arXiv:1906.02975v1 [cs.SD] 7 Jun 2019

AUDIO TAGGING WITH NOISY LABELS AND MINIMAL SUPERVISION

Eduardo Fonseca[1*], *Manoj Plakal*[2], *Frederic Font*[1], *Daniel P. W. Ellis*[2], *Xavier Serra*[1]

[1]Music Technology Group, Universitat Pompeu Fabra, Barcelona {name.surname}@upf.edu
[2] Google, Inc., New York, NY, USA {plakal,dpwe}@google.com

ABSTRACT

This paper introduces Task 2 of the DCASE2019 Challenge, titled "Audio tagging with noisy labels and minimal supervision". This task was hosted on the Kaggle platform as "Freesound Audio Tagging 2019". The task evaluates systems for multi-label audio tagging using a large set of noisy-labeled data, and a much smaller set of manually-labeled data, under a large vocabulary setting of 80 everyday sound classes. In addition, the proposed dataset poses an acoustic mismatch problem between the noisy train set and the test set due to the fact that they come from different web audio sources. This can correspond to a realistic scenario given by the difficulty of gathering large amounts of manually labeled data. We present the task setup, the FSDKaggle2019 dataset prepared for this scientific evaluation, and a baseline system consisting of a convolutional neural network. All these resources are freely available.

Index Terms— Audio tagging, sound event classification, audio dataset, label noise, minimal supervision

1. INTRODUCTION

Environmental sound recognition has gained increasing attention in recent years, encompassing tasks such as acoustic scene classification, sound event detection or audio tagging [1]. The latter is now becoming a popular task partly due to the various audio tagging tasks in the DCASE Challenge editions, and its impact on applications such as automatic description of multimedia, or acoustic monitoring. This paper describes the characteristics, dataset and baseline system of DCASE2019 Task 2 "Audio tagging with noisy labels and minimal supervision".

Everyday sound tagging consists of identifying the sound events present in an audio recording. The most common approach to create sound event taggers relies on supervised learning through labeled audio datasets. New datasets released nowadays tend to be of increasing size in order to allow exploitation of data-driven approaches, e.g., deep learning. However, manual labeling large datasets is expensive and is typically a limiting factor in machine listening; hence, creators are often forced to compromise between dataset size and label quality. The most recent datasets feature larger sizes [2, 3, 4], but their labeling is less precise than that of conventional small and exhaustively labeled datasets [5, 6, 7]. We are, therefore, witnessing a transition in favour of larger datasets that inevitably include some degree of label noise. Likewise, the current trend is to move towards general-purpose sound event recognizers, able to recognize a broad range of everyday sounds. This is favoured by the appearance of the AudioSet Ontology; a hierarchical tree with 627 sound classes encompassing the most common everyday sounds [2]. This implies that not only are we interested in recognizing typical sound sources (e.g., *Bark* or *Car passing by*), but also less usual sound classes (e.g., production mechanisms such as *Fill (with liquid)*). Further, not only are these classes less frequent, but also some of them can be semantically or acoustically similar to others. For instance, *Trickle, dribble* can be similar to *Fill (with liquid)*, or partially overlapped with it. The manual annotation of these more ambiguous categories becomes more difficult, which makes them more prone to labeling errors or inconsistencies. An alternative to gathering data for training general-purpose audio taggers is to retrieve audio and metadata from websites such as Freesound or Flickr. Labels can be inferred automatically by using automated heuristics applied to the metadata, or applying pretrained classifiers on the downloaded audio material. This approach supports rapid collection of large amounts of data, but at the likely cost of a considerable amount of label noise.

In this context, label noise arises as a challenge in general-purpose sound event recognition, with adverse effects including performance drops or increased complexity of models [8]. Likewise, it can hinder proper learning of classifiers based on deep networks [9, 10]. Consequently, coping with label noise could open the door to better sound event classifiers, and could allow the exploitation of large amounts of web audio for training, while reducing manual annotation needs. The topic of *learning with noisy labels* (LNL) is a consolidated research area in computer vision, counting with extensive literature [11, 12, 13, 14, 15, 16, 17]. In sound recognition, however, LNL has received little attention, probably due to the conventional paradigm of learning from small and clean datasets; and only a few works directly address the analysis and mitigation of label noise [4, 18].

