Interspeech 2022
18-22 September 2022, Incheon, Korea

RaDur: A Reference-aware and Duration-robust Network for Target Sound Detection

Dongchao Yang[1†], *Helin Wang*[1†], *Zhongjie Ye*[1], *Yuexian Zou*[1,*], *Wenwu Wang*[2]

[1]ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2]Center for Vision, Speech and Signal Processing, University of Surrey, UK
{dongchao98,zhongjieye}@stu.pku.edu.cn, {wanghl15,zouyx}@pku.edu.cn, w.wang@surrey.ac.uk

Abstract

Target sound detection (TSD) aims to detect the target sound from a mixture audio given the reference information. Previous methods use a conditional network to extract a sound-discriminative embedding from the reference audio, and then use it to detect the target sound from the mixture audio. However, the network performs much differently when using different reference audios (e.g. performs poorly for noisy and short-duration reference audios), and tends to make wrong decisions for transient events (i.e. shorter than 1 second). To overcome these problems, in this paper, we present a reference-aware and duration-robust network (RaDur) for TSD. More specifically, in order to make the network more aware of the reference information, we propose an embedding enhancement module to take into account the mixture audio while generating the embedding, and apply the attention pooling to enhance the features of target sound-related frames and weaken the features of noisy frames. In addition, a duration-robust focal loss is proposed to help model different-duration events. To evaluate our method, we build two TSD datasets based on UrbanSound and Audioset. Extensive experiments show the effectiveness of our methods.
Index Terms: target sound detection, embedding enhancement, reference-aware, duration-robust focal loss

1. Introduction

Human beings have the ability to focus their auditory attention on a particular sound in a multi-source environment, which attracts the related studies in machine hearing. In this paper, we focus on the target sound detection (TSD) task [1], which aims to recognize and localize target sound source within a mixture audio given a reference audio or/and a sound label, e.g. detecting the talking sound within a noisy cafe environment. TSD has many potential applications, such as noise monitoring for smart cities [2] and large-scale multimedia indexing [3]. TSD is similar to sound event detection (SED), however, the difference is that SED aims to classify and localize all pre-defined sound events (*e.g.*, train horn, car alarm) within an audio clip, which has been widely studied [4, 5, 6, 7, 8, 9]. Other related tasks include speaker extraction [10, 11, 12, 13, 14] where the target speech is extracted from a mixture speech given a reference utterance of the target speaker, and acoustic events sound selection (or removal) problems [15, 16, 17]. Different from them, TSD focuses on the detection task, as seen in multimedia retrieval applications where the training data can be more easily obtained.

In a recent work [1], a target sound detection network (TSDNet) is presented, which is composed of a conditional network and a detection network. In TSDNet, a sound-discriminative embedding generated by the conditional network is used as the reference information to guide the detection network for detecting the target sound from the mixture audio. TSDNet provides a good detection performance on a small-scale training dataset (i.e. UrbanSound [18]). However, we observe that TSDNet tends to make wrong detection on short-duration events (such as chop and bouncing), as Figure 2(a) shows. In addition, the performance of detection highly relies on the quality of the reference information, which may be severely degraded when the reference audio is noisy or quite short.

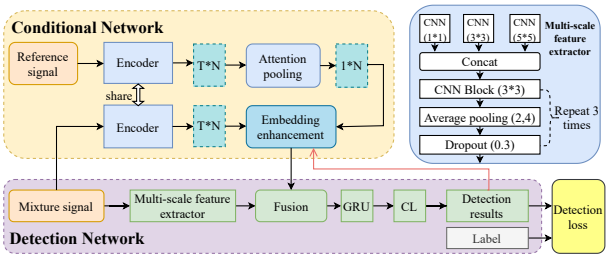

Figure 1: *The architecture of our proposed RaDur. Here, CL denotes the frame-level classification layer, containing two fully-connected layers and one softmax function.*

