FRAUNHOFER FKIE SUBMISSION FOR TASK 5: FEW-SHOT BIOACOUSTIC EVENT DETECTION

Technical Report

Kevin Wilkinghoff and Alessia Cornaggia-Urrigshardt

Fraunhofer FKIE
Fraunhoferstraße 20, 53343 Wachtberg, Germany
{kevin.wilkinghoff,alessia.cornaggia-urrigshardt}@fkie.fraunhofer.de

ABSTRACT

This report describes the Fraunhofer FKIE submission for task 5 "Few-shot Bioacoustic Event Detection" of the DCASE challenge 2023. The submitted system is an adaptation of a few-shot keyword spotting system that uses embeddings with a temporal resolution suitable for template matching with dynamic time warping. The embedding model is trained to not only predict the sound event class but also the temporal position of a segment in a sound event using the angular margin loss TempAdaCos. At inference, embeddings are extracted and segment-wise cosine distances between the recording to be searched in and the provided templates are calculated. The resulting cost matrices are processed by applying a logistic regression model that is trained to discriminate between positive and negative frames. Lastly, dynamic time warping in combination with peak-picking and using a decision threshold is applied to detect on- and offsets of bioacoustic events. As a result, the presented system significantly outperforms both baseline systems.

***Index Terms*—** bioacoustics, sound event detection, few-shot learning, representation learning, machine listening

1. INTRODUCTION

Biologists often collect large amounts of data to increase the likelihood that the events of interest for a specific research project are being captured. Possible ways to do this is to use bioacoustic sensors belonging to a (multisensor) station with a fixed position [1] or sensors attached to an individual specimen [2]. However, since the sensors are not supervised, large parts of the recordings do not contain bioacoustic events or at least not the ones of interest. To reduce manual labeling work and thus the required time biologists need to analyze an audio recording, bioacoustic events should be annotated automatically. This motivates bioacoustic monitoring as an area of machine listening research [3, 4, 5]. The main problems of this field are complex acoustic scenes, varying acoustic conditions and recording devices, very limited amounts of training data (few-shot learning [6]) and very different characteristics such as the duration of animal vocalizations. All of these problems are addressed in the few-shot bioacoustic event detection task of the DCASE challenge 2023 and its predecessors [7, 8]. The goal of these tasks is to detect all vocalizations of an animal together with the corresponding on- and offsets within in a recording of possibly long duration by using only the first five annotated sound events in the same recording belonging to the same species. To train and develop the system, multiple fully labeled recordings containing mostly other animals than those to be detected are provided.

Classically, automatic systems for detecting bioacoustic events are based on template matching with dynamic time warping (DTW) applied to spectral data representations [9, 10] or auto-correlation [11, 12]. Template matching has the advantage that varying lengths between animal calls can be handled effectively and that no training is required. However, in case of changing or difficult acoustic conditions these approaches quickly fail. Moreover, determining a single feature representations suitable for detecting very different species via template matching is a challenge. Modern systems utilize deep learning based models [13]. For few-shot learning, often convolutional neural networks (CNNs) with a prototypical loss [14] are used to learn a suitable embedding space and some form of sliding window is used to detect events [15, 16, 17, 18]. Here, choosing the size of the sliding window is one of the main difficulties because of the strongly varying durations of calls belonging to different species. In [16] an ensemble of multiple models trained on different segment lengths is used and in [18] individual frames of the spectral representations are used directly to avoid the need to choose a specific length. For the DCASE challenge task, both approaches, template matching and a prototypical network, are provided as baseline systems [7, 8].

The contributions of this work are the following. First and foremost, a state-of-the art few-shot bioacoustic event detection system submitted to task 5 of the DCASE challenge 2023 is presented. The system learns embeddings with temporal structure using the TempAdaCos loss [19], which has been developed for few-shot keyword spotting (KWS), and applies DTW to detect bioacoustic events. Hence, the strengths of both bioacoustic event detection approaches, learning embeddings as robust feature representations and applying template matching to effectively detect events, are combined into a single system. To improve the performance for this task, adaptions of the original work are proposed, namely using other input feature representations and a different procedure for calculating cost matrices and detecting on- and offsets of events.

2. PROPOSED SYSTEM

The structure of the proposed system is depicted in Figure 1. Each of the four depicted processing steps will now be described in detail.

