A HYBRID PARAMETRIC-DEEP LEARNING APPROACH FOR SOUND EVENT LOCALIZATION AND DETECTION

Andrés Pérez-López[1,2], *Eduardo Fonseca*[1*], *Xavier Serra*[1]

[1] Music Technology Group, Universitat Pompeu Fabra, Barcelona
{andres.perez, eduardo.fonseca, xavier.serra}@upf.edu
[2] Eurecat, Centre Tecnològic de Catalunya, Barcelona

ABSTRACT

This work describes and discusses an algorithm submitted to the *Sound Event Localization and Detection* Task of DCASE2019 Challenge. The proposed methodology relies on parametric spatial audio analysis for source localization and detection, combined with a deep learning-based monophonic event classifier. The evaluation of the proposed algorithm yields overall results comparable to the baseline system. The main highlight is a reduction of the localization error on the evaluation dataset by a factor of 2.6, compared with the baseline performance.

Index Terms— SELD, parametric spatial audio, deep learning

1. INTRODUCTION

Sound Event Localization and Detection (SELD) refers to the problem of identifying, for each individual event present in a sound field, the temporal activity, spatial location, and sound class to which it belongs. SELD is a current research topic which deals with microphone array processing and sound classification, with potential applications in the fields of signal enhancement, autonomous navigation, acoustic scene description or surveillance, among others.

SELD arises from the combination of two different problems: Sound Event Detection (SED) and Direction of Arrival (DOA) estimation. The number of works in the literature which jointly address SED and DOA problems is relatively small. It is possible to classify them by the type of microphone arrays used: distributed [1, 2, 3] or near-coincident [4, 5, 6]. As mentioned in [6], the usage of near-coincident circular/spherical arrays enables the representation of the sound field in the spatial domain, using the spherical harmonic decomposition, also known as Ambisonics [7, 8]. Such spatial representation allows a flexible, device-independent comparison between methods. Furthermore, the number of commercially available ambisonic microphones has increased in recent years due to their suitability for immersive multimedia applications. Taking advantage of the compact spatial representation provided by the spherical harmonic decomposition, several methods for parametric analysis of the sound field in the ambisonic domain have been proposed [9, 10, 11, 12]. These methods ease sound field segmentation into direct and diffuse components, and further localization of the direct sounds. The advent of deep learning techniques for DOA estimation has also improved the results of traditional methods [6]. However, none of the deep learning-based DOA estimation methods explicitly exploits the spatial parametric analysis. This situation is further extended to the SELD problem, with the exception of [5], where DOAs are estimated from the *active intensity vector* [9].

The motivation for the proposed methodology is two-fold. First, we would like to check whether the usage of spatial parametric analysis in the ambisonic domain can improve the performance of SELD algorithms. Second, temporal information derived by the parametric analysis could be further exploited to estimate event onsets and offsets, thus lightening the event classifier complexity; such reduction might positively impact algorithm's performance.

In what follows, we present the methodology and the architecture of the proposed system (Section 2). Then, we describe the design choices and the experimental setup (Section 3), and discuss the results in the context of DCASE2019 Challenge - Task 3 (Section 4). A summary is presented in Section 5. In order to support open access and reproducibility, all code is freely available at [13].

2. METHOD

The proposed method presents a solution for the SELD problem splitting the task into four different problems: *DOA estimation*, *association*, *beamforming* and *classification*, which will be described in the following subsections. The former three systems follow a heuristic approach—in what follows, they will be jointly referred to as the *parametric front-end*. Conversely, the *classification* system is data-driven, and will be referred to as the *deep learning back-end*. The method architecture is depicted in Figure 1.

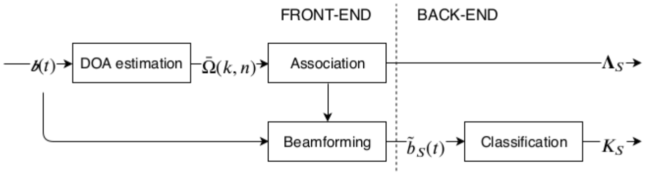


Figure 1: System architecture.

