize new and learned audio classes.

Algorithm 1: Random episodic training strategy. $\mathcal{L}_{CE}(\cdot, \cdot)$ denotes the cross-entropy loss function.

```
Input: $\boldsymbol{D}_{train}^{0}$; A randomly initialized model $\boldsymbol{\mathcal{F}}$ including four modules: $f_\theta$, $g_\phi$, $h_\psi$ and $\boldsymbol{P}_0$. The number of episode in each training iteration $t$.
Output: A trained model.
while not convergence do
    Randomly sample a test subset $\boldsymbol{Q}$ from $\boldsymbol{D}_{train}^{0}$, whose label set is $\boldsymbol{Y}_Q$, and $\boldsymbol{Y}_Q \subset \boldsymbol{Y}_0$;
    Randomly sample $t$ sets of $N$-way $K$-shot training subsets from $\boldsymbol{D}_{train}^{0}$: $\{\boldsymbol{S}_1, \cdots, \boldsymbol{S}_t\}$, whose label sets are $\{\boldsymbol{Y}_{\boldsymbol{S}_1}, \cdots, \boldsymbol{Y}_{\boldsymbol{S}_t}\}$ ;
    for $i$ in $[1, t]$ do
        Keep the prototypes in $\boldsymbol{P}_0$, whose labels are not in $\boldsymbol{Y}_{\boldsymbol{S}_i}$ and treat them as the pseudo-base prototypes $\boldsymbol{P}_0{}'$;
        Use $f_\theta$ and $\boldsymbol{S}_i$ to generate the prototypes of pseudo-new classes $\boldsymbol{P}_{\boldsymbol{S}_i}$;
        Merge $\boldsymbol{P}_0{}'$ and $\boldsymbol{P}_{S_i}$ to form the initial prototypes in an episode $\boldsymbol{P}_{init}$;
        Take $\boldsymbol{P}_{init}$ into DRPM to obtain the refined prototypes: $\boldsymbol{P}_{re} = g_\phi(\boldsymbol{P}_{init}, \boldsymbol{P}_0{}')$
        Obtain predictions $\hat{\boldsymbol{Y}}_Q$ of the test embeddings $f_\theta(\boldsymbol{Q})$ using $\boldsymbol{P}_{re}$ and the RM;
        Calculate and accumulate the loss: $\sum \mathcal{L}_{CE}\left(\hat{\boldsymbol{Y}}_Q, \boldsymbol{Y}_Q\right)$ ;
    Optimize the model with SGD algorithm;
```

3.1. Random episodic training strategy

The random episodic training strategy (RETS) is to organize the training data into a series of simulated tasks and use them to optimize the model. Those tasks mimic the few-shot circumstances encountered in each incremental session. The pseudocode of the RETS is presented in Algorithm 1. In each training iteration, a test subset is randomly selected from $\boldsymbol{D}_{train}^{0}$, which contains samples of all base classes. The test subset set is used as test data in a pseudo-incremental session containing test samples of pseudo-old and pseudo-new classes. Multiple training subsets are randomly selected from $\boldsymbol{D}_{train}^{0}$ and then used to obtain the prototypes of pseudo-new classes. Meanwhile, the prototypes in $\boldsymbol{P}_0$ whose labels are different from those of the pseudo-new classes, are used as the pseudo-base prototypes. Then, the initial pseudo-incremental prototypes are obtained by merging the pseudo-new and pseudo-base prototypes. The refined pseudo-incremental prototypes are obtained using the DRPM. After that, the predictions of the test subset can be made using the EE, RM, and refined pseudo-incremental prototypes. Finally, the cross-entropy losses of multiple episodes are calculated, and the model is optimized via the algorithm of stochastic gradient descent (SGD) [32].

3.2. Dynamic relation projection module

The initial prototypes $\boldsymbol{P}_{init} \in \mathbb{R}^{N_{init} \times d}$ in each incremental session are constructed by merging the new prototypes and the prototypes of previous session $\boldsymbol{P}_{pre} \in \mathbb{R}^{N_{pre} \times d}$. To enhance the discriminability of the initial prototypes, the DRPM employs a set of relation weights $\boldsymbol{M} \in \mathbb{R}^{N_{init} \times N_{pre}}$ to refine the initial prototypes in each incremental session. N_{init} and N_{pre} are the number of audio classes in the current and previous sessions, respectively. The relation weights are obtained by

$$\boldsymbol{M} = f_1\left(\boldsymbol{P}_{init}\right)\left(f_2\left(\boldsymbol{P}_{pre}\right)\right)^T \tag{3}$$

where f_1 and f_2 represent two convolutional blocks for transforming the prototypes to a shared latent space. Each block contains a 1×1 convolutional layer, a batch normalization layer, and a ReLU layer. The value in $\boldsymbol{M}$ represents the similarity between different prototypes, so it can be used to weight the prototypes to achieve refinement of the prototypes. Therefore, the refined prototypes $\boldsymbol{P}_{re} \in \mathbb{R}^{N_{init} \times d}$ is obtained by $\boldsymbol{P}_{re} = \boldsymbol{M}\boldsymbol{P}_{pre}$.