2.1. Calculating input feature representations

To obtain input feature representations for the embedding model, the following processing steps have been carried out. First, the audio signal is normalized to a maximum amplitude of 1, resampled to

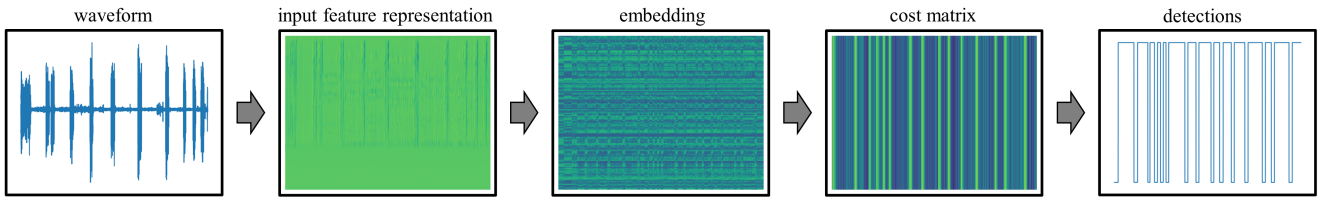

Figure 1: Main processing steps of the proposed few-shot bioacoustic event detection system.

22 050 Hz and high-pass filtered at 50 Hz. Then, Mel-spectrograms are calculated using a window size of 512 and a hop size of 128 with 64 Mel bins to have a sufficiently high time resolution. Last but not least, per-channel energy normalization (PCEN) [20, 21] is applied as done for most other systems such as the prototypical baseline systems [7, 8].

One of the major difficulties to overcome when detecting different bioacoustic events with a single system are the different lengths of individual events ranging from a few milliseconds to several seconds. To be able to handle different lengths, the computed features are divided into overlapping segments along the temporal dimension. As also proposed for the prototypical baseline system [8], we applied an adaptive segment length depending on the size of the five labeled events. More concretely, we used one percent of the maximum of all event lengths while ensuring a minimum segment length of 0.1 s and an overlap of half the chosen segment size. Events with a total length shorter than 0.1 s are zero-padded to the desired segment length. As the result, all feature representations have a size of 66×64. Note that for training the embedding model, not only the class label of the bioacoustic event but also the position of a given segment within an event is used and thus needs to be stored as an additional label. Due to the chosen size of the segment lengths, this corresponds to 200 possible positions that are encoded as categorical labels with 200 entries. For short sound events, multiple or even all positions may be encoded as "active" in the categorical label. More details can be found in [19].

2.2. Extracting embeddings with temporal dimension

To extract embeddings, a very similar procedure as explained in our prior work on few-shot KWS is applied [22, 19]. The model has the modified ResNet architecture used in [22] and is trained by minimizing the angular margin loss (AML) TempAdaCos [19], which teaches the model to learn embeddings with temporal structure. The main differences between using AMLs and prototypical losses [14] for few-shot learning are that for AMLs the embeddings are projected onto the unit sphere and a margin between classes is ensured. Furthermore, for AMLs the randomly initialized class centers of the embedding space are learned during training as trainable model parameters instead of being re-calculated after each epoch as the mean of embeddings belonging to a support set. After training, a mean embedding can be calculated in the same manner regardless of the loss being used. Thus, for balanced classes an AML has no disadvantage over a prototypical loss and one could also easily formulate an angular prototypical loss [23].

For the presented system, the learned embeddings have a feature dimension of 128 and the same temporal dimension ($T = 66$) as the input feature representations. This is achieved by not applying any temporal pooling operation or strides inside the CNN, similarly as proposed in [18] and, before that, in [22]. TempAdaCos consists of a sum of two loss functions, one for predicting the sound event class and one for the temporal position of the segment in a sound event. Let $e_{k,j}$ denote an embedding computed for the jth segment (in temporal order) of the kth training sample. Then, the softmax probability of $e_{k,j}$ belonging to class i_{class} and position i_{pos} is defined as

$$s_{k,j}(i_{\text{class}}, i_{\text{pos}}) := \frac{\exp(\hat{s} \cdot \theta_{k,j}(i_{\text{class}}, i_{\text{pos}}))}{\sum_{j_{\text{class}}=1}^{N_{\text{class}}} \sum_{j_{\text{pos}}=1}^{N_{\text{pos}}} \exp(\hat{s} \cdot \theta_{k,j}(j_{\text{class}}, j_{\text{pos}}))}$$

where $\hat{s}$ denotes the dynamically adaptive scale parameter of the AdaCos loss [24]. The cosine angles between embedding $e_{k,j}$ and class center $c_{i_{\text{class}}, i_{\text{pos}}}$ are defined as the mean cosine similarity over the time dimension:

$$\theta_{k,j}(i_{\text{class}}, i_{\text{pos}}) := \frac{1}{T} \sum_{t=1}^{T} \frac{\langle e_{k,j}(t), c_{i_{\text{class}}, i_{\text{pos}}} \rangle}{\|e_{k,j}(t)\|_2 \|c_{i_{\text{class}}, i_{\text{pos}}}\|_2}.$$

