CROSS-DIMENSIONAL INTERACTION WITH INVERTED RESIDUAL TRIPLET ATTENTION FOR LOW-COMPLEXITY SOUND EVENT DETECTION

Tanmay Khandelwal and Rohan Kumar Das

Fortemedia Singapore, Singapore
f20170106p@alumni.bits-pilani.ac.in, rohankd@fortemedia.com

ABSTRACT

Attention mechanisms have been widely used in a variety of sound event detection (SED) tasks, owing to their ability to build interdependencies among channels or spatial locations. The existing state-of-the-art (SOTA) architectures and attention modules incorporated in SED have a high computational cost in terms of the number of parameters. To address this issue, we propose a lightweight module utilizing triplet attention on an inverted residual network (IRN) referred to as an inverted residual triplet attention module (IRTAM) for replacing the standard 2D convolutional neural network. The IRTAM captures cross-dimensional interdependencies using the rotation operation followed by residual transformations with a three-branch structure embedded in IRN. On DCASE 2022 Task 4 validation set, the proposed lightweight module improves the performance of the baseline by 34.1% in terms of polyphonic sound event detection score and achieves SOTA results with only 27.6% parameters of the baseline.

Index Terms: sound event detection, low-complexity, triplet attention, inverted residual network

1. INTRODUCTION

Sounds help in better understanding our surroundings and in detecting environmental changes. The ability to recognize and classify sound events in our surroundings is inherent in the human body. The sound event detection (SED) systems *automate* this process to detect the sound events to mark their corresponding onset and offset. It has important practical applications as well as theoretical significance and has been applied to audio surveillance in environments such as smart-homes, cities, and monitoring biodiversity.

Real-world audio recordings frequently contain numerous overlapping sound occurrences. Recent advances in predicting and recognizing these overlapping events have shifted from traditional methods like Gaussian mixture models [1], hidden Markov models [2], and support vector machines [3] to advanced deep learning techniques. The recent success of convolutional recurrent neural networks (CRNNs) [4] and transformer [5] structures have achieved state-of-the-art (SOTA) results in the field of SED. These modern, cutting-edge structures demand high computing resources that are beyond the capacity of many embedded and mobile applications. Therefore, reducing the number of parameters in a SED model allows the method to be fit for systems with limited resources while also decreasing the training time.

Most of the previously built systems [6, 7] proposed the use of *depthwise separable convolutions* and showed the system's effectiveness with reduced parameters. Another way to target an effective *low-complexity* SED system is to use *attention mechanisms* [8]. In human perception, attention refers to the process of selectively concentrating on parts of the given information while ignoring the rest. This mechanism aids in the refinement of perceived information while preserving its context. In the case of deep learning systems with a basic building block as the 2D convolutional layer, filters capture local spatial patterns along all input channels and generate feature maps jointly encoding the time-frequency and channel information.

Several works have been aimed at capturing either *spatial* or *channel* attention, done by building dependencies among channels or weighted spatial masks for *spatial* attention. One such promising approach is a component called the squeeze and excitation (SE) [9] block, which can be seamlessly integrated into the convolutional neural network (CNN). This SE block removes the spatial dependency by using global average pooling to learn a channel-specific descriptor, which is then used to rescale the input feature map to highlight only useful channels. The SE block was succeeded by the convolutional block attention module (CBAM) [10], which emphasized the importance of providing robust representative attention by combining *spatial* and *channel* attention. This method of combining *spatial* attention and *channel* attention improved the performance compared to the SE block. However, most attention modules add substantial computational overhead, and stacking these complex modules usually ignores the interdependence between *spatial* dimensions and *channel* dimension of the input feature.

In this work, we devote to incorporating *cross-dimensional* interaction while computing attention weights to provide rich feature representations for *low-complexity* SED systems by a novel inverted residual *triplet attention* module (IRTAM) that uses a three-branch structure, where each branch is responsible for aggregating *cross-dimensional* interactive features. We summarize the major contributions of this work as follows:

- Inspired from *MobileNetV2* [11], we propose to incorporate an inverted residual network (IRN) with a linear bottleneck to replace the standard 2D convolution block. The IRN makes the SED model suitable to be deployed for *real-time* applications on low computational devices.
- We propose to introduce a *triplet attention* [12] module into the IRN at a negligible computational overhead to effectively learn *cross-dimensional* interaction. The attention module is made up of *three branches*, each of which is responsible for capturing the *cross-dimensional* interaction between the input's *spatial* dimensions and *channel* dimension.

