DOMAIN-ADVERSARIAL TRAINING AND TRAINABLE PARALLEL FRONT-END FOR THE DCASE 2020 TASK 4 SOUND EVENT DETECTION CHALLENGE

Samuele Cornell[1], Michel Olvera[2]* Manuel Pariente[2]*, Giovanni Pepe[1]*, Emanuele Principi[1]*, Leonardo Gabrielli[1], Stefano Squartini[1]*

[1] Università Politecnica delle Marche, Dept. Information Engineering, Ancona, Italy, {s.cornell;g.pepe}@pm.univpm.it, {e.principi;l.gabrielli;s.squartini}@univpm.it
[2] INRIA Nancy Grand-Est, Dept. Information and Communication Sciences and Technologies, France {manuel.pariente;michel.olvera}@inria.fr

ABSTRACT

In this paper, we propose several methods for improving Sound Event Detection systems performance in the context of the Detection and Classification of Acoustic Scenes and Events (DCASE) 2020 Task 4 challenge. Our main contributions are in the training techniques, feature pre-processing and prediction post-processing. Given the mismatch between synthetic labelled data and target domain data, we exploit domain adversarial training to improve the network generalization. We show that such technique is especially effective when coupled with dynamic mixing and data augmentation. Together with Hidden Markov Models prediction smoothing, by coupling the challenge baseline with aforementioned techniques we are able to improve event-based macro F_1 score by more than 10% on the development set, without computational overhead at inference time. Moreover, we propose a novel, effective Parallel Per-Channel Energy Normalization front-end layer and show that it brings an additional improvement of more than one percent with minimal computational overhead.

Index Terms— Sound Event Detection, Domain Adversarial Training, Per-Channel Energy Normalization, Semi-Supervised Learning

1. INTRODUCTION

Sound Event Detection (SED) is the task of recognizing the set of active sound events, as well as their onset and offset time, in a given audio recording. Such task is useful for a wide variety of applications such as acoustic monitoring, human-computer interaction, and meeting room transcription [1]. Current state-of-the-art methods [2] for tackling SED are largely based on supervised-learning approaches with Deep-Neural-Networks (DNNs) in which a DNN-based classifier is trained using a strongly-labelled dataset of, possibly co-occurring, audio events. However, acquiring and creating such amount of strongly-labelled data is a costly and tedious procedure as it requires human annotation of onset and offset times for each audio event. On the other hand, weakly-labelled data, for which only the set of active sound events is given, can be acquired much more easily. For this reason, there is a growing interest towards using, less costly, weakly-labelled data and unlabelled data to train SED systems and thus decrease the reliance on strongly-annotated data via Semi-Supervised Learning.

In the wake of this trend, several works have recently appeared in the literature that propose semi-supervised methods for SED with weak labels. As a further confirmation of the trend, recent DCASE editions have focused more and more on using weakly and unlabeled data for SED. The majority of the works proposed in the literature for this task are based on Convolutional Neural Networks (CNNs) [3] or Convolutional Recurrent Neural Networks (CRNNs) [4, 5, 6] coupled with a Multiple Instance Learning pooling method [7] for dealing with weak labels and a consistency loss for exploiting unlabeled data [8]. The winner of the 2019 edition of the DCASE Task 4 challenge used guided learning [9] where two networks are updated simultaneously and the teacher network is larger than the student network. The consistency loss in this case is represented by the cross-entropy between the student outputs and the teacher pseudo-labels.

In this work, we outline our systems and techniques for SED in the context of the DCASE 2020 Task 4 challenge. As classifier, we used the the baseline framework composed of a CRNN coupled with Mean Teacher [10]. Here, however, we explored the possibility of using Domain Adversarial Training [11] for dealing with the different data domains composing the training set and we employed an online augmentation strategy to increase the amount of acoustic diversity of the data by dynamic mixing of synthetic examples, artificial reverberation and various time-frequency transformations. Moreover, we propose a parallel Per-Channel Energy Normalization (PCEN) [12] layer applied to the input features as a trainable dynamic compression strategy. Finally, we replaced the common median filter used to smooth the outputs of the classifier with a Hidden Markov Model-based smoothing.

