ROBUSTNESS OF ADVERSARIAL ATTACKS IN SOUND EVENT CLASSIFICATION

Vinod Subramanian[1,2*], *Emmanouil Benetos*[1†], *Mark Sandler*[1],

[1] Centre for Digital Music, Queen Mary University of London, London, UK
[2] ROLI Ltd., London, UK
{v.subramanian,emmanouil.benetos,mark.sandler}@qmul.ac.uk

ABSTRACT

An adversarial attack is a method to generate perturbations to the input of a machine learning model in order to make the output of the model incorrect. The perturbed inputs are known as adversarial examples. In this paper, we investigate the robustness of adversarial examples to simple input transformations such as mp3 compression, resampling, white noise and reverb in the task of sound event classification. By performing this analysis, we aim to provide insights on strengths and weaknesses in current adversarial attack algorithms as well as provide a baseline for defenses against adversarial attacks. Our work shows that adversarial attacks are not robust to simple input transformations. White noise is the most consistent method to defend against adversarial attacks with a success rate of 73.72% averaged across all models and attack algorithms.

Index Terms— adversarial attacks, deep learning, robust classifiers, sound event classification.

1. INTRODUCTION

Adversarial attacks are algorithms that add imperceptible perturbations to the input signal of a machine learning model in order to generate an incorrect output. The perturbed input signals are called adversarial examples. The existence of adversarial attacks presents a security threat to deep learning models that are used in tasks such as speech recognition and sound event classification, where fooling classifiers can be used to hide malicious content [1, 2]. Adversarial attacks call into question the robustness of machine learning models and whether we can improve them by addressing adversarial attacks.

There is extensive work that investigates the robustness of adversarial attacks against simple input transformations in the task of image recognition. Kurakin, Goodfellow and Bengio [3] apply transformations such as Gaussian noise, JPEG compression etc. to verify the robustness of adversarial attacks. They work towards physical adversarial examples where a photo can be taken of the adversarial example and fool the image recognition model. There is a lot of similar work in image recognition that focuses on different input transformations and their effect on adversarial attacks [4, 5, 6]. Ultimately, this led to 3-d printouts that were adversarial [7], adversarial stickers [8] etc.

In automatic speech recognition, a lot of adversarial attack algorithms have been developed keeping in mind audio specific concerns. Yakura and Samura [9] and Qin et al. [10] developed methods to simulate real world distortion while creating adversarial attacks to make them robust. Liu et al. [11] developed a system to make the process of generating adversarial attacks quicker and quieter. Du et al. [12] developed the Siren Attack that uses Particle Swarm Optimization to speed up generation of adversarial attacks. Besides the Siren Attack, none of these attacks have been tested on other audio tasks such as sound event classification or music classification.

Similarly, most of the work on defenses against adversarial attacks is focused on automatic speech recognition. Research has shown that mp3 compression, band pass filters, adding noise etc. [13, 14, 15, 16] are effective at eliminating adversarial examples in automatic speech recognition. To our knowledge, no work has been done on the effect of input transformations on adversarial attacks in sound event classification. Esmaeilpour et al. [17] developed a support vector machine (SVM) classifier that was more robust to adversarial attacks for sound event classification, but it was at the cost of model performance.

This research aims to establish a body of work that studies the effects of adversarial attacks and defenses in sound event classification. In this paper, we explore simple input transformations such as mp3 compression, resampling, white noise and live reverb as defenses against adversarial attacks across different models. We build off of work done in Subramanian et al. [18] where the performance of popular adversarial attacks was tested against the top submissions to the DCASE 2018 challenge on General purpose audio tagging[1]. We use the adversarial examples generated in that work and run experiments on how robust they are to input transformations.

Our contributions can be summarised as follows: 1. We evaluate the robustness of adversarial examples generated in [18] against simple input transformations. 2. We create a baseline system of defenses against adversarial attacks for sound event classification.

