HIERARCHICAL DETECTION OF SOUND EVENTS AND THEIR LOCALIZATION USING CONVOLUTIONAL NEURAL NETWORKS WITH ADAPTIVE THRESHOLDS

Sotirios Panagiotis Chytas, Gerasimos Potamianos

Department of Electrical and Computer Engineering, University of Thessaly, Volos, Greece

schytas@uth.gr gpotam@ieee.org

ABSTRACT

This paper details our approach to Task 3 of the DCASE'19 Challenge, namely sound event localization and detection (SELD). Our system is based on multi-channel convolutional neural networks (CNNs), combined with data augmentation and ensembling. Specifically, it follows a hierarchical approach that first determines adaptive thresholds for the multi-label sound event detection (SED) problem, based on a CNN operating on spectrograms over long-duration windows. It then exploits the derived thresholds in an ensemble of CNNs operating on raw waveforms over shorter-duration sliding windows to provide event segmentation and labeling. Finally, it employs event localization CNNs to yield direction-of-arrival (DOA) source estimates of the detected sound events. The system is developed and evaluated on the microphone-array set of Task 3. Compared to the baseline of the Challenge organizers, on the development set it achieves relative improvements of 12% in SED error, 2% in F-score, 36% in DOA error, and 3% in the combined SELD metric, but trails significantly in frame-recall, whereas on the evaluation set it achieves relative improvements of 3% in SED, 51% in DOA, and 4% in SELD errors. Overall though, the system lags significantly behind the best Task 3 submission, achieving a combined SELD error of 0.2033 against 0.044 of the latter.

Index Terms— Sound event detection and localization, convolutional neural networks, DCASE19

1. INTRODUCTION

Sound event detection (SED) constitutes an active research area with many applications, such as medical telemonitoring [1] and surveillance [2]. Not surprisingly, SED has been the subject of multiple evaluation campaigns in the literature, including the recent and well-established DCASE Challenges [3–5]. Moreover, alongside SED, in many applications [6, 7] it is also crucial to determine the location or, more coarsely, the direction of arrival (DOA) of each detected sound event source. Thus, in Task 3 of the 2019 DCASE Challenge [8], both problems are considered jointly (SED and DOA estimation of the detected events). The task is referred to as sound event localization and detection (SELD), and it is addressed in an indoors scenario given multi-channel audio.

In this paper, we present our developed SELD system for Task 3 of the 2019 DCASE Challenge [8]. As deep-learning based methods are well-established, outperforming traditional machine learning ones in both SED [9–12] and DOA estimation [13,14], we adopt a deep-learning approach. In particular, we employ convolutional neural networks (CNNs) to first address SED, i.e., determine the existence of each class at each time-frame, and to subsequently estimate the DOA for each of the audio segments predicted to exist. Notably, for SED we follow a hierarchical approach, where, first, a CNN operating over long-duration audio windows determines adaptive thresholds indicating how likely it is for each class to exist, and, subsequently, an ensemble of CNNs operating over shorter-duration windows determines the exact moments each class occurs.

The remainder of the paper is organized as follows: Section 2 overviews the Challenge dataset; Section 3 focuses on the developed SELD system; Section 4 details its evaluation on the Challenge data; and, finally, Section 5 concludes the paper.

2. CHALLENGE DATASET

Task 3 of the 2019 DCASE Challenge provides two datasets of the same indoors sound scene: "Microphone Array" and "Ambisonic" [15]. In this paper, the "Microphone Array" set is employed, containing four-channel directional microphone recordings from a tetrahedral array configuration. The development dataset consists of 400 1-min long recordings sampled at 48 kHz, divided into four cross-validation splits. For the purposes of the Challenge, the given cross-validation split should be used during system development, and the use of external data is not allowed. In addition, the evaluation dataset consists of 100 1-min long recordings. Note also that, in our system, all audio data are downsampled to 16 kHz.

There exist 11 sound event classes, taken from Task 2 of the 2016 DCASE Challenge [4]. The duration of each event segment in the development set ranges from 205ms to 3.335s, and there can be at most two overlapping sounds at any given time. The number of segments is almost the same for all classes, however there exists significant variation in their total durations (see also Fig. 1).