We consider the two-stage system developed by [13–15] for the detection and classification of acoustic scenes and events (DCASE) 2022 Task 4 participation for the studies in this work. We also used *data augmentation* and *adaptive post-processing* techniques to increase the robustness of the developed system.

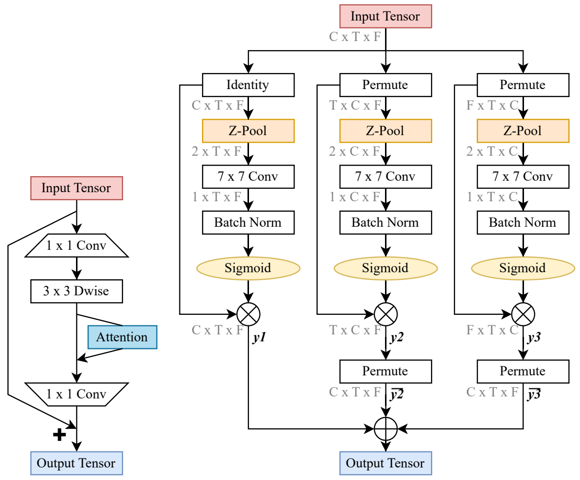
2. SOUND EVENT DETECTION SYSTEM

2.1. Baseline

The baseline [16] architecture, adopted from the DCASE Task 4 Challenge 2022, is a CRNN that combines a CNN and a recurrent neural network (RNN). The CNN block is composed of 7 layers, each with 16, 32, 64, 128, 128, 128, and 128 filters. The kernel size for each convolutional layer is 3×3 and each layer is followed by a Gaussian error linear unit activation and batch normalization. For frequency and temporal pooling, the average pooling layer is employed, and its sizes are [[2, 2], [2, 2], [1, 2], [1, 2], [1, 2], [1, 2], [1, 2]], respectively. The RNN block is made up of two layers of 128 bidirectional gated recurrent units (Bi-GRUs), resulting in a total of 1.1M parameters. After the RNN block comes the attention pooling layer, which is the product of multiplying a linear layer with softmax activations and a linear layer with sigmoid activations. The baseline employs the *mean-teacher* (MT) model, which updates the teacher model's weights using an exponential moving average from the student model.

2.2. Inverted residual network (IRN)

Taking inspiration from *MobileNetV2* [11], we propose to incorporate IRN to replace the standard 2D convolutions, as depicted in Figure 1 (a). The proposed replacement has a distinct property that allows the network expressiveness (encoded by expansion layers) to be separated from its capacity (encoded by bottleneck inputs). Further, it allows lightweight model implementation for low-computational embedded systems. The block uses *depthwise separable convolutions* to replace the fully convolutional operations with a factorized version to split the standard convolution into two separate layers. The block performs three separate convolutions. First, a *pointwise convolution* is used to expand the low-dimensional input feature map to a higher-dimensional space. Followed by a *depthwise convolution*, achieving spatial filtering. Finally, the spatially filtered feature map is projected back to a low-dimensional subspace using another pointwise convolution. Figure 1 (a) shows the residual link between low-dimensional feature maps.



(a) (b)

Figure 1: The proposed architectures (a) IRN with attention module (b) triplet attention module for SED.

2.3. Triplet attention

We propose to incorporate an effective channel attention module, namely, *triplet attention* [12], into the IRN. The *cross-dimensional* interaction is captured by this parameter-free attention mechanism, which may be integrated into other standard networks. In traditional ways of computing channel attention, there is a significant loss of spatial information as the input tensor is spatially decomposed into one pixel per channel by employing global average pooling. This leads to a loss of interdependence between the *channel* dimension and *spatial* dimension on these single pixels. Further, as the *channel* attention and *spatial* attention in [9] are computed independently of each other, the relationship between the two is not considered. To address this issue, we propose capturing *cross-dimensional* interaction with no dimensionality reduction by adding *triplet attention* to the IRN for SED applications.