In this paper, we propose a task, a dataset and a baseline system to foster LNL research in general-purpose sound event tagging. We follow-up on DCASE2018 Task 2 [3], and propose to investigate the scenario where a small set of manually-labeled data is available, along with a larger set of noisy-labeled data, in a multi-label audio tagging setting, and using a vocabulary of 80 classes of everyday sounds. The proposed task addresses two main research problems. The first problem is how to adequately exploit a large quantity of noisy labels, many of which are incorrect and/or incomplete, and how to complement it with the supervision provided by a much smaller amount of reliable manually-labeled data. The second problem is given by the acoustic mismatch between the noisy train set and the test set. Distribution shifts between data have been shown to cause substantial performance drops in machine learning, both for vision [19] and audio [20]. In our case, the noisy train set comes from a different web audio source than the test set, which is sometimes a real-world constraint. This paper is organized as follows. Section 2 provides more details about the task and its experimental setup. Section 3 presents the dataset prepared for the task, and Section 4 describes a baseline system. Final remarks are

[*]This work is partially supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 688382 AudioCommons and a Google Faculty Research Award 2017.

given in Section 5.

NOTE: In the current manuscript, a few details of the dataset and results have been omitted purposefully until the DCASE2019 Challenge has concluded. More information will be available in the official submission for the DCASE2019 Workshop.

2. TASK SETUP

The goal of this task is to predict appropriate labels for each audio clip in a test set. The predictions are to be done at the clip level, i.e., no start/end timestamps for the sound events are required. Some test clips bear one ground truth label while others bear several labels. Hence, the task setup is a *multi-label* classification problem and the systems to be developed can be denoted as multi-label audio tagging systems, as illustrated in Fig. 1. This task was hosted on Kaggle—a platform for machine learning competitions—from April 4th to June 10th 2019, and had the participation of more than 800 teams. The resources associated to this task (dataset download, submission, and leaderboard) can be found on the Kaggle competition page.[1]

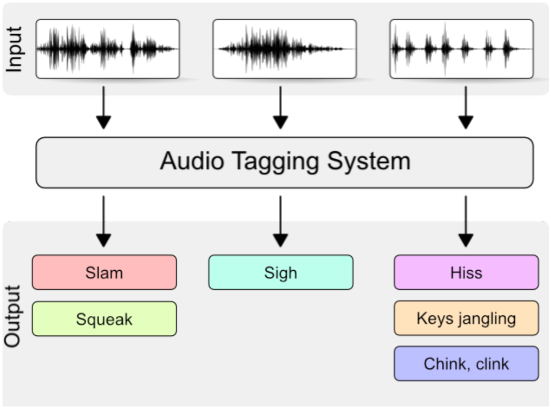

Figure 1: Overview of a multi-label tagging system.

As described in Section 3, the audio data for this task consists of a test set and two train sets: a *curated* train set and a *noisy* train set, that allow to experiment with training data of different levels of reliability and coming from different sources. System evaluation was carried out on Kaggle servers (see Section 2.1) using the test set, which is further split into two divisions, for the *public* and *private* leaderboards. During the competition, the test subset corresponding to the public leaderboard was used to provide live ranking of all participants. To compute the final private leaderboard, at the end of the competition, systems were re-evaluated using the unseen private test set, of which neither the audio nor the labels were given to participants.

2.1. Evaluation Metric and Competition Rules

As the evaluation metric, the task used label-weighted label-ranking average precision (abbreviated as *lwlrap* and pronounced "lol wrap"). This generalizes the mean reciprocal rank used in the 2018 version of the challenge [3] to the case of multiple true labels per test item. A list of relevant labels, ranked according to the classifier, is assembled for each test clip, then the precisions of the sublists formed by truncating at each true label are averaged. The "label-weighted" qualification means that the overall score is the average over all the labels in the test set, with each label receiving equal weight. By contrast, plain *lrap* gives each test item equal weight, thereby discounting the contribution of individual labels when they appear on the same item as multiple other labels. We used label weighting because it allows per-class values to be calculated, and still have the overall metric be expressed as simple average of the per-class metrics (weighted by each label's prior in the test set). A Python implementation of lwlrap is provided in [2].

