FMSG SUBMISSION FOR DCASE 2023 CHALLENGE TASK 4 ON SOUND EVENT DETECTION WITH WEAK LABELS AND SYNTHETIC SOUNDSCAPES

Technical Report

Yang Xiao, Tanmay Khandelwal, and Rohan Kumar Das

Fortemedia Singapore, Singapore

{xiaoyang, rohankd}@fortemedia.com, f20170106p@alumni.bits-pilani.ac.in

ABSTRACT

This report presents the systems developed and submitted by Fortemedia Singapore (FMSG) for DCASE 2023 Task 4A, which focuses on sound event detection with weak labels and synthetic soundscapes. Our approach primarily involves integrating features from Bidirectional Encoder representation from Audio Transformers (BEATs) and frequency dynamic (FDY)-convolutional recurrent neural network (CRNN) into a single-stage setup. We focus on three main directions to enhance our approach. Firstly, we curate an external dataset from AudioSet by establishing relationships between AudioSet sound event categories and the target sound events. Secondly, we utilize multiple aggregation methods to leverage the strengths of different methods. Lastly, we employ the asymmetric focal loss (AFL) function to adjust the training weights based on the model's training difficulty. Additionally, we use data augmentation techniques to prevent overfitting, apply adaptive post-processing methods, and experiment with an ensemble of multiple subsystems to improve the generalization capability of our system. Our method achieves the top PSDS1 and PSDS2 scores of 0.557 and 0.854, respectively, on the development set. Further, on the public evaluation set, our approach achieves the highest PSDS1 and PSDS2 scores of 0.607 and 0.875, respectively.

Index Terms— sound event detection, semi-supervised learning, FDY-CRNN, BEATs, data augmentation

1. INTRODUCTION

Sound event detection (SED) is a task that involves detecting sound events from acoustic signals and accurately classifying them into specific event categories with corresponding timestamps, considering various acoustic environments [1, 2, 3]. DCASE 2023 Task 4A is specifically focused on SED, aiming to detect sound events and their temporal boundaries in both Scenario 1 (react fast) and Scenario 2 (avoid class confusion). The task utilizes a substantial amount of weakly labeled and unlabeled data. It serves as a continuation of DCASE 2022 Task 4. The evaluation for this year's task incorporates a threshold-independent implementation of the polyphonic sound event detection score (PSDS) [4]. Additionally, the baseline approach incorporates the use of Bidirectional Encoder representation from Audio Transformers (BEATs) [5] embeddings.

In this report, we outline our contributions in our submission for DCASE 2023 Task 4A as described:

- We utilize the frame-level embeddings generated by the pretrained BEATs model in late-fusion with the frequency dynamic (FDY)-convolutional recurrent neural network (CRNN) [6] and then fed into the recurrent neural network (RNN) with multi-layer Perceptron (MLP) classifier.
- We propose a method for frame-level concatenation that involves utilizing several aggregation techniques to merge the frame-level outputs, which are then averaged to generate the final output.
- To tackle the prevalent class imbalance in SED datasets, we integrate the use of asymmetrical focal loss (AFL) [7] as a means of addressing this challenge.
- We put forth a method to generate additional potential data from AudioSet [8] by leveraging the mapping relationship between the original 527 sound categories in AudioSet and the 10 target sound event categories.

Additionally, in order to enhance performance, we employ data augmentation techniques, exponential softmax pooling function, apply adaptive median-filtering [9] to smooth the outputs for each class and utilize weakified labels [10, 11] with the weak training method.

2. PROPOSED APPROACH

2.1. Baseline

The baseline architecture, referred to as convolutional recurrent neural network (CRNN)[12], combines convolutional neural network (CNN) and RNN components, as illustrated in Fig.1 (a). The CNN component consists of 7 blocks, with each block having 16, 32, 64, 128, 128, 128, and 128 filters, respectively. Each block utilizes a kernel size of 3×3 and applies average-pooling [13] operations of [2, 2], [2, 2], [2, 1], [2, 1], [2, 1], [2, 1], [2, 1] per layer. The RNN component comprises two layers of 128 bidirectional gated recurrent units (Bi-GRU) [14]. Following the RNN block, an attention pooling layer is employed, involving a linear layer with softmax activations multiplied by a linear layer with sigmoid activations.

2.2. Network

In this work, we employed FDY-CRNN from [6], which uses frequency adaptive kernels to enforce frequency dependency in 2D convolutions. In the baseline CRNN [12] architecture depicted in Figure 1 (a), we replaced the standard 2D convolutional blocks with FDY-convolutional blocks, as illustrated in Figure 1 (b). The CNN part consists of 7 blocks with the same number of filters as in the baseline. In the FDY-convolutional block, batch normalization [15] and gated linear units [16] are used. The RNN part consists of 2 layers of Bi-GRU with 256 hidden units.

