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

Technical Report

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 technical report describes and discusses the algorithm submitted to the *Sound Event Localization and Detection* Task of DCASE2019 Challenge. The proposed methodology combines a parametric spatial audio analysis approach for localization estimation, a simple heuristic for event segmentation, and a deep learning based monophonic event classifier. The evaluation of the proposed algorithm with the development dataset yields overall results slightly outperforming the baseline system. The main highlight is a reduction of the localization error over 65%.

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, its temporal activity, spatial location, and sound class to which it belongs. SELD is an ongoing problem 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 Estimation (DOA). The number of works in the literature jointly addressing SED and DOA problems is relatively small. We can classify them by the type of microphone array used: distributed [1, 2, 3] and 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 among methods. Furthermore, the number of 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 have been proposed for parametric analysis in the Ambisonic domain [9, 10, 11, 12]. These methods are capable of segmentating a sound field into direct and diffuse components, and further estimating the 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 DNN-based DOA estimation methods explicitly exploits the Ambisonic parametric analysis. This situation is further extended to the SELD problem, with the exception of [5], where DOAs are calculated from the *active intensity vector*.

The motivation for the proposed methodology is two-fold. First, we want to check whether the usage of spatial parametric analysis helps in the event classification task. Second, since the parametric analysis is usually performed in the time-frequency (TF) domain, temporal information could be further exploited to derive sound event onsets and offsets, thus lightening the complexity of the event classifier.

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 comparison with a baseline implementation (Section 4). A summary is finally presented in Section 5. In order to support open access and experiment reproducibility, all code produced in this research is freely available at [13].

2. METHODS

The method presented proposes a solution for the SELD problem splitting the task into four different problems: *DOA estimation*, *association*, *beamforming* and *classification*, which are described in the following subsections. The former three systems follow an model-based analytical approach—in what follows, they will be jointly referred to as the *parametric frontend*. Conversely, the *classification* system is data-driven, and will be referred to as the *deep learning backend*. The method architecture is depicted in Figure 1.

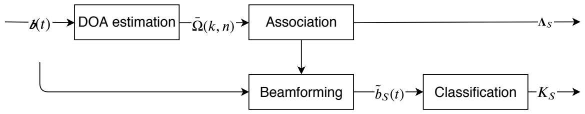

Figure 1: System architecture.

2.1. DOA estimation

The *DOA estimation* system computes the DOAs of the signals most *significant* TF bins, based on a parametric spatial audio analysis of the input signal. A general overview of the system can be found in Figure 2. In a more formal description, given a first order ambisonic time-domain signal $\boldsymbol{b}(t)$[1] with $L = 4$ channels:

$$\boldsymbol{b}(t) = [b_x(t), \sqrt{3}b_y(t), \sqrt{3}b_z(t), \sqrt{3}b_x(t)]^\intercal, \tag{1}$$

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

[1] Considering ACN channel ordering and N3D normalization.

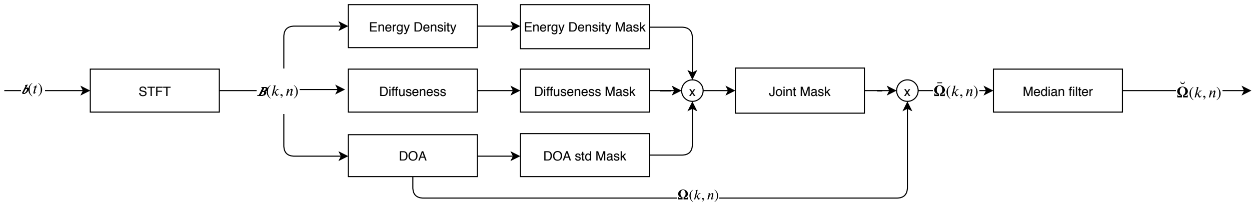

Figure 2: DOA estimation architecture.

First, the input signal is transformed into the short-time frequency domain signal $\boldsymbol{B}(k,n)$ using the Short-Time Fourier Transform (STFT). It is then possible to obtain an estimate of the instantaneous predominant DOA (in azimuth and elevation) at each bin, $\boldsymbol{\Omega}(k,n) = [\varphi(k,n), \theta(k,n)]$:

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

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.

Next, we would like to filter $\boldsymbol{\Omega}(k,n)$ in order to only include information from the bins where a sound event is present. For that goal we compute three different binary masks.