This scientific evaluation was set up as a Kaggle *Kernels-only* competition. This means that all participants had to submit their systems as inference models in Kaggle Kernels (similar to *Jupyter Notebooks*), to be evaluated on Kaggle's servers. In addition, inference run-time was limited to a maximum of one hour in a Kernel with one GPU. This constraint was imposed in order to discourage the usage of large model ensembles. Participants could submit a maximum of two submissions per day, and select two *final* submissions to be considered for the private leaderboard ranking. A detailed description of the task rules can be found in the Rules section of the competition page;[1] the most important points are summarized in the DCASE Challenge page.[3]

2.2. Judges' Award

To complement the leaderboard results of the lwlrap ranking, the task organizers introduced a complementary Judges' Award to promote submissions using novel, problem-specific and efficient approaches. Details about the Judges' Award rules can be found in the Judges' Award section section of the competition page.[1]

3. DATASET

The dataset used is called *FSDKaggle2019*, and it employs audio clips from the following sources:

- Freesound Dataset (FSD): a dataset under development based on Freesound content organized with the AudioSet Ontology [2]. Freesound is a sound sharing site hosting over 400,000 clips uploaded by a community of users, who additionally provide some basic metadata, e.g., tags. [21]. These data are used to create the curated train set and the test set.
- The soundtracks of a pool of Flickr videos taken from the Yahoo Flickr Creative Commons 100M (YFCC100M) dataset [22]. These data are used to create the noisy train set.

FSDKaggle2019 is freely available, all clips are provided as uncompressed PCM 16 bit 44.1 kHz mono audio files, its ground truth labels are provided at the clip-level (i.e., weak labels), and its partitioning is depicted in Fig. 2.

3.1. Curated train set and test set

The first step carried out in the creation of FSDKaggle2019 was the definition of a vocabulary of 80 classes drawn from the AudioSet Ontology [2]. This vocabulary was chosen based on the

[1] https://www.kaggle.com/c/freesound-audio-tagging-2019 Note that the competition name on Kaggle is abbreviated from the full DCASE2019 Challenge task name to "Freesound Audio Tagging 2019".

[2] https://colab.research.google.com/drive/1AgPdhSp7ttY18O3fEoHOQKlt_3HJDLi8

[3] http://dcase.community/challenge2019/task-audio-tagging#task-rules

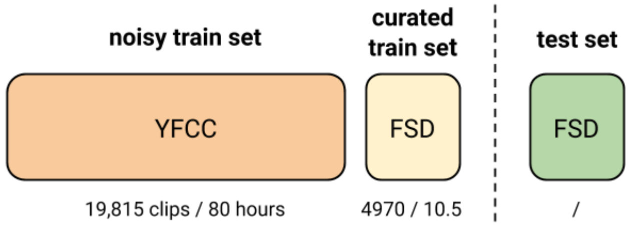

Figure 2: Data split in FSDKaggle2019, including number of clips / duration in hours.

following criteria: *i)* we consider leaf nodes of the AudioSet hierarchy for which there is enough data available in FSD, *ii)* we aim to encompass a diverse range of everyday sounds, and *iii)* we remove clearly isolated classes, promoting confounds between semantically/acoustically similar classes to some extent. The main sound families in the resulting vocabulary are shown in Fig. 3 and the full list of classes is available on [4].

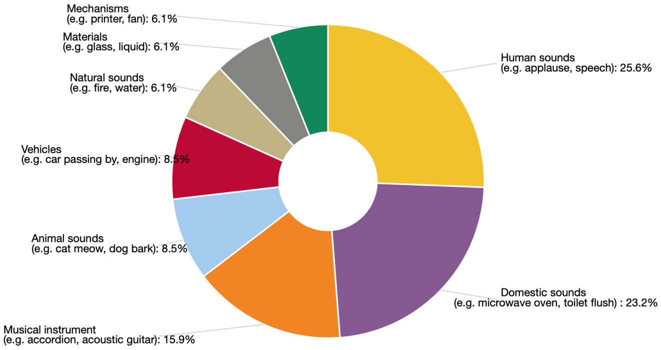

Figure 3: Main sound families in FSDKaggle2019. Circular sector areas correspond to amount of class labels.

