A-CRNN: A DOMAIN ADAPTATION MODEL FOR SOUND EVENT DETECTION

Wei Wei[1], *Hongning Zhu*[2], *Emmanouil Benetos*[3], *and Ye Wang*[1]

[1]School of Computing, National University of Singapore, Singapore
[2]School of Computer Science and Technology, Fudan University, China
[3]School of EECS, Queen Mary University of London, UK

ABSTRACT

This paper presents a domain adaptation model for sound event detection. A common challenge for sound event detection is how to deal with the mismatch among different datasets. Typically, the performance of a model will decrease if it is tested on a dataset which is different from the one that the model is trained on. To address this problem, based on convolutional recurrent neural networks (CRNNs), we propose an adapted CRNN (A-CRNN) as an unsupervised adversarial domain adaptation model for sound event detection. We have collected and annotated a dataset in Singapore with two types of recording devices to complement existing datasets in the research community, especially with respect to domain adaptation. We perform experiments on recordings from different datasets and from different recordings devices. Our experimental results show that the proposed A-CRNN model can achieve a better performance on an unseen dataset in comparison with the baseline non-adapted CRNN model.

***Index Terms*—** sound event detection, domain adaptation, computational sound scene analysis, CRNNs.

1. INTRODUCTION

Sound event detection (SED) is the task which aims at detecting and classifying all sound events in an audio file, such as door slam and keyboard click. For each detected sound event, the system will give an onset time, an offset time and a label for the corresponding sound event class. It has a wide range of applications, including smart homes [1], audio surveillance systems [2], and biodiversity analysis [3]. Currently, SED is still an open research task, especially for *polyphonic* SED, where events from different classes may overlap with each other. Recently, several methods for SED have been proposed based on deep neural networks [4]. Amongst different neural network architectures, convolutional neural networks (CNNs) are the most commonly used ones [5]. Recurrent neural networks (RNNs) are also used in some works [6] to capture the temporal information in the audio signal. Recent research shows the effectiveness of combining CNNs with RNNs for SED [7].

In real-world scenarios, an SED model could be tested on audio data which have a different distribution from the dataset that the model is trained on. The datasets could be different in several aspects, in terms of recording conditions, acoustic environment, and sound sources amongst other factors. Such mismatch amongst datasets causes a *domain shift* [8] and typically results in a degradation of model performance. The domain where the training dataset comes from is referred to as the *source domain*, and the domain where the test dataset comes from is referred to as the *target domain*. In many scenarios such as SED, labelling the target domain samples might not be possible or practical. In this case, a domain adaptation model is needed to address the domain shift problem. When the target domain samples are fully unlabeled, this is an instance of *unsupervised domain adaptation* – see e.g. [9] for an unsupervised domain adaptation model for image classification.

Recently, inspired by generative adversarial networks (GANs) [10], several adversarial-based domain adaptation methods have been proposed to address the domain shift problem [11, 12, 13]. In [14], a general framework for adversarial domain adaptation models is proposed. Based on this framework, an unsupervised domain adaptation model was proposed in [15] for acoustic scene classification.

In this paper, we propose a domain adaptation model for SED. Prior work on domain adaptation mainly focuses on other research fields, such as text sentiment classification [16], speech recognition [17] and image object detection [18]. The model proposed in this paper is based on the work originally proposed in [15] for domain adaptation in acoustic scene classification. Here, we apply and adapt the model to SED, which to the authors' knowledge is the first attempt to construct a domain adaptation model for this task.

The model we propose is trained using labeled data from the source domain and unlabeled data from the target domain. Thus, our model is an unsupervised domain adaptation model [9]. We also propose a new dataset of audio recordings with corresponding sound event annotations collected in Singapore. We consider recordings recorded by different devices as different domains. The main contributions of this paper are as follows:

- We propose the first domain adaptation model for (multi-label) polyphonic SED.
- A new dataset is collected containing recordings recorded by different devices. This is the first real-world dataset for SED suitable for domain adaptation.

2. DATA COLLECTION

2.1. Background and Motivation

DCASE[1], standing for Detection and Classification of Acoustic Scenes and Events, is a community focusing on different computational sound scene analysis tasks. We aim to make the proposed SED dataset as an extension of the dataset for DCASE 2017 Task 3 ('Sound event detection in real life audio') [19], so that the performance of a model on our dataset will be comparable to the results

This work was performed while HZ was visiting the National University of Singapore as a research intern. EB is supported by RAEng Research Fellowship RF/128 and a Turing Fellowship.

[1] `http://dcase.community/`

on the DCASE dataset, whilst being able to carry out research on domain adaptation for this task. Therefore, we follow a similar data collection procedure as the one proposed in [20].

