DESCRIPTION AND ANALYSIS OF NOVELTIES INTRODUCED IN DCASE TASK 4 2022 ON THE BASELINE SYSTEM

Francesca Ronchini[1*], *Samuele Cornell*[2*], *Romain Serizel*[1], *Nicolas Turpault*[1],
Eduardo Fonseca[3], *Daniel P. W. Ellis*[3]

[1] Universite de Lorraine, CNRS, Inria, Loria, Nancy, France
[2]Department of Information Engineering, Universita Politecnica delle Marche, Italy
[3] Google Research, United States

ABSTRACT

The aim of the Detection and Classification of Acoustic Scenes and Events Challenge Task 4 is to evaluate systems for the detection of sound events in domestic environments using an heterogeneous dataset. The systems need to be able to correctly detect the sound events present in a recorded audio clip, as well as localize the events in time. This year's task is a follow-up of DCASE 2021 Task 4, with some important novelties. The goal of this paper is to describe and motivate these new additions, and report an analysis of their impact on the baseline system. We introduced three main novelties: the use of external datasets, including recently released strongly annotated clips from Audioset, the possibility of leveraging pre-trained models, and a new energy consumption metric to raise awareness about the ecological impact of training sound events detectors. The results on the baseline system show that leveraging open-source pretrained on AudioSet improves the results significantly in terms of event classification but not in terms of event segmentation.

Index Terms— Sound event detection, synthetic soundscapes, deep learning, external data, energy consumption

1. INTRODUCTION

Machine Listening has a substantial impact on applications such as noise monitoring in smart cities [1, 2], smart homes and home security solutions [3, 4], health monitoring systems [5], bio-acoustics [6], and hearing aids [7], among others. Sound Event Detection (SED) is one particularly important task in the field of Machine Listening. Its goal is to correctly output the class of different sound events present in an audio clip, together with each sound event's time boundaries [8]. Multiple events can be active simultaneously in each audio recording, but target sound events can also overlap with other non-target sound events. This research area is in continual development, attracting an expanding community.

Since 2018, the Detection and Classification Acoustic Scene and Events (DCASE) Challenge Task 4 evaluates systems for detection of sound events in audio clips recorded in domestic environments. The challenge's main goal is to explore to what extent it is possible to exploit an unbalanced and heterogeneous dataset, e.g. containing a limited subset of weakly labeled and strongly labeled data and a larger amount of unlabeled data. This is especially important as strongly labeled data is expensive and time-consuming to obtain while weakly labeled and in particular unlabeled data are much more accessible and scalable. Moreover, unlabeled data and weakly labeled data raise significantly fewer privacy concerns. In fact, weak labels could be obtained in an automated manner and, as such, human annotation is not required. This allows using on-device federated techniques which are more privacy-friendly.

Unlabeled data is usually leveraged via self-supervised learning. Indeed, past iterations of DCASE Challenge Task 4 have mainly explored this path, with most of the participants employing the mean-teacher technique [9] for this purpose. However, another possible direction is leveraging pre-trained models from related tasks such as Sound Event Classification (SEC) to obtain a SED system, e.g. via pseudo-labeling, fine-tuning or by using the pre-trained models' internal activations as additional high-level features. In the last couple of years several SEC models [10, 11] trained on the large AudioSet [12] have been open-sourced, making this approach particularly appealing. The use of such pre-trained models has been recently boosted by advancements in self-supervised learning, which enables the training in an unsupervised way on massive amounts of unlabeled data. These models can then be fine-tuned for multiple downstream tasks. A shining example in the audio domain is Wav2Vec 2.0 [13]. While powerful, these models are usually expensive to train and run (in particular self-attention based ones such as Wav2Vec 2.0 [13]), and unsuitable for widespread deployment on on-edge devices. On this premise, this year we introduced the following novelties for the DCASE Challenge Task 4:

- we allowed the use of external datasets and embeddings extracted from open-source pre-trained models. Participants were encouraged to propose open-source models to use and external data sources, such as AudioSet [12], which might also include real-word strongly annotations [14].
- we developed a new codebase[1] and new baseline models to encourage participants to explore pre-trained models and such new datasets. We present the results of such baselines in Section 5.
- finally, we introduced a new energy consumption metric based on CodeCarbon toolkit [15]. This direction aims to foster interest among participants in finding new solutions for effective but also efficient SED systems. This was mainly a pilot experiment as the reliability of CodeCarbon across various computational platforms has not been fully investigated. We explain in detail the new metric in Section 3.2, together with the challenges that needs to be addressed in order to meaningfully compare the energy consumption of different models on different hardware

[*] denotes equal contribution.

