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

Many datasets and approaches in ambient sound analysis use weakly labeled data. Weak labels are employed because annotating every data sample with a strong label is too expensive. Yet, their impact on the performance in comparison to strong labels remains unclear. Indeed, weak labels must often be dealt with at the same time as other challenges, namely multiple labels per sample, unbalanced classes and/or overlapping events. In this paper, we formulate a supervised learning problem which involves weak labels. We create a dataset that focuses on the difference between strong and weak labels as opposed to other challenges. We investigate the impact of weak labels when training an embedding or an end-to-end classifier. Different experimental scenarios are discussed to provide insights into which 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 current approaches rely on training "big" classifiers in an end-to-end fashion on large-scale labeled data. An alternative approach is to learn an intermediate representation or embedding of the data that can allow for better generalization, shorter training time and smaller labeled data requirements by separating the time-consuming stage of learning the embedding from the final stage of training a smaller classifier. This approach has been used successfully in various domains related to audio signal processing [2, 3].

Several attempts towards learning meaningful embeddings have been made in the field of ambient sound analysis relying on audio-only [4–6] or audiovisual data [7]. These approaches are either unsupervised or based on a very small amount of labeled data. Some supervised approaches have been proposed that can exploit labeled data to learn an 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 the extension of these approaches to weakly labeled data remains an open issue.

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, accurate labels. Weak labels form a recurring challenge in various machine learning applications [12, 13] including ambient sound analysis [14]. Since obtaining enough strongly labeled data is usually too expensive, a common choice is to gather a sufficient amount of weakly labeled data. For instance, Task 4 of the Detection and Classification of Acoustic Scene and Events (DCASE) challenge[1] has provided weakly labeled data since 2018.

The use of weakly labeled data to train a sound event detection system, which outputs labels together with their time localization in a segment of audio, has been studied in recent years [15–17]. A common approach is to use multi-instance learning [14] to predict strong labels while training on weak labels only. The top performing systems for DCASE Task 4 used a mean-teacher model [18, 19]. However, from the evaluation 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 using Audioset [20] (which is already weakly labeled) and measured the performance degradation when extending the length of the original 10 s segments to 30 s or 60 s. Audioset has the advantage of being real data, but this has the inherent drawback of posing several additional challenges that are difficult to analyze separately: multiple labels per segment, unbalanced classes, and overlapping events. Tokozume et al. [11] used data from UrbanSound8k [21], a synthetic dataset composed of urban sound recordings extracted from Freesound [22] and verified by human annotators. The segments are mostly strongly labeled and last up to 4 s.

In order to properly tackle the challenge of weak labels, we need strongly labeled events in longer recordings so that we can simulate weaker labels in a controlled way. Also, we want a single event per recording so as to focus on the challenge of weak labels and avoid multiple labels, unbalanced classes, and overlapping events which could be considered in future experiments. To do so, we propose to create a synthetic dataset by combining ambient sound events from Freesound with a background sound. 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 several embedding learning methods in order not to depend on 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 embeddings and perform tagging. Section 3 introduces the proposed dataset. Section 4 describes the 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 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).

[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 $\boldsymbol{y}_i = [y_{i,1}, ..., y_{i,K}]$ is a vector containing the labels with $y_{i,k} \in \{0, 1\}$ 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 network is followed by a classifier G which performs audio tagging, i.e., detecting the sound event classes that are present within an audio clip, regardless of their time boundaries. E and G can be trained jointly (end-to-end classifier) or E can be trained separately from G (triplet network or prototypical network). We detail these three approaches below.

2.1. End-to-end classifier

The end-to-end approach consists of jointly training E and G to minimize a classification cost. In the following, we minimize the binary cross-entropy

$$\sum_{k} -y_{i,k} \log(G(E(\boldsymbol{x}_i))_k) - (1 - y_{i,k}) \log(1 - G(E(\boldsymbol{x}_i))_k). \quad (1)$$

Since E is not trained to optimize an explicit distance, there is no such distance between the embeddings learned.