The *triplet attention* is composed of *three parallel branches*, built to capture dependencies between the (C, F), (C, T), and (F, T) of the input feature, where C, F, T represent the channel, frequency, time feature maps, respectively. Two of the branches capture the cross-dimension interaction between the channel dimension C and either the spatial dimension F or T. The last, final branch resembles CBAM, which is used to build *spatial attention*. For each branch, the input is permuted as shown in Figure 1 (b), and then it is passed through *Z-pool*. The *Z-pool* layer reduces the zeroth dimension to two by concatenating average pooling and max pooling across that dimension. This helps to retain a rich representation while shrinking the depth, resulting in less computational requirement. The operation of *Z-pool* is as follows:

$$Z\text{-}pool(x) = [MaxPool_{0d}(x), AvgPool_{0d}(x)] \qquad (1)$$

where *0d* is the 0th-dimension along which the operation is applied and *x* is the input tensor. The resultant from the *Z-pool* is passed through a standard convolutional layer of kernel size 7×7, followed by batch normalization. The attention weights are generated by passing the tensor through a sigmoid function and are applied to the input tensor for the respective branch. The resulting output is then rotated back to its original state to retain the original input shape. The results of all *three branches (y1, y2, y3)* are aggregated with straightforward *averaging* as given below:

$$y = \frac{1}{3}(y1 + \overline{y2} + \overline{y3}) \qquad (2)$$

where $\overline{y2}$ and $\overline{y3}$ represents the 90° clockwise permutation to retain the original input shape of (C × T × F).

2.4. Proposed architecture

We employed the IRN described in Section 2.3 to replace the standard 2D CNNs, which results in a smaller amount of parameters. The *triplet attention* module was plugged in after the *depthwise separable convolution* in the IRN, as shown in Figure 1 (a). This newly generated module is referred to as IRTAM, which enables the acquisition of more blended *cross-dimensional* feature information. The updated architecture has the same number of layers, but the size of the feature map in each module is reduced to 16, 32, 64, 64, 64, 64, and 64, respectively. The updated architecture consists of 2 layers of Bi-GRU with 64 hidden units, resulting in a total of 304k parameters for the entire model compared to the 1.1M parameters in the baseline. In summary, the updated architecture with the proposed replacement has 27.6% of the parameters of the baseline.

Table 1: Summary of DCASE 2022 Task 4 development set.

Clips	Description
10,000	Synthetic strongly labeled data
3,470	Real strongly labeled data (external set)
1,578	Real weakly labeled data
14,412	In-domain unlabeled data
1,168	Real strongly labeled validation data
2,500	Synthetic strongly labeled validation data

3. EXPERIMENTAL SETUP

3.1. Dataset

In our experiments, we used the DCASE 2022 Task 4 dataset, which is identical to the DCASE 2023 Task 4A dataset and consists of 10-second audio clips extracted from AudioSet or constructed using isolated sound events to simulate a domestic environment. The split for the development training set is reported in Table 1. Additionally, the public evaluation ("YouTube" evaluation) collection consists of 692 YouTube clips.

3.2. Pre-processing

The audio clips are first re-sampled at 16 kHz to a mono channel. They are then segmented using a window size of 2048 samples with a hop length of 256 samples. The spectrograms of segmented waveforms are extracted using the short-time Fourier transform. Then, log-mel spectrograms are created by using mel-filters in the frequency domain of 0 to 8 kHz, followed by a logarithmic operation. Silence is used to pad the clips that are less than 10-seconds long.