[1]Available at `github.com/DCASE-REPO/DESED_task/blob/master/recipes/dcase2022_task4_baseline`

platforms.

The goal of this paper is to describe and motivate these new directions of this year's challenge, and to analyse their impact on the baseline performance.

2. DCASE 2022 CHALLENGE TASK 4: NOVELTIES AND MOTIVATIONS

In every edition, the DCASE Challenge Task 4 proposes to focus on new research questions, targeting specific aspects of the SED task that are considered interesting and timely for the research community, with the ultimate goal of advancing the state-of-the-art.

This year, we investigate three main aspects that are motivated by the availability of large scale datasets related to the SED task, the recent popularity of pre-trained generic audio representation and the growing concerns regarding the environmental impact of our digital life. These developments were proposed in order to target three key scientific questions related to SED systems that we believe are worth addressing.

2.1. What is the impact of using external data and pre-trained models on SED systems?

In the DCASE 2021 Challenge Task 4 [16], we found that using a separation model trained in an unsupervised way using MixIT [17] on the massive YFC100m dataset [18] yielded a significant performance boost when used in conjunction with the baseline SED model on the evaluation set even if results on the development set were not that promising. Motivated by such result, this year we allowed participants to also use external data for the purpose of improving SED performance. To encourage participants to explore this path, we also provided a baseline which employs embeddings from two state-of-the-art popular models trained on AudioSet [12]: PANNs [10] and AST [11]. In addition to being allowed to use AudioSet and these two models, participants were granted the use of other pre-trained models such as YAMNet [19] and also datasets not strictly related to SED such as MUSAN [20] and ImageNet (used in AST [11] for example). Participants were also welcomed to propose other pre-trained models and external datasets. The full list of external resources allowed for the task can be found on the task website [2] and in [21, 22].

Each team was allowed to submit four different systems. However, in order to highlight the impact of external resources on SED systems, we required each team to submit at least one system that was not using external data.

2.2. Is strongly annotated real-world data necessary to build an effective SED system?

In Hershey et al. [14] it was found that strong labels (with temporally precise onset and offset together with sound-event class label) can bring substantial benefits in SEC applications even when they are provided for a small fraction of the total data, which can remain weakly labeled (only the sound class label is provided without any temporal precision). This is especially remarkable since manual annotating data is costly and time-consuming, but also bias-prone due to human errors and disagreement on the perception of some sound event onsets and offsets. On the other hand, synthetic data is cheap to obtain, but also inherently mismatched with respect to real-world data, leading to potentially sub-optimal performance [23]. In an attempt to mitigate this mismatch, and make the synthetic audio more realistic, in the past edition, non-target events have been included in the synthetic split of the training dataset [24]. This year, among the allowable external data source, we also included 3470 strongly labeled recorded audio clips coming from AudioSet [14]. The goal is to assess if the substantial improvement observed for SEC in Hershey et al. [14] also translates to SED and to what extent it may be worth spending extra resources for more manual annotation.

2.3. What is the environmental footprint of our SED systems?

Current state-of-the-art SED systems heavily rely on deep learning. Numerous recent works [25, 26] have raised concerns about the massive environmental costs of training deep learning models with large amount of parameters on massive amounts of data. For example, focusing on the audio domain, in Parcollet et al. [26], a study on the carbon footprint for ASR training was performed using the CodeCarbon toolkit [3], a software package that estimates the amount of energy consumption and carbon dioxide produced by the cloud or personal computing resources used to execute the code. One of the key takeaways from this work is the tremendous inefficiency of many current ASR state-of-the-art models that trade off significant energy consumption (and thus pollution) for a marginal increase in performance that is likely not significant in actual deployment scenarios. These results raise important questions regarding the direction of ASR research, and suggests that the blind pursuit of the best possible performance in spite of the energy efficiency is likely not worth from a practical standpoint.

