Limitations of weak labels for embedding and tagging

Nicolas Turpault, Romain Serizel, Emmanuel Vincent

LIMITATIONS OF WEAK LABELS FOR EMBEDDING AND TAGGING

Nicolas Turpault *Romain Serizel* *Emmanuel Vincent*

Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France

ABSTRACT

While many datasets and approaches in ambient sound analysis use weakly labeled data, the impact of weak labels on the performance in comparison to strong labels remains unclear. Indeed, weakly labeled data is usually used because it is too expensive to annotate every data with a strong label and for some use cases strong labels are not sure to give better results. Moreover, weak labels are usually mixed with various other challenges like multilabels, unbalanced classes, overlapping events. In this paper, we formulate a supervised problem which involves weak labels. We create a dataset that focuses on difference between strong and weak labels. We investigate the impact of weak labels when training an embedding or an end-to-end classifier. Different experimental scenarios are discussed to give insights into which type of applications are most sensitive to weakly labeled data.

***Index Terms*—** weak labels, triplet loss, prototypical network, audio tagging, audio embedding

1. INTRODUCTION

Sound carries a lot of information that can provide important information on our environment. In recent years, interest in ambient sound analysis has grown in particular due to the numerous potential applications [1]. Most of the current approaches rely heavily on annotations and complex classifiers. An alternative approach is to learn an intermediate representation or embedding of the data that can allow for a better generalization, reduced training time and amount of annotated data needed by separating the time consuming part of learning the embedding from the training of the final classification or regression. This approach has been used successfully in various domains related to audio signal processing [2, 3].

Multiple attempts have been made in the particular domain of ambient sound either relying only on audio [4–6] or co-training with visualization [7]. However, these works are unsupervised or exploit only a very limited amount of labeled data. Some supervised approaches have been proposed that can allow for exploiting annotated data to learn embedding with a classifier that is later truncated [8] or to learn the embedding using sampling methods like triplet networks [9] or prototypical networks [10]. Tokozume et. al [11] explained how embeddings can be learned by mixing two examples and predicting the ratio of the mix. However it is not demonstrated how these approach could be extended to weak labels.

Weakly labeled data is a recurrent problem in various applications [12, 13] including ambient sound analysis [14], as data is usually available but expensive to annotate. Since 2018, Detection and Classification of Acoustic Scene and Events (DCASE) challenge task 4[1] is providing weakly labeled data. As it is usually too expensive to have enough strongly labeled data, it is a common choice to annotate a sufficient amount of weakly labeled data. Weak labels consist of indicating the presence of a label in a segment without any information about the number of instances or their time localization in the recording. This can be considered as introducing noise in the labels as opposed to strong labels that can be considered as clean and accurate labels.

Using weakly labeled data to get a sound event detection system, which is the prediction of labels with their time localization in a segment of audio, has been studied in the recent years [15–17]. A common approach is to use multi instance learning [14] to predict strong labels while training only on weak labels. The top performing systems with DCASE task 4 used a mean-teacher model [18, 19]. However, from the reports, it is hard to analyze how much weakly labeled data and the strongly labeled data is used for training and the impact of this distribution on the performance.

Shah et al. [17] analyzed the impact of weak labels by using Audioset [20] (which is already weakly labeled) and extended the length of the 10 s segments to 30 s and 60 s to see how performance are degraded. Audioset has the advantage of being real data, but the counterpart is to pose multiple problems that are difficult to analyze separately: multilabeled data, overlapping labels and weak labels.

Tokozume et. al [11] used data from UrbanSound8k [21] a synthetic dataset composed of urban sounds extract from Freesound [22] and verified by human annotators. The segments are mostly strongly labeled and can last up to 4 s.