2.2. Triplet network

In the triplet network based approach, we consider triplets $(\boldsymbol{x}^{\mathrm{a}}, \boldsymbol{x}^{\mathrm{p}}, \boldsymbol{x}^{\mathrm{n}})$ where the anchor $\boldsymbol{x}^{\mathrm{a}}$ is any sample from the training dataset, the positive example $\boldsymbol{x}^{\mathrm{p}}$ is a random sample with the same label as the anchor and the negative example $\boldsymbol{x}^{\mathrm{n}}$ is a random sample with a label different from that of the anchor. The embedding network E is trained by minimizing the triplet loss [24]

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

where $[.]_+$ is the hinge loss, $||.||_2$ is the L_2 norm, and δ is a margin parameter. The triplet loss aims to find a meaningful embedding space in which the anchor and the positive example are closer than the anchor and the negative example. The margin corresponds to the difference of the distance between the anchor and the negative example and the distance between the anchor and the positive example (in the embedding space). The larger the margin, the further the negative example will be. Since the margin depends on distances between the embeddings, in order for it to make sense, the embeddings must be normalized before computing the distances.

2.3. Prototypical network

In the prototypical network based approach, the data are sampled in a specific manner. Each batch contains J classes (in the following, $J = K$), and m training data points $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ for each class. Among them, m_s points are called support points and are used to generate a *prototype* of the class. The prototype vector for class j is the average of the embeddings of the support points:

$$\boldsymbol{c}_j = \frac{1}{m_j} \sum_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathscr{D}_j} E(\boldsymbol{x}_i) \quad (3)$$

where $\mathscr{D}_j$ is the set of support points of that class. The remaining $m_q = m - m_s$ points are called query points and are used to train the embedding network E. The loss function to be minimized is the sum over all queries of the cross-entropy between the class label of a given query and the softmax over the distances between the embedding of that query and the prototypes of all classes. This results in a soft assignment of each query point to one of the J classes based on these distances.

Class	Dev		Eval
	Train	Valid	
Alarm/bell/ringing	177	13	63
Blender	89	9	27
Cat	78	10	26
Dishes	99	10	34
Dog	121	15	43
Electric shaver/toothbrush	51	5	17
Frying	56	8	17
Running water	59	9	20
Speech	117	11	47
Vacuum cleaner	62	10	20
Total	909	100	314

Table 1: Unique Freesound sound events used in each set.

2.4. Classification of the embeddings

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

3. DATASETS

In order to analyze the impact of weak labels independently of other challenges, we introduce two new datasets called the weak annotation analysis (WAA) dataset and the 200 ms dataset. These datasets are generated from the DESED dataset [23] and contain 10 s sound clips generated by mixing foreground sound events from Freesound [25, 26] with backgrounds from SINS [27] and MUSAN [28]. The synthetic data for training and validation use the foreground sound events and backgrounds from the DESED synthetic development dataset. The sound clips in our evaluation set use the foreground sound events and backgrounds from the synthetic evaluation set in DESED. Both the training set, the validation set and the evaluation set were created using Scaper [29]. We further made sure that the foreground sound events do not overlap between the training set and the validation set and that foreground files from the same Freesound user do belong to the same set. Since we want to focus on the problem of weak labels, we created sound clips with a single event and a signal-to-noise ratio (SNR) between the foreground sound event and the background uniformly drawn between 6 dB and 30 dB.

3.1. WAA dataset

The WAA dataset contains 2,700 clips for training, 300 for validation and 750 for evaluation. It is composed of the 10 sound event classes of the DESED dataset. The number of unique Freesound sound events used to generate each subset is presented in Table 1. Each of the subsets is balanced, i.e., it contains the same number of clips for each class.

The effective duration of each sound event depends on the duration of the isolated event and its onset time within the clip (sound events which are longer than the time remaining until the end of

the clip are cut down). Figure 1 shows the distribution of sound event durations for each class in the training set. We can distinguish two categories of sound event classes: short events (Dishes, Speech, Alarm/bell/ringing, Dog, Cat) and long events (Blender, Electric shaver/toothbrush, Frying, Vacuum cleaner, Running water). However, the duration still varies within each category.

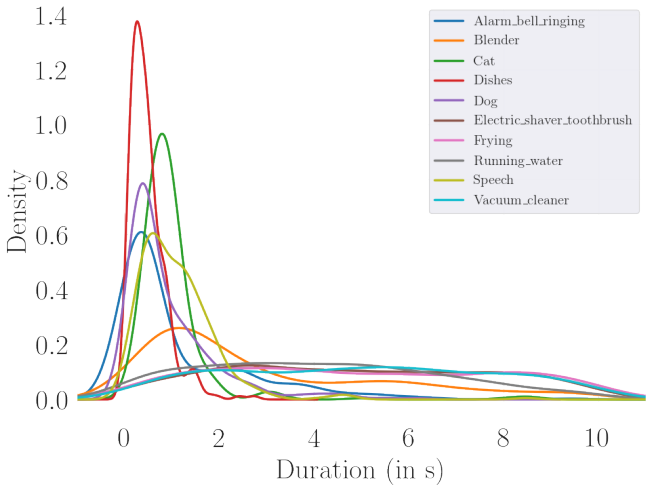

Fig. 1: Kernel density estimates of the duration of sound events for each of the 10 classes in the training dataset.