The motivation to propose this dataset is that although there are several publicly available datasets on SED, none of them have addressed the problem of domain adaptation. Also, most existing datasets have been compiled in Europe, where the acoustic environments and sound source characteristics could be different from Asian countries. Therefore, we propose our own dataset recorded in Singapore, targeting the domain adaptation task for SED.

2.2. Audio Recording

To make sure that the dataset can be useful for evaluating domain adaptation across recording devices, two different devices are used to perform simultaneous audio recording. The devices used for high-quality audio recording are a Roland CS-10EM binaural microphone and a Zoom H5 portable recorder, which are similar to the recording devices for collecting the DCASE 2017 Task 3 dataset [19]. The device used for low-quality audio recording is an iPhone XS smartphone. At the same time, a Valore MAXIMAL action camera is used for video recording in order to facilitate the annotation work. Each recording is around 5 minutes long. In total, the dataset contains 100 recordings, at approximately 9 hours total duration.

2.3. Annotation and Post-Processing

Five classes of sound events are annotated for this dataset: car, children, large vehicle, people speaking and people walking. These are the same classes as DCASE 2017 Task 3 [19], except for the breaks squeaking event which is not included in this dataset, considering that it is not commonly seen compared with other five sound event classes. Weather conditions (in our case, whether it is raining or not) under which each recording is recorded are also collected and annotated in the metadata file to address the problem of mismatch between domains in terms of different types of background noise.

The high-quality recordings are recorded using 44.1kHz sampling rate in WAV format and the low-quality recordings are automatically resampled to 48kHz by the recording device (iPhone XS) in M4A format. During post-processing, we convert the low-quality recordings to WAV format and resample them to 44.1kHz sampling rate. Aside from this, high-quality and low-quality recordings are temporally aligned so that two recordings recorded simultaneously will share the same annotation file.

2.4. Domain Adaptation

By proposing this dataset, the domain adaptation problem for SED can be addressed in the following three aspects:

- Mismatch of the recording conditions, which is addressed by using different types of recording equipment.
- Mismatch of sound event characteristics and acoustic environments, which can be addressed by comparing our dataset recorded in Singapore with datasets which are recorded in other countries, such as the DCASE 2017 Task 3 dataset [19].
- Mismatch of the background noise, which is addressed by recording under different weather conditions.

In this paper, we carry out domain adaptation experiments for the first two aspects. The mismatch of the recording conditions is addressed by performing experiments over our dataset with high-quality recordings and low-quality recordings. The mismatch of the sound event characteristics is addressed by performing experiments over our dataset, which is recorded in Singapore, and the DCASE 2017 Task 3 dataset [19], which is recorded in Finland.

3. DOMAIN ADAPTATION MODEL

In this section, we propose an adversarial domain adaptation model for SED. Our model is an unsupervised model which means no labeled data from the target domain will be used to train the model. We follow the general framework proposed in [14] and the weights for source and target models are not tied.

3.1. Overview

We use $(X_s, Y_s) = \{(X_s^1, Y_s^1), \dots, (X_s^{N_s}, Y_s^{N_s})\}$ to denote labeled data from the source domain, where N_s is the number of source domain samples. Suppose that there are K classes of sound events in total, then $Y_s \in \{1, 2, \dots, K\}$. Similarly, unlabeled data from the target domain is denoted as $X_t = \{X_t^1, \dots, X_t^{N_t}\}$, where N_t is the number of target domain samples. Our goal is to train a representation mapping M_s for the source domain and another representation mapping M_t for the target domain. We want to train M_s and M_t so that the distance between the source domain representation $M_s(X_s)$ and the target domain representation $M_t(X_t)$ is minimized. After that, the classifier C which is trained only using the source domain data (X_s, Y_s) can be applied to the target domain data X_t as well.

3.2. Detailed Steps

There are three steps to build and test this domain adaptation model: pre-training, adversarial training and testing.