In a second step, we did a mapping of Freesound clips to the selected 80 class labels. To this end, a set of keywords was defined connecting the user-provided Freesound tags with the AudioSet labels. Using this mapping, for every class, we retrieved the audio clips that feature at least one of the defined keywords among their tags. This process led to a number of automatically-generated *candidate annotations* indicating the potential presence of a sound class in an audio clip, essentially weak labels. Nonetheless, in some audio clips the target signal fills the clip length almost completely, which can be considered as a strong label. Subsequently, the candidate annotations were human-validated using a *validation* task deployed in *Freesound Annotator*,[5] an online platform for the collaborative creation of open audio datasets [23]. In this task, users verify the presence/absence of a candidate sound class in an audio clip with a *rating* mechanism. Most provided labels have inter-annotator agreement but not all of them. The outcome is a set of clips where the corresponding label(s) are correct; nevertheless, it can happen that a few of these audio clips present additional acoustic material beyond the provided label(s).

The resulting data was split into a train set and a test set. We refer to this train set as *curated* in order to distinguish it from the noisy set described in Section 3.2. To mitigate train-test contamination, the split was carried out considering clip uploaders in Freesound. We allocated all audio clips uploaded from the same user into either the curated train set or the test set, so that the sets are disjoint at the Freesound user level. The partition proportion was defined to limit the supervision provided in the curated train set, thus promoting approaches to deal with label noise.

Table 1: Main stats of the different sets in FSDKaggle2019. *A few classes have slightly less than 75 clips. Stats for the test set will be available in the official submission for the DCASE2019 Workshop.

Aspect	**curated train**	**noisy train**	**test**
Clips/class	∼75*	300	
Total clips	4970	19,815	
Labels/clip	1.2	1.2	
Clip length	∼0.3 - 30s	∼15s	
Total duration	∼10.5h	∼80h	

As a final step, the test set was further annotated using a label *generation* tool [24], in which *i)* pre-existent labels can be revalidated, and *ii)* potentially missing labels can be added through exploration of the AudioSet Ontology. The outcome is a set of exhaustively labeled clips where the label(s) are correct and complete considering the target vocabulary; nevertheless, it can happen that a few clips present additional (unlabeled) acoustic material out of the vocabulary.

The main characteristics of the curated train set, noisy train set and test set are listed in Table 1. The curated train set consists of 4970 clips with a total of 5752 labels. Labels per clip ranges from 1 to 6 with a mean of 1.2. Clip length ranges from 0.3s to 30 due to the diversity of the sound classes and the preferences of Freesound users when recording/uploading sounds.

3.2. Noisy train set

The noisy training set was prepared using the YFCC100M dataset [22], which has the advantages of *i)* being a very large and diverse dataset that is not correlated with Freesound in acoustics or domain, and *ii)* offering permissive Creative Commons licenses that allow ease of use, modification, and redistribution. The original dataset contained ∼99M photos and ∼793K videos from ∼581K Flickr users. We dropped videos with licenses that disallowed making derivatives or commercial use, videos that were no longer available, and videos with audio decode errors that we could not transcode, leaving us with ∼201K 44.1 kHz mono WAV files. Video length varied with a maximum of 20 minutes, and a mean of ∼37s and median of ∼20s.

The Flickr video metadata (title, description, tags) proved to be too sparse to meaningfully map to our class vocabulary. Therefore, we used a content-based approach where we generated video-level predictions from a variety of pre-trained audio models. Hence, the label noise can vary widely in amount and type depending on the class, including in- and out-of-vocabulary noises [4].

As listed in Table 1, the noisy train set consists of 19,815 clips with a total of 24,000 labels (300 * 80). Labels per clip ranges from 1 to 7 with a mean of 1.2. Clip length ranges from 1s to 15s (by construction), with a mean of 14.5s. Therefore, the per-class training data distribution in FSDKaggle2019 is, for most of the classes, 300 clips from the noisy set and 75 clips from the curated set. This means 80% noisy / 20% curated at the clip level, while at the duration level the proportion is more extreme considering the variable-length clips. Since most of the train data come from YFCC, acoustic domain mismatch between the train and test set can be expected.