Each segment is associated with an elevation and an azimuth value. Elevation values lie within the [-40°, 40°] range, while azimuth values are within [-180°, 170°], both at a resolution of 10°.

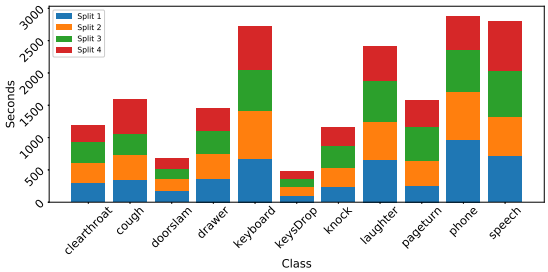

Figure 1: Class total durations in the four development set splits.

3. SYSTEM DESCRIPTION

In our method, we first address the SED sub-task and then the DOA one. Specifically, we develop a hierarchical approach to the for-

https://doi.org/10.33682/c6q0-wv87

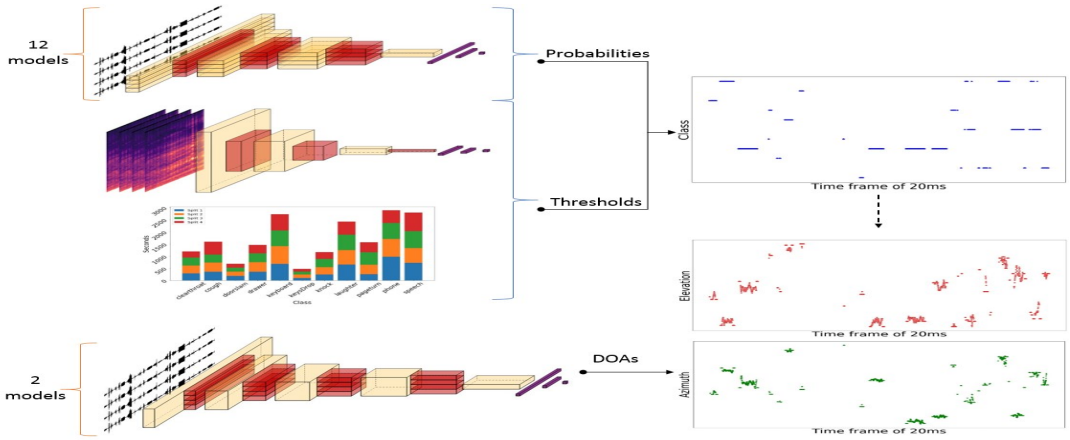

Figure 2: Overview diagram of the developed system for Task 3 of DCASE'19 (CNNs are drawn using the PlotNeuralNet software [16]).

mer, determining the existence of each sound event class at each time-frame. For this purpose, first a "*long SED model*" estimates adaptive thresholds for each class, also taking into account the class prior probabilities. Then, an ensemble of "*short SED models*" determines the exact time-frames each class exists, exploiting the aforementioned thresholds. Following SED, we utilize a *DOA model* to localize the source of each detected event, estimating its elevation and azimuth values. All models are multi-channel CNNs, operating on raw waveforms or spectrograms over sliding windows of different durations. A schematic of the system is provided in Fig. 2.

3.1. Short SED models

We create an ensemble of multi-channel CNNs (12 in total, as explained in the next paragraph), all with the architecture of Table 1. These operate on raw audio waveforms over short-duration windows of 100ms or 200ms, with these values determined after experimenting with various window lengths on the development set. We do not apply any preprocessing to the four channels (other than their downsampling to 16 kHz), and we use all four microphone data streams as input to the CNNs. The output layers of the models have 11 neurons (same as the number of sound event classes), each providing the probability of its corresponding class, following sigmoid activation. Note that, during training, windows with no sound events are kept, and windows with overlapping events are assigned to all occurring events inside them (maximum of two), while they slide in steps equal to half their duration, i.e. by 50 or 100ms. All CNNs are trained with a binary cross-entropy objective using the Adam optimizer and early stopping to prevent overfitting, employing the Keras API for development [17].