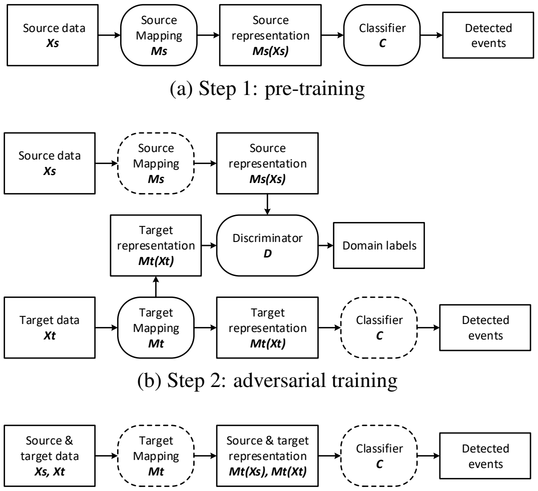
In the first step, we train a source domain representation mapping M_s and a classifier C, using labeled data (X_s, Y_s) from the source domain. The goal of the step is to obtain a representation mapping M_s and a classifier C which can work on source domain data. If $C(M_s(X_s))$ denotes the output of applying the classifier C on the source domain representation $M_s(X_s)$, the loss function for training M_s and C is then defined as follows:

$$\min_{M_s, C} L_s = -\frac{1}{N_s} \sum_{n=1}^{N_s} \sum_{k=1}^{K} \mathbb{1}_{[k=Y_s^n]} \log C(M_s(X_s^n)), \qquad (1)$$

where $\mathbb{1}_{[k=Y_s^n]} = 1$ if $k = Y_s^n$ and 0 otherwise.

The second step is the adversarial training step. Since we are dealing with the source and target domains which have the same label space, the classifier remains fixed in this step. Only the target domain representation mapping M_t is trained in an adversarial way (similar with GANs [10]) to map the target domain data to the same feature space as the source domain representation. A domain discriminator D is introduced in this step to perform domain classification on $M_s(X_s)$ and $M_t(X_t)$, i.e. the output of D is binary, indicating whether the input is from the source or the target domain. We perform alternating optimization to train the discriminator D and the target domain representation mapping M_t. To train the discriminator D, we use the following loss:

$$\begin{aligned}\min_{D} L_d = & - \frac{1}{N_s} \sum_{n=1}^{N_s} \log D(M_s(X_s^n)) \\ & - \frac{1}{N_t} \sum_{n=1}^{N_t} \log(1 - D(M_t(X_t^n))). \qquad (2)\end{aligned}$$



(a) Step 1: pre-training

(b) Step 2: adversarial training

(c) Step 3: testing

Fig. 1: Detailed steps of the domain adaptation model. Models in dashed boxes are not trained in the respective step.

When training M_t, we use the following loss:

$$\begin{aligned}\min_{M_t} L_t = & -\frac{1}{N_t}\sum_{n=1}^{N_t}\log D(M_t(X_t^n)) \\ & -\frac{1}{N_s}\sum_{n=1}^{N_s}\sum_{k=1}^{K}\mathbb{1}_{[k=Y_s^n]}\log C(M_t(X_s^n)). \quad (3)\end{aligned}$$

Eq. (3) does not follow the framework proposed in [14]. The first term in Eq. (3) maximizes the domain confusion while the second term minimizes the classification error on the source domain data X_s using the target mapping M_t. This is because we observe that by only maximizing the domain confusion, the training process is not stable and the model gives a worse detection accuracy on both the source and target domain. Therefore, we adopt the method proposed in [15] which adds an extra classification error term to the loss function. This gives a more stable adversarial training process and a better detection accuracy on both the source and target domains.

In the final testing step, sound event detection for both the source and target domain data is performed using the target domain representation mapping M_t and the classifier C.

Fig. 1 illustrates how these three steps work. Models with dashed lines (e.g. source mapping and classifier in step 2) are fixed during that step.

3.3. Model Architecture

The architecture of our model is as follows. Since our domain adaptation framework does not depend on a certain model architecture, we use a convolutional recurrent neural network (CRNN) [21] as a base model, which is currently the state-of-the-art for SED [22]. The input of the model is 40 Mel-band energies calculated by applying a 2048 samples Hamming window with 50% overlap. The Mel-band energies are then divided into several sequences with a length of 256. Samples with multiple channels are averaged to a single channel. Therefore, the shape of the input is 256 × 40. For source and target mappings, the models share the same architecture, consisting of three convolutional layers, all of which have 128 filters with the size of $(3, 3)$. Each convolutional layer is followed by a max pooling layer. The size of the kernels for the three max pooling layers is $\{(1, 5), (1, 2), (1, 2)\}$. We only apply max pooling to the frequency dimension because we do not want any information in the time dimension to be lost. The classifier is an RNN classifier which has two recurrent layers with a size of 32. The recurrent layers are followed by two fully connected layers with a size of $\{16, 5\}$. We use ReLU as the activation function for the output of all layers, except for the output layer which uses sigmoid as the activation function. We also apply a 50% dropout rate for all the layers. For the discriminator, it has three RNN layers with a size of 32, followed by three hidden fully connected layers with 64, 64 and 16 nodes. The output layer of the discriminator has one node with a sigmoid activation function.