Other datasets are available, but to tackle the problem of weak labels, we need strongly labeled data in longer recordings so we can adjust how "weak" we want the label to be. We want a single event per recording in order to focus on the weakly labeled data problem and avoid overlapping events, multilabel problems or unbalanced classes. These problems could be added in further experiments. To do so, we propose to create a synthetic dataset by combining an ambient sound events from Freesound with a background. This approach is directly inspired by the Desed synthetic dataset [23].

Our contributions in this paper are the formulation of the problem specific to supervised learning using weak labels for embedding learning and tagging, and the experimental analysis using different embedding learning methods to not be dependent to a specific method. The dataset and the code are available [2].

The remainder of this manuscript is organized as follows. Section 2 describes the different methods to learn embedding and predict tags. Section 3 describes the dataset followed by section 4 describing the different experiments. Section 5 discusses the results and conclusions are provided in section 6.

This work was made with the support of the French National Research Agency, in the framework of the project LEAUDS "Learning to understand audio scenes" (ANR-18-CE23-0020) and the French region Grand-Est. Experiments presented in this paper were carried out using the Grid5000 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`).

[1] http://dcase.community/challenge2019/task-sound-event-detection-in-domestic-environments

[2] https://github.com/turpaultn/walle

2. LEARNING EMBEDDINGS

Let $\mathscr{C}$ be a set of K classes. We have a dataset $\mathscr{D} = \{(\boldsymbol{x_i}, \boldsymbol{y_i})\}_{i=1}^{N}$ where $\boldsymbol{x_i}$ is a time-frequency representation of the input data and $\boldsymbol{y_i} = [e_{i_1}, ..., e_{i_K}]$ is a vector containing the labels with $e_k \in \{0, 1\}$ indicating whether the sound event class k is present in the clip or not. Our goal is to learn an embedding E that can easily discriminate the classes $k \in \mathscr{C}$. The embedding model is followed by a classifier G which performs audio tagging. That is detecting the sound event classes that are present within an audio clip, regardless of the events time boundaries. E and G can be trained jointly (end-to-end classifier) or E can be trained separately from G (triplets or prototypical network).

2.1. End-to-end classifier

In this scenario E and G are trained jointly to optimize a classification cost. As E is not trained to directly optimize a specific cost function there is no explicit distance between the embeddings learned. We optimize our model over its parameters θ to minimize binary cross-entropy:

$$L_\theta = \min_\theta(-(\boldsymbol{y_i}\log(B_\theta(\boldsymbol{x_i})) + (1-\boldsymbol{y_i})\log(1-B_\theta(\boldsymbol{x_i})))). \quad (1)$$

where B is the concatenation of E and G and θ its parameters.

2.2. Triplet loss

For this model, we sample a triplet $(\boldsymbol{x^a}, \boldsymbol{x^p}, \boldsymbol{x^n})$ representing (anchor, positive, negative) where the anchor is an example from the dataset, the positive is an example from the dataset with the same label as the anchor and the negative is an example from the dataset with a label different from that of the anchor. The aim of the triplet loss [24] is to find meaningful embedding space of the data where the anchor and the positive example closer than a negative example and the anchor. The cost function we optimize is then:

$$\sum_{\boldsymbol{x_i} \in \mathscr{D}} \left[||E_\phi(\boldsymbol{x_i^a}) - E(\boldsymbol{x_i^p})||_2^2 - ||E_\phi(\boldsymbol{x_i^a}) - E(\boldsymbol{x_i^n})||_2^2 + \delta\right]_+, \quad (2)$$

where ϕ are the trainable parameters of E, $[]_+$ is the hinge loss, $||.||_2$ is the L_2 norm and δ is the margin. The margin corresponds to difference between the distance between the anchor and the negative and the distance between the anchor and the positive (in the embedding space). The larger the margin δ is the further the negative will be. As this margin depends on distances between the embeddings, in order for the margin to make sense the embeddings have to be normalized before computing the distances.