[4] https://www.kaggle.com/c/freesound-audio-tagging-2019/data

[5] https://annotator.freesound.org

4. BASELINE SYSTEM

4.1. Model Architecture and Training

The baseline system uses a convolutional neural network that takes log mel spectrogram patches as input and produces predicted scores for 80 classes. We use an efficient MobileNet v1 [25] architecture which lets us fit comfortably within the inference time limits of the challenge. Incoming audio (always 44.1 kHz mono) is divided into overlapping windows of size 1s with a hop of 0.5s. These windows are decomposed with a short-time Fourier transform using 25ms windows every 10ms. The resulting spectrogram is mapped into 96 mel-spaced frequency bins covering 20 Hz to 20 kHz, and the magnitude of each bin is log-transformed after adding a small offset to avoid numerical issues. The model consists of 1 convolutional layer followed by 13 separable convolution layers (which give MobileNets their compactness) followed by either max or average pooling, and an 80-way logistic classifier layer. The model contains ∼3.3M weights and, by comparison, is ∼8x smaller than a ResNet-50 while using ∼4x less compute. Detailed documentation is available in the public release of the baseline system code [6].

To combat label noise, we use dropout (inserted before the classifier layer) as well as label smoothing [26] which replaces each label's target value with a hyperparameter-controlled blend of the original value and 0.5 (representing a uniform probability). To combat the domain mismatch between the test set and noisy train set, we use transfer learning by training a model on the noisy set first to learn a representation, and then use a checkpoint from that run to warm-start training on the curated train set. In addition, we used batch normalization, exponential learning rate decay, and the Adam optimizer. We trained models on the curated data alone, the noisy data alone, the curated and noisy combined, noisy first followed by warm-started curated, as well as a weighted version of warm-started training that we describe in the next Section.

4.2. Results

Table 2 shows the lwlrap values produced by our various baseline models when evaluated on the entire test set (i.e., including both the public and private splits). Each row lists the best lwlrap obtained from a small grid search that varied maximum vs average pooling, learning rate, label smoothing, dropout, learning rate decay, and whether or not we used batch normalization. The baseline system that we settled on was ”Warm-started curated”, which achieved a lwlrap of 0.537 on the public test set (see the publicly released baseline code for hyperparameter choices).

Table 2: Baseline system results.

Training	**lwlrap**
Curated only	0.542
Noisy only	0.312
Curated + Noisy	0.522
Warm-started curated	0.546
Weighted warm-started curated	0.561

Comparing ”Noisy only” and ”Curated + Noisy” to ”Curated only” shows a large domain mismatch where we hurt our performance when we blindly add more data. A transfer learning approach of warm-starting the curated training with a noisily trained model gives us a small boost in performance.

We conducted a class-based analysis of the results in a public Colab notebook[7] where we look at the best and worst classes of each model and the correlations between various pairs of models. It becomes evident that, at least for our baseline models, there are classes where using curated data alone is better while there are other classes where the noisy data is better. One simple way to incorporate this in training is to use the ratio of noisy to curated lwlraps as a per-class weight during noisy training to boost classes that have value and suppress classes that don't. When we warmstart with this weighted noisy model, we get a further boost in performance. This optimization is not included in the released baseline.

5. CONCLUSION

In this paper, we have described the task setup, dataset, and baseline system of DCASE2019 Task 2 “Audio tagging with noisy labels and minimal supervision”. This task was hosted on the Kaggle platform as “Freesound Audio Tagging 2019”. The goal is to adequately exploit a large set of noisy-labeled data and a small quantity of manually-labeled data, in a multi-label audio tagging setting with a vocabulary of 80 everyday sound classes. In addition, the dataset poses an acoustic mismatch problem between the noisy train set and the test set due to the fact that they come from different web audio sources. We believe this can correspond to a realistic scenario given the difficulty of gathering large amounts of manually labeled data. Lwlrap is proposed as evaluation metric, and baseline experiments indicate that leveraging noisy-labeled data with distribution shifts for sound event tagging can be challenging. The FSDKaggle2019 dataset and the baseline system proposed are freely available and not limited for use within the competition.