In order to have more segments with overlapping sounds, we employ data augmentation as follows: we add segments, each belonging to only one class, two at a time. Concerning the Challenge evaluation metrics, we observed that datasets with more overlapping segments tend to yield better frame-recall results, while data with less overlapping segments tend to perform better in terms of SED error and F-score. As we wish to improve all three metrics simultaneously, we choose to create different models, trained on data with various degrees of artificial overlap, and then ensemble them. Thus, we create six datasets, having 0%, 5%, 10%, 20%, 30%, and 40% extra overlapping segments, and we train two different CNNs on each (i.e. with input window sizes of 100ms and 200ms length), thus resulting to 12 models. The process is repeated for each of the four given development data splits.

3.2. Long SED model

A major issue in multi-label problems concerns the choice of class thresholds, used to decide if a class exists or not. A simple approach is to set all thresholds to 0.5, as in the Challenge baseline system [18], however their careful tuning may yield significant improvements. For example, in [9] exhaustive search is utilized to yield a single optimal threshold for all classes, whereas in [10, 11] separate thresholds are employed for each class, found by exhaustive search. Nevertheless, both approaches may be prone to overfitting due to the exhaustive search used.

To prevent overfitting, we opt to create a SED model operating on longer-duration data windows. Our motivation stems from the expectation that such a model will provide a "bigger picture" concerning class existence, and thus can help in determining class thresholds adaptively. These can then be utilized in conjunction with the outputs of the short SED models to predict the exact time-frames in which each sound event occurs.

For this purpose, we create a multi-channel CNN that operates on power spectrograms over signal windows of one-second duration (sliding in 100ms steps during training), with 128×32-dimensional spectrograms generated by libROSA [19] under its default parameters. We use all available channels, ending up with four spectrograms as input. For data augmentation, we consider all permutations of the four channels, resulting in 24 times more training data. Details of the long SED model architecture are provided in Table 2.

3.3. Adaptive thresholds and SED predictions

To determine the class thresholds, we work with a time-resolution of 20ms, exploiting the long SED model predictions. These fine-resolution predictions are obtained by averaging the coarser-resolution probabilities of each class over all 1s-long windows that contain the given 20ms time-frame, while sliding by 200ms.

A first approach is to simply set the desired thresholds to

$$\theta_c^t = 1 - \mathrm{lp}_c^t \,, \tag{1}$$

where lp_c^t denotes the long SED model prediction (probability) of class c at time-frame t, and θ_c^t is the corresponding threshold. In general, however, we do not wish the thresholds to be too close to 1, in order to guard against false negatives of the long SED model. Thus, we choose to smooth (1) by multiplying the thresholds with a number within the [0.6, 0.9] range. This number is different for each class, and it is based on its total duration in the training data

Input (4 x segment size)
100 filters, Conv 1x10, ReLU MaxPool 1x5
200 filters, Conv 1x10, ReLU MaxPool 1x6
300 filters, Conv 1x10, ReLU MaxPool 1x7
500 filters, Conv 4x1, ReLU
Flatten Dropout 0.6
1000 neurons, Dense, ReLU Dropout 0.3
11 neurons, Dense, sigmoid

Table 1: Architecture of the short SED models. Segment sizes are 1600 for 100ms windows and 3200 for 200ms ones.

Input (4x128x32)
40 filters, Conv 1x6x1, ReLU MaxPool 1x3x1
60 filters, Conv 1x1x6, ReLU MaxPool 1x1x3
80 filters, Conv 1x6x6, ReLU MaxPool 1x3x3
Flatten Dropout 0.5
500 neurons, Dense, ReLU Dropout 0.3
11 neurons, Dense, sigmoid

Table 2: Long SED model architecture.

Input (4 x segment size)
100 filters, Conv 4x10 (same padding), ReLU MaxPool 1x3
200 filters, Conv 4x10 (same padding), ReLU MaxPool 1x5
300 filters, Conv 4x10 (same padding), ReLU MaxPool 1x5
400 filters, Conv 4x10 (same padding), ReLU MaxPool 1x5
500 filters, Conv 4x1 (same padding), ReLU
Flatten Dropout 0.5
1000 neurons, Dense, ReLU Dropout 0.3
22 neurons, Dense, linear

Table 3: Architecture of the DOA models.