The first one is the *energy density mask*, which is used as an activity detector. The energy density $E(k,n)$ [9] is an estimator of the total energy of a given TF bin:

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

being c the speed of sound. A gaussian adaptive thresholding algorithm is then applied to $E(k,n)$, in order to avoid suppressing events with lower energy, which might occur with a fixed threshold value. The energy density mask is thus defined as the bins with an energy density level higher than the local threshold.

The *diffuseness mask* selects the TF bins in which the transmitted energy is high, which is the case of the sources' direct path. Diffuseness $\Psi(k,n)$ is defined in [9] as:

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

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

The third mask is the *DOA variance mask*. It tries to select the frequency bins with small standard deviation[2] with respect to their neighbor bins. TF regions with a significant contribution of one direct source will have small values, while the standard deviation for isotropic diffuse fields will be maximum [12].

The three masks are then applied to the DOA estimation, obtaining the TF-filtered DOAs $\bar{\boldsymbol{\Omega}}(k,n)$. As a last step, a median filter is applied to $\bar{\boldsymbol{\Omega}}(k,n)$, which helps to remove spurious TF bins, and to improve the consistency of the DOA estimation. The median is computed for a given bin only if the ratio of valid bins in its vicinity is greater than a given threshold B_{min}. The final DOA estimation result is referred to as $\breve{\boldsymbol{\Omega}}(k,n)$.

2.2. Association

The association step (depicted in Figure 3) tackles the problem of assigning the time-frequency-space observation $\breve{\boldsymbol{\Omega}}(k,n)$, to a set of sound events, each one having a specific onset time, offset time and angular position. First, the DOA estimates are regrouped into the required hop size $h = 0.02$ seconds, with the restriction of only considering the DOA estimates for a given analysis window if its number is greater than a minimum K_{min}. In what follows, each segment of length h will be called a *frame* and represented by m.

Next, a space-frequency clustering procedure is applied. For a given frame M, the standard deviation σ of all DOA estimates $\breve{\boldsymbol{\Omega}}(k,M)$ is computed, and the result is used to determine the overlapping amount $o(M)$: 1, if $\sigma_\varphi/2 + \sigma_\theta < \sigma_{max}$ (an angular standard deviation threshold), or 2 otherwise. If $o(M) = 1$, then the clustered DOA value for that frame, $\boldsymbol{\Omega}_{\text{cluster}}(M)$, is assigned to the median of all DOA estimates. Conversely, when $o(M) = 2$, the DOAs are clustered using a modified version of K-Means which minimizes the central angle instead of the euclidean distance. The value of $\boldsymbol{\Omega}_{cluster}(M)$ in that case is assigned to the tuple given by the K-Means centroids.

The following step is the grouping of clustered DOA values into events. Let's 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}.

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

2.3. Beamforming

Once the event annotations are ready, we can use them to segmentate the input signal in time and space domains. By doing so, it is possible to extract a monophonic signal estimation of each event, $b_S(t)$, as the signal captured by a virtual first-order microphone:

$$\tilde{b}_S(t) = \sum^{L} \boldsymbol{\alpha}\boldsymbol{Y}(\boldsymbol{\Omega}_S)\boldsymbol{b}(t), \tag{5}$$

where $\boldsymbol{Y}(\boldsymbol{\Omega}_S)$ is the set of real-valued spherical harmonics up to order L evaluated at the position $\boldsymbol{\Omega}_S$, and $\boldsymbol{\alpha}$ defines the virtual microphone directivity. In this work we have chosen a hypercardioid pattern, $\boldsymbol{\alpha} = [1,1,1,1]^\intercal$.

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

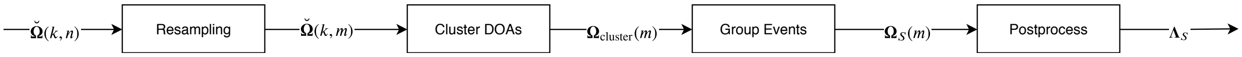
$\breve{\mathbf{\Omega}}(k,n)$
Resampling
$\breve{\mathbf{\Omega}}(k,m)$
Cluster DOAs
$\mathbf{\Omega}_{\text{cluster}}(m)$
Group Events
$\mathbf{\Omega}_S(m)$
Postprocess
$\mathbf{\Lambda}_S$

Figure 3: Association architecture.