Motivated by this study, this year we added support in the DCASE Challenge 2022 Task 4 baseline and repository for energy consumption benchmarking based on CodeCarbon. Participants were encouraged to submit their CodeCarbon estimated kilowatt-hour kWh energy consumption figures for each submission, both for model training and, contrary to Parcollet et al. [26], also inference on evaluation data. In fact, we argue that the energy consumption in inference is more important than the training one as the model could end up being deployed on thousands of devices and ran for years, with cumulative energy consumption quickly outpacing training phase. We describe more in detail the new CodeCarbon energy consumption metric and report energy consumption figures for this year baselines in Section 3.2.

3. DCASE 2022 CHALLENGE TASK 4 BASELINE SYSTEM

This year's challenge baseline is based on a convolutional recurrent neural network (CRNN) but includes the main novelty of having the possibility of using features extracted from pre-trained models. CRNN was already found to be the architecture, mainly taken from [27], is composed of a CNN module followed by a 2-layers bi-directional gated recurrent unit (biGRU). The CNN has 7 layers, each composed of batch normalization, gated linear unit and dropout. Input features are Log-Mel Filterbank Energies extracted with a 128 ms window and 16 ms stride. The model is trained with the mean-teacher strategy [9, 27] on audio data resampled at 16 kHz and outputs one frame-wise prediction each 64 ms. To leverage more effectively weakly and unlabeled labeled data, attention pooling is employed, as outlined in [27], to derive clip-level predictions

[2] dcase.community/challenge2022/task-sound-event-detection-in-domestic-environments#external-data-resources

[3] https://codecarbon.io/

from frame-level predictions. From 2020, small changes, such as MixUp [28], and hyper-parameters improvements have been implemented on the baseline system, based on top-ranked systems submitted every years from participants.

3.1. Integration with Pre-Trained Models

As mentioned in Section 2.1, among the main novelties in this year challenge is the allowance of pre-trained models and external data. Participants were encouraged to explore this direction by an expanded codebase in the official challenge repository. We implemented support for two different pre-trained models: PANNs and AST, which we use to extract embeddings to aid in the SED task. In detail, these embeddings are late-fused into the CRNN classifier described previously, before the biGRU module, by combining them with the features extracted by the CNN module. During training the pre-trained model is kept frozen, and it is used only as an embeddings extractor. The CRNN SED classifier is instead trained with the mean-teacher strategy as outlined before.

We consider two types of embeddings extracted from pre-trained models internal activations: global embeddings, which are extracted at clip-level $\in \mathbb{R}^{C}$, and frame-wise embeddings, which instead are frame-wise $\in \mathbb{R}^{S_e \times C}$, where C is the dimension of each vector and S_e is the length of the embeddings sequence.

Regarding PANNs, global embeddings are extracted after the mean and max pooling layers, following the original work [10]. The frame-wise embeddings instead are extracted from the third convolutional block, after dropout. Regarding AST, also in this case we follow the original work for global embeddings [11] and derive them from the layer before the classification head. Frame-wise embeddings instead come from the last transformer layer.

As illustrated in Figure 1, left panel, global embeddings are simply fused with CNN features $\in \mathbb{R}^{S_c \times D}$ via concatenation at each time-step $s \in [1, \dots S_c]$. In this work $D = 128$. We firstly map the embeddings to same dimensionality as the CNN features D and apply layer-normalization. Then these are concatenated with the CNN features at each time-step along the channel dimension, obtaining a tensor $\mathbb{R}^{S_c \times 2D}$, another dense layer plus layer-normalization is used to shrink back the representation to $\mathbb{R}^{S_c \times D}$ before the biGRU. For frame-wise embeddings they cannot be concatenated right away as usually $S_e \neq S_c$, since PANNs, AST and the CNN front-end have different pooling factors. Thus we employ a single-layer BiGRU to encode the frame-wise embeddings into a representation with fixed dimensionality. As illustrated in Figure 1, right panel, we use the biGRU encoder last step output $\in \mathbb{R}^{H}$, where H is 1024 channels, twice the hidden size of the biGRU encoder, here we use 512 neurons. This encoded representation is then fed to the same pipeline employed for global embeddings.