To address these issues, in this paper, we propose a reference-aware and duration-robust network (called RaDur) for target sound detection task. More specifically, we design an embedding enhancement module in the conditional network, which utilizes the frames of mixture audio related to the reference information to enhance the embedding. We employ the attention pooling function to guide the conditional network to attend to target-related frames and ignore noisy frames or interference. In addition, we apply a multi-scale feature extractor to extract characteristics of events with different duration and we propose a duration-aware focal loss to solve the problems induced by short-duration events. To evaluate our method, we use URBAN-TSD dataset [1] and establish a new large-scale dataset (Audioset-TSD) based on Audioset [3]. The experiments show that our proposed method provide 6.6% and 16.7% improvement for the segment-based and event-based F scores on the URBAN-TSD dataset, and 23.8% and 8.0% improvement for the segment-based and event-based F scores on the Audioset-TSD dataset.

2. Proposed Method

The architecture of our proposed network (RaDur) is shown in Fig. 1, which is composed of two parts: a conditional network and a detection network. In the conditional network, we propose an embedding enhancement module and use the attention

† Indicates equal contribution.
* Corresponding Author.

10.21437/Interspeech.2022-433

pooling function to obtain more sound-discriminative embedding. In the detection network, we apply multi-scale feature extractor to generate a multiple view of the mixture audio with respect to different-duration events. In addition, a duration-aware focal loss function is proposed to facilitate the modelling of short-duration events. The details are as follows.

2.1. Conditional Network

The conditional network aims to extract a sound-discriminative embedding vector from the reference audio. Similar to the previous work [1], we adopt a VGG-like convolutional neural network (CNN) model [19] for the conditional network, which uses the log-mel spectrogram as input and consists of 5 convolutional blocks with 64, 128, 256, 512, 1024 output channels, respectively. Lastly, we use a fully-connected layer to output the conditional embedding vector with a fixed dimension of 128.

We observe that the quality of the embedding is crucial to TSD. If the reference audio contains noise or if the duration of the event within the reference audio is shorter than 1 second, the quality of the embedding and thereby the performance of TSD will be degraded. To make the embedding more discriminative and robust to noise and other interference, we propose an attention pooling function and an embedding enhancement (EE) module to enhance the embedding.

2.1.1. Attention Pooling Function

We find that many frames of the reference audio do not include the information of the target sound, instead, these frames may include noises or other interference information. As a result, these target-irrelevant frames may affect the distinguishability of the embedding. Inspired by the temporal attention pooling operations in the audio classification tasks [20, 8, 21], we replace the global pooling layer in [1] with an attention pooling (AP) function which enables the network to attend to the frames containing the reference information. More specifically, given the input representation of the reference audio $\boldsymbol{x}_r \in \mathcal{R}^{t_r \times f_r}$ where t_r and f_r denote the number of time frames and the number of frequency bands, respectively, we can get the deep time-frequency feature $\boldsymbol{E}_r \in \mathcal{R}^{t' \times C_r}$ from the encoder f_{re}.

$$\boldsymbol{E}_r = f_{re}(\boldsymbol{x}_r; \theta_r) \tag{1}$$

where t' denotes the number of feature frames, C_r denotes the dimension of the feature for each frame and θ_r is the parameters of the encoder. Then we use the global average pooling f_{GAP} to get the global feature $\boldsymbol{e}_g \in \mathcal{R}^{C_r}$.

$$\boldsymbol{e}_g = f_{GAP}(\boldsymbol{E}_r) \tag{2}$$

After that, we use the global feature $\boldsymbol{e}_g$ as query, $\boldsymbol{E}_r$ as key to calculate attention weights of all the frames $\boldsymbol{a} \in \mathcal{R}^{t'}$. Finally, we get the final embedding $\boldsymbol{e}_f \in \mathcal{R}^{C_r}$ according to the attention weights.