2.1. DOA estimation

The *DOA estimation* system (Figure 2) is based on parametric time-frequency (TF) spatial audio analysis. Let us consider a first-order ($L = 1$) ambisonic signal vector $\boldsymbol{b}(t)$ with N3D normalization [14]:

$$\boldsymbol{b}(t) = [b_w(t), \sqrt{3}b_x(t), \sqrt{3}b_y(t), \sqrt{3}b_z(t)]. \quad (1)$$

From its short-time frequency domain representation $\boldsymbol{B}(k,n)$, the instantaneous DOA at each TF bin $\boldsymbol{\Omega}(k,n)$ can be estimated as:

$$\begin{gathered}\boldsymbol{I}(k,n) = -\frac{1}{Z_0}\mathbb{R}\{[B_x(k,n), B_y(k,n), B_z(k,n)]B_w(k,n)^*\},\\ \boldsymbol{\Omega}(k,n) = [\varphi(k,n), \theta(k,n)] = \angle(-\boldsymbol{I}(k,n)),\end{gathered} \quad (2)$$

[*] E.F. is partially supported by a Google Faculty Research Award 2018.

https://doi.org/10.33682/v1za-0k45

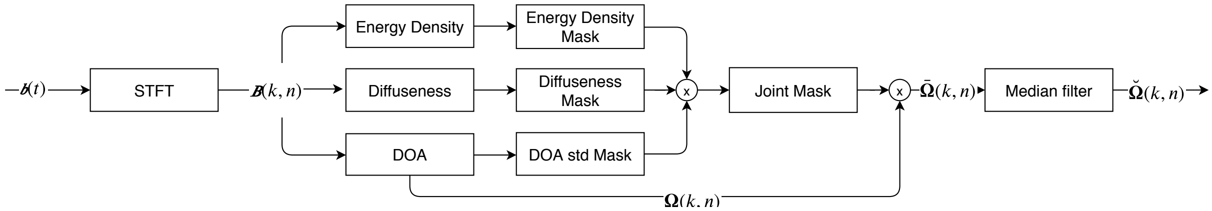


Figure 2: DOA estimation architecture.

where $\boldsymbol{I}(k,n)$ stands for the *active intensity vector* [9], Z_0 is the characteristic impedance of the medium, * represents the complex conjugate operator, and $\angle$ is the spherical coordinates angle operator, expressed in terms of azimuth φ and elevation θ.

It is desirable to identify the TF regions of $\boldsymbol{\Omega}(k,n)$ which carry information from the sound events, and discard the rest. Three binary masks are computed with that aim. The first mask is the *energy density mask*, which is used as an activity detector. The energy density $E(k,n)$ is defined as in [15] :

$$E(k,n) = \frac{|B_w(k,n)|^2 + ||[B_x(k,n), B_y(k,n), B_z(k,n)]||^2}{2Z_0 c}, \tag{3}$$

with c being the sound speed. A gaussian adaptive thresholding algorithm is then applied to $E(k,n)$, which selects TF bins with local maximum energy density, as expected from direct sounds.

The *diffuseness mask* selects the TF bins with high energy propagation. Diffuseness $\Psi(k,n)$ is defined in [16] as:

$$\Psi(k,n) = 1 - ||\,\langle\, \boldsymbol{I}(k,n) \,\rangle\,|| / (c \,\langle\, E(k,n) \,\rangle), \tag{4}$$

where $\langle\,\cdot\,\rangle$ represents the temporal expected value.

The third mask is the *DOA variance mask*. It tries to select TF regions with small standard deviation[1] with respect to their neighbor bins—a characteristic of sound fields with low diffuseness [12].

The three masks are then applied to the DOA estimation, obtaining the TF-filtered DOAs $\bar{\boldsymbol{\Omega}}(k,n)$. Finally, a median filter is applied, with the aim of improving DOA estimation consistency and removing spurious TF bins. The median filter is applied in a TF bin belonging to $\bar{\boldsymbol{\Omega}}(k,n)$ only if the number of TF bins belonging to $\bar{\boldsymbol{\Omega}}(k,n)$ in its vicinity is greater than a given threshold B_{min}. The resulting filtered DOA estimation is referred to as $\breve{\boldsymbol{\Omega}}(k,n)$.