3.2. CodeCarbon metric

As mentioned in Section 2.1, this year we proposed to optionally report the energy consumption in kWh at training and test time. The goal here is to raise awareness regarding the environmental footprint of machine listening systems and SED systems in particular. Since every participant uses a different hardware platform to perform training and inference, absolute energy consumption figures are not directly comparable and cannot be used to assess each system efficiency. To allow for a more fair and meaningful comparison we asked for the participants to also provide, for the dev-test and eval datasets, the energy consumption obtained when inference is performed with the CRNN baseline system. The CRNN baseline kWh can then be used effectively as a common measure, mitigating the factors of variations from hardware difference. As such, in order to analyze the systems performance in terms of SED together with their energy consumption, we used a tentative, trivial energy weighted polyphonic sound detection score (EW-PSDS):

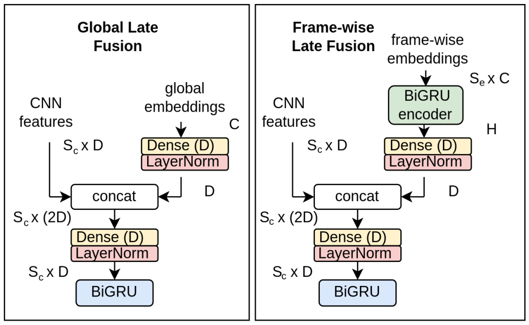

Figure 1: Combining pre-trained models embeddings with the CRNN Baseline via late-fusion before the biGRU module. Left: fusion with global embeddings. Right: fusion with frame-wise embeddings. We report the tensor dimensions as *sequenceLength* × *Channels*.

$$\text{EW-PSDS} = \text{PSDS} * \frac{\text{kWh}_{\text{baseline}}}{\text{kWh}_{\text{submission}}} \tag{1}$$

where PSDS is the polyphonic sound event detection scores [29], $\text{kWh}_{\text{baseline}}$ is the energy consumption reported for the baseline, and $\text{kWh}_{\text{submission}}$ is the energy consumption of the submitted system. Since providing energy consumption was not mandatory for participants, this initial experiment aims to provide insights to design more reliable protocols to obtain energy consumption report from challenge participants and more reliable metrics to report SED performance and energy consumption altogether. In particular, the proposed metric is very challenge-centric as it systematically relies on the energy consumption of the baseline as reference. Additionally, early results have shown that it is heavily biased by large energy consumption differences.

4. CHALLENGE DATASETS AND EVALUATION METRICS

The dataset considered on this paper is the DESED dataset[4] [30, 31], which is the same as provided for the DCASE 2021 Challenge Task 4. It is composed of 10 seconds length audio clips either recorded in a domestic environment or synthesized to reproduce such an environment[5]. The synthetic part of the dataset is generated with Scaper [32], a Python library for soundscape synthesis and augmentation.

The foreground events (both target and non-target) are obtained from the Freesound Dataset (FSD50k) [33], while the background sounds are obtained from the SINS dataset (activity class "other") [34] and TUT scenes 2016 development dataset [35]. The event co-occurences are computed on a set of strong annotations from Audioset [14]. More information regarding the generation of the DESED dataset can be found in Ronchini et al. [24].

[4] https://project.inria.fr/desed/

[5] For a detailed description of the DESED dataset and how it is generated the reader is referred to the original DESED article [31] and DCASE 2021 task 4 webpage: `http://dcase.community/challenge2021`

	PSDS-1 ↑		PSDS-2 ↑	
	dev-test	eval	dev-test	eval
Baseline	0.336	0.315	0.536	0.543
w. AudioSet strong	0.351	**0.345**	0.552	0.540
w. AST-frame	0.313	0.290	**0.722**	**0.678**
w. AST-global	0.205	0.192	0.369	0.305
w. PANNs-frame	0.354	0.304	0.635	0.597
w. PANNs-global	**0.375**	0.308	0.668	0.584

Table 1: Results for the baseline system with additional external data or pre-trained models embeddings. We report PSDS for two application scenarios as described in Section 4.

	kWh ↓		EW-PSDS-1 ↑		EW-PSDS-2 ↑	
	Dev-test	Eval	Dev-test	Eval	Dev-test	Eval
Baseline	**0.030**	**0.617**	**0.336**	**0.315**	**0.536**	**0.543**
w. AST-frame	0.061	0.901	0.149	0.198	0.344	0.464
w. AST-global	0.063	0.873	0.097	0.136	0.181	0.215
w. PANNs-frame	0.045	0.713	0.236	0.263	0.423	0.516
w. PANNs-global	0.045	0.724	0.262	0.285	0.445	0.497

Table 2: Energy consumption (kWh) and Energy Weighted PSDS scores obtained on an Nvidia A100 GPU during inference on dev-test and evaluation.

