SEGMENT-LEVEL METRIC LEARNING FOR FEW-SHOT BIOACOUSTIC EVENT DETECTION

Haohe Liu[1], *Xubo Liu*[1], *Xinhao Mei*[1], *Qiuqiang Kong*[2], *Wenwu Wang*[1], *Mark D. Plumbley*[1]

[1] Centre for Vision, Speech, and Signal Processing (CVSSP), University of Surrey, UK
[2] Speech, Audio, and Music Intelligence (SAMI) Group, ByteDance, China

ABSTRACT

Few-shot bioacoustic event detection is a task that detects the occurrence time of a novel sound given a few examples. Previous methods employ metric learning to build a latent space with the labeled part of different sound classes, also known as positive events. In this study, we propose a segment-level few-shot learning framework that utilizes both the positive and negative events during model optimization. Training with negative events, which are larger in volume than positive events, can increase the generalization ability of the model. In addition, we use transductive learning on the validation set during training for better adaptation to novel classes. We conduct ablation studies on our proposed method with different setups on input features, training data, and hyper-parameters. Our final system achieves an F-measure of 62.73 on the DCASE 2022 challenge task 5 (DCASE2022-T5) validation set, outperforming the performance of the baseline prototypical network 34.02 by a large margin. Using the proposed method, our submitted system ranks 2nd in DCASE2022-T5 with an F-measure of 48.2 on the evaluation set. The code of this paper is open-sourced[1].

Index Terms— few-shot learning, transductive learning, metric learning, audio event detection

1. INTRODUCTION

Few-shot learning (FSL) [1] is a machine learning problem that makes predictions based on the training data that contains limited information. Sound event detection (SED) [2] is a task that locates the onset and offset of certain sound classes. By combining the idea of FSL with SED [3], a system can detect a new type of sound with only a few examples. Few-shot SED is useful for audio data labeling, especially when the user needs to detect a new type of sound.

Most prior studies use a prototypical network [4] as the main architecture [5, 6, 7, 8]. Yang et al. [5] propose a mutual learning framework that employs transductive learning to iteratively improve the feature extractor and classifier, where transductive learning means the model has access to the test set without labels during the training process. A smoother manifold of embedding space can help extend the decision boundary and reduce the noise in data representation [9]. Tang et al. [6] propose to use embedding propagation [9] in few-shot SED to learn a smoother manifold by interpolating between the model output features based on a similarity graph. Data augmentations such as spec-augment and mixup are used in the method described in [7, 8]. There is also a spectrogram-cross-correlation-based method called template matching [10], which performs detection based on the normalized cross-correlation between example sound event and unlabeled data.

[1] https://github.com/haoheliu/DCASE_2022_Task_5

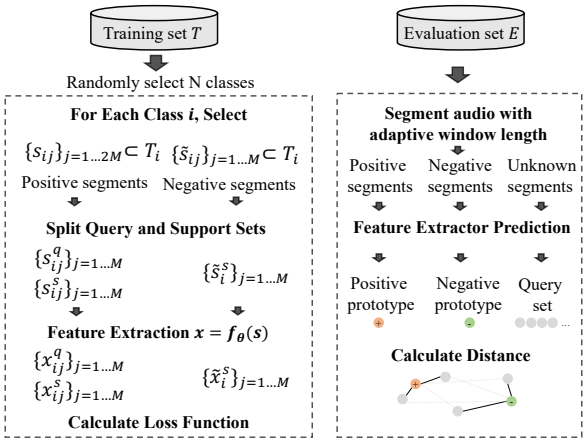
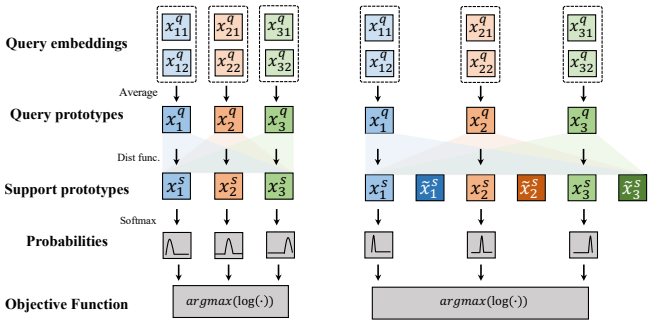


Figure 1: Training and evaluation procedure of the N-way-M-shot segment-level metric learning. M denotes the number of segments or embeddings.