2.2. Association

The association step (Figure 3) tackles the problem of assigning the time-frequency-space observation $\breve{\boldsymbol{\Omega}}(k,n)$ to a set of events, each one having a specific onset, offset and location. First, DOA estimates are resampled into *frames* of the task's required length (0.02 s). In what follows, frames will be represented by index m. An additional constraint is applied: for a given window n_0, the DOA estimates $\breve{\boldsymbol{\Omega}}(k,n_0)$ are assigned to the corresponding frame m_0 only if the number of estimates is greater than a threshold K_{min}.

Next, the standard deviation in azimuth (σ_φ) and elevation (σ_θ) of the frame-based DOA estimates $\breve{\boldsymbol{\Omega}}(k,m)$ are compared to a threshold value (σ_{max}), and the result is used to estimate the frame-based event overlapping amount $o(m)$:

$$o(m) = \begin{cases} 1, & \text{if } \sigma_\varphi/2 + \sigma_\theta < \sigma_{max}, \\ 2, & \text{otherwise.} \end{cases} \tag{5}$$

The clustered values $\boldsymbol{\Omega}_{\text{cluster}}(m)$ are then computed as the $K = o(m)$ centroids of $\breve{\boldsymbol{\Omega}}(k,m)$, using a modified version of K-Means which minimizes the central angle distance. Notice that, for $o(m) = 1$, the operation is equivalent to the median.

The following step is the grouping of clustered DOA values into events. Let us define $\boldsymbol{\Omega}_S(m)$ as the frame-wise DOA estimations belonging to the event S. A given clustered DOA estimation $\boldsymbol{\Omega}_{\text{cluster}}(m)$ belongs to the event S if the following criteria are met:

- The central angle between $\boldsymbol{\Omega}_{\text{cluster}}(m)$ and the median of $\boldsymbol{\Omega}_S(m)$ is smaller than a given threshold d_{max}^{ANGLE}, and
- The frame distance between M and the closest frame of $\boldsymbol{\Omega}_S(m)$ is smaller than a given threshold d_{max}^{FRAME}.

The resulting DOAs $\boldsymbol{\Omega}_S(m)$ are subject to a postprocessing step with the purpose of delaying event onsets in frames where $o(m) > 2$, and discarding events shorter than a given minimum length. Finally, the frame-based event estimations are converted into *metadata annotations* in the form $\boldsymbol{\Lambda}_S = (\boldsymbol{\Omega}_S, \text{onset}_S, \text{offset}_S)$.

2.3. Beamforming

The last step performed in the front-end is the input signal segmentation. The spatial and temporal information provided by the annotations $\boldsymbol{\Lambda}_S$ are used to produce monophonic signal estimations of the events, $\tilde{b}_S(t)$, as the signals captured by a virtual hypercardioid:

$$\tilde{b}_S(t) = \boldsymbol{Y}(\boldsymbol{\Omega}_S)\boldsymbol{b}^{\mathsf{T}}(t), \tag{6}$$

where $\boldsymbol{Y}(\boldsymbol{\Omega}_S) = [Y_w(\boldsymbol{\Omega}_S), Y_x(\boldsymbol{\Omega}_S), Y_y(\boldsymbol{\Omega}_S), Y_z(\boldsymbol{\Omega}_S)]$ is the set of real-valued spherical harmonics up to order $L = 1$ evaluated at $\boldsymbol{\Omega}_S$.

2.4. Deep learning classification back-end

The parametric front-end performs DOA estimation, temporal activity detection and time/space segmentation, and produces monophonic estimations of the events, $\tilde{b}_S(t)$. Then, the back-end classifies the resulting signals as belonging to one of a target set of 11 classes. Therefore, the multi-task nature of the front-end allows us to define the back-end classification task as a simple multi-class problem, even though the original SELD task is multi-label. It must be noted, however, that due to the limited directivity of the first-order beamformer, the resulting monophonic signals can present a certain leakage from additional sound sources when two events overlap, even when the annotations $\boldsymbol{\Lambda}_S$ are perfectly estimated.

The classification method is divided into two stages. First, the incoming signal is transformed into the log-mel spectrogram and split into TF patches. Then, the TF patches are fed into a single-mode based on a Convolutional Recurrent Neural Network (CRNN), which outputs probabilities for event classes $k \in \{1...K\}$, with $K = 11$. Predictions are done at the event-level (not at the frame level), since the temporal activities have been already determined by the front-end.