4. EXPERIMENTS

In this section, we present the experimental results of our proposed domain adaptation model on different datasets as the source and target domains. Our proposed dataset (referred to as the 'SG' dataset) has two subsets, one with high-quality recordings ('SG-high' dataset), the other one with low-quality recordings ('SG-low' dataset). The dataset for DCASE 2017 Task 3 will be referred to as the DCASE dataset. Our proposed domain adaptation model will be referred to as A-CRNN, standing for adapted CRNN.

The SG dataset (both SG-high and SG-low) is manually split into training, validation and test subsets and each of them contains 60%, 20% and 20% of the recordings respectively. For the DCASE dataset, we use the original splits proposed in [20]. Hyperparameters are determined using the validation set and the model with the best performance on the validation set is applied on the test set to report the results.

4.1. Evaluation Metrics

We use the segment-based F-score (segment size: 1 second) and error rate for evaluation [23]. An ideal SED system should have an F-score of 1 and an error rate of 0.

4.2. Experiment Results

4.2.1. Overall Performance

In this section, we report the performance of A-CRNN. All results are presented in Tables 1-4. In each table, the first row presents the results of the pre-trained non-adapted model, CRNN, which is trained only using the labeled data from the source domain. The second row presents results of the adapted model, A-CRNN, which is trained based on the pre-trained non-adapted CRNN model, using labeled source domain data and unlabeled target domain data.

Tables 1 and 2 present the results of the experiments where the source domain is the SG-high dataset, and the target domain is the SG-low dataset (Table 1) and the DCASE dataset (Table 2). Tables 3 and 4 present the results of the experiments where the source domain is the DCASE dataset, and the target domain is the SG-high dataset (Table 3) and the SG-low dataset (Table 4). Note a different train-validation-test split is used for the Table 1 experiment to avoid using aligned recordings for the source and target domains.

From Table 1 to Table 4, we can see that on the source domain, compared with the non-adapted CRNN model, the performance of A-CRNN only drops marginally, and there is a small improvement of the performance on the source domain in Table 1. On the target domain, the performance of A-CRNN clearly improves after adaptation.

Table 1: Results (Source: SG-high; Target: SG-low)

Model	Source domain		Target domain	
	F-score	Error rate	F-score	Error rate
CRNN	0.583	0.620	0.442	0.743
A-CRNN	0.590	0.609	0.480	0.688

Table 2: Results (Source: SG-high; Target: DCASE)

Model	Source domain		Target domain	
	F-score	Error rate	F-score	Error rate
CRNN	0.470	0.793	0.256	0.947
A-CRNN	0.427	0.869	0.458	0.826

Among the four experiments, the first one (SG-high to SG-low) is the easiest one, because the only difference between the source and target domains is the recording quality. That is why the performance on the source domain increases for this experiment only.

For the remaining experiments, in the second (SG-high to DCASE) and the third experiments (DCASE to SG-high), the source and target domain recordings have different sound event characteristics and acoustic environments which is harder for domain adaptation compared to the first experiment. The last one (DCASE to SG-low) would be the most difficult experiment for domain adaptation, for the reason that it covers all the domain adaptation aspects in the first three experiments. That would also explain why the experimental results in Table 4 show the smallest improvement on the target domain, compared with Tables 2 and 3.

4.2.2. *Class-wise Performance*

Tables 5 and 6 present results in terms of the class-wise F-score on the various target domains. From the tables we can see that after adaptation, the improvement is most significant for the car class, for all the experiments. We assume that it is because among all five event classes, the car class is a relatively simple one, especially compared with children and people speaking classes - car sounds mostly correspond to low-frequency sound events without complex temporal structures.

It can also be noticed that, for the large vehicle class, the F-score for the non-adapted model (CRNN) is better than the adapted model (A-CRNN) for three of the experiments. For other classes, the results are not consistent among the four experiments. For example, for the people speaking class the F-score improves after adaptation for the first two experiments (in Table 5) but drops for the last two experiments (in Table 6).

5. CONCLUSIONS

In this paper, we propose the first real-world dataset for SED which addresses the problem of domain adaptation. The recordings are collected by different devices which can be considered as different domains. We also propose A-CRNN, an unsupervised adversarial domain adaptation model based on a CRNN baseline model. Results show that the proposed A-CRNN model reports improved target domain performance when using a different dataset or a different recording device, with a small decrease in source domain performance.