2.3. Prototypical network

In a prototypical network, we sample the data in a batch in a specific manner. Each batch contains examples from J classes, and we sample m points for each these class (In our case, $J = K$ but it is not mandatory). m_s of the m points $(\boldsymbol{x_s}, \boldsymbol{y_s})$ are called support points and will be used to generate a *prototype* of the class. The prototype of each class is the mean vector of the embedded support points to its class:

$$c_j = \frac{1}{|\mathscr{D}_j|} \sum_{(\boldsymbol{x_s}, \boldsymbol{y_s}) \in \mathscr{D}_j} E_\phi(\boldsymbol{x_s}) \quad (3)$$

where $\mathscr{D}_j$ is the subset of the m_s support points from the class j and $|\mathscr{D}_j|$ is the number of support points from the class j.

Class	Dev		Eval
	Train	Valid	
Alarm/bell/ringing	178	12	63
Blender	89	9	27
Cat	78	10	26
Dishes	99	10	34
Dog	122	14	43
Electric shaver/toothbrush	51	5	17
Frying	56	8	17
Running water	59	9	20
Speech	117	11	47
Vacuum cleaner	64	10	20
Total	314	1378	2124

Table 1: Unique Freesound sound events used in each set

The remaining $m_q = m - m_s$ points $(\boldsymbol{x_q}, \boldsymbol{y_q})$ are called the queries and are used to adapt the model parameters. We try to assign each of the $\boldsymbol{x_q}$ to one of the J classes by comparing their embedding to the prototype corresponding to each class. The loss function optimized is the cross-entropy between a class label and the softmax over the distances between the embedding of query x_q and the prototypes.

2.4. Classification on the embeddings

Once the embeddings are learned with the triplet network or the prototypical network, we can learn a classifier G from these embeddings by optimizing the cross-netropy cost funtion defined in (1).

3. DATASET

The dataset used in this paper is inspired by the DESED dataset [23]. We created 10 s sound clips using foreground sound events from the Freesound dataset [25, 26] and backgrounds from SINS [27] and MUSAN [28]. The synthetic data for training and validation use the same foreground sound events and backgrounds as the DESED synthetic development dataset. The sound clips in our evaluation set use the same foreground sound events and backgrounds as the evaluation set in DESED. Both the training set, the validation set and the evaluation set are created using Scaper [29]. We further make sure that the foreground sound events and the backgrounds do not overlap between the training set and the validation set. Since we want to focus on the problem of weak labels we create sound clips with a single event and a SNR between the foreground sound event and the background is uniformly drawn between 6 dB and 30 dB.

3.1. WAA dataset

We create a dataset with 2,700 files for training, 300 for validation and 750 for evaluation. We called this dataset the weakly annotations analysis (WAA) dataset. It is composed of the 10 sound event classes of the DESED dataset. The number of unique Freesound sound events used to generate this subset is presented in table 1. Each of the subsets are balanced, so they have the same number of clips for each class. The duration of the sound events within this subset is constrained by the duration of the isolated sound event and the position of the sound event onset within the clips (when the sound event is possibly longer than the clip).

In Figure 1, we present the kernel density estimation of the duration for each sound event class in the training set. We can iden-

tify 2 groups of sound event classes, one represents the short events (Dishes, Speech, Alarm/bell/ringing, Dog, Cat) and the other one represents long events (Blender, Electric shaver/toothbrush, Frying, Vacuum cleaner, Running water). However, the duration still varies within the groups. To mitigate these aspects we propose to consider a fixed duration scenario.

3.1.1. WAA1: fixed sized segments

In this scenario, we assume that we know the labels, and divide the sound clips into fixed-size segments that contains either the full sound event or part of it when the event is long. The label associated with each segment then becomes weak when the sound event is shorter than the segment. In order to keep a consistent subset size, we keep only one segment per original file. When the segment length is decreasing we are more confident that most of the frames within the segment do actually contain the sound event. When the segment duration is increasing it increases the number of frames where the sound event is not present but the label still indicates it is present.