[1] In this work, all statistical operators for angular position refer to the 2π-*periodic* operator for azimuth, and the standard operator for elevation.

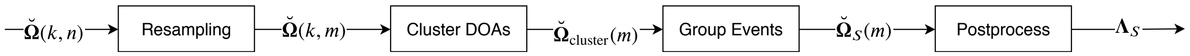

$\breve{\boldsymbol{\Omega}}(k,n)$
Resampling
$\breve{\boldsymbol{\Omega}}(k,m)$
Cluster DOAs
$\breve{\boldsymbol{\Omega}}_{\text{cluster}}(m)$
Group Events
$\breve{\boldsymbol{\Omega}}_S(m)$
Postprocess
$\boldsymbol{\Lambda}_S$

Figure 3: Association architecture.

The proposed CRNN is depicted in Figure 4. It presents three convolutional *blocks* to extract local features from the input representation. Each convolutional block consists of one convolutional layer, after which the resulting feature maps are passed through a ReLU non-linearity [17]. This is followed by a max-pooling operation to downsample the feature maps and add invariance along the frequency dimension. The target classes vary to a large extent in terms of their temporal dynamics, with some of them being rather impulsive (e.g., *Door slam*), while others being more stationary (e.g., *Phone ringing*). Therefore, after stacking the feature maps resulting from the convolutional blocks, this representation is fed into one bidirectional recurrent layer in order to model discriminative temporal structures. The recurrent layer is followed by a Fully Connected (FC) layer, and finally a 11-way softmax classifier layer produces the event-level probabilities. Dropout is applied extensively. The loss function used is categorical cross-entropy. The model has ∼175k weights.

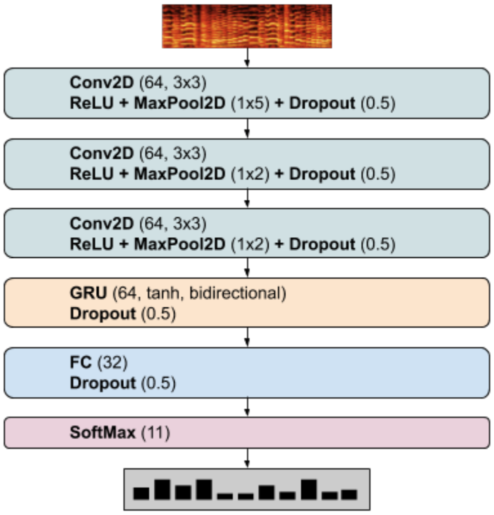

Conv2D (64, 3x3)
ReLU + MaxPool2D (1x5) + Dropout (0.5)
Conv2D (64, 3x3)
ReLU + MaxPool2D (1x2) + Dropout (0.5)
Conv2D (64, 3x3)
ReLU + MaxPool2D (1x2) + Dropout (0.5)
GRU (64, tanh, bidirectional)
Dropout (0.5)
FC (32)
Dropout (0.5)
SoftMax (11)

Figure 4: Back-end architecture.

3. EXPERIMENTS

3.1. Dataset, evaluation metrics and baseline system

We use the TAU Spatial Sound Events 2019 - Ambisonic, which provides first-order ambisonic recordings. Details about the recording format and dataset specifications can be found in [18]. The dataset features a vocabulary of 11 classes encompassing human sounds and sound events typically found in indoor office environments. The dataset is split into a development and evaluation sets. The development set consists of a four fold cross-validation setup.

The SELD task is evaluated with individual metrics for SED (F-score (*F*) and error rate (*ER*) calculated in one-second segments) and DOA estimation (DOA error (*DOA*) and frame recall (*FR*) calculated frame-wise) [6]. The *SELD score* is an averaged summary of the system performance.

The baseline system features a CRNN that jointly performs DOA and SED through multi-task learning [6]. Baseline results are shown in Table 2.

3.2. Parametric front-end

Based on the method's exploratory analysis, we propose the following set of parameter values, which are shown in Table 1.

Table 1: Parameter values for the selected configuration. Top: *DOA analysis* parameters. Bottom: *Association* parameters.