Metric learning [11] refers to learning a distance function and feature space for a task. Previous metric-learning-based studies [5, 6] usually optimize the model with the labeled positive events, by grouping and separating the latent prototypes of the events with the same and different classes, respectively. The audio chunks that do not contain target events, which we refer to as negative events, are larger in volume but receive less attention. For example, in the DCASE 2022 task 5 development set [10], the duration of the negative events is 19.18 hours, accounting for 91.3% of the training data with a total duration of 21 hours.

In this paper, we propose a segment-level metric learning method that achieves state-of-the-art results on the few-shot bioacoustic detection task. As shown in Figure 1, our system operates on a segment level. Each sound event can contain multiple segments. We train a feature extraction network that maps the segments into latent embeddings, which are averaged into prototypes to represent different sound classes. To learn a robust latent space, we use a transductive learning scheme and propose to build contrastive loss with negative events. We also improve our method by using feature selection, data augmentation, and post-processing. We perform ablation studies to measure the effectiveness of each component. Our proposed method achieves an F-measure of 62.73 on the DCASE task 5 validation set.

This paper will be organized as follows. Section 2 provides an overview of our system. Section 3 introduces our methodology. Section 4 discusses the experimental setup. Section 5 reports the result and the ablation studies. Section 6 summarizes this work and provides a conclusion.



(a) Previous method. (b) Proposed method.

Figure 2: This figure illustrates N-way-M-shot metric learning when N=3 and M=2. (a) Visualization of the previous method, which only uses positive classes. (b) The proposed metric learning with the negative segments we used in our system. Support embeddings are omitted for simplicity. x_i and $\widetilde{x}_i$ stand for the positive and negative prototypes of class i.

2. SYSTEM OVERVIEW

We build our system using a prototypical network [4], which is widely used for metric-based few-shot learning. The training data $\mathbb{T} = (\mathbb{S}_i, \mathbb{Y}_i)|_{i=1}^{N_{\text{train}}}$ contains audio feature set $\mathbb{S}_i = \{s_i|y_i = 1\} \sqcup \{\widetilde{s}_i|y_i = 0\}$ and its corresponding label set $\mathbb{Y}_i = \{y_i|y_i \in \{0, 1\}\}$, where $\{s_i\}$ and $\{\widetilde{s}_i\}$ are the sets of positive and negative segments for class i, respectively, and N_{train} is the total number of training classes. The evaluation dataset $\mathbb{E} = (\mathbb{S}'_i, \mathbb{Y}'_i)|_{i=1}^{N_{\text{eval}}}$ also contains an audio feature set $\mathbb{S}'_i = \{s'_i\}$ and a label set $\mathbb{Y}'_i = \{y'_i\}$, where N_{eval} is the number of classes in the evalution set, $|\mathbb{S}'_i| = L_i$ and $|\mathbb{Y}'_i| = K$. Here we have $L_i \geq K$ because the evaluation set is partially labeled with only first K events. The validation set has the same structure as the evaluation set. The objective of our system is properly mapping different audio features into a latent embedding within a high-dimensional space, where similar audio features are closer together.

We use episodic training [12] to optimize our system in an N-way-M-shot way. As illustrated in Figure 2(a), N-way-M-shot means each training batch will select data from N classes. And for each classes i, the system will randomly select M segments $\{s^s_{ij}\}_{j=1...M}$ as support segments and another M segments $\{s^q_{ij}\}_{j=1...M}$ as query segments. All the segments in different classes have the same length. Then a feature extraction network (Section 3.1) will map these segments into fix-length embeddings, which are later averaged into query prototypes x^q_i and supporting prototypes x^s_i. The system is optimized by minimizing the distance between the query and support prototypes with the same class. To build a robust latent space and generalize better to the new class, we propose to use the negative event in metric learning and the transductive learning scheme in Section 3.2 and 3.3.

During evaluations, the audio file will be segmented using a sliding window with an adaptive segment length (Section 3.4). The segments in the labeled parts will be used to build positive and negative prototypes, which are treated as the latent representation of the positive and negative events in an audio file. The segments in the unlabeled part are the query set, which can be classified by calculating and comparing the distance with the positive and negative prototype (Section 3.5). And if the probability of one query belonging to a positive prototype is greater than a threshold h, it will be classified as positive. Consecutive positive predictions will be merged into one single event.