3.3. Two-stage system for SED

We incorporate the two-stage system developed by [13–15] for DCASE 2022 Task 4 participation, depicted in Figure 2. In this system, Stage-1 focuses on audio-tagging (AT), whereas Stage-2 improves SED by using the reliable *pseudo-labels* generated by Stage-1. To extract the embeddings in Stage-1, we used a CNN-14-based pre-trained audio neural network [17] as the feature extractor. The embeddings extracted are fed into the Bi-GRU, which has 2 layers with 1024 hidden units. Stage-1 is trained using a strongly labeled set converted into weak predictions referred to as a *weakified* set, a weakly labeled set, and an unlabeled set with 64 mel-bins, to improve AT performance, as shown in Figure 2. Additionally, the AT system (Stage-1) predicted unlabeled set and employed those as *pseudo-weak* labels in Stage-2 training with 128 mel-bins. In Stage-2, we used the proposed lightweight architecture with 304k parameters described in Section 2.4. It is trained on a *pseudo-weakly* labeled set in addition to the strongly labeled and the weakly labeled set in a supervised manner. In training, the weak and *pseudo-weak* sets were merged. Both strongly and weakly labeled samples were assigned a weight of 1 using the baseline system's loss functions.

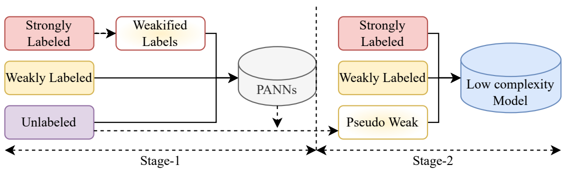


Figure 2: Two-stage system, with Stage-1 focusing on AT and Stage-2 focusing on SED.

3.4. Training process

For all experiments, the batch size is 48 (1/4 strong set, 1/4 weak set, 1/2 unlabeled set). We employed the Adam optimizer with a maximum learning rate of 0.001 and a learning rate ramp-up over the first 50 epochs of the optimization process. A total of 100 epochs are used to train Stage-1, and 200 epochs are used to train Stage-2. The weak training data was used to generate a 90% training set and a 10% validation set. Then the validation is performed on the 10% held-out weak subset and on the strongly labeled synthetic validation set. The system was built with PyTorch Lightning and trained on an NVIDIA Quadro RTX 5000 GPU.

3.5. Additional methods

We used several *data augmentation* techniques to artificially generate more data and improve the model's robustness during the training in both stages. We employed time-masking [18], frame-shifting, mixup [16], and Gaussian noise addition in Stage-1 and time-masking, frame-shifting, mixup and frequency-masking [18] in Stage-2. We also adopted *adaptive post-processing* [19] in all the experiments, where the median filter window sizes are different for each event category, calculated *heuristically* based on the varying length of each event in real life. Furthermore, for each class, we used *probability value correction* [20], in which we multiplied the probability generated by the model by a magnification factor to correct the probability to a maximum value of 1. For inference *temperature tuning* as in [21] a temperature factor of 2.1 is employed. In our final developed system, we also used the *external set* released by DCASE 2022 Task 4 organizers during training, with each stage employing MT and *interpolation consistency training* (ICT) [22] to utilize the unlabeled training data.

3.6. Evaluation metric

In our studies, we used polyphonic sound event detection scores (PSDS) [23] introduced in the DCASE 2022 Task 4 as a performance metric to evaluate the systems. The PSDS is more resistant to labeling subjectivity, allowing for the interpretation of both the ground truth and the detection of temporal structure. It computes a *single* PSDS using polyphonic receiver operating characteristic curves, allowing for comparison regardless of the operating point. Furthermore, it can be customized for a variety of applications, ensuring that the desired user experience is achieved. As a result, it *overcomes* the limitations of traditional event F-scores based on collars. We compute the PSDS in our studies using *two* different scenarios that emphasize different system properties. Scenario-1 requires the system to respond quickly to event detection, focusing on the temporal localization of the sound event. Scenario-2, on the other hand, focuses on preventing class confusion rather than reaction time. The greater the values for PSDS1 and PSDS2, the better for both scenarios. Notably, the PSDS metric employed here adheres to the DCASE 2022 Task 4 protocol and differs from the threshold-independent PSDS used in DCASE 2023 Task 4A.

4. RESULTS AND ANALYSIS

4.1. Proposed IRTAM

We consider the two-stage framework described in the previous section for our studies with *low-complexity* SED systems.