Parameter	Unit	Value
STFT window size	sample	256
analysis frequency range	Hz	[0,8000]
time average vicinity radius r	bin	10
diffuseness mask threshold Ψ_{max}	-	0.5
energy density filter length	bin	11
std mask vicinity radius	bin	2
std mask normalized threshold	-	0.15
median filter minimum ratio B_{min}	-	0.5
median filter vicinity radius (k,n)	bin	(20, 20)
resampling minimum valid bins K_{min}	bin	1
overlapping std threshold σ_{max}	degree	10
grouping maximum angle d_{max}^{ANGLE}	degree	20
grouping maximum distance d_{max}^{FRAME}	frame	20
event minimum length	frame	8

3.3. Deep learning classification back-end

We use the provided four fold cross-validation setup. Training and validation stages use the outcome of an *ideal* front-end, where the groundtruth DOA estimation and activation times are used to feed the beamformer for time-space segmentation. Conversely, we test the trained models with the signals coming from the *complete* front-end described in Section 2. We conducted a set of preliminary experiments with different types of networks including a VGG-like net, a less deep CNN [19], a Mobilenetv1 [20] and a CRNN [21]. The latter was found to stand out, and we explore certain facets of the CRNN architecture and the learning pipeline.

Sound events in the dataset last from ∼ 0.2 to 3.3 s. First, clips shorter than 2s are replicated to meet this length. Then, we compute TF patches of log-mel spectrograms of $T = 50$ frames (1 s) and $F = 64$ bands. The values come from the exploration of $T \in \{25, 50, 75, 100\}$ and $F \in \{40, 64, 96, 128\}$. $T = 50$ is the top performing value, roughly coinciding with the median event duration. In turn, more than 64 bands provide inconsistent improvements, at the cost of increasing the number of network weights.

Several variants of the CRNN architecture were explored until reaching the network of Figure 4. This included a small grid search over number of CNN filters, CNN filter size and shape, number of GRU units, number of FC units, dropout [22], learning rate, and the usage of Batch Normalization (BN) [23]. Network extensions (involving more weights) were considered only if providing major improvements, as a measure against overfitting. The main takeaways are: *i)* squared 3x3 filters provide better results than larger filters, *ii)* dropout of 0.5 is critical for overfitting mitigation, *iii)* more than one recurrent layer does not yield improvements, while slowing down training, and *iv)* surprisingly, slightly better performance is attained without BN nor pre-activation [24]. For all experiments, the batch size was 100 and Adam optimizer was used [25] with initial learning rate of 0.001, halved each time the validation accuracy plateaus for 5 epochs. Earlystopping was adopted with a patience of 15 epochs, monitoring validation accuracy. Prediction for every event was obtained by computing predictions at the patch level, and aggregating them with the geometric mean to produce a clip-level prediction.

Finally, we apply *mixup* [26] as data augmentation technique. Mixup consists in creating virtual training examples through linear interpolations in the feature space, assuming that they correspond to linear interpolations in the label space. Essentially, virtual TF patches are created on the fly as convex combinations of the input training patches, with a hyper-parameter α controlling the interpolation strength. Mixup has been proven successful for sound event classification, even in adverse conditions of corrupted labels [27]. It seems appropriate for this task since the front-end outcome can present leakage due to overlapping sources, effectively mixing two sources while only one training label is available, which can be understood as a form of label noise [19]. Experiments revealed that mixup with $\alpha = 0.1$ boosted testing accuracy in $\sim 1.5\%$.

4. RESULTS AND DISCUSSION

Table 2: Results for development (top) and evaluation (bottom) sets.