(class prior), meaning that less frequent classes tend to have lower thresholds. The resulting thresholds are given by

$$\theta_c^t = \left(1 - \mathrm{lp}_c^t\right)\left(0.6 + 0.3\,\frac{p_c - p_{\min}}{p_{\max} - p_{\min}}\right), \qquad (2)$$

where p_c denotes the prior of class c (based on duration), while $p_{\min}$ and $p_{\max}$ are the minimum and maximum of all class priors.

The desired SED results are finally derived at a time-resolution of 20ms, by employing the ensemble of the 12 short SED models of Section 3.2 and the adaptive thresholds of (2). Specifically, let sp_c^t denote the combined short model prediction of class c at time-frame t. This is estimated for each of the 12 CNNs by averaging the class probabilities over all windows (of length 100 or 200ms, depending on the model) that contain the given 20ms time-frame, while sliding by a 20ms step. The resulting estimates are then averaged over all 12 models of the CNN ensemble to yield sp_c^t. As a final step, class c is detected at time-frame t, whenever $\mathrm{sp}_c^t \geq \theta_c^t$.

3.4. DOA models

Following SED, we proceed to the DOA sub-task. For this purpose, and similarly to the short SED models, we create short models for DOA estimation that provide 22 numbers at their output layer, i.e. the elevation and azimuth for each of the 11 classes. The goal is, given a raw multi-channel audio segment of short duration, to predict the DOA of each class, no matter if it exists or not (SED results will determine what to keep). Specifically, we create two CNNs, with their architecture detailed in Table 3. The CNNs operate on four channels of raw audio over windows of 100ms or 200ms in duration that, during model training, slide at steps of 50ms or 100ms, respectively. For training the two networks, we use the same data as in the SED sub-task, but exclude audio with no sound events, as such data are not associated with DOA values. We employ the mean squared error loss as training objective, but slightly modified, as we calculate it only in the 2 (in the case of one class) or the 4 (for two overlapping classes) output neurons of interest. As before, we use the Adam optimizer and early stopping to prevent overfitting.

DOA estimation occurs at a time resolution of 20ms, first by averaging the elevation and azimuth predictions for the 20ms time-frame of interest within the model sliding windows, and subsequently averaging the predictions across the two models. A problem arises in this approach towards the boundaries of each segment. To prevent noisy DOA estimates there, these are smoothed by setting predictions for the first and last 300ms of each segment to the minimum or maximum of that sub-segment (depending on the relative position to the zero), thus preventing steep DOA ascents or descents. An example of this process is depicted in Fig. 3.

3.5. Submitted systems

Following the above process, we create a total of four SELD systems, each trained on one of the four given cross-validation development data splits. We then combine these four systems in two ways, thus providing two submissions to the Challenge, resulting from: (a) their average; and (b) their weighted average. In both cases, averaging occurs at the sub-component level across the four systems (e.g., each short SED CNN is averaged across the four systems first, before model ensembling). Particularly in the weighted averaging

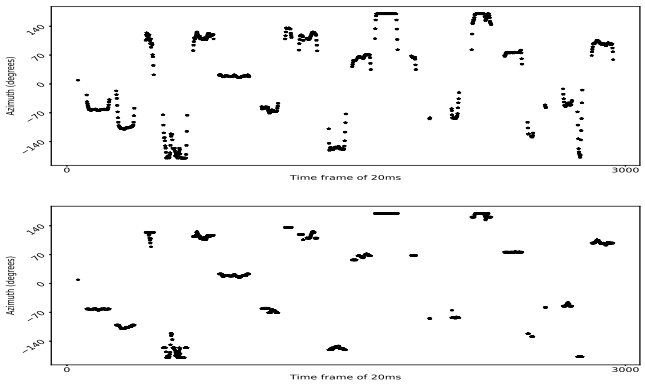

Figure 3: Example of DOA (here, azimuth) estimate smoothing at segment edges: (top) before smoothing; (bottom) after smoothing.