In order to control the "weakness" of the labels, we cut the 10 s clips into shorter fixed-sized segments. Specifically, we assume that we know the labels, and divided the sound clips into segments that contain either the full sound event or part of it when the event is long. In order to keep a consistent subset size, we kept only one segment per original 10 s clip. When the segment duration is small, most of the frames within the segment do actually contain the sound event therefore the label can be considered as strong. When the segment duration is larger, the number of frames where the sound event is absent increases therefore the label becomes weaker.

We set the segment durations to 200 ms, 1 s, and 10 s in practice. Since most of the events are longer than 200 ms, the 200 ms subset is almost entirely strongly labeled. The experiments using 1 s segments introduce some weak labels for short sound events. When the segment duration is increased to 10 s the proportion of labels that can be considered as weak increases 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 which we call the 200 ms dataset. To create this dataset, we used the same association of foreground sound events and background files as in the WAA dataset. However, we cut the duration of the foreground sound events down 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 events. Therefore, even though the bias induced by sound event duration is greatly reduced, there are still about $\frac{1}{4}$ of the Dishes events for which the models may exhibit a different behavior.

4. EXPERIMENTS

4.1. Feature extraction

The sound clips are single-channel and sampled at 44.1 kHz. We first resample them 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 features with 64 mel bands.

4.2. Model and parameters

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

The classifier G is a fully connected layer of size 32 with leaky ReLU activation followed by an output layer of size 10 with sigmoid activation which predicts the sound event class. The sigmoid is used here in order to allow for a future extension to multi-label classification. 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 for early stopping

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

$$\boldsymbol{c}_k = \frac{1}{|\mathscr{D}_k|} \sum_{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathscr{D}_k} E(\boldsymbol{x}_k) \tag{4}$$

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 $|\mathscr{D}_k|$ denotes the size of that subset. 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 $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ whether the closest centroid $\boldsymbol{c}_k$ is the centroid corresponding to the sound event class that is present in the sound clip:

$$F(\boldsymbol{x}_i) = \begin{cases} 1 & \text{if } \arg\min_k ||E(\boldsymbol{x}_i) - \boldsymbol{c}_k||_2^2 = \arg\max_k(\boldsymbol{y}_i) \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

This metric 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 the classes.

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

Method	Training segment	Test segment		
		200 ms	1 s	10 s
Classifier	200 ms	45.8±2.9	29.6±1.7	3.7±0.5
	1 s	44.2±1.8	47.4±3.2	12.7±2.4
	10 s	39.8±1.9	49.3±3.2	36.7±3.8
Triplets	200 ms	42.5±1.0	2.6±0.4	0.0±0.0
	1 s	39.1±2.4	28.9±2.7	0.1±0.1
	10 s	0.0±0.0	0.0±0.0	0.0±0.0
Prototypes	200 ms	41.2±3.5	9.4±2.7	0.0±0.0
	1 s	38.8±1.8	36.1±2.1	1.1±1.3
	10 s	0.0±0.0	0.0±0.0	0.0±0.0

Table 2: F-measure (%) achieved on the 200 ms dataset.

Method	Training segment	Test segment		
		200 ms	1 s	10 s
Classifier	200 ms	45.8±2.9	49.0±4.1	26.8±3.1
	1 s	46.9±1.2	57.5±2.5	38.0±1.9
	10 s	40.2±2.0	54.2±0.7	51.0±2.3
Triplets	200 ms	42.5±1.0	38.2±3.6	11.7±3.2
	1 s	41.7±7.0	44.8±10.9	18.3±7.3
	10 s	9.1±3.2	10.2±2.0	2.8±0.7
Prototypes	200 ms	41.2±3.5	36.1±7.3	9.5±4.3
	1 s	45.2±0.4	52.4±3.9	22.0±3.4
	10 s	29.9±6.2	35.8±10.9	28.6±11.0

Table 3: F-measure (%) achieved on the WAA dataset.

5. RESULTS

We have described three different methods to learn embeddings and perform sound event tagging. In this section, we report the results achieved by the final classifier G on the evaluation set in terms of the F-measure. Note that, since we are using a sigmoid output for each class, a classifier that performs randomly will likely predict that no class is active. Therefore, the F-score can be as low as 0%, rather than 10% as usually expected for a 10-class classification problem (this would be the case if we were using a softmax output). Also note that we are not interested in the absolute performance of the three methods but in their behavior relatively to weakly labeled data.