For this scenario, we use segments of 200 ms, 1 s and 10 s. As most of the events are longer than 0.2 s the 200 ms subset is almost like the strongly labeled set. The experiments using 1 s segments introduce some weak labels for short sound events. When the segment is increased to 10 s the number of labels that can be considered as week is increasing too.

3.2. 200 ms dataset

In order to study the impact of the weak labels on the embedding quality regardless of the event duration, we derived an additional set from the original isolated sound events. To create this dataset, we use the same association of foreground sound events and background files as in the WAA dataset. However, in this scenario, we we limit the duration of the foreground sound events to 200 ms. In this dataset, only 122 clips have a foreground event that lasts less than 200 ms. The distribution of sound events that are shorter than 200 ms is as follows: 60 Dishes, 18 Speech, 18 Dog, 13 Alarm_bell_ringing, 10 Running_water and 3 Cat. Therefore, even if we greatly reduced the bias induced by sound event duration, there is still about $\frac{1}{4}$ of the Dishes events for which the models may exhibit a different behavior.

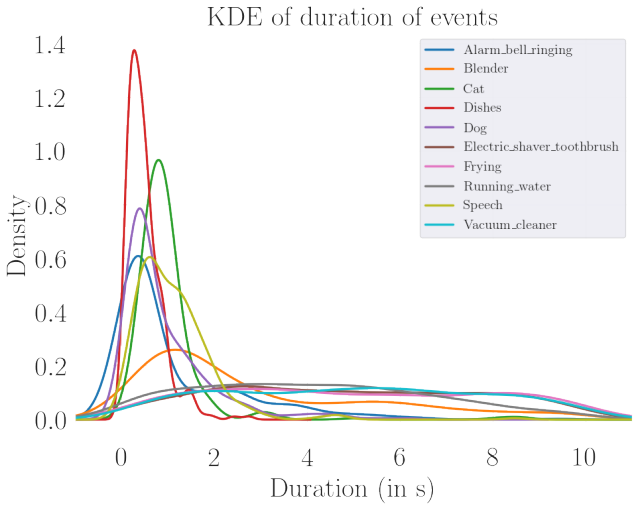

Fig. 1: KDE of the 10 classes used in the dataset.

4. EXPERIMENTS

4.1. Feature extraction

The sound clips are mono-channel and sampled at 44,1 kHz, resampled at 16 kHz. We then compute the short-time Fourier transform on 25 ms windows with a step size of 10 ms. We finally compute log-mel spectrograms features with 64 mel bands.

4.2. Model and parameters

Our embedding model E is a convolutional neural network (CNN) with 4 layers. We average the output of the CNN along the time axis to obtain a single embedding vector. When considering 1 s or 10 s segments in input, the CNN is still applied to obtain embeddings representing 200 ms that are then averaged over time to obtain an embedding representing the whole segment. We consider averaging for aggregation here as we want the approach to be extensible to the multi-event case (which is not compatible with, e.g., a maximum-based aggregation). The final output is a vector of dimension 130 regardless of the duration of the input clip.

The classifier G is a fully connected layer of size 32 with leaky relu activations followed by an output layer of size 10 with sigmoid activations which predicts the sound event class. Sigmoid is used to be able to allow for an extension to the multi-label problem. The architecture of the model (number of convolution layers, dropout, embedding size, number of fully connected layers) has been optimized on the end-to-end classifier with the Asynchronous Successive Halving Algorithm (the asynchronous version of Hyperband) [30], using Orion[3]. We use the same architectures for all experiments.

