CURRICULUM LEARNING WITH AUDIO DOMAIN DATA AUGMENTATION FOR SOUND EVENT LOCALIZATION AND DETECTION

Technical Report

Ricardo Falcon-Perez

Aalto University
Department of Signal Processing and Acoustics
Espoo, Finland
ricardo.falconperez@aalto.fi

ABSTRACT

In this report we explore a variety of data augmentation techniques in audio domain, along with a curriculum learning approach, for sound event localizaiton and detection (SELD) tasks. We focus our work on two areas: 1) techniques that modify timbral of temporal characteristics of all channels simultaneously, such as equalization or added noise; 2) methods that transform the spatial impression of the full sound scene, such as directional loudness modifications. We test the approach on models using either time-frequency or raw audio features, trained and evaluated on the STARSS22: Sony-TAU Realistic Spatial Soundscapes 2022 dataset. Although the proposed system struggles to beat the official benchmark system, the augmentation techniques show improvements over our non-augmented baseline.

Index Terms— sound event localization and detection, data augmentation, audio signal processing

1. INTRODUCTION

Soudn event localization and detection (SELD) is a dual task where the goal is to detect, classify and localize distinct sound events happening in a 3D spatial recording or a sound scene. SELD can be understood as an extension to the Sound Event Detection (SED) task with the addition of the localization problem. In addition, This tasks is related to other fields such as image segmentation [1] or object tracking [2], where the objective is also to detect and localize objects in a scene, although the modality of the input data are images or video instead of audio.

One of the main issues with SELD tasks, is the limited amount of labeled data available, as well as the high density of data. Moreover, there is usually a distribution shift across datasets due to the acoustical conditions of the recordings. Usually, simulated datasets present large diversity of acoustical environments at the cost of diversity and naturalness of the sound events. On the other hand, real-life recordings are often smaller and recorded in a handful of rooms, limiting the variety of acoustical properties, but having much richer and natural events.

Therefore, data augmentation has been very popular and crucial for most systems solving SELD tasks. Typically, most systems based on time-frequency representations borrow data augmentation techniques from the computer vision domain, including methods such as spec augment [3], random cropping, or scaling. More recently, data augmentation techniques that work directly on audio domain have been applied [4, 5]. However, these audio domain augmentations are usually just casually mentioned, with little details about their hyperparameters. There is a vast literature on audio signal processing effects [6], and the configuration of each effect can have a dramatic impact of the perception of the sound.

On the other hand, the process of applying data augmentation itself is also understudied. The authors of [7] found that data augmentation works best when the techniques transforms the data strongly, while preserving the task-specific semantic meaning close to the original data. More recently, [8] presented a general framework to improve the results and stability of SED systems, mainly by structured training approach that combines models pretrained in large datasets as initialization, balanced sampling with augmentation, and the use of ensemble networks.

In this work we present a framework for training SELD systems using data augmentation in audio domain, and a curriculum learning procedure to control the amount of augmentation. This approach is independent of the feature representation used, therefore it is applicable to both time domain models such as SampleCNN , [9] or custom 1D CNNs [10], as well as models based on time-frequency input features, including spectrograms or learned 2D representations. Furthermore, our proposed technique could be applied to systems using audio in both FOA or Mic formats, for most cases [1]. However, due to time constraints we only present results from FOA format in this paper.

We also provide code for the implementation of our full system, including the custom data loading, the models, and the augmentation techniques [2]

2. METHOD

The overall process is described in figure 1. For each training iteration, first a minibatch of raw audio signals is augmented by uniformly randomly selecting M out of N possible augmentation methods (from a predetermined set of augmentations). Each of these augmentation techniques has its own parameters (e.g. frequency for a low pass filter) that are randomly selected at each iteration from fixed set of hyperparameters (e.g. uniform distribution

[1] Some data augmentation techniques used here are exclusive to FOA format (e.g. Spatial Mixup), while the effectiveness of others could vary significantly depending on the input format.