Table 3: Results (Source: DCASE; Target: SG-high)

Model	Source domain		Target domain	
	F-score	Error rate	F-score	Error rate
CRNN	0.528	0.705	0.163	1.072
A-CRNN	0.514	0.716	0.301	0.960

Table 4: Results (Source: DCASE; Target: SG-low)

Model	Source domain		Target domain	
	F-score	Error rate	F-score	Error rate
CRNN	0.528	0.705	0.223	1.097
A-CRNN	0.511	0.757	0.295	0.936

Table 5: Class-wise F-score on the target domain

Class name	SG-high to SG-low		SG-high to DCASE	
	CRNN	A-CRNN	CRNN	A-CRNN
car	0.448	0.498	0.289	0.638
children	0.406	0.490	0.226	0.250
large vehicle	0.502	0.534	0.397	0.197
people speaking	0.467	0.514	0.087	0.118
people walking	0.008	0.103	0.000	0.000

Table 6: Class-wise F-score on the target domain

Class name	DCASE to SG-high		DCASE to SG-low	
	CRNN	A-CRNN	CRNN	A-CRNN
car	0.256	0.473	0.357	0.479
children	0.072	0.005	0.235	0.005
large vehicle	0.119	0.004	0.109	0.024
people speaking	0.297	0.056	0.334	0.149
people walking	0.081	0.096	0.082	0.104

Although the A-CRNN has shown improved detection accuracy on the target domain compared with the CRNN baseline, there still exist many future research directions. First, further experiments can be done using different non-adapted model architectures for pre-training. Theoretically, our method should work regardless of the baseline model architecture. Besides, as shown in Section 4.2.2, performance on some of the classes is not improving after adaptation. This is because the model proposed in this paper is quite a general domain adaptation model and more work could be done to focus on specific sound event characters, such as temporal structures, to improve performance on these classes.

There are also some other domain shift aspects that have not been addressed in this paper. For example, the source and target domains might have different label spaces. Besides, our model is an unsupervised domain adaptation model while sometimes a few labeled target domain samples are available, in which case a semi-supervised domain adaptation model is needed.

6. REFERENCES

[1] Sacha Krstulović, "Audio event recognition in the smart home," in *Computational Analysis of Sound Scenes and Events*, pp. 335–371. Springer, 2018.

[2] Aki Harma, Martin F McKinney, and Janto Skowronek, "Automatic surveillance of the acoustic activity in our living environment," in *2005 IEEE International Conference on Multimedia and Expo*. IEEE, 2005, pp. 4–pp.

[3] Dan Stowell, Emmanouil Benetos, and Lisa F Gill, "On-bird sound recordings: automatic acoustic recognition of activities and contexts," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1193–1206, 2017.

[4] Annamaria Mesaros, Toni Heittola, Emmanouil Benetos, Peter Foster, Mathieu Lagrange, Tuomas Virtanen, and Mark D Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 2, pp. 379–393, 2018.

[5] Haomin Zhang, Ian McLoughlin, and Yan Song, "Robust sound event recognition using convolutional neural networks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 559–563.

[6] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6440–6444.

[7] Emre Cakir, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas 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.

[8] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1-2, pp. 151–175, 2010.

[9] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, 2015, pp. 1180–1189.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[11] Ming-Yu Liu and Oncel Tuzel, "Coupled generative adversarial networks," in *Advances in neural information processing systems*, 2016, pp. 469–477.

[12] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4068–4076.

[13] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto, "Few-shot adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, 2017, pp. 6670–6680.

[14] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," in *Computer Vision and Pattern Recognition (CVPR)*, 2017, vol. 1, p. 4.

[15] Shayan Gharib, Konstantinos Drossos, Emre Cakir, Dmitriy Serdyuk, and Tuomas Virtanen, "Unsupervised adversarial domain adaptation for acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 138–142.

[16] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 513–520.

[17] Jun Deng, Zixing Zhang, Florian Eyben, and Björn Schuller, "Autoencoder-based unsupervised domain adaptation for speech emotion recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1068–1072, 2014.

[18] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *2011 International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 999–1006.

[19] Annamaria Mesaros and Toni Heittola, "Sound event detection in real life audio," http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-sound-event-detection-in-real-life-audio, 2017, [Online; accessed 18-October-2019].

[20] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 1128–1132.

[21] Sharath Adavanne, Giambattista Parascandolo, Pasi Pertila, Toni Heittola, and Tuomas Virtanen, "Sound event detection in multichannel audio using spatial and harmonic features," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, September 2016, pp. 6–10.

[22] Annamaria Mesaros and Toni Heittola, "Sound event detection in real life audio - challenge results," http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-sound-event-detection-in-real-life-audio-results, 2017, [Online; accessed 18-October-2019].

[23] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, pp. 162, 2016.