4.3. Validation of the embeddings

When the embeddings are learned separately we perform early stopping based on a measure related to the quality of the embeddings. We propose a metric that relies the mean value of the embeddings for each sound event class:

$$\boldsymbol{c_k} = \frac{1}{|\mathscr{D}_k|} \sum_{(\boldsymbol{x_i}, \boldsymbol{y_i}) \in \mathscr{D}_k} E_\phi(\boldsymbol{x_k}) \tag{4}$$

where $\mathscr{D}_k$ is the subset of $\mathscr{D}$ that contains the points $(\boldsymbol{x_i}, \boldsymbol{y_i})$ from the class k and $|\mathscr{D}_k|$ is the number of points from the class k in $\mathscr{D}$. Note that this is similar to the prototypes (3) but computed on the whole training set.

In order to measure the quality of an embedding, we then define a metric that indicates for each example $(\boldsymbol{x_i}, \boldsymbol{y_i})$ if the closest center $\boldsymbol{c_k}$ is the center related to the sound event class that is present in the sound clip:

$$F(\boldsymbol{x_i}) = \begin{cases} 1, & \text{if } argmin_k(||E_\phi(\boldsymbol{x_i}) - \boldsymbol{c}_k||_2^2) = argmax_k(\boldsymbol{y_i}) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

This measure is motivated by the fact that we want embeddings which are grouped into separable clusters. Indeed, if every point of each class is closer to the mean of its class, we should be able to separate each class.

[3]https://github.com/Epistimio/orion

Method	Training time	Testing time		
		0.2	1.0	10.0
Classifier	0.2	45.8±2.9	29.6±1.7	3.7±0.5
	1.0	44.2±1.8	47.4±3.2	12.7±2.4
	10.0	39.8±1.9	49.3±3.2	36.7±3.8
Triplets	0.2	42.5±1.0	2.6±0.4	0.0±0.0
	1.0	39.1±2.4	28.9±2.7	0.1±0.1
	10.0	0.0±0.0	0.0±0.0	0.0±0.0
Prototypes	0.2	41.2±3.5	9.4±2.7	0.0±0.0
	1.0	38.8±1.8	36.1±2.1	1.1±1.3
	10.0	0.0±0.0	0.0±0.0	0.0±0.0

Table 2: F-measure results on the 200 ms dataset (in %)

Method	Training Time	Testing time		
		0.2	1.0	10.0
Classifier	0.2	45.8±2.9	49.0±4.1	26.8±3.1
	1.0	46.9±1.2	57.5±2.5	38.0±1.9
	10.0	40.2±2.0	54.2±0.7	51.0±2.3
Triplets	0.2	42.5±1.0	38.2±3.6	11.7±3.2
	1.0	41.7±7.0	44.8±10.9	18.3±7.3
	10.0	9.1±3.2	10.2±2.0	2.8±0.7
Prototypes	0.2	41.2±3.5	36.1±7.3	9.5±4.3
	1.0	45.2±0.4	52.4±3.9	22.0±3.4
	10.0	29.9±6.2	35.8±10.9	28.6±11.0

Table 3: F-measure results on the WAA1 dataset (in %)

5. RESULTS

It has to be noted that we describe 3 different methods to learn embeddings and perform sound event tagging. In this work, we do not want to compare them in terms of absolute performance but we compare their behavior relatively to weakly labeled data. In this section we report the measure of the final classifier learned on the embeddings or the classifier itself to have an actual comparison. The metric used in this work is a F-score measure. When having random predictions, since we are using a sigmoid output for each class, so the model can predict a 0 for each class (no event), it is likely that the model will actually not predict any class because the loss will be smaller than predicting a random class. That is why it does not perform 10% as expected when we have 10 classes (this would be the case if we were using a softmax output) but a F-score of 0% as you will see in Tables 2 and 3.