This paper is organized as follows: Section 2 gives an overview of the DCASE 2020 Task 4 challenge. In Section 3 we outline our proposed techniques to tackle the Sound Event Detection problem (DCASE 2020 Task 4 Scenario 1). We present an in-depth explanation of such proposed techniques in Sub-Sections 3.1 to 3.3 and then in Section 4 we perform an in depth ablation study and show the results obtained. Finally, in Section 5 we draw conclusions and trace possible future research directions.

2. THE DCASE 2020 TASK 4 CHALLENGE

In this work, we conducted experiments and tried new techniques in the context of The DCASE 2020 Task 4 Sound Event Detection

*Equal contributions

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

Challenge. The DCASE 2020 Task 4 challenge offers the opportunity to tackle Sound Event Detection (SED) in domestic environments facing real-world issues such as weakly-annotated data, unlabeled data and only a very small corpus of strongly annotated, synthetic data. More specifically, the ultimate goal of the challenge is to develop a SED system being able to tag onset and offset of 10 different sound events classes: Speech, Dog, Cat, Alarm bell/ringing, Dishes, Frying, Blender, Running water, Vacuum cleaner, Electric shaver/toothbrush. The datasets made available are unbalanced and diverse. Specifically, the DESED [13] dataset offers, for training, weakly labelled and unlabelled real soundscapes, and isolated synthetic events with strong labels. The SINS [14] and TUT Acoustic scenes 2017 [15] datasets offer background noise. The FUSS [16] source separation dataset offers isolated events but no annotations. Moreover it does come from a completely different domain as it is aimed at arbitrary sound separation [17].

3. PROPOSED TECHNIQUES

In this section we illustrate the techniques we employed in our submitted systems for DCASE 2020 Task 4 Sound Event Detection challenge. We started from the baseline [10] and kept the CRNN-based architecture as well as the mean-teacher training scheme with same network and optimization hyper-parameters[1]. As said, our main contributions are in the training procedure, feature preprocessing and in prediction post-processing and smoothing.

3.1. Domain-Adversarial Training

The training dataset includes data from different domains, including real and synthetic ones. Moreover, strong annotations are provided only for synthetic data and only a portion of real recording is provided with weak annotations. Instead, test and development set include only data from real-world recordings. This is problematic, as the training and the target domains are different and the SED system must generalize well enough by learning from the synthetic out of domain examples.

Domain Adversarial Training [11] (DAT) provides a solution to this by enforcing a model to learn features that are invariant to the change of domains. This is achieved by embedding the domain adaptation process into the training procedure by adding, to the original architecture, a branch with a gradient reversal layer followed by a domain classifier. The added branch is only used at training time and then dropped at test-time, so there is no computational overhead at run-time. During training, both the network and the added domain classifier are jointly optimized. The gradient reversal layer encourages the feature extraction stage of the original architecture to work adversarially to the added domain classifier by extracting features that are domain-invariant and thus maximize the loss of the domain classification task. We thus employ DAT to enforce learning of features invariant between the synthetic examples domain and real-world recordings domain, reducing in this way the chance of overfitting the strong-labelled synthetic examples.

For the adversarial branch, we employed a modified version of Conv-TasNet [18] separator network available in the Asteroid source separation toolkit [19]. In our modified version, instead of outputting a mask for each transformed-domain feature bin, the separator network outputs a probability on the whole input example by using mean pooling. In fact, the network must classify whether the input example belongs to synthetic examples or to weak/unlabeled examples. The fully convolutional architecture of Conv-Tasnet along with skip connections helped gradient propagation from discriminator to the main network. The adversarial branch was placed in parallel to the RNN block after the CNN layers in the CRNN architecture. The adopted training scheme is illustrated in Figure 1.