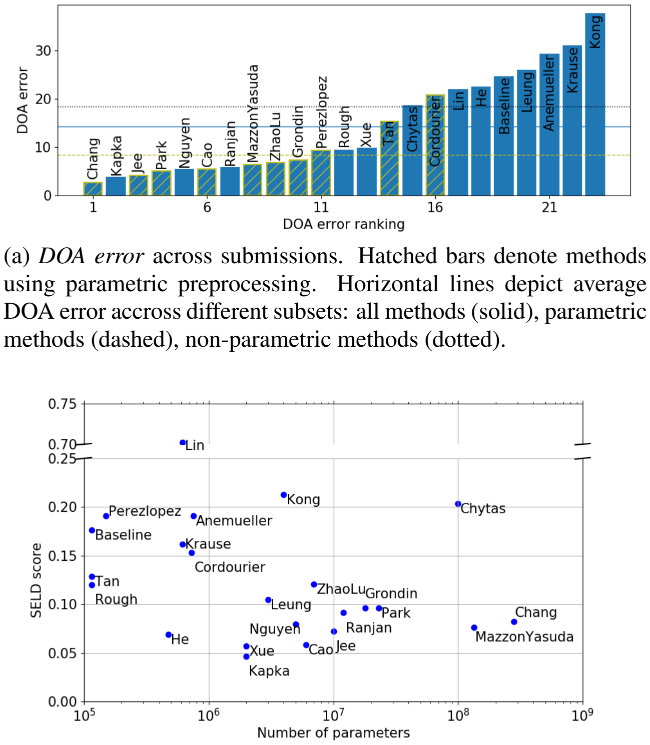
Method	*ER*	*F*	*DOA*	*FR*	*SELD*
Baseline	0.34	79.9%	28.5°	85.4%	0.2113
Proposed	0.32	79.7%	9.1°	76.4%	**0.2026**
Ideal front-end	0.08	93.2%	∼ 0°	∼ 100%	0.0379
Baseline	0.28	85.4%	24.6°	85.7%	0.1764
Proposed	0.29	82.1%	9.3°	75.8%	**0.1907**

Table 2 shows the results of the proposed method for both development and evaluation sets, compared to the baseline. Focusing on evaluation results, our method and the baseline obtain similar performance in SED (*ER* and *F*). However, there is a clear difference in the DOA metrics: in our method, *DOA* error is reduced by a factor of 2.6, but *FR* is $\sim$ 10 points worst. In terms of *SELD score*, our method performs slightly worse than the baseline in evaluation mode, while marginally outperforming it in development mode.

The most relevant observation is the great improvement in *DOA* error. Results suggest that using spatial audio parametric analysis as a preprocessing step can help to substantially improve localization. Figure 5a provides further evidence for this argument: Challenge methods using some kind of parametric preprocessing (*GCC-PHAT* with the microphone dataset, and *Intensity Vector-Based* in ambisonics) obtained in average better *DOA* error results.

Conversely, the front-end fails regarding *FR*. This is probably due to the complexity added by the association step [6], and its lack of robustness under highly reverberant scenarios. Including spectral information at the grouping stage might help to improve *FR* — such information could be provided by the classification back-end, in a similar approach to the baseline system. Another option would be the usage of more sophisticated source counting methods [28, 29].

In order to gain a better insight of the classification back-end performance, Table 2 shows the method results when the testing clips are obtained by feeding the beamformer with groundtruth annotations (*ideal* front-end). In this ideal scenario of DOA performance, the SED metrics show a significant boost. This result suggests that the low *FR* given by the front-end has a severe impact on the back-end performance. Yet, the proposed system reaches similar performance to the baseline system in terms of SED metrics.



(a) *DOA error* across submissions. Hatched bars denote methods using parametric preprocessing. Horizontal lines depict average DOA error accross different subsets: all methods (solid), parametric methods (dashed), non-parametric methods (dotted).

(b) *SELD score* versus complexity.

Figure 5: DCASE2019 Challenge Task 3 results, evaluation set.

Finally, we would like to discuss algorithm complexity among Challenge methods. As depicted in Figure 5b, there is a general trend towards architectures with very high number of weights, as a consequence of the usage of ensembles and large capacity networks. Specifically, 66% of submitted methods employ 1M weights or more, 30% employ 10M or more, and 15% employ 100M or more. Such complexities are several orders of magnitude greater than the baseline (150k weights) or the proposed method ($\sim$175k weights). In this context, our method represents a low-complexity solution to the SELD problem, featuring a number of parameters and a performance comparable to the baseline method.