As evaluation metrics we mainly use polyphonic sound event detection scores (PSDS) [29] and consider two different applications scenarios. The first scenario targets the need of the systems to accurately detect the onset and offset of the sound event, while the second scenario penalizes more confusion between classes, but the temporal localization is less crucial [23].

5. IMPACT OF 2022 CHALLENGE NOVELTIES ON BASELINE SYSTEM

Tables 1 and 2 report the results of this study. In Table 1 we study the impact of pre-trained models and additional strongly labeled data on the challenge baseline SED system described in Section 3. Firstly, we can observe that the addition of strongly-labeled data from Audioset, as expected, improves the PSDS-1 but has practically no effect on PSDS-2. Instead, the use of embeddings from pre-trained models seems to have the opposite trend, it bring benefits mainly to PSDS-2, with AST-frame and PANNs-global coming on top. This is intuitive as PANNs and AST are trained to perform SEC. AST-global seems to perform very poorly. This is explained by the fact that the embeddings in this model are taken directly before the final classification output projection and thus the representation may be too much biased towards SEC. On the other hand, as far as PANNs embeddings are concerned, global embeddings offer the best performance overall but the two are very close. Note that differently from AST, the global embeddings in PANNs are not taken from the layer just before the final linear classification projection but rather after the mean pooling layer.

The takeaway from these experiments is that downstream performance can heavily depend on the choice of the layer at which the pre-trained model embeddings are extracted. Instead of considering just one layer, future works could consider fusing features from many different layers, for example via self-attention.

In Table 2 we compare the energy consumption of the different architectural variations of the baseline system: PANNs versus AST and frame versus global embeddings. In detail we report kWh using CodeCarbon as explained in Section 3.2 on dev-test and evaluation. We also report the relative EW-PSDS scores so that we can take into account each model efficiency. We can notice that AST models consistently require more energy in inference than PANNs. This is in accordance with the number of parameters 80M for PANNs and 88.1M for AST, and with the fact that AST are based on self-attention which is more computationally expensive than convolution. Secondly we can notice that there is little to no difference between frame and global embeddings. This is unexpected because we know that the baseline with frame-wise embeddings should be slightly more computational demanding due to the biGRU encoder as illustrated in Figure 1. This is likely due to the fact that the pre-trained model embedding extraction likely lead the energy consumption figure and CodeCarbon is simply not sensitive enough to pick up this subtle architectural difference. A more fair measure would be floating points operations (FLOP) for each frame prediction in output, however this has also the drawback that it is not easy to derive in a scalable way for multiple different models as current open source toolkits lack support for many operations.

Regarding the energy-weighted PSDS measures we can see that overall the most efficient system seems to be the “plain” baseline. This is because the pre-trained models used are much more computational intensive than the baseline, with one order of magnitude more parameters. More sophisticated techniques such as distillation could however mitigate this and drive down such energy consumption in inference. On the other hand, the EW-PSDS definition in Section 3.2 may in fact be too aggressive and too penalizing regarding energy consumption. Nevertheless indeed the simple CRNN baseline still performs fairly good, it is remarkable that, for example in Table 1, the best results for PSDS-1 are obtained without any additional pretrained model.

6. CONCLUSION

In this paper we presented the developments introduced in this year’s DCASE Challenge Task 4 and analyzed their impact on the baseline performance. We focused on the possibility to use external datasets, in particular AudioSet with strong annotations, embeddings extracted with pre-trained models and on monitoring the energy consumption of the different systems at inference time. We have shown that using recorded (in-domain) clips from AudioSet with strong annotations together with synthetic soundscapes improves the PSDS-1 performance which focuses on accurately localizing the sound events in time. Using embeddings extracted from pre-trained SEC models (like PANNs or AST) improves the performance in terms of PSDS-2 which focuses on the accurate estimation on the sound event class (with loose constraints on the time localization). Yet, there are still some open questions regarding the possibility to efficiently exploit these external resources. Indeed, the baseline remains competitive with the systems using external data or pre-trained models while being substantially simpler. This aspect also reflect in terms of energy consumption of the different systems. In particular, the EW-PSDS of the models with pre-trained models is consistently lower than that of the baseline. Finally, the way monitoring energy consumption was introduced here remains naive and should be consolidated but the preliminary results open the way to considering SED systems under another angle.