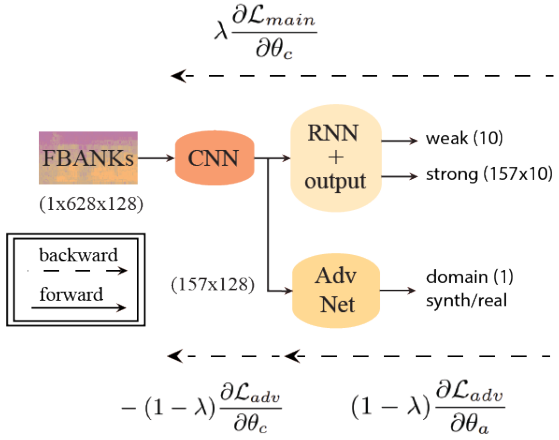

Figure 1: Domain adversarial training scheme.

The CRNN and adversarial branches, are then updated in two different steps adversarially. We denote with $\mathcal{L}_{main}$ the loss for the CRNN training, comprised of strongly labeled loss, weak labeled loss and consistency loss between teacher and student. Thus for the CRNN the update rule for its parameters θ_c is:

$$\theta_c \leftarrow \theta_c - \alpha \left(\lambda \frac{\partial \mathcal{L}_{main}}{\partial \theta_c} - (1-\lambda) \frac{\partial \mathcal{L}_{adv}}{\partial \theta_c} \right), \quad (1)$$

where $\mathcal{L}_{adv}$ is the binary cross-entropy loss for the adversarial network, λ is an hyper-parameter which controls the relative magnitude of the two losses and α is the learning rate. Differently, for the adversarial network with parameters θ_a the update rule is:

$$\theta_a \leftarrow \theta_a - \alpha(1-\lambda) \frac{\partial \mathcal{L}_{adv}}{\partial \theta_a}. \quad (2)$$

In our implementation we did not use the gradient reversal layer proposed by [11], but a two step optimizing procedure like the one used in Generative Adversarial Networks [20]. We found this two-step approach to give better results than the gradient reversal layer approach, as it leads to more stable gradients during training. We tuned λ on the development set and found that a value of 0.1 gave the best results.

3.2. Dynamic Mixing and Augmentation

Because of the limited amount of acoustic diversity in DESED synthetic examples, we also employed an online augmentation strategy. Each synthetic training example is constructed at training time by randomly sampling from one to five random foregrounds and one background file from SINS. We apply reverberation to each source independently by using FUSS [16] Room Impulse Responses (RIRs). Then we apply a random time-domain augmentation chain with different effects to each source, with a maximum of two random cascaded effects: additive noise bursts, additive sine bursts, time-varying comb filters, compression, pitch shifting, low-pass and high-pass filtering.

[1]Code for replicating this work is publicly available at `https://github.com/popcornell/UNIVPM-INRIA-DCASE2020`.

Finally we mix the foregrounds and background. The level for each foreground is randomly sampled between -35 dB and 0 dB while the background is constrained to be at max 5 dB over the foreground which has minimum level. On the feature domain, we add Gaussian noise with SNR between -30 dB and 10 dB and we employ SpecAugment [21]. This procedure ensures a virtually infinite amount of different strongly labelled data.

For weak and unlabeled data, we use a slightly different augmentation scheme as the foregrounds and backgrounds are not available. We only randomly add an additional background from SINS to the original mixture with 50% probability and employ only the aforementioned feature domain augmentations. In fact, we found that using time-domain augmentations on this data worsened the performance as the network failed to generalize to development set when weak and unlabeled data was strongly augmented.

3.3. Parallel Per-Channel Energy Normalization

We experimented with Per-Channel Energy Normalization (PCEN) [12] as a trainable dynamic compression strategy. This technique is able to enhance transient audio events while transforming many soundscape noise patterns into additive white Gaussian noise, improving the robustness of audio classification algorithms in presence of background noise [22] with minimal computational overhead. It is defined as

$$\text{PCEN}(t,f) = \left(\frac{E(t,f)}{(\epsilon + M(t,f))^{\alpha}} + \delta \right)^{r} - \delta^{r}, \qquad (3)$$

where t and f denote time and Mel frequency band index, α, ϵ, r and δ are positive constants and $E(t,f)$ denotes filter bank energy used as feature representation. $M(t,f)$ is a smoothed version of $E(t,f)$, which is computed using a first-order infinite impulse response filter (IIR) as $M(t,f) = (1-s)M(t-1,f) + sE(t,f)$, with s the smoothing coefficient.