[2] Code available at: `https://github.com/rfalcon100/seld_dcase2022_ric`

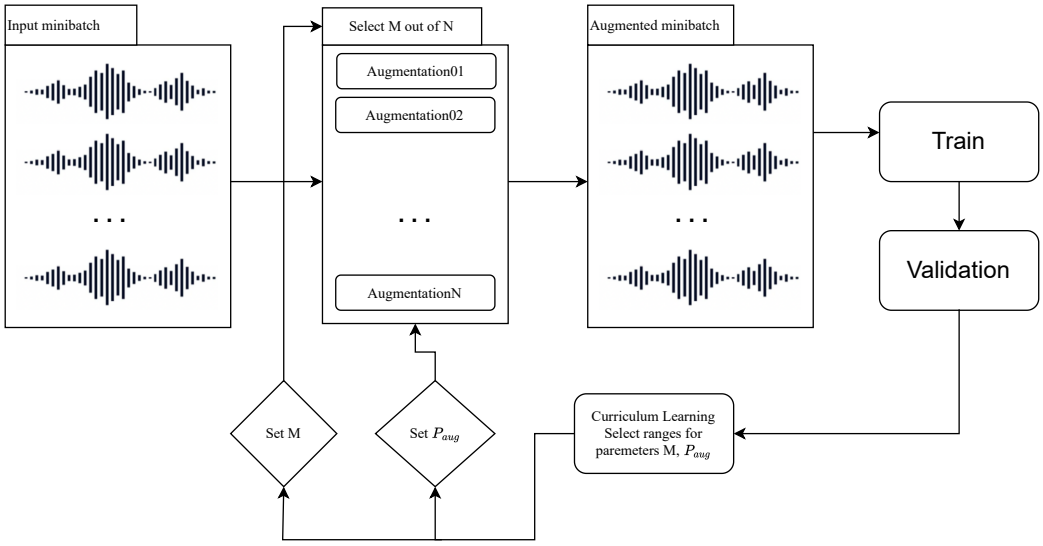

Figure 1: Our proposed approach on how to apply data augmentation and curriculum learning. For details refer to section 2

covering 1000-5000 Hz for the same low pass filter). The augmentations are applied sequentially in a pre-determined order, and with a different selection of parameters for each example in the minibatch (per example, to all channels), but with the same selection of augmentation methods. Next, after applying the augmentation to the minibatch, the training iteration continues as usual (forward and backward passes). Then, a curriculum learning rule updates the maximum number of augmentations M, as well as the parameters for each augmentation, based on some metric obtained during the validation step.

2.1. Data Augmentation

The main goal of this work was to explore data augmentation methods on audio domain. For this purpose, we analyzed a few common audio signal processing methods that transform the audio signal while preserving the overall semantic content intact. The main motivation is that changes applied to the overall spectral and timbral features, should not impact neither the detection or localization of sound events. However, this assumption is not true if the transformation is too extreme. For example, a low pass filter where the cutoff frequency is set too low, will remove all sound in the middle and high frequencies, making detection impossible for events that have little low frequency content. Similarly, a hard spatial filter that heavily suppresses large areas of the soundfield will make localization virtually impossible for sounds arriving from the suppressed regions. Therefore selecting both suitable transformations and their hyperparameters parameters is important.

The data augmentation techniques used, and the initial hyperparameter sets are:

- **Spatial Mixup (Directional Loudness)** Spatial Mixup is a data augmentation technique that transforms the spatial characteristics of audio signals encoded in spherical harmonics domain. More precisely, the method applies selective directional gains to emphasize or suppress the signal in certain directions. However, unlike a traditional beamformer, the suppression is subtle and similar to soft spatial filtering, in the order of only a few dB. For this work, we use the same hyperparameter set as in [11], with the soft spherical caps. For computational efficiency [3], this transformation is applied to the whole minibatch with the same settings.
- **Gain**: Apply gain to the input signal to change the overall loudness. We uniform randomly select gains from [-15, 6] dB.
- **Polarity Inversion**: Random inverse the polarity of all channels. Applied to either none, or all channels together.
- **Pitch Shift**: Pitch shifting by time stretching complex spectrogram of the input signal and then applying the inverse FFT. Therefore the audio pitch is changed but its length stays the same. We limit the range of pitch shifting to $\pm$ 2 semitones only. It should be noted that in some cases, pitch shifting can significantly modify the spatial characteristics of the signal, does affecting DOA estimations.
- **Colored Noise**: Adding noise to the signal helps the networks learn how to ignore it. Here we add colored noise, by filtering white noise. This means that the frequency content of the noise is randomly selected. The SNR of the added noise is uniform randomly selected from [30, 2] dB. Therefore, in cases of low SNR, the noise is very prominent.
- **Random Equalization**: Equalization changes the frequency content of a signal, by either suppressing of enhancing certain frequencies. This can be very useful in circumstances where the signal contain unwanted noise that do not contribute to the content, for example, low frequency rumble. The motivation here is to not only remove undesired components, but to teach the networks to extract meaningful characteristics of the sound events even when the frequency content is very different. In particular, we apply a 4 band parametric filter based on windowed sinc filters with linear phase. During training, we uniform randomly selected up to 4 filters to apply to the input signal. The filters can be either low pass (LP), high pass (LP),

[3] This is a limitation in our implementation that requires computing the spherical harmonics in CPU instead of GPU. Nevertheless, this could be improved by moving this computation directly to the GPU.

band stop (BS), or band pass (BP). In some cases, the equalization can be quite extreme where the output sounds significantly different to the input signal, making it difficult even for human listeners to classify the events. The parameters of each filter are also randomly selected from:

1. **LP** : min-freq = 1000 Hz, max-freq = 5000 Hz
2. **HP** : min-freq = 250 Hz, max-freq = 1500 Hz
3. **BP** : min-freq = 400 Hz, max-freq = 4000 Hz, Q = [0.5, 1.5]
4. **BS** : min-freq = 400 Hz, max-freq = 4000 Hz, Q = [0.5, 1.5]

- **Limiter** In some cases, the resulting augmented signal can have amplitudes that go beyond the [-1, 1] range. Instead of normalizing the signal to constrain to this range, we apply a simple brickwall limiter, where all values that exceed a certain threshold are set to either 1 or -1. In some cases this can introduce clipping and distortion, which might actually be useful as data augmentation. We set the threshold to $\pm$ 0.99 .

2.2. Curriculum Learning

In general, curriculum learning refers to the process of increasing the difficulty of the learning tasks as the training progresses. The motivation is that by first focusing on simple problems, the optimization of the networks is easier and smoother. By gradually introducing more difficult examples, the optimization process remains stable. This in practice helps with generalization as it is less likely that the model will learn spurious correlation from the data, that are not representative of the true task [12]. Curriculum learning is a large field with many variants, but it is commonly applied in two ways: 1) by increasing the difficulty of the training data as training progresses; or 2) by increasing the model complexity. In this work we focus on the former.

That said, for curriculum learning to work properly, there needs to be an effective mechanism to rank the difficulty of the training data, and an efficient way to sample accordingly. This is not a trivial problem, and some alternatives have been suggested in the literate. For audio tasks, one could consider for example analyzing the signal-to-noise ratio of the input audio, or using a surrogate, basic model as a classifier for example difficulty.

In this work we instead use data augmentation as the mechanism to control and rank data difficulty. The main idea is that the original, non-augmented data is considered a clean, while increasingly augmenting (and transforming) the data increases the difficulty. In this sense, the amount and strength of the audio signal processing methods mentioned in section 2.1 are increased during training.