$$\boldsymbol{Q} = \boldsymbol{e}_g \boldsymbol{W}_q, \boldsymbol{K} = \boldsymbol{E}_r \boldsymbol{W}_k \tag{3}$$

$$\boldsymbol{a} = softmax(\frac{\boldsymbol{K}\boldsymbol{Q}^{\boldsymbol{T}}}{\sqrt{C_r}}) \tag{4}$$

$$\boldsymbol{E}_f = \boldsymbol{E}_r \otimes \boldsymbol{a}, \boldsymbol{e}_f = \sum_{i=1}^{t'} E_f^i \tag{5}$$

where $\boldsymbol{W}_q \in \mathcal{R}^{C_r \times C_q}$ and $\boldsymbol{W}_k \in \mathcal{R}^{C_r \times C_k}$ denote the learnable weights, and $C_q = C_k$. $\boldsymbol{E}_f = \{E_f^1, E_f^2, ..., E_f^{t'}\} \in \mathcal{R}^{t' \times C_r}$ denotes the weighted feature and $\otimes$ indicates the element-wise multiplication of a matrix and a vector.

2.1.2. Embedding Enhancement Module

We can mitigate the problem caused by noise and short-duration events using the attention pooling, but the quality of some reference audios is still poor. From the experiments, we find that if we use several reference audios or directly use the target audio as the reference audio, we can get a much better performance. However, we cannot get the target audio in practical applications, and it is sometimes hard to collect multiple audios for some unseen classes [22, 23]. To further improve the quality of the embedding without using extra data, in this paper, we propose to use the mixture audio to enhance the embedding and present an embedding enhancement (EE) module which works at different training stages. The core idea is to use the detection results of the previous training stages (*i.e.* previous epochs in our experiments) to select frames from the features of the mixture audio that contain the characteristics of the target event. After that, we use these frames as the additional reference information to enhance the quality of the embedding.

To be more specific, we denote f_d as the detection network whose inputs are the mixture audio $\boldsymbol{x}_m \in \mathcal{R}^{t_m \times f_m}$ and the embedding $\boldsymbol{e}_f$, where t_m and f_m denote the number of time frames and the number of frequency bands, respectively. Hence, we can get the detection results of the previous stage $\hat{\boldsymbol{y}} \in \mathcal{R}^{t'}$ by

$$\hat{\boldsymbol{y}} = f_d(\boldsymbol{x}_m, \boldsymbol{e}_f; \theta_d) \tag{6}$$

where θ_d denotes the parameters of the detection network, and the details of the detection network are introduced in Section 2.2. Next, we can get the deep time-frequency feature of the mixture audio $\boldsymbol{E}_m \in \mathcal{R}^{t' \times C_r}$ from the encoder f_{re} similarly to (1).

$$\boldsymbol{E}_m = f_{re}(\boldsymbol{x}_m; \theta_r) \tag{7}$$

Then for the current training stage, we select top-k frames of the feature of the mixture audio according to the detection results $\hat{\boldsymbol{y}}$. Note that $k \ll t'$, and we can get the selected feature $\boldsymbol{E}'_m \in \mathcal{R}^{k \times C_r}$ and the corresponding the detection scores $\hat{\boldsymbol{y}}' \in \mathcal{R}^k$. Finally, the selected feature is used to enhance the reference embedding $\boldsymbol{e}_f$ by the attention method.

$$\boldsymbol{Q}' = \boldsymbol{e}_f \boldsymbol{W}'_q, \boldsymbol{K}' = \boldsymbol{E}'_m \boldsymbol{W}'_k \tag{8}$$

$$\boldsymbol{a}' = softmax(\frac{\boldsymbol{K}'\boldsymbol{Q}'^T}{\sqrt{C_r}}) \tag{9}$$

$$\boldsymbol{E}_s = \boldsymbol{E}'_m \otimes \boldsymbol{a}', \boldsymbol{e}'_f = \sum_{i=1}^{k} E_s^i \tag{10}$$

where $\boldsymbol{W}'_q$ and $\boldsymbol{W}'_k$ denote the learnable weights. $\boldsymbol{e}'_f \in \mathcal{R}^{C_r}$ is the enhanced embedding, $\boldsymbol{a}' \in \mathcal{R}^k$ is the attention weights of selected frames, and $\boldsymbol{E}_s = \{E_s^1, E_s^2, ..., E_f^k\} \in \mathcal{R}^{k \times C_r}$ denotes the weighted selected feature. As this EE module utilizes the previous training stage to guide the current stage, we argue that in the first several epochs, the EE module is not required, while it turns out to be important with the increase in the number of training epochs. Thus, we set the first 10 epochs as the warm-up stage in which the EE module is not applied, while in the following epochs, the results from the previous epoch are used to enhance the current epoch. In addition, if there is no target sound happening in the mixture audio, the EE module still makes the embedding attend to the mixture audio, which may interfere the original reference audio. Therefore, we set a hyper-parameter τ to control the EE module. Here, τ is a threshold which is used to filter the detection scores $\hat{\boldsymbol{y}}'$. Finally,