We present the results on the 200 ms dataset in Table 2. Training the models with 200 ms segments (first row for each method), and predicting aggregated embeddings for 1 s or 10 s segments containing background noise does not work well. Therefore, if we have a strongly labeled training dataset and we train a model on a good segmentation, we cannot expect it to predict accurately the labels on unsegmented data. Training the models on weakly labeled 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 embedding methods based on triplets and prototypes than on the end-to-end classifier, possibly because when using segments that are longer than the actual event we are mostly learning embeddings for the background noise. By contrast, training the end-to-end classifier on 1 s segments actually improves the performance when testing on 200 ms segments. This could be due to the noisy frames acting as a regularization to the model which sees a small amount of data (especially in the case of short segments).

We present the results on the WAA dataset in Table 3. We can see that, for each of the models, training on 1 s segments and testing on 1 s segments gives the best results. We can relate this observation 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 segments remains small. This is confirmed when comparing these results with those in Table 2. On the 200 ms dataset, changing the duration of the training segments from 200 ms to 1 s and testing on 200 ms segments degrades the performance because 9 frames out of 10 during training contain noise. On the WAA dataset, when we test on 200 ms segments, training on 1 s segments performs at least as well as training on 200 ms segments (triplets) and often performs event better (end-to-end classifier and prototypes). The latter 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 segments, performance with the embedding-based methods degrades severely while the end-to-end classifier still performs well. Indeed, learning embeddings on longer segments becomes very complicated probably because in most cases the segments then contain mostly noise. The embeddings learned at segment level are then probably representing the background noise more than the foreground sound event class that is hardly present within the segment. The training of the classifier on the other hand is based on a decision about the class present in the segment. Since the background noise is not a class, the classifier cannot be biased towards it and it remains more robust to loose 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 learning on shorter duration segments reduces this bias. However, the amount of information contained in the segments can then become insufficient for accurate sound tagging. We also showed 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 event duration. The work so far has focused on clips with a single event but real scenarios often include several sound events that possibly overlap. The impact of these aspects will also have to be investigated.

7. REFERENCES

[1] T. Virtanen, M. Plumbley, and D. P. W. Ellis, *Computational Analysis of Sound Scenes and Events*, Springer, 2017.

[2] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from

multiple self-supervised tasks," in *Interspeech*, 2019, pp. 161–165.

[3] L. van der Maaten and K. Q. Weinberger, "Stochastic triplet embedding," in *IEEE International Workshop on Machine Learning for Signal Processing*, 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] M. Cartwright, J. Cramer, J. Salamon, and J. P. Bello, "TRICYCLE: Audio representation learning from sensor network data using self-supervision," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2019, pp. 278–282.

[6] J. Pons, J. Serrà, and X. Serra, "Training neural audio classifiers with few data," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 16–20.

[7] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 3852–3856.

[8] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 131–135.

[9] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[10] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017.

[11] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," in *International Conference on Learning Representations*, 2018.

[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, 2017.

[13] J. Schlüter, "Learning to pinpoint singing voice from weakly labeled examples," in *International Society for Music Information Retrieval Conference*, 2016, pp. 44–50.

[14] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *24th ACM International Conference on Multimedia*, 2016, pp. 1038–1047.

[15] 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.

[16] R. Serizel and N. Turpault, "Sound event detection from partially annotated data: Trends and challenges," in *IcETRAN Conference*, 2019.

[17] A. Shah, A. Kumar, A. G. Hauptmann, and B. Raj, "A closer look at weak label learning for audio events," *arXiv:1804.09288*, 2018.

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

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

[20] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 776–780.

[21] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM international conference on Multimedia*, 2014, pp. 1041–1044.

[22] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *ACM International Conference on Multimedia*, 2013, pp. 411–412.

[23] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *4th Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019, pp. 253–257.

[24] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[25] E. Fonseca, J. Pons, X. Favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *International Society for Music Information Retrieval Conference*, 2017, pp. 486–493.

[26] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *ACM International Conference on Multimedia*, 2013, pp. 411–412.

[27] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, 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," in *2nd Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017, pp. 32–36.

[28] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv:1510.08484v1*, 2015.

[29] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2017, pp. 344–348.

[30] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, and A. Talwalkar, "Massively parallel hyperparameter tuning," *arXiv:1810.05934*, 2018.