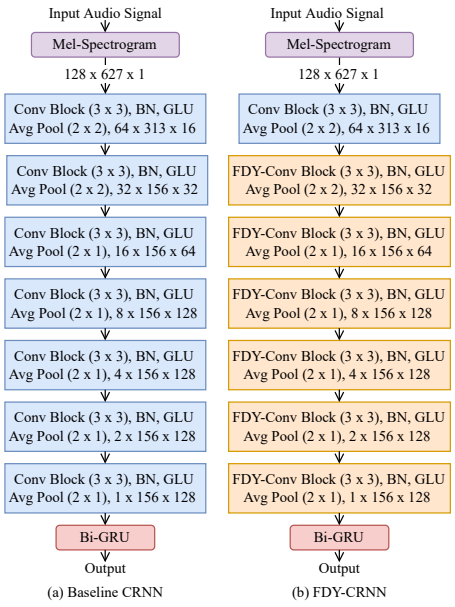

Figure 1: *Architecture of (a) CRNN (Baseline) (b) FDY-CRNN.*

2.3. Pretrained model

In this work, we utilize the pretrained BEATs [5] model that has achieved state-of-the-art performance on AudioSet with an mAP of 0.486. It is an iterative self-supervised framework for audio representation learning, utilizing an acoustic tokenizer and an audio semi-supervised learning model. Unlike previous models, BEATs employs a self-distilled tokenizer for converting audio signals into discrete labels. We use it to construct frame-level embeddings of size 768, which aligns with the recently released baseline approach.

2.4. Aggregation

In our proposed approach, we extract the frame-level embeddings from the BEATS, then combine them with the FDY-CRNN in a late-fusion fashion as represented in Figure 2. To make sure these embeddings match the output of the CNN part of our model in terms of temporal dimension, we use the technique called aggregation. The

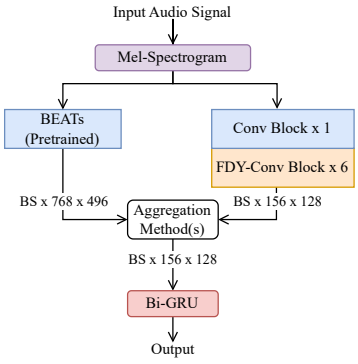

Figure 2: *Concatenation of frame-wise embeddings from BEATs and features from FDY-CRNN, where BS denotes batch size.*

embeddings can be integrated with the output of the CNN part of our model using several aggregation methods:

- **Frame:** It involves taking the last state of an RNN that has been fed as the sequence of embeddings.
- **Interpolate:** It uses nearest-neighbour interpolation to adjust the time resolution.
- **Adaptive average pooling (pool1d):** It divides the embeddings into several regions and calculates the average of each region to create a specified output size.
- **Adaptive max pooling:** It is similar to adaptive average pooling, but instead of taking the average of each region, it takes the maximum value.
- **Conv1d:** It is an aggregation strategy that can learn local temporal patterns in the embeddings by using a 1D convolution layer, which can help the model to understand the structure of the embeddings better along the time dimension.

To leverage the strengths of different aggregation methods and save rich local information from the embeddings, we also propose an approach to use multiple aggregation methods. First iterates over a list of aggregation methods, each aggregation shares the same embedding and CNN feature as the input to fuse. The aggregated input is then passed through an individual RNN layer to generate the frame-wise score. We collect the frame-wise score from the iteration of aggregation methods and average it as the final frame-wise output.

2.5. Data augmentation

During the training process, we employed multiple data augmentation techniques. These techniques include time-masking (TM) [17], frequency-masking (FM) [17], mixup [12, 18], and filter augmentation (FA) [6]. TM involves applying weights to specific time-frequency representation bins, while FM entails blocking a portion of the frequency spectrum of an audio signal by setting those frequency components to zero or attenuating them. Mixup randomly combines selected samples using a mixing parameter, facilitating linear interpolation to enhance the model's robustness. Additionally, FA utilizes varying weights on random frequency regions, which has demonstrated a significant improvement in SED performance.