we use a fusion layer (1D convolutional layer) to integrate the original embedding and the enhanced embedding.

$$\hat{y}_i' = \begin{cases} 0, & \text{if } \hat{y}_i' < \tau \\ \hat{y}_i', & \text{if } \hat{y}_i' \geq \tau \end{cases}, \boldsymbol{a}'' = \boldsymbol{a}' \otimes \hat{\boldsymbol{y}}' \tag{11}$$

$$\boldsymbol{E}_s' = \boldsymbol{E}_m' \otimes \boldsymbol{a}'', \boldsymbol{e}_f'' = \sum_{i=1}^{k} E_s'^i \tag{12}$$

$$\boldsymbol{e}_f^\star = Conv1d(\boldsymbol{e}_f) \otimes Conv1d(\boldsymbol{e}_f'') \tag{13}$$

where $\boldsymbol{e}_f^\star \in \mathcal{R}^{C_r}$ is the modified enhanced embedding, $\boldsymbol{a}'' \in \mathcal{R}^k$ is the modified attention weights of selected frames, and $\boldsymbol{E}_s' = \{E_s'^1, E_s'^2, ..., E_f'^k\} \in \mathcal{R}^{k \times C_r}$ denotes the modified weighted selected feature. The whole process for generating the enhanced embedding is summarized in Algorithm 1.

Algorithm 1 Embedding Generation.

Input:
The input representation of the reference audio $\boldsymbol{x}_r$; The input representation of the mixture audio $\boldsymbol{x}_m$;
Output: The enhanced embedding $\boldsymbol{e}_f^\star$;
1: Get the original embedding $\boldsymbol{e}_f$ using equation (1)-(5);
2: Get the feature of the mixture audio $\boldsymbol{E}_m$ using equation (7);
3: Get the detection results of the previous stage $\hat{\boldsymbol{y}}$ using equation (6);
4: Select top-k frames from $\boldsymbol{E}_m$, and get the selected feature $\boldsymbol{E}_m'$ and the corresponding detection scores $\hat{\boldsymbol{y}}$;
5: Calculate the attention weights $\boldsymbol{a}'$ using equation (8)-(9);
6: Modify the attention weights and get $\boldsymbol{a}''$ using equation (11);
7: Calculate $\boldsymbol{e}_f^\star$ according to equation (12)-(13);
8: **return** $\boldsymbol{e}_f^\star$;

2.2. Detection Network

The detection network is composed of three parts: (1) Multi-scale feature extractor: a CNN-based structure aiming to extract the acoustic representation of the mixture audio. As Figure 1 shows, to effectively capture both the global and local information which is good for the long and transient events, multi-scale CNNs with varied kernel sizes 1×1, 3×3 and 5×5 are employed [24] in the first layer with the 64 output channels. Besides, we use gated linear units (GLUs) [25] to replace ReLU [26] activation in this layer. After that, we concatenate the output of multi-scale CNNs, and feed it to three CNN blocks with the output channels of 128, 256 and 512 respectively to get the final representation. (2) Bi-GRU: one Bi-GRU layer with 512 units is used to capture temporal dependencies and integrate the conditional embedding. (3) Frame-level classification layer: it composes of two fully-connected layers with 256 hidden units and a softmax function, which is used to get the frame-level predictions.