3. METHODOLOGY

3.1. Feature extraction network

Our feature extraction network f_θ is a convolutional neural network (CNN) based architecture that maps the audio feature s into a latent embedding x. In a similar way to the architecture proposed by [13], the network f_θ consists of three convolutional blocks with hidden channels of sizes 64, 128, and 64. Each convolutional block consists of three two-dimensional CNN layers with batch normalization and leaky rectified linear unit activations [14]. As a common trick in CNN-based network [13, 15], we apply 2×2 max-pooling after each block for downsampling and enlarging the reception field. The input and output of each convolutional block have a residual connection processed by a downsampling CNN layer. In order to maintain the same output dimension with different input lengths, we apply an adaptive average pooling at the end of the network. The final output feature map after adaptive pooling is a $C \times T \times F$ size block, which is the final latent embedding of s.

3.2. Segment-level metric learning

We propose to utilize negative segments within negative events during model optimization to learn a more robust representation, as illustrated in Figure 2(b). In a similar way to [3], we first divide the audio features into segments with equal length for metric learning. Then f_θ maps all the segments into latent embeddings. During optimization, we will calculate the class probabilities distributions of the query prototype x^q_i, which involves the distance calculation with all the positive and negative support prototypes. In this case, the model can learn a larger amount of contrastive information from the negative events on building the latent space. Specifically, we first calculate a distance matrix $\mathbf{D} = [\mathbf{d}^{(1)}, \mathbf{d}^{(2)}, ..., \mathbf{d}^{(N)}]^T$ according to Equation 1,

$$\mathbf{d}^{(i)}_{2j} = \|x^q_i - x^s_j\|_2, \ \mathbf{d}^{(i)}_{2j+1} = \|x^q_i - \widetilde{x}^s_j\|_2, \tag{1}$$

where $\mathbf{d}^{(i)} \in \mathbb{R}^{2N}$ stands for the distance between x^q_i and $2N$ support prototypes, and $\widetilde{x}^s_j$ denotes the support prototype for the negative events of class j. Then we optimize our model by maximizing the probability that x^q_i is close to the positive support prototype of class i, x^s_i, given by

$$\mathbf{d}'^{(i)} = \log(\text{Softmax}(-\mathbf{d}^{(i)})), \ l = \arg\max_\theta(\Sigma^N_{i=1}(\mathbf{d}'^{(i)}_{2i}), \tag{2}$$

where $0 \leq i, j \leq N, i, j \in \mathbb{N}$, and l is the objective function. Note that the learning process does not involve the query prototypes for negative events $\widetilde{x}^q_i$, because $\widetilde{x}^q_i$ and $\widetilde{x}^s_i$ are not guaranteed to have the same type of sound.

Data balancing is important in this task because different sound classes have different total durations [10]. In order to balance between classes, we sample each class with equal probability during the episodic training. In this way, the model has equal probabilities to attend to each class and will be less prone to overfitting [16].

3.3. Transductive learning

We adopt a transductive learning [17] approach during training, which means our model will be optimized both on the fully-labeled training set and the partially labeled evaluation data. Each file in

evaluation data has first K labeled events for a particular type of sound. We treat these K events as positive events and the remaining K chunks of audio in the labeled part as negative events. In the evaluation set, although the sound class of each file is not available, files with the same sound class should be in the same subfolder, and we treat each subfolder of the evaluation set as a different sound class. Even though the files within each subfolder may not always contain the same target sound, our experiment shows transductive learning in this way can still help the model gain better adaptation to the evaluation set (Section 3).

3.4. Adaptive segment length

We use the same segment length among all classes during training for the convenience of batch processing. But during evaluation, using the same segment length is not ideal. For example, using a segment length that is too long or too short will tend to have a high false negative rate or false positive rate, respectively. In the evaluation set, different animal or bird species have drastically different lengths of vocalization, ranging from 30 milliseconds to 5 seconds. Thus we choose to use adaptive segment lengths during evaluation.

$t_{\max}$ (s)	[0,0.1]	(0.1,0.4]	(0.4,0.8]	(0.8,3.0]	(3.0, ∞)
Length	8	$t_{\max}$	$t_{\max}$ / 2	$t_{\max}$ / 4	$t_{\max}$ / 8

Table 1: The segment length we use on dividing the evaluation audio file for different values of $t_{\max}$.