4. Experiments

4.1. Experimental setup

The Nsynth-100 and FSC-89 datasets[1] are derived from the public audio corpora of Nsynth [33] and FSD-MIX-CLIPS [34], respectively. Statistics of the experimental datasets are shown in Table. 1. The baseline methods of Finetune [28], ICaRL [29], and DFSL [25] are re-implemented with open source codes according to the problem setup of FCAC. For a fair comparison, the backbone of the ResNet18 [35] is used as the EE for all methods. Meanwhile, FBFs is used in all methods. The RM used in the proposed method consists of three 1×1 convolutional layers with a batch normalization, a ReLU layer, and a dropout layer between every two convolutional layers. Accuracy in each session $\mathcal{A}_l$, average accuracy over all sessions $AA = \frac{\sum_{l=0}^{L} \mathcal{A}_l}{L}$, and performance dropping rate $PD = \mathcal{A}_0 - \mathcal{A}_L$ [26] are used as performance metrics. $\mathcal{A}_0$ is the accuracy in the base session, and $\mathcal{A}_L$ is the accuracy in the last incremental session. All results are obtained by averaging the outputs of 100 trials.

Table 1: *Statistics of the experimental datasets. NBS and NNS denotes the number of training, validation and testing samples per base and new class, respectively.*

	NSynth-100	FSC-89
Type of sound	Musical instruments	Sound events
No. of classes	55 (base), 45 (new)	59 (base), 30 (new)
NBS	200, 100, 100	800, 200, 200
NNS	100 / none / 100	500 / none / 200
Duration per sample	4 seconds	1 second
Sampling frequency	16 kHz	44.1 kHz

4.2. Results and discussions

We first perform comparison experiments on the two audio datasets. The results obtained by different methods on the Nsynth-100 and the FSC-89 are shown in Table 2 and Table 3, respectively. As shown in Table 2, the accuracies achieved by our method are higher than that of three baseline methods in all nine incremental sessions. The AA obtained by our method is 89.26%, which is 29.44%, 14.72%, and 4.37% higher than that of the Finetune, ICaRL, and DFSL methods, respectively. In addition, the PD obtained by our method is 20.27%, which

[1] https://github.com/chester-w-xie/FCAC_datasets

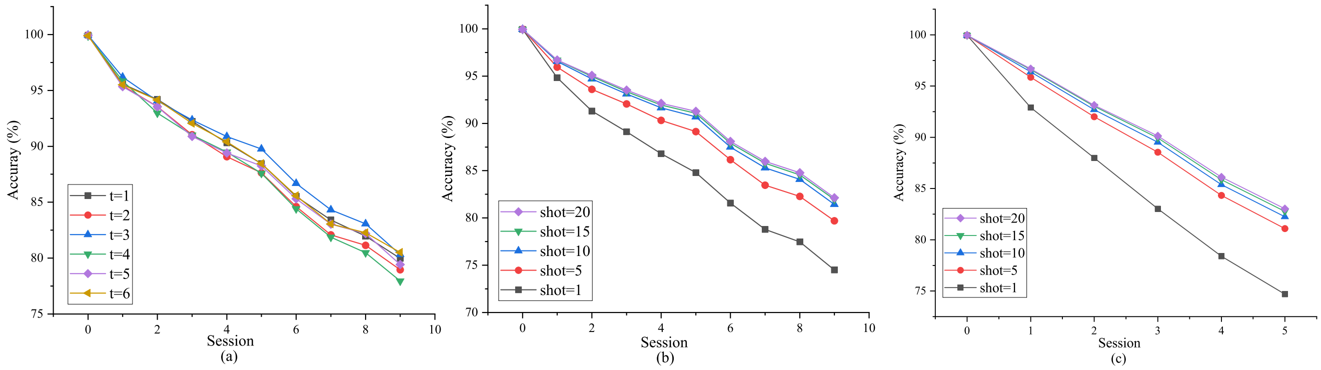

Figure 2: *Performance analysis of our method with different settings on the Nsynth-100. (a) The impact of the episode number in Algorithm 1. (b) Performance of our method in the setting of 5-way with different numbers of shots. (c) Performance of our method in the setting of 9-way with different numbers of shots.*