7. REFERENCES

[1] G. Ciaburro and G. Iannace, "Improving smart cities safety using sound events detection based on deep neural network algorithms," in *Informatics*, 2020.

[2] P. Zinemanas, P. Cancela, and M. Rocamora, "Mavd: A dataset for sound event detection in urban environments." *DCASE Workshop*, 2019.

[3] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection in domestic environments," *arXiv preprint arXiv:1807.10501*, 2018.

[4] J. Ebbers and R. Haeb-Umbach, "Self-trained audio tagging and sound event detection in domestic environments," in *DCASE Workshop*, 2021.

[5] R. M. Alsina-Pagès, J. Navarro, F. Alías, and M. Hervás, "homesound: Real-time audio event detection based on high performance computing for behaviour and surveillance remote monitoring," *Sensors*, 2017.

[6] V. Lostanlen, J. Salamon, and A. e. a. Farnsworth, "Robust sound event detection in bioacoustic sensor networks," *PloS one*, 2019.

[7] A. Hüwel, K. Adiloğlu, and J.-H. Bach, "Hearing aid research data set for acoustic environment recognition," in *Proc. of ICASSP*, 2020.

[8] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, 2021.

[9] 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*, 2017.

[10] Q. Kong, Y. Cao, T. Iqbal, and Y. e. a. Wang, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE TASLP*, 2020.

[11] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," *arXiv preprint arXiv:2104.01778*, 2021.

[12] J. F. Gemmeke, D. P. W. Ellis, and D. e. a. Freedman, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. of ICASSP*, 2017.

[13] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, 2020.

[14] S. Hershey, D. P. Ellis, and E. e. a. Fonseca, "The benefit of temporally-strong labels in audio event classification," in *Proc. of ICASSP*, 2021.

[15] V. Schmidt, K. Goyal, and A. J. et al., "CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing," 2021.

[16] F. Ronchini, S. Cornell, and N. e. a. Turpault, "DCASE 2021 Task 4 Challenge," https://dcase.community/challenge2021, 2021.

[17] S. Wisdom, E. Tzinis, and H. e. a. Erdogan, "Unsupervised sound separation using mixture invariant training," *Advances in Neural Information Processing Systems*, 2020.

[18] B. Thomee, D. A. Shamma, and G. e. a. Friedland, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, 2016.

[19] S. Hershey, S. Chaudhuri, and D. P. e. a. Ellis, "Cnn architectures for large-scale audio classification," in *Proc. of ICASSP*, 2017.

[20] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[21] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *DCASE Workshop*, 2019.

[22] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *Proc. of ICASSP* , 2020.

[23] F. Ronchini and R. Serizel, "A benchmark of state-of-the-art sound event detection systems evaluated on synthetic soundscapes," in *Proc. of ICASSP*, 2022.

[24] F. Ronchini, R. Serizel, N. Turpault, and S. Cornell, "The impact of non-target events in synthetic soundscapes for sound event detection," *arXiv preprint arXiv:2109.14061*, 2021.

[25] P. Henderson, J. Hu, and J. a. a. Romoff, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *Journal of Machine Learning Research*, 2020.

[26] T. Parcollet and M. Ravanelli, "The energy and carbon footprint of training end-to-end speech recognizers," 2021.

[27] L. JiaKai, "Mean teacher convolution system for dcase 2018 task 4," DCASE2018 Challenge, Tech. Rep., 2018.

[28] H. Zhang, M. Cisse, and Y. N. e. a. Dauphin, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[29] Ç. Bilen, G. Ferroni, and F. e. a. Tuveri, "A framework for the robust evaluation of sound event detection," in *Proc. of ICASSP*, 2020.

[30] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *Proc. of ICASSP*, 2020.

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

[32] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. of WASPAA*, 2017.

[33] E. Fonseca, X. Favory, J. Pons, and e. a. Font, "Fsd50k: an open dataset of human-labeled sound events," *arXiv preprint arXiv:2010.00475*, 2020.

[34] G. Dekkers, S. Lauwereins, and T. et al., "The sins database for detection of daily activities in a home environment using an acoustic sensor network," in *DCASE Workshop*, 2017.

[35] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *EUSIPCO*, 2016.