2.3. Loss Function

TSDNet [1] uses the binary cross entropy (BCE) loss as the training objective. In this paper, we find that using the focal loss [27] can get better performance especially on Audioset-TSD dataset. Furthermore, to solve the problems caused by short-duration events, a duration-aware focal loss function is proposed, which is based on the focal loss, defined by:

$$\mathcal{L}_{focal} = -\beta y(1-\hat{y})^\gamma log(\hat{y}) - (1-\beta)(1-y)\hat{y}^\gamma log(1-\hat{y}) \tag{14}$$

where β and γ are hyper-parameters. $\hat{y}$ and y denote the frame-level prediction probability and label, respectively. The focal loss is a dynamically scaled cross entropy loss, where the scaling factor decays to zero as confidence in the correct class increases. Hence, the scaling factor can automatically down-weight the contributions of easy examples during training, which allows the model to focus on hard examples. To further improve the performance of the transient events, we propose the duration-aware focal loss function (Du-Focal Loss), which works by applying different weights to events of different duration.

$$\mathcal{L}_{Du-focal} = (1 + \alpha \frac{e^{-w} - e^{-w_{max}}}{e^{-w_{min}} - e^{-w_{max}}}) \mathcal{L}_{focal} \tag{15}$$

where $w \in [0, 10]$ denotes the average duration of each event in the training dataset, $w_{max} = 0$ is the maximum value and $w_{min} = 10$ is the minimum value. We use $\frac{e^{-w} - e^{-w_{max}}}{e^{-w_{min}} - e^{-w_{max}}}$ to scale the values to $[0, 1]$. α is a hyper-parameter which controls the ratio of the extra weights of the loss.

3. Audioset-TSD Dataset

We build a new TSD dataset based on Audioset [28]. Here, we use the strong-labelled data [29] which includes 94,126 training clips and 16,118 test clips, and they come from 456 different classes. We choose 192 different classes to build a large-scaled TSD dataset, named as Audioset-TSD dataset. The process for building the TSD dataset is the same as the previous work [1]. Note that Audioset-TSD dataset also includes negative samples, in which the mixture audio does not contain the target sound. In summary, Audioset-TSD dataset includes 10-second 490,336 training, 40,185 validation and 83,334 test clips. We take the reference audio for each category directly from the Audioset training set. We preferentially select the audio clips that do not contain interference from other events, but they may still contain background noises.

As a result, compared with URBAN-TSD dataset [1] which contains 10-class 48,489 samples, the Audioset-TSD dataset contains more data with a bigger number of classes. In addition, the clips contain noises and interference from other events. Therefore, it is a more challenging dataset.

4. Experiments

4.1. Experimental setups

Dataset. We evaluate our methods on Audioset-TSD and URBAN-TSD [1] datasets.
Metrics. We use the segment-based F-measure and event-based F-measure [30] as the evaluation metrics, which are the most commonly used metrics for sound event detection. All the F-scores are macro-averaged.
Preprocessing. The pre-extracted log mel-spectrograms with a window of 1024 samples and hop length of 320 samples are used in our experiments. The number of Mel bands is set to 64 and the size of log mel spectrogram is 1001×64.
Training details. In the training phase, the Adam optimizer [31] is employed as the optimizer with a learning rate of 1×10^{-3}. Batch size is set to 64 and training takes 50 epochs.

Table 1: *The comparison between TSDNet and Radur on the URBAN-TSD dataset. S-F and E-F denote segment- and event-based F score. We did the experiments three times and report the mean value.*

Model	Loss	AP	EE	S-F	E-F
TSDNet [1]	BCE			69.3	30.6
Radur	BCE			71.5	33.3
Radur	BCE	✓		71.7	33.5
Radur	BCE	✓	✓	73.5	35.2
Radur	Focal	✓	✓	**73.9**	**35.7**
Radur	Du-Focal	✓	✓	73.8	**35.7**

Table 2: *The comparison between TSDNet and Radur on the Audioset-TSD dataset.*

Model	Loss	AP	EE	S-F	E-F
TSDNet [1]	BCE			49.5	48.6
Radur	BCE			51.3	50.0
Radur	BCE	✓		52.5	51.4
Radur	BCE	✓	✓	54.4	52.2
Radur	Focal	✓	✓	58.3	51.1
Radur	Du-Focal	✓	✓	**61.3**	**52.5**

Following [1], the dimension of the embedding vector is 128 and we use the multiplication fusion manner. We choose top-2 frames for the EE module. Unless specifically stated, the hyper-parameters α, β, γ and τ are set to 1.5, 0.65, 2 and 0.7 respectively, empirically based on the validation set.