While the PCEN operation can be helpful to enhance some sound events in domestic environments, the filtering operation involved in (3) can have a negative impact in sound classes with slowly varying spectro-temporal characteristics, for instance, vacuum cleaner or blender events [23]. Therefore, instead of learning the parameters of a single transformation that finds a trade-off between standing out fast transition sounds and not degrading the quality of stationary-like sounds, we propose to learn several PCEN transformations in parallel, using a Parallel-PCEN trainable front-end (PPCEN). We hypothesize that each individual layer can focus on the specifics of certain groups of sounds. The output of each layer is given as feature channels to the CRNN model and jointly optimize the parameters of such PPCEN front-end layers using backpropagation.

In trainable PCEN we jointly optimize parameter α, δ and r, but instead of learning the smoothing coefficient s, we predetermine two smoothing coefficients $s_1 = 0.015$ and $s_2 = 0.25$ and learn a combination of the smoother outputs.

In Figure 2 we can see the output of our proposed 2-layers PPCEN front-end when it is fed an example with speech and vacuum cleaner events. The vacuum cleaner is present in the whole audio clip, while speech appears in three separate sound events. It can be seen that the first PCEN layer captures also more slow-varying events. In fact, the background noise and the vacuum cleaner harmonics can be clearly distinguished and are enhanced with respect to the original log-Mel features. On the other hand, the second PCEN layer focuses only on events with faster onset such as speech.

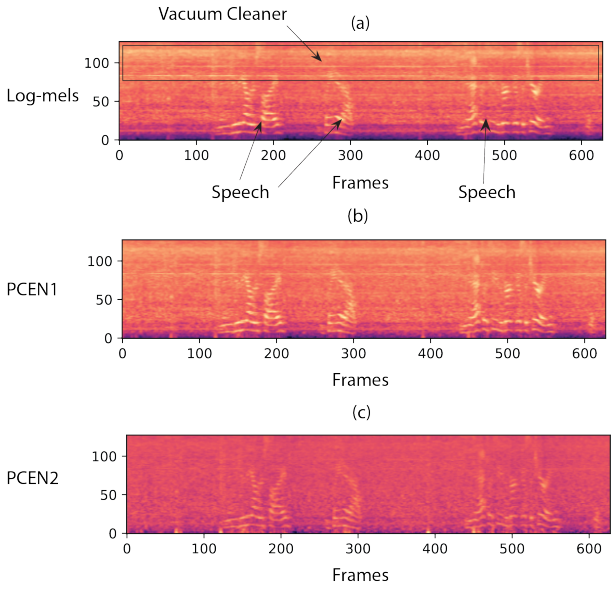

Figure 2: Output of the PPCEN layer: (a) original mixture log-Mels, (b) first PCEN layer, (c) second PCEN layer. The two parallel layers capture different spectro-temporal dynamics.

3.4. Hidden-Markov-Model Smoothing

A popular approach used to reduce the number of spurious detections is to apply median filtering to the outputs of the classifier. Here, instead, final predictions are obtained by using a Hidden Markov Model (HMM) with two states for each class. The silence self-loop transition probability was tied to be the same for all HMMs. We tuned the self-loop transition probabilities for every class and silence on the development set using a 50% split by using Random Forest and with the objective of maximizing the event-based F_1 macro-average score of the trained SED model. Once found the optimal parameters for the HMMs transition probabilities, inference is performed by running Viterbi decoding on the CRNN-obtained emission probabilities for each class. This approach is different to what has been proposed in [24] as we employ only two states for each sound class HMM and, instead of fixing transition probabilities and learning corresponding emission probabilities, as in a full fledged hybrid DNN-HMM system, we tune the HMMs transition probabilities with fixed emission probabilities obtained by the pre-trained SED classifier.