Table 2: Performance comparison showing the importance of the proposed method on DCASE 2022 Task 4 validation set.

System	PSDS1	PSDS2	#Parameters
Baseline	0.351	0.552	1.1M
IRN	0.343	0.519	301k
IRN + SE	0.359	0.521	442k
IRN + CA	0.419	0.694	333k
IRN + Triplet Attention (IRTAM)	**0.440**	**0.708**	304k
+ data augmentation	0.446	0.702	304k
+ external set	0.457	0.712	304k
+ ICT	0.471	0.710	304k
+ median filtering	0.480	0.727	304k
+ probability correction	**0.483**	**0.728**	304k

First, in Stage-2 used for inference, we replace the standard 2D CNNs in the baseline with the proposed IRN described in Section 2.3, resulting in a reduction of parameters from 1.1M in the baseline to 301k. From Table 2, we observe a minor degradation in the performance with a decrease in PSDS1 from 0.351 to 0.343 and in PSDS2 from 0.552 to 0.519 owing to the reduction in the number of parameters. Following our proposed design, we next incorporate an attention module in the IRN after the *depthwise separable convolution* layer to assist the model in learning the *frequency-dependent* patterns and feature *interdependencies* between channels and time-frequency locations.

We are also interested in comparing the performance of the proposed IRTAM (IRN + *triplet attention*) with widely popular SE attention and another recent method, namely, coordinate attention (CA) [24] incorporated in IRN. From Table 2, we observe that the SED performance increases with the introduction of both SE and CA modules. However, on comparing their performance to our proposed IRTAM, we find that IRN with *triplet attention* (IRTAM) performs better than both the other attention modules considered. It is also observed that the increase in the number of parameters for IRTAM is very negligible compared to that with SE and CA. Thus, these studies show the effectiveness of the proposed *low-complexity* IRTAM module, specifically due to the introduction of *triplet attention*, for capturing *cross-dimensional* interaction in SED models. Further, we show the contribution of each additional method discussed in Section 3.5 to apply on the proposed developed system to achieve the final PSDS1 of 0.483 and PSDS2 of 0.728 on the validation set, giving a 34.1% increase compared to the baseline in terms of both *PSDS metrics*.

4.2. Ablation study on triplet attention branches

With the use of a *three-branch* structure, we verify that it is important to capture the *cross-dimensional* interaction between (T, F), (T, C), and (C, F). In Table 3, we compare the results when each branch is turned on, represented by the combination given in each row, to analyze the influence of the branches in the *triplet attention* module. As can be seen, the findings corroborate our understanding that individual and pair branch interaction is inferior to the performance of *triplet attention*, which involves all *three branches* being active.

4.3. System comparison

To further assess the efficacy of the proposed module, the system is also compared with the top-ranked *single* (without ensemble) systems submitted to DCASE 2022 Task 4. In Table 4, the scores for the other systems are directly taken from their cited technical reports released in the challenge. The proposed *low-complexity* system surpasses systems with large parameters and gets close to the top-ranked system, which has 15.4M parameters while having just 304k parameters. We also note that the proposed attention module is network-independent and can be employed in any model to replace standard convolutions with the IRTAM block. Furthermore, on the public evaluation set, the final system with the proposed IRTAM achieved a PSDS1 of 0.488 and a PSDS2 of 0.720, in contrast to the baseline system having a PSDS1 of 0.387 and a PSDS2 of 0.592.

Table 3: Ablation study to show the gain of each branch in the triplet attention on DCASE 2022 Task 4 validation set, where (x,y) is the interplay between dimensions x and y to compute attention weights and aggregated average.

Branch Interaction	PSDS1	PSDS2	#Parameters
(F,T)	0.420	0.643	304k
(C,T)	0.410	0.614	304k
(C,F)	0.424	0.657	304k
((F,T), (C,T))	0.480	0.723	304k
((F,T), (C,F))	0.468	0.716	304k
((C,T), (C,F))	0.459	0.730	304k
((F,T), (C,T), (C,F))	**0.483**	**0.728**	304k

Table 4: Comparison with top-ranked single systems (without ensemble) from DCASE Task 4 2022 on the validation set.