As for standard angular margin losses, the class centers $c_{i_{\text{class}}, i_{\text{pos}}}$ do not have a temporal dimension and are randomly initialized parameters of the model adapted during training. Now, the probability of embedding $e_{k,j}$ belonging to class i_{class} is set to be $\sum_{i_{\text{pos}}=1}^{N_{\text{pos}}} s_{k,j}(i_{\text{class}}, i_{\text{pos}})$. Similarly, the probability of embedding $e_{k,j}$ belonging to position i_{pos} is equal to $\sum_{i_{\text{class}}=1}^{N_{\text{class}}} s_{k,j}(i_{\text{class}}, i_{\text{pos}})$. The embedding model is trained to solve both classification tasks simultaneously by minimizing the sum of both corresponding categorical cross-entropies. After training, all resulting two-dimensional embeddings belonging to overlapping segments are combined into one large two-dimensional embedding by taking the mean of all individual frames belonging to exactly the same temporal position. More technical details about TempAdaCos and how to combine the embeddings can be found in [19].

The embedding model is trained using all annotated positive and negative events of the training set as also done in [17]. Furthermore, temporally reversed segments of the training samples are used for training as proposed in [19]. For each original class, all temporally reversed segments belonging to this class are labeled as belonging to a specific, newly introduced class, doubling the number of sound event classes. The position of temporally-reversed segments is not of interest and encoded to have the same value for each categorically labeled position, i.e. N_{pos}^{-1}. For clarification we note that time-reversed versions of the acoustic events are included here in order to train the network to better recognize the correct temporal structure of the acoustic events. Consequently, the time-reversed events are *not* considered to belong to the same class as the original, non reversed, events As a result, training the embedding model can be seen as a combination of supervised and self-supervised learning because for every segment the model needs to

predict 1) its sound event class (supervised) as well as the 2) relative position in the non-segmented sound event (self-supervised) and 3) determine whether the temporal order is reversed or not (self-supervised). This leads to learning more meaningful embeddings than when only using a supervised training procedure and significantly improves few-shot sound event detection performance [19]. For data augmentation, mixup [25] with a mixing coefficient drawn from a uniform distribution and SpecAugment [26] have been applied. Since using an adaptive scale parameter for TempAdaCos as proposed in [24] led to numerical issues during training, we used ArcFace [27], i.e. TempArcFace, with a margin of 0.2 and a fixed scale parameter $\hat{s} = \sqrt{2} \cdot \log(N_{\text{class}} - 1)$ instead. To train the embedding model, undersampling with respect to the sound event class labels as implemented in [28] has been used. The model has been trained for 10000 epochs with a batch size of 64 using Adam [29].

2.3. Calculating cost matrices

When interpreting animal calls as keywords, the bioacoustic event detection task can be viewed as finding *any* keyword emitted by a certain animal instead of finding specific keywords. Thus, the global structure can strongly vary between different calls. Therefore, we do not directly match the templates to the corresponding recording by using frame-wise cosine distances to compute a cost matrix as done for KWS but use the steps shown in Figure 2. Note that local temporal structures of the input feature representations are captured by the CNN and thus are still being used as they are contained in the embeddings.