4. RESULTS AND DISCUSSION

Hereafter we report our results obtained on DCASE2020 Task 4 SED for our proposed techniques and systems. Each system we have submitted is comprised of combinations of aforementioned techniques applied to the baseline system.

4.1. Performance of Different Acoustic Front-Ends

In Table 1 we report performance results on the development set according to different acoustic front-ends. As PCEN strictly requires non-negative inputs, we perform global instance min-max scaling instead of z-score normalization over the whole dataset as in the baseline system and no data augmentation was used. The same configuration was used for log-Mel energies.

Table 1: Performance comparison per acoustic front-end.

Acoustic front-end	Event macro F_1 score
Log-Mel	35.93
Fixed PCEN 1	28.71
Fixed PCEN 2	35.42
Fixed PCEN 3	36.84
Trainable PCEN	36.92
PPCEN on Mel	38.20
PPCEN on log-Mel	39.34

Fixed PCEN 1 uses parameters $\epsilon = 10^{-6}$, $\alpha = 0.98$, $\delta = 2$ and $T = 400$ms. This setup degrades sound classes intrinsically stationary yielding a sub-optimal performance. Fixed PCEN 2 uses same parameters except for α, which is set to 0.8. This setup no longer suppresses slower transients and performance is similar to log-Mel. Fixed PCEN 3 also uses $\alpha = 0.8$, but the time constant is set to 60 ms. This configuration seems to favor all sound classes in the task and brings a modest improvement over log-Mel.

By jointly optimizing the PCEN parameters with the cost objective (Trainable PCEN), a slightly better score than Fixed PCEN 3 is achieved. Training a second layer on parallel (PPCEN on Mel) brings a little more than 1% improvement over a single PCEN layer. This setup enhances sounds with faster modulations of foreground in one layer and sounds with slower modulations in the second one. Finally, a similar setup with log-Mel energies (PPCEN on log-Mel) achieves the best performance over the common log-Mel front-end by 3.41%. We found that two layers were sufficient to give significant performance improvement and that adding more layers did not bring additional appreciable improvement.

4.2. Ablation Study

In Table 2 we compare results obtained on development set by the challenge baseline system and results obtained by adding the proposed techniques to the challenge baseline.

We can see that PPCEN alone is able to bring substantial improvement in performance. Instead, online mixing and augmentation (Augm) brings modest performance improvement on its own. We suspect this is due to the fact that the online generated examples are only partially representative of true target-domain sound events and thus the network fails to generalize to real-world data. It however brings significant benefits when it is coupled with DAT. When PPCEN is combined with data-augmentation we observe a drop in performance. We hypothesize this is due to the fact that strong augmented synthetic examples have a significant different spectro-temporal characteristics and, by fitting the PPCEN also on this data, generalization to real-world data can be hampered.

On the other hand, HMM smoothing alone is able to constantly give at least two points performance improvement on all systems.

Finally adding all proposed techniques to the baseline system yields a significant improvement of more than 10 % points over the baseline model, without significant additional computational overhead. It is worth noting that adding PPCEN on top of DAT leads only to a modest improvement while adding PPCEN on top of the baseline significantly improves the results. This is due to the fact that, PPCEN seems to be affected negatively by data-augmentation and DAT is only partially able to overcome this by forcing PCEN layers produce same output distribution for augmented synthetic and real-world examples.

Table 2: Ablation study for proposed techniques (development set).

Method	Event macro F_1 score
Baseline	34.8
+HMM	37.13
+PPCEN	39.93
+PPCEN+HMM	43.69
+Augm	37.31
+DAT+Augm	40.91
+PPCEN+Augm	33.20
+DAT+Augm+HMM	45.20
+PPCEN+DAT+Augm+HMM	46.30

Table 3: Performance on development and evaluation sets.

Method	Event macro F_1 score		PSDS
	dev	eval	dev
Baseline	34.8	34.9	0.61
PPCEN+HMM	43.69	42.6	0.63
DAT+HMM	45.20	42.0	0.68
Ensemble DAT+PPCEN+HMM	46.17	44.4	0.69
Ensemble DAT+PPCEN+HMM 2	47.44	43.2	0.69

4.3. Challenge Results

Hereafter, in Table 3, we report results obtained by our submitted systems on the development and evaluation sets. We submitted a total of four different SED systems: two single systems and two ensemble systems.