4.2. Experimental results

We evaluate our proposed Radur and the state-of-the-art method [1] on URBAN-TSD and Audioset-TSD datasets, and report the experimental results in Tables 1 and 2. By comparing row 1 and row 2 in Tables 1 and 2, we can see that RaDur has better performance thanks to the multi-scale convolutional scheme and deeper network structure. By comparing row 2 and row 3 in Table 2, we can see that the AP module leads to about 1.4% improvement on the event-based F1 score, which means that the AP module can improve the quality of reference embedding. In addition, the comparison between row 3 and 4 shows the effectiveness of the EE module. We can see that the focal loss performs better than the BCE loss especially on the Audioset-TSD dataset, and our proposed Du-Focal loss further improves the performance on the Audioset-TSD dataset. Instead, the Du-Focal loss does not bring improvement on the URBAN-TSD dataset, for the reason that all of the events in this dataset have similar duration (1-2 seconds).

To further analyze the influence of event duration on the performance, we divide the 192 events into 5 groups according to the average duration in the test set. As Figure 2 (a) shows, the duration of the first group is from 0 seconds to 1 seconds, which represents the transient events. The duration of the last group is from 7 seconds to 10 seconds, which represents the long events. We can find that detecting transient events is the most difficult task. Because of the multi-scale feature extractor and duration-aware focal loss, RaDur provides significant improvement on transient events. However, the detection results for transient and long events still have a large gap, which deserves further study in our future work.

Table 3: *Ablation studies on hyper-parameter τ on the URBAN-TSD dataset.*

Model	τ	Segment-based F-score	Event-based F-score
Radur	0.5	71.4	32.9
Radur	0.6	72.6	34.1
Radur	0.7	**73.5**	**35.2**
Radur	0.8	73.1	34.9
Radur	0.9	71.6	33.8

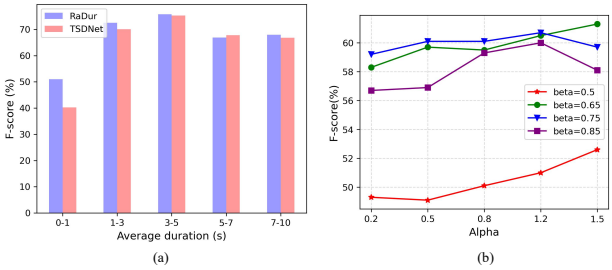

Figure 2: *(a) shows the performance comparison between TSDNet and RaDur on events of different duration. (b) shows the ablation study on two hyper-parameters (α and β) of the Du-Focal loss. Note that F-score is segment-based.*

4.3. Ablation Studies

We first conduct ablation studies to investigate the effect of hyper-parameter τ on the EE module and report experimental results in Table 3. We can see that the performance gradually improves with τ increasing from 0.5 to 0.7. After that, the performance begins to decrease. The experimental phenomenon meets our hypothesis: If τ is smaller than 0.6, we may select non-target feature frame, which may influence the original embedding. On the contrary, if τ is larger that 0.8, most of the frames will be filtered, then the EE module may not work.

We also conduct ablation studies to investigate the influence of two of the hyper-parameters (α and β) in the Du-Focal loss. As shown in Figure 2 (b), the value of β is very important, and β should be larger than 0.5 because there are more negative sample frames than positive ones in the Audioset-TSD dataset. Furthermore, as the value of α increases, the F-score is boosted. However, we can see that if $\beta > 0.65$ and $\alpha > 1.2$, the performance will drop. We argue that if β and α are both set too large, the ratios of negative samples will be too small, as a result, the negative samples may be easily ignored in the training process.