2.4. Deep Learning Classification Backend

The parametric frontend performs DOA estimation, temporal detection and time/space segmentation, the output being a monophonic signal containing, in theory, one single event. The goal of the backend is to classify this incoming signal as belonging to one of a target set of 11 sound categories. Therefore, the multi-task nature of the parametric frontend allows us to define the backend classification task as a simple multi-class problem (even though the original SELD task is a multi-label one). It must be noted, however, that due to the limited spatial directivity of the first-order beamformer, the resulting monophonic signal can present a certain degree of leakage from additional sound sources when two events overlap, even when the annotation $\mathbf{\Lambda}_S$ is perfectly estimated.

The proposed classification method is divided into two stages. First, the incoming signal is transformed into log-mel spectrogram, which is split into TF patches of $\mathbb{R}^{T\times F}$ (see Sec 3.3). For the second stage, we decided to use a single model based on a Convolutional Recurrent Neural Network (CRNN), fed by the TF patches, and outputting 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 onset/offset times have been determined by the frontend.

The proposed CRNN architecture is depicted in Fig 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 [14]. 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, 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 to 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 probability predictions. Dropout is applied after the max-pooling operations in the convolutional blocks, and also after the recurrent layer and the FC layer. Given the formulated multi-class problem, the loss function used for training is categorical cross-entropy. The model has ∼175k weights.

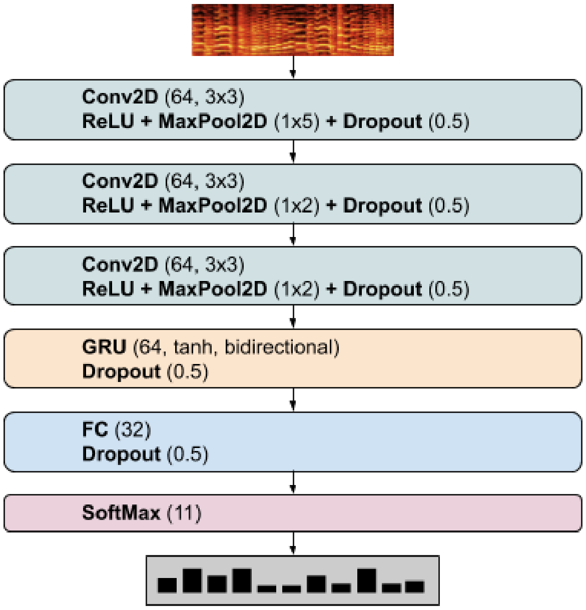
Conv2D (64, 3x3)
ReLU + MaxPool2D (1x5) + Dropout (0.5)
Conv2D (64, 3x3)
ReLU + MaxPool2D (1x2) + Dropout (0.5)
Conv2D (64, 3x3)
ReLU + MaxPool2D (1x2) + Dropout (0.5)
GRU (64, tanh, bidirectional)
Dropout (0.5)
FC (32)
Dropout (0.5)
SoftMax (11)

Figure 4: Backend model 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 (FOA) recordings[3]. Details about the recording format and dataset specifications can be found in [15]. The dataset features a vocabulary of 11 classes encompassing human sounds and sound events typically occurring in indoor office environments. The dataset is split into a development and evaluation sets. The development set consists of a pre-defined four fold cross-validation setup. The SELD task is evaluated with individual metrics for SED and DOA estimation. SED is evaluated with F-score (F) and error rate (ER) calculated in one-second segments, while DOA estimation is evaluated with two frame-wise metrics: DOA error (DOA) and frame recall (FR) [15]. 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 Frontend

Based on the method's exploratory analysis, we propose the following set of parameter values, which are shown in Table 1. In general, the selected values follow a *permissive* approach: most of parameters have relatively low values (e.g. Ψ_{max}, K_{min}, σ_{max}). The only exception is the median filter, which features a very large window size, and is responsible for TF filtering to a great extent.

3.3. Deep Learning Classification Backend

We use the provided four fold cross-validation setup. We train and validate using the outcome of an *ideal* frontend, where the ground truth DOA estimation and onset/offset times are used as inputs to the beamformer for time/space segmentation. Conversely, we test the trained models with the signals coming from the *complete* frontend described in Section 2. We conduct a set of preliminary experiments with different types of networks including a VGG-like net, a less deeper CNN [16], a Mobilenet v1 [17] and a CRNN [18]. The latter is found to stand out, and we explore certain facets of the CRNN architecture and the learning pipeline. A number of decisions are taken to mitigate the risk of overfitting given data scarcity.

Sound events in the dataset last from ∼ 0.2 to 3.3s. First, clips shorter than 2s are replicated to this length. Then, we compute TF patches of log-mel spectrograms of 1s (equivalent to $T = 50$ frames) and $F = 64$ bands. This is the result of exploration of $T \in \{25, 50, 75, 100\}$ and $F \in \{40, 64, 96, 128\}$. $T = 50$ is the

[3] Compatibility with the *Microphone dataset* could be straightforwardly accomplished with the usage of proper filters.