Table 2: *Results obtained by different methods on the Nsynth-100.*

Method	Accuracy in each session (%) ↑										AA ↑	PD ↓
	0	1	2	3	4	5	6	7	8	9		
Finetune	99.96	84.73	76.92	71.06	63.18	40.15	47.99	38.33	38.01	37.86	59.82	62.10
ICaRL	**99.98**	93.30	88.88	84.82	79.57	67.65	65.92	59.65	54.62	51.01	74.54	48.97
DFSL	99.91	95.36	91.51	88.02	84.30	81.97	79.08	77.12	76.55	75.14	84.89	24.77
Proposed	99.96	**95.95**	**93.60**	**92.06**	**90.32**	**89.14**	**86.16**	**83.47**	**82.28**	**79.69**	**89.26**	**20.27**

is 41.83%, 28.70%, and 4.5% lower than that of the Finetune, ICaRL, and DFSL methods, respectively. Because the Finetune method uses a few training samples of new classes to retrain the model, the model overfits the new classes and catastrophically forgets the learned classes. As a result, the Finetune method performs worse than other methods. With data retention and knowledge distillation strategies, the ICaRL method obtains higher classification accuracies than the Finetune method. However, since the ICaRL method also employs the retraining strategy to expand the classifier, the model cannot effectively remember the learned classes' knowledge when new classes appear continuously. Therefore, with the increase of new classes, its performance worsens. To adapt the DFSL method for the FCAC problem, we reuse an ABWG trained in the base session to generate new classification weights in each incremental session. The parameters of the ABWG were frozen in all incremental sessions for avoiding interference caused by the retraining strategy. However, without a powerful EE and the refinement of the classification weights, the model's classification performance can hardly be improved. In contrast, the proposed method employs the RETS to train the model to improve its generalization ability, which enables EE to generate the initial prototypes with strong discriminability. Moreover, the proposed method employs the DRPM to refine the prototypes in incremental sessions to further enhance their discriminability .

Table 3: *Results obtained by different methods on the FSC-89.*

Method	Accuracy in each session (%)↑							AA↑	PD↓
	0	1	2	3	4	5	6		
Finetune	42.28	31.67	29.11	24.63	21.74	20.17	16.70	26.61	25.58
ICaRL	**42.43**	36.25	33.00	26.60	26.63	23.76	20.71	29.91	21.72
DFSL	42.36	35.23	32.72	31.21	29.79	28.51	27.40	32.46	14.96
Proposed	42.04	**39.95**	**37.01**	**34.68**	**32.97**	**31.45**	**30.09**	**35.46**	**11.95**

Since the FSC-89 is more complex than the Nsynth-100 in terms of background noise, inter-class confusion, and intra-class inconsistency, all methods obtain lower accuracy on the FSC-89 than on the Nsynth-100, as shown in Table 2 and Table 3. However, the advantages of the proposed method over the baseline methods still hold for the FSC-89 in terms of accuracy, AA, and PD.

To show the performance of the proposed method with different experimental settings, we conducted three performance analysis experiments on the Nsynth-100. Fig. 2 (a) shows the impact of different numbers of episode t on the performance of the proposed method. The results show that the proposed method obtains the best overall performance when $t = 3$. Fig. 2 (b) and Fig. 2 (c) present the accuracy obtained by the proposed method with different numbers of shots. By comparing Fig. 2 (b) and Fig. 2 (c), it can be found that more shots lead to better performance. Moreover, when the model needs to recognize more new audio classes within a session, its classification performance decreases more severely, which is the reason why the curve in Fig. 2 (c) is sharper than that in Fig. 2 (b).

5. Conclusions

This paper addresses a new problem of audio classification termed few-shot classes-incremental audio classification. A method to expand the model with adaptively refined prototypes is proposed for the problem. We also adapt the methods of finetune, ICaRL, and DFSL to the FCAC problem and use them as the baseline methods. Experimental results on the NSynth-100 and the FSC-89 datasets indicate that our method achieves state-of-the-art performance. This work can be a benchmark for future studies of the FCAC problem. Our future works include designing a more effective prototype refinement strategy and evaluating the methods with more comprehensive experiments.

6. References

[1] W. Xie, Q. He, H. Yan, and Y. Li, "Acoustic scene classification using deep cnns with time-frequency representations," in *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, 2021, pp. 1325–1329.

[2] W. Xie, Q. He, Z. Yu, and Y. Li, "Deep mutual attention network for acoustic scene classification," *Digital Signal Processing*, vol. 123, p. 103450, 2022.

[3] N. W. Hasan, A. S. Saudi, M. I. Khalil, and H. M. Abbas, "A genetic algorithm approach to automate architecture design for acoustic scene classification," *IEEE Transactions on Evolutionary Computation*, 2022, early access.

[4] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[5] K. Imoto, S. Mishima, Y. Arai, and R. Kondo, "Impact of data imbalance caused by inactive frames and difference in sound duration on sound event detection performance," *Applied Acoustics*, vol. 196, p. 108882, 2022.