As shown in Table 1, we set different segment length for each audio file based on the max length of the labeled events $t_{\max} = \max(t_1, ..., t_K)$, where $t_1, ..., t_K$ denotes the duration of the K labeled positive events. We set the hop length as one-third of the window length. Note the parameters here are chosen by experience and not necessarily optimal.

3.5. Positive and negative prototypes

During the evaluation, we assume the first K labeled positive events do not contain too much variety, therefore we calculate the positive prototype by averaging the embeddings of the labeled positive segments. By comparison, building negative prototypes is more tricky because negative segments can contain many different kinds of sounds. So simply averaging all the negative embeddings would result in a sub-optimal representation of negative prototypes. To address these challenges, we choose to run our evaluation six times, each selecting 30 randomly selected negative segments within the labeled negative parts, and we average the predicted probabilities across time of six runs as the final prediction. The negative prototype in each run can have a chance to represent different sounds. This process is similar to the random subspace method [18], in which the ensemble of several estimators trained with a different subset of training data can outperform a single estimator optimized on full training data.

4. EXPERIMENTS

4.1. Dataset

DCASE2022-T5 The DCASE 2022 task 5 dataset[2] contains a training set, a validation set, and an official evaluation set. The training and validation set are both fully labeled. The official evaluation set has the labels of the first five positive events. Our result on the evaluation set is available on the DCASE 2022 Challenge result page[3]. The full label of the official evaluation set is not released at the time of writing, hence, we mainly report the result on the validation set in this paper. The validation during training is not meant to pick the best model. That's because we perform validation in a different way from evaluation. Similar to the training process, we calculate validation accuracy on a fix-length segment level without adaptive segment length. Therefore the best model on validation does not necessarily perform the best during evaluation. Nevertheless, we use the same validation process in our experiments, so the comparisons are fair in different settings. There are also similar ideas in [19, 20], which utilize the evaluation set for validations.

AudioSet-Aminal-SL AudioSet [21] is a large-scale dataset for audio research [13, 22]. Considering that the training set of DCASE2022-T5 only contains 47 different sound classes, we choose to use the strongly labeled part of the AudioSet dataset[4] to augment training data with a wider variety of sounds. To alleviate the domain mismatch problem, we only use sound labels that are related to animal vocalizations and do not overlap with other non-animal sounds. After data cleaning, we have 1796 pieces of audio with 37 classes from AudioSet. However, even if the sounds have the same label in the AudioSet, they can still sound very different. To alleviate this problem, we treat each audio file in AudioSet as its own class, so we have 1796 classes in this dataset, which is named AudioSet-Aminal-SL, where SL means strongly labeled. To balance the 1796 classes and 47 classes in AudioSet-Aminal-SL and DCASE2022-T5, we choose half classes from each dataset during episodic training.

4.2. Evaluation metric

We use the F-measure score, the official evaluation metric provided by the organizers of DCASE task 5, as our main evaluation metric. We also report system performance with the Polyphonic Sound Detection Score (PSDS) [23], which is a robust intersection-based sound event detection evaluation metric. In PSDS, we set the detection tolerance criterion (DTC) and the ground truth intersection criterion (GTC) to 0.5, and the maximum effective false positive rate to 100.0. Other parameters like the cross-trigger tolerance criterion (CTTC) are not used because our task is not polyphonic detection.

4.3. Experimental setup

Following [5], all the audio data are resampled to a 22.5 kHz sampling rate. The input feature of our system is the stack of PCEN [24] and ΔMFCC [25] features. In the short-time Fourier transform, we set the window length as 1024 and the hop size as 256. We set the mel-frequency dimension as 128. The input length of our model during training is 0.2 seconds. If the sound event is less than 0.2 seconds, zero-padding will be applied. The size of the embedding mentioned in Section 3.1 is 2048, in which $C = 64, T = 4, F = 8$. All the experiments use an initial learning rate of 0.001 with 0.65 exponential decay every 10 epochs. We perform validation after every epoch. We perform validation in a 3-way-5-shot manner since there are only three classes (HB, ME, PB) in the validation set. We will stop model training if the validation accuracy does not improve

[2] https://zenodo.org/record/6482837

[3] https://dcase.community/challenge2022/task-few-shot-bioacoustic-event-detection-results