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

Parameter	Unit	Value
sampling rate	Hz	48000
STFT window size	sample	256
STFT window overlap	sample	128
STFT window type	-	Hann
minimum STFT frequency	Hz	0
maximum STFT frequency	Hz	15000
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^{ANGLE}_{max}	degree	20
grouping maximum distance d^{FRAME}_{max}	frame	20
event minimum length	frame	8

top performing value, roughly coinciding with the median duration of events in the dataset; more than 64 bands provide inconsistent improvements while increasing the number of network weights.

Several variations of the CRNN architecture are explored until reaching the network of Fig 4. This includes a small grid search over number of CNN filters, CNN filter size and shape, number of GRU units, number of FC units, dropout [19], learning rate, and whether or not to use Batch Normalization (BN) [20]. Network extensions (involving more weights) are 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 in mitigating overfitting, *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 [21]. For all experiments, the batch size is 100 and Adam optimizer is used [22] with initial learning rate of 0.001, halved each time the validation accuracy plateaus for 5 epochs. Earlystopping is adopted with a patience of 15 epochs monitoring validation accuracy. Prediction for every event is obtained by computing predictions at the patch level, and aggregating them with geometric mean to produce a clip-level prediction.

Finally, we apply *mixup* [23] 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 [24]. It seems appropriate for this task since the frontend 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 [16]. Experiments revealed that mixup with $\alpha = 0.1$ boosted testing accuracy in $\sim 1.5\%$.

4. RESULTS AND DISCUSSION

Table 2 shows the evaluation results of the proposed method for the development dataset, compared with the baseline. The proposed method and the baseline obtain very similar results in SED (*ER* and *F*). However, there is a clear difference in the DOA metrics. The *DOA error* in the proposed method is reduced by a factor over 3, which represents an improvement of 68% with respect to the baseline. *FR*, by contrast, is 9 points worst in the proposed method. In summary, the *SELD* score for both methods is very similar, with the proposed method slightly outperforming the baseline.

The most significant conclusion is the great improvement in *DOA error*. The result suggests that using spatial audio parametric analysis as a preprocessing step can substantially improve the localization error. This fact may be explained taking into account the great compression performed to the input spectrograms, which eliminates most of non-relevant TF information.

Conversely, the frontend fails with respect to *FR*. This is probably due to the added complexity and the misbehaviour of the *association* algorithm [6]. One of the main problems is the lack of robustness against reverberation: strong early reflections might be incorrectly characterised as individual sources, thus potentially leading to underestimation of existing overlapping sources. The inclusion of spectral information might help to disambiguate in this case—such information could be provided in parallel by the *classifier*, in a similar approach to the baseline system. Another possibility includes more sophisticated source counting methods [11, 25].

To better analyze the performance of the classification backend, Table 2 shows the results when the testing clips are obtained using the groundtruth DOAs and onset/offset times as inputs to the beamformer (*ideal* frontend). In this ideal scenario of DOA performance, the classification metrics show a significant boost. This suggests that the low *FR* given by the frontend (*i.e.*, events being partially or totally missed) has a severe impact on the backend classification performance. Yet, the proposed system reaches similar performance to the baseline system in terms of SED metrics.

Table 2: Evaluation results on development set.

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 frontend	0.08	93.2%	$\sim 0°$	$\sim 100\%$	0.0379

5. CONCLUSION

We have presented a novel approach for the SELD task. Our method performs spatial parametric analysis and a simple association methodology to estimate the DOAs, onsets and offsets of the sound events. This information is used to filter the input signal in time and space, and the resulting monophonic event estimation is fed into a CRNN which predicts the class to which the event belongs. In this way, the classification problem is handled from a simple multi-class perspective. The proposed methodology is able to obtain slightly better overall results than the baseline system. The localization performance is greatly improved while the detection and classification performance suffers from the loss of Frame Recall. The improvement of this metric in the proposed system could lead to promising SELD scores, suggesting that signal preprocessing through spatial parametric analysis might improve the classification task.

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] 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.

[15] S. Adavanne, A. Politis, and T. Virtanen, "A multiroom 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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

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

[24] E. Fonseca, F. Font, and X. Serra, "Model-agnostic approaches to handling noisy labels when training sound event classifiers," in *Submitted to WASPAA 2019*, New York, US, 2019.

[25] 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.