5. CONCLUSION

We present a novel approach for the SELD task. Our method relies on spatial parametric analysis for the computation of event DOAs, onsets and offsets. This information is used to filter the input signals in time and space, and the resulting event estimations are fed into a CRNN which predicts the class to which the events belong; the classification problem is thereby handled from a simple multiclass perspective. The proposed method is able to obtain an overall performance comparable to the baseline system. The localization accuracy achieved by our method greatly improves the baseline performance, suggesting that spatial parametric analysis might enhance performance of SELD algorithms. Moreover, detection and classification performance in our method suffers from a low Frame Recall; improving this metric could lead to promising SELD scores.

6. REFERENCES

[1] C. Grobler, C. P. Kruger, B. J. Silva, and G. P. Hancke, "Sound based localization and identification in industrial environments," in *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2017, pp. 6119–6124.

[2] T. Butko, F. G. Pla, C. Segura, C. Nadeu, and J. Hernando, "Two-source acoustic event detection and localization: Online implementation in a smart-room," in *2011 19th European Signal Processing Conference*. IEEE, 2011, pp. 1317–1321.

[3] R. Chakraborty and C. Nadeu, "Sound-model-based acoustic source localization using distributed microphone arrays," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 619–623.

[4] T. Hirvonen, "Classification of spatial audio location and content using convolutional neural networks," in *Audio Engineering Society Convention 138*. Audio Engineering Society, 2015.

[5] K. Lopatka, J. Kotus, and A. Czyzewski, "Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations," *Multimedia Tools and Applications*, vol. 75, no. 17, pp. 10 407–10 439, 2016.

[6] 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*, pp. 1–1, 2018.

[7] M. A. Gerzon, "Periphony: With-height sound reproduction," *Journal of the Audio Engineering Society*, vol. 21, no. 1, pp. 2–10, 1973.

[8] J. Daniel, "Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia," 2000.

[9] V. Pulkki, "Directional audio coding in spatial sound reproduction and stereo upmixing," in *Audio Engineering Society Conference: 28th International Conference: The Future of Audio Technology–Surround and Beyond*. Audio Engineering Society, 2006.

[10] S. Berge and N. Barrett, "High angular resolution planewave expansion," in *Proc. of the 2nd International Symposium on Ambisonics and Spherical Acoustics May*, 2010, pp. 6–7.

[11] A. Politis, S. Tervo, and V. Pulkki, "COMPASS: Coding and Multidirectional Parameterization of Ambisonic Sound Scenes," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, no. May, pp. 6802–6806, 2018.

[12] V. Pulkki, S. Delikaris-Manias, and A. Politis, *Parametric time-frequency domain spatial audio*. Wiley Online Library, 2018.

[13] https://github.com/andresperezlopez/DCASE2019_task3.

[14] T. Carpentier, "Normalization schemes in ambisonic: does it matter?" in *Audio Engineering Society Convention 142*. Audio Engineering Society, 2017.

[15] D. Stanzial, N. Prodi, and G. Schiffrer, "Reactive acoustic intensity for general fields and energy polarization," *The Journal of the Acoustical Society of America*, vol. 99, no. 4, pp. 1868–1876, 1996.

[16] J. Merimaa and V. Pulkki, "Spatial impulse response rendering i: Analysis and synthesis," *Journal of the Audio Engineering Society*, vol. 53, no. 12, pp. 1115–1127, 2005.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

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

[19] E. Fonseca, M. Plakal, D. P. W. Ellis, F. Font, X. Favory, and X. Serra, "Learning sound event classifiers from web audio with noisy labels," in *Proc. IEEE ICASSP 2019*, Brighton, UK, 2019.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.

[21] E. Cakır, 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.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

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

[24] E. Fonseca, R. Gong, and X. Serra, "A simple fusion of deep and shallow learning for acoustic scene classification," in *Proceedings of the 15th Sound & Music Computing Conference (SMC 2018)*, Limassol, Cyprus, 2018.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR 2015*. [Online]. Available: https://arxiv.org/abs/1412.6980

[26] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[27] E. Fonseca, F. Font, and X. Serra, "Model-agnostic approaches to handling noisy labels when training sound event classifiers," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, US, 2019.

[28] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the number of clusters in n-way probabilistic clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2006–2021, 2010.

[29] N. Stefanakis, D. Pavlidi, and A. Mouchtaris, "Perpendicular Cross-Spectra Fusion for Sound Source Localization with a Planar Microphone Array," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 25, no. 9, pp. 1517–1531, 2017.