[6] https://github.com/DCASE-REPO/dcase2019_task2_baseline

[7] https://colab.research.google.com/drive/1vIpAWCGSHyYYLblbtMzAKLNTkHcr1SMK

6. ACKNOWLEDGMENT

We thank Addison Howard and Walter Reade of Kaggle for their invaluable assistance with the task design and Kaggle platform, and everyone who contributed to FSDKaggle2019 with annotations. Eduardo Fonseca is also grateful for the GPU donated by NVidia.

7. REFERENCES

[1] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analysis of Sound Scenes and Events*. Springer, 2018.

[2] 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 *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[3] E. Fonseca, M. Plakal, F. Font, D. P. Ellis, X. Favory, J. Pons, and X. Serra, “General-purpose tagging of Freesound audio with AudioSet labels: task description, dataset, and baseline,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*.

[4] E. Fonseca, M. Plakal, D. P. W. Ellis, F. Font, X. Favory, and X. Serra, “Learning Sound Event Classifiers from Web Audio with Noisy Labels,” in *Proc. IEEE ICASSP 2019*, Brighton, UK, 2019.

[5] J. Salamon, C. Jacoby, and J. P. Bello, “A dataset and taxonomy for urban sound research,” in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 1041–1044.

[6] K. J. Piczak, “ESC: Dataset for environmental sound classification,” in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2015, pp. 1015–1018.

[7] P. Foster, S. Sigtia, S. Krstulovic, J. Barker, and M. D. Plumbley, “CHiME-home: A dataset for sound source recognition in a domestic environment,” in *Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2015.

[8] B. Frénay and M. Verleysen, “Classification in the presence of label noise: a survey,” *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, 2014.

[9] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, *et al.*, “A closer look at memorization in deep networks,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 233–242.

[10] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” *arXiv preprint arXiv:1611.03530*, 2016.

[11] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” in *Advances in Neural Information Processing Systems*, 2018, pp. 8527–8537.

[12] E. Malach and S. Shalev-Shwartz, “Decoupling” when to update” from ”how to update”,” in *Advances in Neural Information Processing Systems*, 2017, pp. 960–970.

[13] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, “Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels,” *arXiv preprint arXiv:1712.05055*, 2017.

[14] J. Goldberger and E. Ben-Reuven, “Training deep neural-networks using a noise adaptation layer,” 2016.

[15] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. J. Belongie, “Learning from noisy large-scale datasets with minimal supervision.” in *Proceedings of CVPR*, 2017, pp. 6575–6583.

[16] Z. Zhang and M. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels,” in *Advances in Neural Information Processing Systems*, 2018.

[17] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, “Making deep neural networks robust to label noise: A loss correction approach,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1944–1952.

[18] A. Kumar, A. Shah, A. Hauptmann, and B. Raj, “Learning sound events from webly labeled data,” *arXiv preprint arXiv:1811.09967*, 2018.

[19] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, “Do ImageNet Classifiers Generalize to ImageNet?” *arXiv preprint arXiv:1902.10811*, 2019.

[20] A. Mesaros, T. Heittola, and T. Virtanen, “Acoustic scene classification: an overview of DCASE 2017 challenge entries,” in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, pp. 411–415.

[21] F. Font, G. Roma, and X. Serra, “Freesound technical demo,” in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2013, pp. 411–412.

[22] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, “YFCC100M: The New Data in Multimedia Research,” *Communications of the ACM*, vol. 59, no. 2, Feb. 2016.

[23] 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 *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, Suzhou, China, 2017, pp. 486–493.

[24] X. Favory, E. Fonseca, F. Font, and X. Serra, “Facilitating the manual annotation of sounds when using large taxonomies,” in *Proceedings of the 23rd Conference of Open Innovations Association FRUCT*. FRUCT Oy, 2018, p. 60.

[25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,*, 2016. [Online]. Available: http://arxiv.org/abs/1512.00567