[4] https://research.google.com/audioset

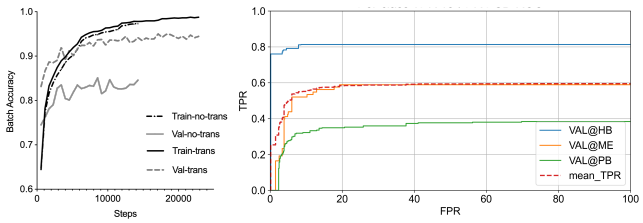
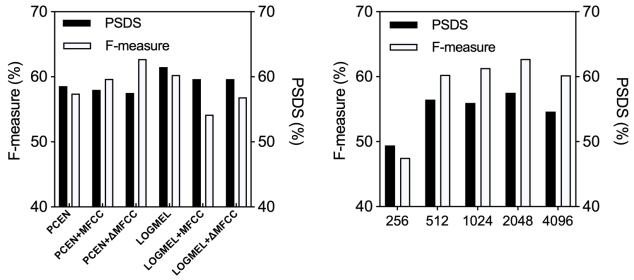


Figure 3: The training and validation accuracy at different training steps, both with and without transductive learning, are shown in the left figure. The right figure is the PSD-ROC curve of our proposed system. *HB*, *ME*, and *PB* are three subsets in the validation set. The area under *mean_TPR* curve is PSDS, which indicates the system's overall performance. TPR and FPR stand for true positive rate and false positive rate, respectively.



(a) Input features (b) Embedding dimensions

Figure 4: Ablation study on (a) input features; and (b) embedding dimension. We report the PSDS and F-measure on the DCASE2022-T5 validation dataset.

for 10 consecutive epochs. And the model with the best validation accuracy is used for calculating metrics scores. To make full use of training data, we implement a dynamic data loader that generates training data with a random starting time on the fly. We assume the duration of one vocalization for a certain animal does not vary significantly. Therefore, we design the post-processing strategy for a sound class based on maximum length of positive event $t_{\max} = \max(t_1, ..., t_K)$. We will remove a positive detection if its length is smaller than $\alpha * t_{\max}$ or greater than $\beta * t_{\max}$. We use different combinations of $\beta = 2.0, \alpha = [0.1, 0.2, ..., 0.9], h = [0.0, 0.05, ..., 0.95]$ to calculate data points [23], draw the PSD-ROC curve, and calculate PSDS. We choose the best F-measure among all β, α, h combinations as the final F-measure.

5. RESULT

Method	Pre.	Rec.	F-measure	PSDS
Template Matching [10]	2.42	18.32	4.28	N/A
ProtoNet (official) [10]	36.34	24.96	29.59	N/A
ProtoNet (our impl)	23.26	63.27	34.02	46.10
Proposed	69.30	57.30	**62.73**	**57.52**

Table 2: Comparisons with baseline template matching and prototypical network methods. *Pre.* and *Rec.* stand for precision and recall, respectively. The first two methods [10] did not report PSDS results. All the metrics are written in percentages.

The performance of our system on the validation set is reported in Table 2. The F-measure score of template matching and our re-implemented prototypical network baseline [10] is 4.28 and 34.02, respectively. Our system outperforms the baselines by a large margin with an F-measure score of 62.73 and a PSDS of 57.52.

Setting	F-measure (%)	PSDS (%)
Proposed	**62.73**	**57.52**
w/o Negative contrast	55.25	54.95
w/o Transductive learning	56.37	54.50
w/o Post processing	57.27	55.90

Table 3: Ablation study of the proposed method.

Training data	F-measure (%)	PSDS (%)
DCASE	**62.73**	57.52
AudioSet-Aminal-SL	46.83	51.00
AudioSet-Aminal-SL & DCASE	58.48	**58.77**

Table 4: A study on using different training datasets. DCASE stands for the DCASE2022-T5 dataset.

As is shown in Figure 3, using transductive learning can significantly improve the validation accuracy. And the class-wise ROC indicates the *HB* class, which is mostly mosquito sounds, is the easiest one to detect. Class *PB* is the hardest class perhaps because it mainly consists of sparse bird calls with strong background noise. Class *ME* achieves an average performance in the validation set.

We perform a study on the effect of the input feature. As shown in Figure 4(a), the performance of F-measure and PSDS is not always consistent, and we use F-measure to guide our selection considering it is widely used in prior studies [10]. By comparing the F-measure score, PCEN+ΔMFCC appears to be a good feature combination on the validation set. We also compare different embedding dimension in Figure 4(b). We change the dimension by altering the dimension of F in the adaptive average pooling. We notice a dimension of 512 can considerably improve over 256, and 2048 has the best performance among all the settings.