5. Conclusions

We have presented an improved TSDNet (RaDur) by modelling short-duration events and enhancing the discriminating ability of the embedding vectors. In the future work, we will explore the extension of novel classes. The source code and dataset of this work have been released.[1]

6. Acknowledgements

This paper was partially supported by Shenzhen Science & Technology Research Program (No: GXWD20201231165807007-20200814115301001; No: JSGG20191129105421211) and NSFC (No: 62176008).

[1] https://github.com/yangdongchao/RaDur

7. References

[1] D. Yang, H. Wang, Y. Zou, and C. Weng, "Detect what you want: Target sound detection," *arXiv preprint arXiv:2112.10153*, 2021.

[2] J. P. Bello, C. Silva, O. Nov, R. DuBois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "Sonyc: A system for the monitoring, analysis and mitigation of urban noise pollution," *arXiv preprint arXiv:1805.00889*, 2018.

[3] S. Hershey, S. Chaudhuri, D. Ellis, J. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "CNN architectures for large-scale audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 131–135.

[4] H. Dinkel, M. Wu, and K. Yu, "Towards duration robust weakly supervised sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 887–900, 2021.

[5] L. Lin, X. Wang, H. Liu, and Y. Qian, "Specialized decision surface and disentangled feature for weakly-supervised polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1466–1478, 2020.

[6] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound event detection of weakly labelled data with cnn-transformer and automatic threshold optimization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2450–2460, 2020.

[7] 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.

[8] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 31–35.

[9] I. Martín-Morató, A. Mesaros, T. Heittola, T. Virtanen, M. Cobos, and F. Ferri, "Sound event envelope estimation in polyphonic mixtures," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 935–939.

[10] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking," *Proc. Interspeech*, pp. 2728–2732, 2019.

[11] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "Multi-stage speaker extraction with utterance and frame-level reference signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6109–6113.

[12] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černockỳ, "Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

[13] M. Borsdorf, C. Xu, H. Li, and T. Schultz, "Universal speaker extraction in the presence and absence of target speakers for speech of one and two talkers," in *Proc. Interspeech*, 2021, pp. 1469–1473.

[14] Z. Pan, M. Ge, and H. Li, "Usev: Universal speaker extraction with visual cue," *arXiv preprint arXiv:2109.14831*, 2021.

[15] T. Ochiai, M. Delcroix, Y. Koizumi, H. Ito, K. Kinoshita, and S. Araki, "Listen to what you want: Neural network-based universal sound selector," *Proc. Interspeech*, pp. 1441–1445, 2020.

[16] M. Delcroix, J. B. Vázquez, T. Ochiai, K. Kinoshita, and S. Araki, "Few-shot learning of new sound classes for target sound extraction," *arXiv preprint arXiv:2106.07144*, 2021.

[17] Y. Okamoto, S. Horiguchi, M. Yamamoto, K. Imoto, and Y. Kawaguchi, "Environmental sound extraction using onomatopoeia," *arXiv preprint arXiv:2112.00209*, 2021.

[18] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, pp. 1041–1044.

[19] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. 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.

[20] H. Wang, Y. Zou, D. Chong, and W. Wang, "Environmental sound classification with parallel temporal-spectral attention," in *Proc. Interspeech*, 2020, pp. 821–825.

[21] Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang, and M. D. Plumbley, "Weakly labelled audioset tagging with attention neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1791–1802, 2019.

[22] 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 (ICASSP)*. IEEE, 2020, pp. 81–85.

[23] D. Yang, H. Wang, Y. Zou, Z. Ye, and W. Wang, "A mutual learning framework for few-shot sound event detection," *arXiv preprint arXiv:2110.04474*, 2021.

[24] Y. Xian, Y. Sun, W. Wang, and S. M. Naqvi, "Multi-scale residual convolutional encoder decoder with bidirectional long short-term memory for single channel speech enhancement," in *IEEE European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 431–435.

[25] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 121–125.

[26] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[28] J. Gemmeke, D. 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 (ICASSP)*. IEEE, 2017, pp. 776–780.

[29] S. Hershey, D. P. Ellis, E. Fonseca, A. Jansen, C. Liu, R. C. Moore, and M. Plakal, "The benefit of temporally-strong labels in audio event classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 366–370.

[30] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

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