In [18], it has been shown that calibrating the distances by differentiating between positive and negative frames is highly beneficial to improve the performance [18]. Here, negative frames belong to the spectral features of annotated events and negative frames to the spectral features between these annotated events. Instead of using a softmax function to calibrate the similarity scores, we used a logistic regression model. More concretely, we used maximum and minimum of cosine similarities between positive and negative frames from the labeled parts of a recording, and the parts of the recordings to be searched through for animal calls. This translates to obtaining 4 dimensional features for each time step as input features for the logistic regression model. The logistic regression model is trained to predict whether a time step belongs to a target event or not with L^2 regularization and balanced class weights as implemented in scikit-learn [30]. Last but not least, the ratio of the positive and negative log-likelihoods from the logistic regression model are used as cost values instead of only using one of the probabilities. All steps of computing cost matrices are visualized in Figure 2. Note that when training the logistic regression model, the mean of the positive frames is taken to not only have similarity scores that are equal to one and thus are far too optimistic. At inference, for each time step the maximum cosine similarity belonging to each template is taken instead. The idea is that we want to compute cost matrices for DTW but, as explained above, not enforce the same global temporal structure of the templates. To still be able to apply DTW, a template size greater than one is simulated by simply repeating the costs of each time step until the desired size is reached. Choosing a size related to the real template sizes appears to be a reasonable choice because this means that detected events need to have a similar size as the five labeled events. For the challenge, we submitted several systems with different sizes as stated in subsection 2.5.

2.4. Detecting acoustic events

To detect acoustic events as well as their on- and offsets, DTW with allowed steps of $(1, 1)$, $(1, 2)$ and $(2, 1)$ is used. This means that the length of detected events can only be between 50% and 200% of the chosen template size and helps to reduce false positive detections. A larger template size can be considered more conservative but most likely increases the number of false negative detections. The entire proces of computing cost matrices is illustrated in Figure 3. First, the accumulated cost matrix with each entry being normalized by the corresponding path length is calculated. The element-wise negative of the last row is used as a matching function. Then, a chosen decision threshold is used to only consider matches with a score above this threshold. Next, peak-picking with a width of 8 and a distance of 2 is applied. At each peak, optimal warping paths are computed whose start- and endpoints are the on- and offsets of acoustic events. To make up for the temporal fuzziness caused by the size of the spectral window, on- and offsets are moved by $\frac{4 \cdot 128}{22050}$ to the left and $\frac{12 \cdot 128}{22050}$ to the right, respectively. Last but not least, events with invalid start- and endpoints are removed and events with a margin less than $\frac{8 \cdot 128}{22050}$ between them are merged. Note that by merging events with overlapping paths, detected events can also be longer than 200% of the chosen template size but still not be shorter than 50%.

2.5. Submissions

In total, we submitted four different systems each with different hyperparameter settings shown in Table 1. As explained above, by design our system cannot detect events shorter than half of the chosen template size. Thus, this hyperparameter has a strong impact on the performance which is the reason why we dedicated three submissions to altering the template size and only a single one to changing the decision threshold.

Table 1: Parameter settings of the submitted systems.

System	Template Size for DTW	Threshold
Submission 1	0.5·mean size of five shots	-0.05
Submission 2	$0.5 \cdot$ (mean + minimum) size of five shots	-0.05
Submission 3	$0.5 \cdot$ (mean + maximum) size of five shots	-0.05
Submission 4	0.5·mean size of five shots	-0.075

3. RESULTS

The results obtained on the validation set with our four submitted systems and the two baseline systems of the challenge are shown in Table 2. Since the validation set of the DCASE 2023 challenge is the same as the validation set of last year's edition of the challenge, the performances of the three top-performing systems of DCASE 2022 have also been included for comparison. It can be seen that our proposed systems all significantly outperform both baseline systems and reach a similar F-measure as the three top-performing systems of last year. Note that in past editions of this challenge task the ranking of systems according to the performance obtained on the validation set was often very different from the final ranking on the evaluation set. Thus, one should not draw conclusions too quickly but only view the performance on the validation set as a rough prediction. Furthermore, this year it is explicitly not allowed to use an ensemble of multiple models for prediction, which is known to