We present the performance on the 200 ms dataset in Table 2. Training the model with 200 ms clips (first line of each method), and predicting aggregated embeddings of 1 s or 10 s containing background noise, does not work well. So, if we have a strongly labeled training data and train a model on a good segmentation we cannot expect to predict accurately labels on non segmented data. Training the models on weakly labels data also has an impact even when we test the models on already segmented data (first column of the table). This impact is a lot more negative on the embeddings method using sampling than the end-to-end classifier possibly because when using clips that are longer than the actual event we are mostly learning embedding for the background noise. On the other hand, training the end-to-end classifier on 1 s clips actually improves the performance when we test on 200 ms clips. This could be due to the noisy frames acting as a regularization to the model which sees a limited number of data (especially when we take short segments).

We present the performance on the WAA1 dataset in Table 3. We can identify that for each of the models, training on 1 s clips and testing on 1 s clips gives the best results. We can relate this result to Figure 1. As we can see, the duration of most of the short sounds is around 1 s so the bias introduced when training on 1 s clips remains small. This is confirmed when comparing to the results presented in Table 2. Indeed changing the duration of the training clips from 200 ms to 1 s and testing on 200 ms clips was degrading the performance in the 200ms dataset because 9 frames out of 10 during training were then containing noise. On the WAA dataset when we test on 200 ms clips, training on 1 s performs at least as well as training on 200 ms clips (triplets) and often performs event better (end-to-end classifier and prototypes). This later aspect also indicates that 200 ms is probably not long enough to accurately identify the sound classes.

We can also assume that 1 s is sufficient to get enough information about long events. When we train on 10 s clips, performance with the embedding-based methods degrades severely while the end-to-end classifier remains good predictions. Indeed, learning embedding on longer clips becomes very complicated probably because in most cases the clips then contain mostly noise. The embedding learned at clip level are then probably closer to an embedding of the background noise than from an embedding of a sound event class that is hardly represented within the clip. The training of the classifier on the other hand is based on a decision about the class present in the clip. As the background noise is not a class then the classifier cannot be biased towards this and remains more robust when loosing segmentation.

6. CONCLUSION

In this paper, we studied the impact of learning embeddings for audio tagging on weakly labeled data. We proposed two complementary datasets composed of synthetic sound clips. We showed that weak labels degrade the performance slightly when using an end-to-end classifier trained in a discriminative manner. Learning embeddings by sampling and comparing distances (protypical network, triplet loss) is very sensitive to the bias introduced when using weak labels (i.e., several frames within the clip actually do not contain the sound event class but just some background). We observed that operation on clips of small duration reduces this bias. However, the amount of information contained in the clips can them become insufficient to obtain an accurate sound tagging. We also shown that using clips that are too long (both at training and test time) is introducing too much bias and that embedding-based methods then become unreliable. This work could be extended by analyzing more in detail the impact of the sound events duration. The work so far was focused on clips with a single event but real scenarios often include several sound events possibly overlapping. The impact of these aspects would also have to be investigated.

7. REFERENCES

[1] Tuomas Virtanen, Mark Plumbley, and Dan Ellis, *Computational Analysis of Sound Scenes and Events*, Springer, 2017.

[2] Santiago Pascual, Mirco Ravanelli, Joan Serr, Antonio Bonafonte, and Yoshua Bengio, "Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks," in *Proc. Interspeech*, 2019, pp. 161–165.

[3] Laurens van der Maaten and Kilian Weinberger, "Stochastic triplet embedding," in *IEEE International Workshop on Machine Learning for Signal Processing*, Sept. 2012, pp. 1–6.

[4] Y. Xu, Q. Huang, W. Wang, P. Foster, S. Sigtia, P. J. B. Jackson, and M. D. Plumbley, "Unsupervised Feature Learning Based on Deep Models for Environmental Audio Tagging," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1230–1241, 2017.

[5] Mark Cartwright, Jason Cramer, Justin Salamon, and Juan Pablo Bello, "TRICYCLE: Audio representation learning from sensor network data using self-supervision," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, p. 5, 2019.

[6] Jordi Pons, Joan Serr, and Xavier Serra, "Training Neural Audio Classifiers with Few Data," in *In Proc. ICASSP*, May 2019, pp. 16–20.