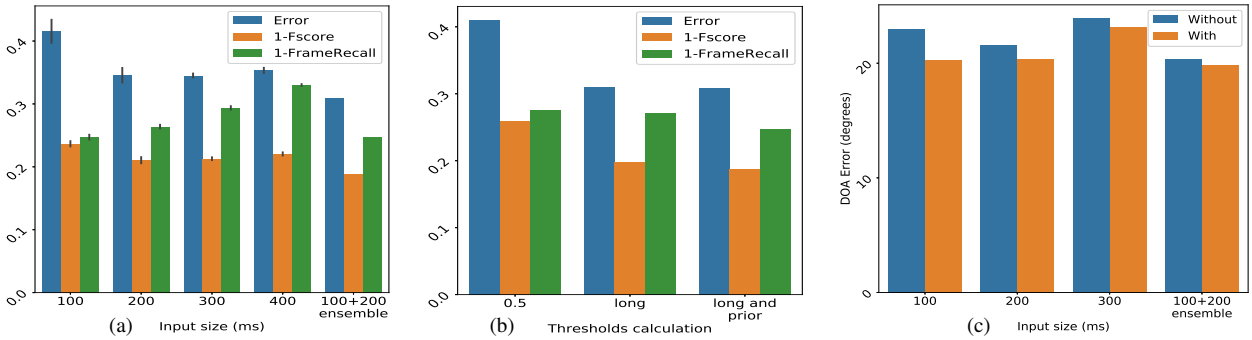

Figure 4: Evaluation of design choices of the proposed system components on the Challenge development set, namely of various: (a) short SED model window durations; (b) class threshold estimation approaches; (c) DOA model window durations with or without smoothing.

case, the performance of the four systems on the corresponding fold test-set is taken into account, based on the appropriate metrics (i.e., the average of the three SED metrics or the DOA error).

4. RESULTS

4.1. Development set results

We first present in Table 4 (top) our system performance on the development set (over its four data splits, thus there is a single result according to the evaluation paradigm), in terms of the four Challenge metrics and their combination (SELD score). We can readily observe that, compared to the Challenge baseline, our system achieves a 3% relative reduction in the SELD score (from 0.22 to 0.213). In terms of the individual metrics, the system yields relative improvements of 12% in SED error, 2% in F-score, 36% in DOA error (from 30.8°to 19.8°), but trails significantly in the frame-recall metric, where it achieves only 75.3% vs. 84.0% for the baseline.

In Fig. 4 we present results to highlight performance differences between the various design choices of our developed system components. First, in Fig. 4(a) we depict performance of the short SED models of Section 3.1 and their ensembles in terms of SED error, F-score, and frame-recall (difference from 1 is shown for the latter two). We also depict results for additional window sizes, namely 300ms and 400ms. Each bar shows results of the ensemble of six models, trained on various data augmented sets (from 0% to 40%, as discussed in Section 3.1), with the error bars indicating the range of the individual model results. Note that the 12-model ensemble results are also shown ("100+200 ensemble"). We observe that shorter window sizes (100ms) yield the best results in frame-recall, mainly because two sounds may overlap for very short periods of time, but have much worse results in SED error and F-score, because short windows may not carry adequate class information. On the other hand, medium window sizes (200ms) yield the best results in SED error and F-score, but worse frame-recall as they may fail to detect very short segments. Combining the two window sizes by model ensembling exploits the relative advantages of both, improving SED error and F-score significantly, but at minor detriment in frame-recall. Longer windows (e.g. 300ms or 400ms sizes) significantly degrade frame-recall, thus are not used in our system. Next, in Fig. 4(b) we examine the effect of class thresholds to SED performance. Thresholds fixed to 0.5 for all classes perform the worst, whereas adaptive thresholds estimated by means of (1) – labeled as "long" in the graph, perform better in all three metrics (SED error, F-score, and frame-recall). Results further improve when adaptive thresholds are computed by (2) – labeled as "long and prior" in the bar-plot. Finally, in Fig. 4(c) we consider the DOA estimation component. There, we can readily observe the importance of DOA estimate smoothing, as systems "without" smoothing perform significantly worse than systems "with" it. Also DOA models operating on windows of 100ms or 200ms in duration outperform systems built on 300ms windows. The ensemble of both 100ms and 200ms systems performs even better in terms of the DOA error metric.