We perform ablations on each of the components we proposed. As shown in Table 3, if we remove the negative segments, the performance drops considerably. The trend is the same with transductive learning and post-processing. We also study the effect of training data. In Table 4, we can see that the best F-measure score is achieved using the DCASE2022-T5 only. Using AudioSet-SL leads to an F-measure of 46.83 and a PSDS of 51.00. By combining two datasets we got an F-measure of 58.48 and a best PSDS of 58.77. We hypothesize that the degradation of F-measure using AudioSet is caused by domain mismatch on training data. However, combining two datasets yield the best PSDS, which means using AudioSet data can lead to a general improvement across all threshold and post-processing settings instead of getting a single best system with a high F-measure. This indicates that PSDS might be a suitable metric for the community to reference in this task.

6. CONCLUSIONS

This paper proposes a new framework for few-shot sound event detection. Our proposed metric learning with negative segments and the transductive learning scheme can significantly improve model performance. On the input feature, our experiment shows that PCEN with ΔMFCC yields the best performance in our settings. Our result also indicates that PSDS might be a useful metric to evaluate the model's overall performance by considering multiple thresholds and post-processing settings.

7. ACKNOWLEDGMENT

This research was partly supported by a Newton Institutional Links Award from the British Council (Grant number 623805725), BBC Research and Development, Engineering and Physical Sciences Research Council (EPSRC) Grant EP/T019751/1 "AI for Sound", and a PhD scholarship from the Centre for Vision, Speech and Signal Processing (CVSSP), Faculty of Engineering and Physical Science (FEPS), University of Surrey. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising.

8. REFERENCES

[1] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," *International Conference on Learning Representations*, 2017.

[2] 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.

[3] Y. Wang, J. Salamon, N. J. Bryan, and J. P. Bello, "Few-shot sound event detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 81–85.

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

[5] D. Yang, H. Wang, Y. Zou, Z. Ye, and W. Wang, "A mutual learning framework for few-shot sound event detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2022, pp. 811–815.

[6] T. Tang, Y. Liang, and Y. Long, "Two improved architectures based on prototype network for few-shot bioacoustic event detection," DCASE2021 Challenge, Tech. Rep., 2021.

[7] Y. Zhang, J. Wang, D. Zhang, and F. Deng, "Few-shot bioacoustic event detection using prototypical network with background classs," DCASE2021 Challenge, Tech. Rep., 2021.

[8] M. Anderson and N. Harte, "Bioacoustic event detection with prototypical networks and data augmentation," *arXiv:2112.09006*, 2021.

[9] P. Rodríguez, I. Laradji, A. Drouin, and A. Lacoste, "Embedding propagation: Smoother manifold for few-shot classification," in *European Conference on Computer Vision*. Springer, 2020, pp. 121–138.

[10] 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.

[11] B. Kulis *et al.*, "Metric learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2013.

[12] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1446–1455.

[13] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[14] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv:1505.00853*, 2015.

[15] H. Liu, L. Xie, J. Wu, and G. Yang, "Channel-wise subband input for better voice and accompaniment separation on high resolution music," *Proc. Interspeech 2020*, pp. 1241–1245, 2020.

[16] Y. Gong, Y.-A. Chung, and J. Glass, "Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3292–3306, 2021.

[17] M. Boudiaf, H. Kervadec, Z. I. Masud, P. Piantanida, I. Ben Ayed, and J. Dolz, "Few-shot segmentation without meta-learning: A good transductive inference is all you need?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 979–13 988.

[18] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[19] Q. Kong, Y. Cao, H. Liu, K. Choi, and Y. Wang, "Decoupling magnitude and phase estimation with deep resunet for music source separation," *arXiv:2109.05418*, 2021.

[20] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio spectrogram transformer," *arXiv:2104.01778*, 2021.

[21] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 776–780.

[22] Q. Kong, H. Liu, X. Du, L. Chen, R. Xia, and Y. Wang, "Speech enhancement with weakly labelled data from audioset," *arXiv:2102.09971*, 2021.

[23] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 61–65.

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

[25] M. A. Hossan, S. Memon, and M. A. Gregory, "A novel approach for mfcc feature extraction," in *2010 4th International Conference on Signal Processing and Communication Systems*, 2010, pp. 946–953.