[7] Jason Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello, "Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings," in *In Proc. ICASSP*, May 2019, pp. 3852–3856, ISSN: 2379-190X, 1520-6149.

[8] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson, "CNN Architectures for Large-Scale Audio Classification," in *In Proc. ICASSP*, 2017.

[9] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Journal of Machine Learning Research*, pp. 207–244, 2009.

[10] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical Networks for Few-shot Learning," in *Advances in Neural Information Processing Systems*, 2017.

[11] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada, "Learning from between-class examples for deep sound recognition," in *ICLR*, 2018, p. 13.

[12] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao, "Learning from Weak and Noisy Labels for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 486–500, Mar. 2017.

[13] Jan Schlter, "Learning to Pinpoint Singing Voice from Weakly Labeled Examples.," in *ISMIR*, 2016, pp. 44–50.

[14] Anurag Kumar and Bhiksha Raj, "Audio Event Detection Using Weakly Labeled Data," in *Proceedings of the 24th ACM International Conference on Multimedia*, New York, NY, USA., 2016, MM '16, pp. 1038–1047, ACM.

[15] Brian McFee, Justin Salamon, and Juan Pablo Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE TRANSACTIONS ON AUDIO*, p. 14, 2018.

[16] Romain Serizel and Nicolas Turpault, "Sound Event Detection from Partially Annotated Data: Trends and Challenges," in *IcETRAN conference*, Srebrno Jezero, Serbia, 2019.

[17] Ankit Shah, Anurag Kumar, Alexander G. Hauptmann, and Bhiksha Raj, "A Closer Look at Weak Label Learning for Audio Events," *arXiv:1804.09288*, Apr. 2018.

[18] Liwei Lin and Xiangdong Wang, "Guided learning convolution system for DCASE 2019 task 4," Tech. Rep., Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, June 2019.

[19] Lionel Delphin-Poulat and Cyril Plapous, "Mean teacher with data augmentation for DCASE 2019 task 4," Tech. Rep., Orange Labs Lannion, France, June 2019.

[20] Jort F. Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *In Proc. ICASSP*, 2017.

[21] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello, "A Dataset and Taxonomy for Urban Sound Research," 2014, pp. 1041–1044, ACM Press.

[22] Frederic Font, Gerard Roma, and Xavier Serra, "Freesound Technical Demo," in *ACM International Conference on Multimedia (MM13)*, Barcelona, Spain, Oct. 2013, pp. 411–412, ACM.

[23] Nicolas Turpault, Romain Serizel, Ankit Parag Shah, and Justin Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Accepted to DCASE2019 Workshop*, NY, USA, 2019, vol. 17.

[24] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu, "Learning Fine-Grained Image Similarity with Deep Ranking," in *In Proc. CVPR*, Columbus, OH, USA, 2014, pp. 1386–1393, IEEE.

[25] Eduardo Fonseca, Jordi Pons, Xavier Favory, Frederic Font, Dmitry Bogdanov, Andrés Ferraro, Sergio Oramas, Alastair Porter, and Xavier Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *Proc. ISMIR*, Suzhou, China, 2017, pp. 486–493.

[26] Frederic Font, Gerard Roma, and Xavier Serra, "Freesound technical demo," in *Proc. ACMM*. ACM, 2013, pp. 411–412.

[27] Gert Dekkers, Steven Lauwereins, Bart Thoen, Mulu Weldegebreal Adhana, Henk Brouckxon, Toon van Waterschoot, Bart Vanrumste, Marian Verhelst, and Peter Karsmakers, "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," in *Proc. DCASE Workshop*, November 2017, pp. 32–36.

[28] David Snyder, Guoguo Chen, and Daniel Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[29] Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. WASPAA*. IEEE, 2017, pp. 344–348.

[30] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar, "Massively Parallel Hyperparameter Tuning," *arXiv:1810.05934*, Oct. 2018.