2.6. Loss function

The AFL [7] function is utilized to regulate the training weight based on the difficulty or ease of training the model. It calculates a value for each data point, considering the target sound event (y_k) and the predicted sound event (p_k). The AFL function is expressed as follows:

$$l_{AFL}(p, y) = \sum_{n=1}^{K} [(1 - p_k)^{\gamma} y_k \ln p_k + (p_k)^{\zeta} (1 - y_k) \ln(1 - p_k)] \quad (1)$$

here, the parameters γ and ζ are hyperparameters provided as input to the loss function. They control the weighting of active and inactive frames, influencing the contribution of each data point to the overall loss.

2.7. Curated set

In this subsection, we introduce the proposed method to utilize more potential data from AudioSet. First, we establish a map-

ping relationship between AudioSet's original sound event 527 categories and the 10 target sound event categories. Following the work in [19], we derive the mapping rate for each of the 527 categories by observing their frequency in association with the target sound categories. We set a threshold of 80% to consider the mapping relations above the threshold, and then those are used to select the additional strongly labeled data from Audioset. However, we noticed an imbalance in the data as there is a very high number of clips labeled with "Speech". To address this, we removed all the clips that are only labeled as "Speech". Then we converted the labels from AudioSet to the 10 target sound categories of DCASE 2023 Task4. In this way, we curated an additional strongly labeled set of 2,161 clips. We merge them with the provided AudioSet external strong-label audio clips for training the models.

2.8. Median filtering (MF)

In all experiments, we employed adaptive median filtering (MF) technique [9]. This approach involved the application of median filters with varying window sizes, denoted as Win, based on the duration of real-life event categories c. The specific window sizes for each event category are presented below:

$$Win_c = duration_c \times \beta_c \quad (2)$$

In order to handle event categories with significant duration variation, we employed a dynamic approach by setting the median duration $duration_c$ as the reference. For this purpose, we initially set the parameter β_c=$\frac{1}{3}$ and fine-tuned the window sizes based on the development set, ensuring optimal performance.

3. EXPERIMENTAL SETUP

3.1. Dataset and feature extraction

This work utilizes the DCASE 2023 Task 4A dataset for the detection and classification of acoustic scenes and events. The dataset consists of 10-second audio clips. The development training set is further divided into the mentioned subsets:

- 1,578 real recordings with weak annotations
- 14,412 real recordings, unlabeled in the domain training set
- 10,000 synthetic recordings with strong annotations [20].
- 3,470 real recordings + 2,161 curated real recordings as described in Section 2.7 (External Set), both sourced from AudioSet
- 1,168 real recordings with strong annotations (Validation Set)

All these audio clips are resampled to a 16 kHz mono channel using Librosa. They are segmented with a window size of 2048 samples and a hop length of 256 samples. Short-time Fourier transform was applied to extract spectrograms. Mel-filters are then used to create log-mel spectrograms spanning from 0 to 8 kHz. Clips shorter than 10 seconds are padded with silence if needed.

3.2. Training method

For all the experiments, a batch size of 96 was used, comprising 1/4 of the strong set, 1/4 of the weak set, and 1/2 of the unlabeled set. The Adam [21] optimizer was employed with a learning rate of 0.001. An exponential warmup was applied for the initial 50 epochs, and no early stopping was implemented during the training process.

3.3. Evaluation metric

The evaluation of our systems was based on the recently introduced threshold-independent [4] implementation of the polyphonic sound event detection scores (PSDS), defined in [22]. We conducted evaluations on two different scenarios to highlight distinct system properties. In Scenario-1, the system's ability to react promptly upon sound event detection was emphasized, with a focus on the temporal localization of the sound event. In contrast, Scenario-2 placed less importance on reaction time and more on avoiding class confusion.

3.4. Ensemble systems

Ensemble modeling is a technique that leverages the strengths of multiple models to improve overall performance and enhance the generalization capability of a system. By combining the predictions from different models, ensemble methods can effectively reduce individual model biases and errors, leading to more accurate and robust results. In the context of our system, the use of ensemble modeling plays a crucial role in achieving superior performance. By grouping together the best-performing models (E-1, E-2, E-3, and E-4) as described below, we are able to capitalize on their respective strengths and expertise in different aspects of the task.

- **E-1:** It is an ensemble of the best 5 models based on their PSDS1 score obtained from the public evaluation set.
- **E-2:** It is an ensemble of the top 10 models based on their PSDS1 score obtained from the development set.
- **E-3:** It is an ensemble of the top 4 models when ranked by their PSDS2 score obtained from the development set.
- **E-4:** It is an ensemble of the top 10 models based on their PSDS2 scores obtained from the public evaluation set.

These groupings work collaboratively to extract the best aspects from the highest-performing models. To generate final predictions, we aggregate the individual predictions from all the models and calculate their average. This approach ensures that every model contributes to the overall performance of the ensemble system.