Regarding single-systems, we submitted one system with PPCEN and HMM (PPCEN+HMM), and another one with DAT and HMM (DAT+HMM). Both improve substantially over the baseline system, with DAT+HMM achieving the highest score both for development but not for test. In fact the PPCEN+HMM system appears to generalize slightly better to evaluation set. We did not submit the combination of PPCEN+DAT+HMM reported in Table 3 and discussed in previous Section 4.2 due to lack of time.

We also submitted two ensemble systems derived from a combination of PPCEN and DAT systems. For these two submissions we used the same models. The only difference is in different HMM transition probabilities. We used an ensemble of three different single systems: PPCEN and two DAT models from two different training runs. To obtain emission probabilities we simply averaged the outputs of the different models. It can be seen that the second ensemble model (DAT+PPCEN+HMM 2) achieves higher score on development but has worse performance on test. This shows that HMM transition probabilities tuning can have a substantial impact on the final system performance and can be prone to overfitting.

5. CONCLUSIONS

In this work we outlined our proposed techniques for tackling the 2020 DCASE Task 4 challenge. Instead of focusing on exploring new neural architectures we experimented directly with the baseline SED system. We showed that, the addition of PPCEN front-end feature pre-processing, Domain Adversarial Training and online data augmentation and mixing can bring substantial benefits with minimal computational overhead at inference time. As another, parallel, contribution we also showed that HMM smoothing alone can greatly improve performance of the systems by refining network predictions. Future work could explore how the proposed techniques fare when coupled with a stronger baseline system.

6. REFERENCES

[1] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational analysis of sound scenes and events*. Springer, 2018.

[2] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2015, pp. 1–6.

[3] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2180–2193, 2018.

[4] E. Çakır, 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.

[5] W. Wei, H. Zhu, E. Benetos, and Y. Wang, "A-crnn: A domain adaptation model for sound event detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 276–280.

[6] J. Yan, Y. Song, L. Dai, and I. McLoughlin, "Task-aware mean teacher method for large scale weakly labeled semi-supervised sound event detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 326–330.

[7] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 31–35.

[8] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 1195–1204.

[9] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided learning for weakly-labeled semi-supervised sound event detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 626–630.

[10] L. Delphin-Poulat and C. Plapous, "Mean teacher with data agumentation for dcase 2019 task 4," in *DCASE 2019 Tech Report*, 2019.

[11] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[12] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5670–5674.

[13] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[14] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, B. Van den Bergh, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, "The sins database for detection of daily activities in a home environment using an acoustic sensor network," *Detection and Classification of Acoustic Scenes and Events 2017*, 2017.

[15] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*, Budapest, Hungary, 2016.

[16] S. Wisdom, H. Erdogan, D. P. W. Ellis, R. Serizel, N. Turpault, E. Fonseca, J. Salamon, P. Seetharaman, and J. R. Hershey, "What's all the fuss about free universal sound separation data?" in *in preparation*, 2020.

[17] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R. Hershey, "Universal sound separation," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 175–179.

[18] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[19] M. Pariente, S. Cornell, J. Cosentino, S. Sivasankaran, E. Tzinis, J. Heitkaemper, M. Olvera, F.-R. Stöter, M. Hu, J. M. Martín-Doñas, *et al.*, "Asteroid: the pytorch-based audio source separation toolkit for researchers," *arXiv preprint arXiv:2005.04132*, 2020.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[21] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[22] V. Lostanlen, J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling, and J. P. Bello, "Per-channel energy normalization: Why and how," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39–43, 2018.

[23] M. Olvera, E. Vincent, R. Serizel, and G. Gasso, "Foreground-background ambient sound scene separation," *arXiv preprint arXiv:2005.07006*, 2020.

[24] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, and K. Takeda, "Bidirectional lstm-hmm hybrid system for polyphonic sound event detection," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, 2016, pp. 35–39.