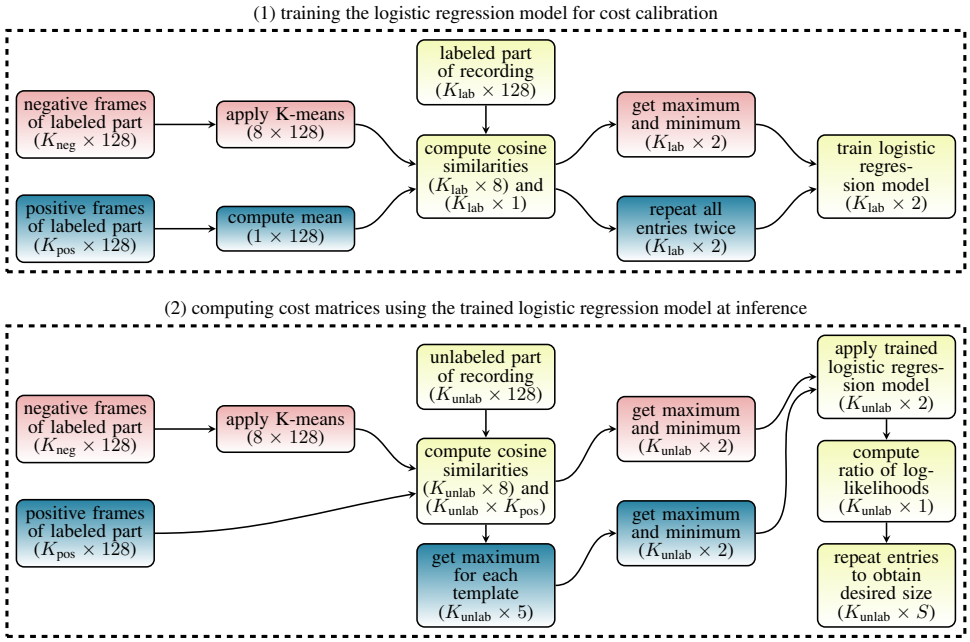

Figure 2: Illustration of the proposed steps for calculating cost matrices. Subfigure (1) contains all steps to train the cost calibration model and subfigure (2) contains all steps to obtain cost matrices at inference. Blocks colored in red are only related to negative frames, blocks colored in blue are only related to positive frames and yellow blocks are related to both.

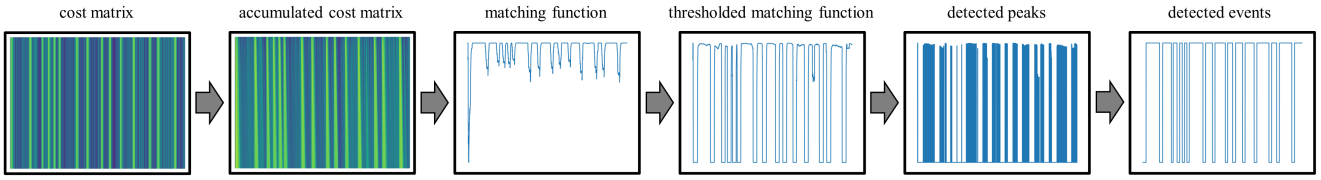

Figure 3: Illustration of the proposed procedure for detecting on- and offsets of the acoustic events using DTW applied to the cost matrices and additional post-processing.

Table 2: Performance of both baseline systems, the three top-ranked systems of the DCASE Challenge 2022 and our submitted systems on the validation set.

System	Precision	Recall	F-measure
Template Matching (Baseline)	2.4%	18.3%	4.3%
Prototypical Network (Baseline)	36.3%	25.0%	29.6%
DCASE 2022 Rank 1 [18]	77.5%	71.5%	74.4%
DCASE 2022 Rank 2 [17]	55.0%	45.9%	50.0%
DCASE 2022 Rank 3 [16]	unknown	unknown	60.0%
Submission 1	70.2%	58.4%	63.7%
Submission 2	74.3%	54.9%	63.2%
Submission 3	79.1%	46.1%	58.2%
Submission 4	62.4%	62.3%	62.4%

increase performance, whereas it was allowed last year. Taken this into account, it appears that our proposed system also reaches state-of-the-art performance. The final results of the challenge will give more insights on whether this claim is actually true.

4. CONCLUSIONS

In this work, a few-shot bioacoustic event detection system for task 5 of the DCASE challenge 2023 has been presented. The system is based on extracting embeddings with temporal structure, calculating suitable cost matrices and detecting on- and offsets of events with dynamic time warping. In experiments, it has been shown that the proposed approach significantly outperforms both baseline systems of the challenge and reaches a similar performance as the three top-performing systems of last year's challenge on the validation set. For future work, it is planned to conduct several ablation studies to justify the proposed design choices and compare the performance of the system to other submitted systems on the evaluation set of this DCASE task.

5. REFERENCES

[1] J. W. Wägele, *et al.*, "Towards a multisensor station for automated biodiversity monitoring," *Basic and Applied Ecology*, vol. 59, pp. 105–138, 2022.

[2] D. Stowell, E. Benetos, and L. F. Gill, "On-bird sound recordings: Automatic acoustic recognition of activities and contexts," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 6, pp. 1193–1206, 2017.