4. RESULTS AND ANALYSIS

In this section, we present the findings of the 8 submitted systems, which include 4 single-systems and 4 ensemble systems. The results of the challenge baselines are also discussed in brief.

4.1. Baseline

We first present the outcomes challenge baselines provided by the organizers, which are reported in Table 1. Baseline-1 corresponds to the CRNN baseline described in Section 2.1 without any external dataset, while Baseline-2 corresponds to the CRNN base-

Table 1: *Performance of the baseline systems provided for DCASE 2023 Task 4A on the development set.*

System	PSDS1	PSDS2
Baseline-1	0.359	0.562
Baseline-2	0.364	0.576
Baseline (BEATs)-1	0.500	0.762
Baseline (BEATs)-2	0.491	0.787

Table 2: *Performance in terms of PSDS1 and PSDS2 of different single-systems with various configurations on the development set, including the number of stages, BEATs utilization, aggregation methods (Agg), time-masking (TM), frequency-masking (FM), filter augmentation (FA), mixup, external set integration (Ext), exponential softmax (ES), asymmetrical focal loss (AFL), and median filtering (MF).*

System	Stages	BEATs	Agg	TM	FM	FA	Mixup	Ext	ES	AFL	MF	PSDS1	PSDS2
S-1	2			✓	✓	✓	✓			✓	✓	0.464	0.711
S-2	1	✓	all-5	✓	✓		✓	✓			✓	**0.543**	0.801
S-3	1	✓	pool1d	✓	✓		✓		✓	✓	✓	0.098	**0.845**
S-4	1	✓	pool1d	✓	✓		✓	✓		✓	✓	0.539	0.793

Table 3: *Performance of the single-systems provided for DCASE 2023 Task 4A on the public evaluation set*

System	PSDS1	PSDS2
S-1	0.455	0.705
S-2	0.566	0.850
S-3	0.102	0.849
S-4	**0.602**	**0.862**

line with 3,470 real strong clips from AudioSet, mentioned in Section 3.1. Additionally, the table presents the results for the newly introduced baselines that utilize the pretrained BEATs model. Baseline (BEATs)-1 without any external set achieved a PSDS1 score of 0.500 and a PSDS2 score of 0.762, showing improved performance compared to the previous baselines. Similarly, Baseline (BEATs)-2, which utilizes the 3,470 real strong clips from AudioSet achieved a PSDS1 score of 0.491 and a PSDS2 score of 0.787. These results demonstrate the effectiveness of incorporating the pretrained BEATs model in the baselines, leading to improved performance in terms of both PSDS1 and PSDS2 metrics.

4.2. Single-systems

In this subsection, we provide a comprehensive overview of the different system configurations and their corresponding performances as reported in Table 2, facilitating straightforward comparison and analysis. The S-1 system, which builds upon our prior work [10, 11, 23], adopts a two-stage setup without utilizing any external datasets or embeddings. Significantly surpassing both Baseline-1 and Baseline-2, this system achieves a remarkable PSDS1 score of 0.464 and a PSDS2 score of 0.711. The S-2 system incorporates BEATs and combines all-5 of the aggregation methods (Agg) outlined in Section 2.4. It also integrates the complete external set (Ext), encompassing the external AudioSet and curated data. Overall, the S-2 system outperforms the BEATs-based baselines reported in Table 1, achieving notable scores of 0.543 and 0.801 for PSDS1 and PSDS2, respectively.

Similarly, leveraging the pretrained BEATs model, the S-3 system concentrates on maximizing the PSDS2 score. It employs the pool1d aggregation method and also incorporates weakified labels [10, 11], utilizing weak training with an exponential softmax (ES) function. By incorporating AFL with γ=0.125 and ζ=4, the S-3 system achieves the highest PSDS2 score of 0.845 on the development set. Lastly, the S-4 system combines the pool1d aggregation method with BEATs and integrates the external set (Ext). Additionally, it incorporates AFL with γ=0.625 and ζ=1, enhancing its performance. As depicted in Table 3, which contains the reported scores for the single-systems on the public evaluation set, we observe that our single-system S-4 achieves the highest PSDS1 score of 0.602 and PSDS2 score of 0.862 on the public evaluation set.