The update schedule and criteria for the curriculum learning is also important. In this work we use a simple fixed schedule, were the curriculum is increased during each validation step. This is explained in section 3.2. O

3. EXPERIMENTS

3.1. Models

We evaluate the proposed approach using the official baseline provided in the DCASE 2022 challenge as reference, and two different models using either raw audio or time-frequency input representations:

1. **CRNN10** We use a CRNN based on [13], a time-frequency model composed of blocks of 2D convolutions with batch normalization as feature extraction, and a bidirectional GRU module for temporal processing. The input features are complex STFT (computed using a frame size of 1024, hop size of 240 samples, for a total input length of 2.55 seconds) of the FOA input signals, concatenated with the intensity vector for total input size of 7 channels. The total number of parameters is around 4.7 million.
2. **SSELDnet** We use the model from [9]. Based on a SampleCNN [14] this consists of squeeze-and-excitation residual blocks of 1D convolutional layers, followed by a Conformer [15] module that applies temporal attention. The inputs are 6 seconds long frames of audio at the original sampling frequency of 24000 Hz, using only the FOA signals. The total number of parameters is around 2.1 million .

3.2. Training

During training, for each iteration we first build a minibatch of frames of raw audio of the selected length depending on the model, and their corresponding labels. Each frame is a a random slice of different audio recordings (wav files), so that no recording appears twice on the same batch. Then each batch is augmented following the procedure described in section 2. We set M to 0 at the beginning of training, and increase this value up to 5 (out of 9 possible augmentations) by 1 during each validation step. All augmentations are performed online, on GPU, in a non-differentiable manner. This means that augmentation does not propagate gradients. We use the implementations provided by [16] or [11], with a few custom modifications and extensions.

We train all models for 200,000 iterations of batch size 32, with validation every 10,000 steps. The training objective is the MSE of the ACCDOA vectors (class-wise) [17], and we use an ADAM optimizer with a learning starting at $1e-5$ with a warmup stage of 5 validation steps up to $1e-3$, followed by a reduce on plateau scheduler, with decay rate of 0.9 and patience of 3 steps.

We follow the data split of the development set proposed in [18], using both simulated and recorded data for training, and only the test split of the development set for validation. We report the same evaluation metrics as the official DCASE 2022 challenge, of the test split for the best validation step of all models based on the $\mathcal{E}_{\text{SELD}} \downarrow$, defined as the normalized mean of all other metrics.

3.3. Results

Table 1 shows the results on the test split of the evaluation dataset. First, compared to the official baseline, our results are not that good. However, it should be noted that the official baseline utilizes multi track ACCDOA with ADPIT training, which accommodates for multiple overlapping sources at the same time, while all our experiments utilize the simple ACCDOA targets. Therefore, a large drop of performance is expected, because we ignore polyphony. Secondly, both our tested models show moderate yet consistent improvements over all metrics, when using the proposed data augmentation and training scheme. This effects seems to be stronger in both localization metrics. Finally, the time-frequency model outperformed the time domain model, despite having a shorter receptive field. Although the former has twice as many parameters.

The results presented here are limited and more experiments are needed to show the true value of the approach, and how it behaves

Table 1: Performance of the proposed approach. Note: the baseline uses a different loss function.

System	$ER_{LD} \downarrow$	$F_{LD} \uparrow$	$LE_{CD} \downarrow$	$LR_{CD} \uparrow$	$\mathcal{E}_{SELD} \downarrow$
Baseline ([19])	**0.710**	21.0	29.3	**46.0**	**0.458**
CRNN10	0.78	15.0	40.4	26.0	0.686
CRNN10 w/aug	0.74	**23.0**	**27.4**	45.0	0.553
SSELDnet	0.8	13.0	61.1	25.0	0.690
SSELDnet w/aug	0.750	19.0	49.3	38.8	0.555

with different types of models.