[3] R. Bardeli, D. Wolff, F. Kurth, M. Koch, K.-H. Tauchert, and K.-H. Frommolt, "Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring," *Pattern Recognit. Lett.*, vol. 31, no. 12, pp. 1524–1534, 2010.

[4] D. Stowell, M. Wood, Y. Stylianou, and H. Glotin, "Bird detection in audio: A survey and a challenge," in *26th International Workshop on Machine Learning for Signal Processing*. IEEE, 2016, pp. 1–6.

[5] I. Nolasco, *et al.*, "Learning to detect an animal sound from five examples," 2023, arXiv:2305.13210.

[6] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 63:1–63:34, 2021.

[7] V. Morfi, *et al.*, "Few-shot bioacoustic event detection: A new task at the DCASE 2021 challenge," in *6th Workshop on Detection and Classification of Acoustic Scenes and Events*, 2021, pp. 145–149.

[8] I. Nolasco, *et al.*, "Few-shot bioacoustic event detection at the DCASE 2022 challenge," in *7th Workshop on Detection and Classification of Acoustic Scenes and Events*, 2022, pp. 136–140.

[9] K. Kaewtip, L. N. Tan, A. Alwan, and C. E. Taylor, "A robust automatic bird phrase classifier using dynamic time-warping with prominent region identification," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 768–772.

[10] K. Kaewtip, A. Alwan, C. O'Reilly, and C. E. Taylor, "A robust automatic birdsong phrase classification: A template-based approach," *The Journal of the Acoustical Society of America*, vol. 140, no. 5, pp. 3691–3701, 2016.

[11] F. Kurth, "Robust detection of multiple bioacoustic events with repetitive structures," in *17th Annual Conference of the International Speech Communication Association*. ISCA, 2016, pp. 2631–2635.

[12] F. Kurth and K. Wilkinghoff, "Robust detection of jittered multiply repeating audio events using iterated time-warped ACF," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2018, pp. 276–280.

[13] M. Lasseck, "Acoustic bird detection with deep convolutional neural networks," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2018, pp. 143–147.

[14] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017, pp. 4077–4087.

[15] R. Li, J. Liang, and H. Phan, "Few-shot bioacoustic event detection: Enhanced classifiers for prototypical networks," in *7th Workshop on Detection and Classification of Acoustic Scenes and Events*. Tampere University, 2022.

[16] J. Martinsson, M. Willbo, A. Pirinen, O. Mogren, and M. Sandsten, "Few-shot bioacoustic event detection using an event-length adapted ensemble of prototypical networks," in *7th Workshop on Detection and Classification of Acoustic Scenes and Events*. Tampere University, 2022.

[17] H. Liu, X. Liu, X. Mei, Q. Kong, W. Wang, and M. D. Plumbley, "Segment-level metric learning for few-shot bioacoustic event detection," in *7th Workshop on Detection and Classification of Acoustic Scenes and Events*. Tampere University, 2022.

[18] J. Tang, *et al.*, "Few-shot embedding learning and event filtering for bioacoustic event detection technical report," DCASE2022 Challenge, Tech. Rep., 2022.

[19] K. Wilkinghoff and A. Cornaggia-Urrigshardt, "TempAdaCos: Learning temporally structured embeddings for few-shot keyword spotting with dynamic time warping," 2023, arXiv:2305.10816.

[20] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 5670–5674.

[21] V. Lostanlen, *et al.*, "Per-channel energy normalization: Why and how," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 39–43, 2019.

[22] K. Wilkinghoff, A. Cornaggia-Urrigshardt, and F. Gökgöz, "Two-dimensional embeddings for low-resource keyword spotting based on dynamic time warping," in *14th ITG Conference on Speech Communication*. VDE, 2021, pp. 9–13.

[23] J. S. Chung, *et al.*, "In defence of metric learning for speaker recognition," in *21st Annual Conference of the International Speech Communication Association*. ISCA, 2020, pp. 2977–2981.

[24] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "AdaCos: Adaptively scaling cosine logits for effectively learning deep face representations," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2019, pp. 10 823–10 832.

[25] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *6th International Conference on Learning Representations*, 2018.

[26] D. S. Park, *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *20th Annual Conference of the International Speech Communication Association*. ISCA, 2019, pp. 2613–2617.

[27] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2019, pp. 4690–4699.

[28] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, pp. 17:1–17:5, 2017.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, 2015.

[30] F. Pedregosa, *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.