Table 4: *Performance of our submitted ensemble systems on the development and public evaluation set of DCASE 2023 Task 4A.*

System	Development set		Public evaluation set	
	PSDS1	PSDS2	PSDS1	PSDS2
E-1	0.544	0.801	**0.607**	0.863
E-2	**0.557**	0.812	0.595	0.873
E-3	0.098	**0.854**	0.082	0.848
E-4	0.551	0.813	0.592	**0.875**

4.3. Ensemble systems

The performance of the submitted ensemble systems, presented in Table 4, highlights their comparative results in relation to DCASE 2023 Task 4A on both the development and the public evaluation sets. As discussed in Section 3.4, the ensemble system E-1 combines the best 5 models employing AFL with γ=0.625 and ζ=1. This ensemble system considers the entire external set and incorporates diverse variations of the aggregation methods outlined in Section 2.4, resulting in the highest PSDS1 score of 0.607 on the public evaluation set. Ensemble system E-2, comprising the top 10 models from the development set, employs different combinations of aggregation techniques, filter augmentation in a few selected systems, and AFL with γ=0.625 and ζ=1 in certain models. These variations collectively contribute to the highest PSDS1 score of 0.557 on the development set. Furthermore, ensemble E-3 utilizes weakified labels, the weak training method, and an exponential softmax function in conjunction with AFL using γ=0.125 and ζ=4, leading to the highest PSDS2 score of 0.854 on the development set. Similarly, ensemble system E-4 prioritize on PSDS2 score and then incorporate variations in aggregation methods, filter augmentation in some systems, and AFL with γ=0.625 and ζ=1 in certain models. These variations culminate in the highest PSDS2 score of 0.875 on the public evaluation set.

5. CONCLUSION

This technical report describes our submission for the DCASE 2023 Task 4A. Our approach demonstrates the effectiveness of a unified framework that combines frame-level embeddings from pretrained BEATs with features extracted from FDY-CRNN. We have implemented several techniques to further improve the performance of our system, including the curation of an external dataset from AudioSet, utilization of various feature fusion methods, and integration of the AFL function. Additionally, we have employed data augmentation techniques, adaptive median filtering, and performed ensemble of multiple subsystems using our developed systems. On the development set, our ensemble systems obtain the best PSDS1 and PSDS2 of 0.557 and 0.854, respectively. In addition, we achieved the best PSDS1 score of 0.607 and PSDS2 score of 0.875 on the public evaluation set.

6. REFERENCES

[1] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "SONYC: A system for monitoring, analyzing, and mitigating urban noise pollution," *Communications of the ACM*, vol. 62, no. 2, pp. 68–77, 2019.

[2] J. P. Bello, C. Mydlarz, and J. Salamon, "Sound analysis in smart cities," *Springer International Publishing*, pp. 373–397, 2018.

[3] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadakis, and A. Bauer, "Monitoring activities of daily living in smart homes: Understanding human behavior," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 81–94, 2016.

[4] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold-independent evaluation of sound event detection scores," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1021–1025, 2022.

[5] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," 2022. [Online]. Available: https://arxiv.org/abs/2212.09058

[6] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, "Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection," 2022. [Online]. Available: https://arxiv.org/abs/2203.15296

[7] K. Imoto, S. Mishima, Y. Arai, and R. Kondo, "Impact of sound duration and inactive frames on sound event detection performance," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 860–864, 2021.

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

[9] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution-augmented transformer for semi-supervised sound event detection," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2020.

[10] T. Khandelwal, R. K. Das, A. Koh, and E. S. Chng, "FMSG-NTU submission for DCASE 2022 Task 4 on sound event detection in domestic environments," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[11] ——, "Leveraging audio-tagging assisted sound event detection using weakified strong labels and frequency dynamic convolutions," *IEEE Statistical Signal Processing Workshop*, 2023.

[12] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for DCASE 2019 task 4 technical report," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2019.

[13] C.-Y. Lee, P. Gallagher, and Z. Tu, "Generalizing pooling functions in CNNs: Mixed, gated, and tree," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2017.

[14] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

[15] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating deep network training by reducing internal covariate shift," *International Conference on Machine Learning (ICML)*, pp. 448–456, 2015.

[16] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *International Conference on Machine Learning (ICML)*, pp. 933–941, 2017.

[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *Interspeech*, pp. 2613–2617, 2019.

[18] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017. [Online]. Available: https://arxiv.org/abs/1710.09412

[19] K. He, X. Shu, S. Jia, and Y. He, "Semi-supervised sound event detection system for DCASE 2022 Task 4," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[20] F. Ronchini, R. Serizel, N. Turpault, and S. Cornell, "The impact of non-target events in synthetic soundscapes for sound event detection," *Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop*, pp. 115–119, 2021.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[22] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 61–65, 2020.

[23] T. Khandelwal and R. K. Das, "Dynamic thresholding on fixmatch with weak and strong data augmentations for sound event detection," *International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 428–432, 2022.