[6] S. Park, D. K. Han, and M. Elhilali, "Cross-referencing self-training network for sound event detection in audio mixtures," *IEEE Transactions on Multimedia*, 2022, early access.

[7] J. Cramer, V. Lostanlen, A. Farnsworth, J. Salamon, and J. P. Bello, "Chirping up the right tree: Incorporating biological taxonomies into deep bioacoustic classifiers," in *2020 ICASSP*, 2020, pp. 901–905.

[8] D. Teixeira, S. Linke, R. Hill, M. Maron, and B. J. van Rensburg, "Fledge or fail: Nest monitoring of endangered black-cockatoos using bioacoustics and open-source call recognition," *Ecological Informatics*, vol. 69, p. 101656, 2022.

[9] B. B. Gatto, J. G. Colonna, E. M. dos Santos, A. L. Koerich, and K. Fukui, "Discriminative singular spectrum classifier with applications on bioacoustic signal recognition," *Digital Signal Processing*, vol. 133, p. 103858, 2023.

[10] R. Cosbey, A. Wusterbarth, and B. Hutchinson, "Deep learning for classroom activity detection from audio," in *2019 ICASSP*, 2019, pp. 3727–3731.

[11] G. Morrone, D. Michelsanti, Z.-H. Tan, and J. Jensen, "Audio-visual speech inpainting with deep learning," in *2021 ICASSP*, 2021, pp. 6653–6657.

[12] Z. Zhao, Q. Liu, W. Cao, D. Lian, and Z. He, "Self-guided information for few-shot classification," *Pattern Recognition*, vol. 131, p. 108880, 2022.

[13] Y. Chen, B. Liu, Z. Zhang, and H.-S. Kim, "An end-to-end deep learning framework for multiple audio source separation and localization," in *2022 ICASSP*, 2022, pp. 736–740.

[14] K. K. Teh and H. D. Tran, "Open-set audio classification with limited training resources based on augmentation enhanced variational auto-encoder gan with detection-classification joint training." in *Interspeech*, 2021, pp. 4169–4173.

[15] T. F. Tavares, "Open-set classification approaches to automatic bird song identification: towards non-invasive wildlife monitoring in brazilian fauna," *IEEE Latin America Transactions*, vol. 20, no. 11, pp. 2388–2394, 2022.

[16] H.-P. Huang, K. C. Puvvada, M. Sun, and C. Wang, "Unsupervised and semi-supervised few-shot acoustic event classification," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 331–335.

[17] Y. Wang and D. V. Anderson, "Hybrid attention-based prototypical networks for few-shot sound classification," in *2022 ICASSP*, 2022, pp. 651–655.

[18] S.-F. Huang, C.-J. Lin, D.-R. Liu, Y.-C. Chen, and H.-y. Lee, "Meta-tts: Meta-learning for few-shot speaker adaptive text-to-speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1558–1571, 2022.

[19] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 183–12 192.

[20] K. Zhu, Y. Cao, W. Zhai, J. Cheng, and Z.-J. Zha, "Self-promoted prototype refinement for few-shot class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6801–6810.

[21] V. Morfi, I. Nolasco, V. Lostanlen, S. Singh, A. Strandburg-Peshkin, L. F. Gill, H. Pamula, D. Benvent, and D. Stowell, "Few-shot bioacoustic event detection: A new task at the dcase 2021 challenge." in *DCASE*, 2021, pp. 145–149.

[22] D. Yang, H. Wang, Y. Zou, Z. Ye, and W. Wang, "A mutual learning framework for few-shot sound event detection," in *2022 ICASSP*, 2022, pp. 811–815.

[23] H. Liu, X. Liu, X. Mei, Q. Kong, W. Wang, and M. D. Plumbley, "Surrey system for dcase 2022 task 5 : Few-shot bioacoustic event detection with segment-level metric learning technical report," DCASE2022 Challenge, Tech. Rep., June 2022.

[24] Y. Wang, N. J. Bryan, M. Cartwright, J. P. Bello, and J. Salamon, "Few-shot continual learning for audio classification," in *2021 ICASSP*, 2021, pp. 321–325.

[25] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4367–4375.

[26] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, and Y. Xu, "Few-shot incremental learning with continually evolved classifiers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 455–12 464.

[27] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[28] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, and D.-C. Zhan, "Pycil: A python toolbox for class-incremental learning," *SCIENCE CHINA Information Sciences*, 2022.

[29] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[30] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, L. Ma, S. Pu, and D.-C. Zhan, "Forward compatible few-shot class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9046–9056.

[31] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[32] Y. Lei and K. Tang, "Learning rates for stochastic gradient descent with nonconvex objectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4505–4511, 2021.

[33] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.

[34] Y. Wang, N. J. Bryan, J. Salamon, M. Cartwright, and J. P. Bello, "Who calls the shots? rethinking few-shot learning for audio," in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021, pp. 36–40.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.