4. CONCLUSIONS AND DISCUSSION

We propose a simple framework to train neural-network-based systems for sound event localization and detection tasks. The framework consists of on-the-fly data augmentation applied to raw audio signals, with a curriculum learning schedule that increases the amount of augmentation over time. In addition, we present a set of hyperparemeters for the augmentation techniques used, that aim to increase robustness and generalization of the network predictions. The augmentation scheme proposed is applied during training, without the need to preprocess features, at a small performance cost, but fast enough because it runs on GPU. Although our submission to the DCASE2022 task 3 struggles to beat the baseline, we find that data augmentation improves our own implementation with no augmentation.

The data augmentation techniques presented are quite sensitive to hyperpameters, and further work is needed to find good sets of values. In addition, there is a large number of raw audio data augmentation methods not considered in this paper. Among these, some of the most promising could be dynamic range compression or expansion, as well as non-linear processing like virtual analog emulations. Finally, the curriculum learning schedule presented here could be explored in more detail, to find a more robust update rule for the augmentation techniques.

5. REFERENCES

[1] S. P. Mary, Ankayarkanni, U. Nandini, Sathyabama, and S. Aravindhan, "A survey on image segmentation using deep learning," *Journal of Physics: Conference Series*, vol. 1712, no. 1, p. 012016, dec 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1712/1/012016

[2] F. Porikli and A. Yilmaz, *Object Detection and Tracking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 3–41. [Online]. Available: https://doi.org/10.1007/978-3-642-28598-1_1

[3] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. of INTERSPEECH*, 2019.

[4] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks." in *Proc. of ISMIR*, 2015.

[5] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, 2017.

[6] U. Zolzer, *DAFX: Digital Audio Effects*. England: John Wiley & Sons, Ltd., 2002.

[7] R. G. Lopes, S. J. Smullin, E. D. Cubuk, and E. Dyer, "Tradeoffs in data augmentation: An empirical study," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=ZcKPWuhG6wy

[8] D. Tompkins, K. Kumar, and J. Wu, "Maximizing audio event detection model performance on small datasets through knowledge transfer, data augmentation, and pretraining: an ablation study," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 1016–1020.

[9] H. Daolang and P. Ricardo, "Sseldnet: A fully end-to-end sample-level framework for sound event localization and detection," DCASE2021 Challenge, Tech. Rep., November 2021.

[10] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 892–900.

[11] R. Falcon-Perez, K. Shimada, Y. Koyama, S. Takahashi, and Y. Mitsufuji, "Spatial mixup: Directional loudness modification as data augmentation for sound event localization and detection," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 431–435.

[12] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum learning: A survey," *Int. J. Comput. Vision*, vol. 130, no. 6, p. 1526–1565, jun 2022. [Online]. Available: https://doi.org/10.1007/s11263-022-01611-x

[13] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. D. Plumbley, "Polyphonic sound event detection and localization using a two-stage strategy," in *Proc. of DCASE Workshop*, 2019.

[14] J. Lee, J. Park, K. L. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, no. 1, 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/1/150

[15] A. Gulati, C.-C. Chiu, J. Qin, J. Yu, N. Parmar, R. Pang, S. Wang, W. Han, Y. Wu, Y. Zhang, and Z. Zhang, Eds., *Conformer: Convolution-augmented Transformer for Speech Recognition*, 2020.

[16] Asteroid-Team, "Asteroid-team/torch-audiomentations: Fast audio data augmentation in pytorch. inspired by audiomentations. useful for deep learning." [Online]. Available: https://github.com/asteroid-team/torch-audiomentations

[17] K. Shimada, N. Takahashi, S. Takahashi, and Y. Mitsufuji, "Sound event localization and detection using activity-coupled cartesian doa vector and rd3net," DCASE2020 Challenge, Tech. Rep., 2020.

[18] A. Politis, K. Shimada, P. Sudarsanam, S. Adavanne, D. Krause, Y. Koyama, N. Takahashi, S. Takahashi, Y. Mitsufuji, and T. Virtanen, "Starss22: A dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," 2022. [Online]. Available: https://arxiv.org/abs/2206.01948

[19] 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, March 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8567942