set	system	SED error	F-score	frame-recall	DOA error	SELD score
dev	proposed	**0.309**	**81.2%**	75.3%	**19.8°**	**0.213**
	baseline *	0.350	80.1%	**84.0%**	30.8°	0.220
eval	proposed (a)	**0.29**	82.4%	75.6%	**18.6°**	**0.2033**
	proposed (b)	**0.29**	82.3%	75.7%	18.7°	0.2034
	baseline *	0.30	**83.2%**	**83.4%**	38.1°	0.2114
	best	0.08	94.7%	96.8%	3.7°	0.044

Table 4: Comparison of our system on the development (dev) and evaluation set (eval) of DCASE'19 Task 3 against the Challenge baseline (*: on Microphone Array data), in terms of the five task metrics. Performance of the best-scoring submission is also shown.

4.2. Evaluation set results

Finally, in Table 4 (bottom) we present our system performance on the Challenge evaluation set. Both system variants of Section 4.2 are shown: (a) averaging; and (b) weighted averaging. They perform similarly, with variant (a) being slightly superior. Compared to the baseline, it yields a slight 4% relative reduction in the SELD metric (from 0.2114 to 0.2033), with the greatest improvement in the DOA error metric (51% relative reduction, from 38.1°to 18.6°). It should be noted however that the proposed system lags significantly behind the best overall submission in Task 3 in all metrics.

5. CONCLUSIONS

We presented a SELD system for Task 3 of the DCASE'19 Challenge using CNNs only, separately addressing SED and DOA estimation, while making no explicit assumptions about the maximum possible number of overlapping segments. We followed a hierarchical approach to SED, first determining adaptive class thresholds based on a CNN operating over longer windows, which we then utilized in an ensemble of CNNs operating on shorter waveforms, also exploiting data augmentation in their training. Our system outperformed the baseline, particularly in DOA error, exhibiting consistent performance across development and evaluation sets, but trailed the best Challenge submission considerably.

6. REFERENCES

[1] N. C. Phuong and T. D. Dat, "Sound classification for event detection: Application into medical telemonitoring," in *Proc. International Conference on Computing, Management and Telecommunications (ComManTel)*, 2013, pp. 330–333.

[2] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, 2005, pp. 1306–1309.

[3] http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-sound-event-detection-in-real-life-audio.

[4] http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-synthetic-audio.

[5] http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-real-life-audio.

[6] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio surveillance: A systematic review," *ACM Computing Surveys*, vol. 48, no. 4, pp. 52:1–52:46, 2016.

[7] C. J. Grobler, C. P. Kruger, B. J. Silva, and G. P. Hancke, "Sound based localization and identification in industrial environments," in *Proc. Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2017, pp. 6119–6124.

[8] http://dcase.community/challenge2019/task-sound-event-localization-and-detection.

[9] Y. Chen, Y. Zhang, and Z. Duan, "DCASE2017 sound event detection using convolutional neural network," DCASE2017 Challenge, Tech. Rep., 2017.

[10] I.-Y. Jeong, S. Lee, Y. Han, and K. Lee, "Audio event detection using multiple-input convolutional neural network," DCASE2017 Challenge, Tech. Rep., 2017.

[11] C.-H. Wang, J.-K. You, and Y.-W. Liu, "Sound event detection from real-life audio by training a long short-term memory network with mono and stereo features," DCASE2017 Challenge, Tech. Rep., 2017.

[12] R. Lu and Z. Duan, "Bidirectional GRU for sound event detection," DCASE2017 Challenge, Tech. Rep., 2017.

[13] X. Zhang and D. Wang, "Deep learning based binaural speech separation in reverberant environments," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1075–1084, 2017.

[14] R. Takeda and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 405–409.

[15] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and detection," in *Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019. [Online]. Available: https://arxiv.org/abs/1905.08546

[16] https://github.com/HarisIqbal88/PlotNeuralNet.

[17] https://keras.io/.

[18] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2019.

[19] https://librosa.github.io/librosa/.