System	PSDS1	PSDS2	#Parameters
Ebbers-UPB-task4 [25]	0.505	0.807	15.4M
Proposed	**0.483**	**0.728**	**304k**
Zhang-UCAS-task4 [26]	0.459	0.672	11M
Kim-GIST-task4 [27]	0.455	0.670	1M
Dinkel-XiaoRice-task4 [28]	0.425	0.644	37M

5. CONCLUSION

In this work, we proposed an inverted residual network with *triplet attention* as a module referred to as IRTAM to replace the standard 2D convolutional neural networks for SED applications. The proposed *low-complexity* attention module was designed to capture *cross-dimensional* interaction with minimal computational overhead. To show the effectiveness of the developed lightweight architecture employing IRTAM, we considered the DCASE 2022 Task 4 dataset for the studies. Our findings demonstrated the efficacy of incorporating *cross-dimensional* interaction in SED applications by improving the baseline by 34.1% and significantly outperforming some other attention modules in both aspects of the *PSDS metric*. Furthermore, our ablation study validated the relevance of capturing *cross-dimensional* interaction using a *three-branch* structure and showed overall effectiveness by achieving comparable results to systems with a large number of parameters. It is also worth noting that the proposed system contains only 27.6% of the baseline parameters, making the model suitable for *low-complexity* SED applications. We intend to extend the proposed IRTAM to larger model sizes in the future.

6. REFERENCES

[1] L. Vuegen, B. V. D. Broeck, P. Karsmakers, J. F. Gemmeke, B. Vanrumste, and H. V. hamme, "An MFCC-GMM approach for event detection and classification," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013.

[2] X. Zhuang, X. Zhou, T. S. Huang, and M. Hasegawa-Johnson, "Feature analysis and selection for acoustic event detection," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 17–20, 2008.

[3] L. Lu, F. Ge, Q. Zhao, and Y. Yan, "A SVM-based audio event detection system," *International Conference on Electrical and Control Engineering (ICECE)*, pp. 292–295, 2010.

[4] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[5] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 646–650, 2022.

[6] K. Drossos, S. I. Mimilakis, S. Gharib, Y. Li, and T. Virtanen, "Sound event detection with depthwise separable and dilated convolutions," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2020.

[7] T. Khandelwal, R. K. Das, and E. S. Chng, "Is your baby fine at home? baby cry sound detection in domestic environments," *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 275–280, 2022.

[8] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, "Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection," *Interspeech*, pp. 2763–2767, 2022.

[9] W. Xia and K. Koishida, "Sound event detection in multichannel audio using convolutional time-frequency-channel squeeze and excitation," *Interspeech*, pp. 3629–3633, 2019.

[10] S. Woo, J. Park, J.-Y. Lee, and I.-S. Kweon, "CBAM: Convolutional block attention module," *European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

[11] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[12] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, "Rotate to attend: Convolutional triplet attention module," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 3138–3147, 2021.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] S. Mizobuchi, H. Ohashi, A. Izumi, and N. Kodama, "Mizobuchi PCO team's submission for DCASE 2022 Task 4 sound event detection using external resources," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[21] X. Zheng, H. Chen, and Y. Song, "Zheng USTC team's submission for DCASE 2021 Task 4 - semi-supervised sound event detection," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2021.

[22] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3635–3641, 2019.

[23] 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.

[24] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13 708–13 717, 2021.

[25] J. Ebbers and R. Haeb-Umbach, "Pre-training and self-training for sound event detection in domestic environments," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[26] S. Xiao, "Pretrained models in sound event detection for DCASE 2022 challenge Task 4," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[27] J. W. Kim, G. W. Lee, H. K. Kim, Y. S. Seo, and I. H. Song, "Semi-supervised learning-based sound event detection using frequency-channel-wise selective kernel for DCASE challenge 2022 Task 4," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.

[28] H. Dinkel, Z. Yan, Y. Wang, M. Song, J. Zhang, and W. Wang, "A large multi-modal ensemble for sound event detection," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2022.