2. METHODOLOGY

2.1. Adversarial attacks

We use a subset of the adversarial attacks in Subramanian et al. [18], the attacks we ignore are the two weaker baseline attacks. All of the attacks used are white box attacks meaning that they have full information of the model they are attacking. The attacks fall into two categories, untargeted and targeted attack. A targeted attack is when the algorithm fools the output classifier to a specific predetermined class. An untargeted attack is when the algorithm reduces the confidence of the current class until the classifier is fooled. The adversarial attacks used in this work are as follows:

[*] This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No. 765068.

[†] supported by RAEng Research Fellowship RF/128.

[1] http://dcase.community/challenge2018/task-general-purpose-audio-tagging-results

https://doi.org/10.33682/sp9n-qk06

1. **L-BFGS:** Szegedy et al. [19] introduced one of the first methods for generating adversarial attacks, it is a targeted attack.

   Assume a classifier denoted as $f : \mathbf{R}^m \rightarrow \{1...k\}$ with a loss function $loss_f$. For a given input $x \in \mathbf{R}^m$ and target $t \in \{1....k\}$ we aim to identify the value of perturbation r as formulated below:

   $$\begin{aligned} &\text{Minimize } \|r\|_2 \text{ under the conditions:} \\ &f(x+r) = t \\ &x + r \in [0,1]^m \end{aligned}$$

   The box constraint on $x+r$ is to prevent clipping. The exact computation of this problem is difficult so it is approximated using the box constrained L-BFGS algorithm. So the new equation to minimize is:

   $$c|r| + loss_f(x+r, l) \quad \text{under the conditions } x + r \in [0,1]^m$$

2. **DeepFool**: DeepFool attack was introduced by Moozavidezfooli et al. [20]. It is an untargeted attack where the algorithm iteratively linearizes the deep learning model to generate perturbations to fool the classifier. We show how the Deepfool classifier works in the case of a binary classifier, in order to understand how it scales to the multi-class problem we recommend the readers look at the original paper.

   We use the same terminology as defined above for the L-BFGS algorithm. In this case we start by assuming we have a linear classifier f so the relationship between the input x and output can be written as:

   $$f(x) = \omega^T x + b$$

   Here, ω is the weight matrix and b is the bias added. In a binary linear classifier there is a hyper-plane $\mathcal{H}$ that separates the two classes. The hyper-plane is defined such that $x \in \mathcal{H} \rightarrow f(x) = 0$ So, to generate an adversarial attack you need to create a perturbation that pushes the input to the other side of the hyper-plane. This perturbation corresponds to the orthogonal projection of the input onto this hyper-plane. The perturbation for a particular input x_0 denoted by $r(x_0)$ can be computed using the formula:

   $$r(x_0) = -\frac{f(x_0)}{\|\omega\|_2^2}\omega$$

   In the case of a general differentiable binary classifier the model is linearized iteratively and the perturbation is calculated using the formula given above.

3. **Carlini and Wagner -** Carlini and Wagner introduce a strong set of attacks based on the L_0, L_2 and L_∞ distance [21]. This can be used as a targeted and untargeted attack. The problem for adversarial attacks is formulated the same way as Szegedy et al. [19]. The classifier is denoted as C with input x, c is constant:

   $$\begin{aligned} \text{Minimize } & D(x, x+\delta) + c.f(x+\delta) \\ & \text{such that } x + \delta \in [0,1]^n \end{aligned}$$

   Here D is a distance function that is either the L_0, L_2 or L_∞ norm and f is an objective that simplifies the problem such that:

   $$\begin{aligned} C(x+\delta) = t \text{ is true if } f(x+\delta) \leq 0 \\ f(x') = (max(Z(x')_i) - Z(x')_t)^+ \; i \neq t \end{aligned}$$

   In the equation for f, Z denotes the penultimate layer of the classifier and t is the target class. This is just one example of the function f many other functions work and can be found in the paper [21]. In this work we use the L_2 version of the Carlini and Wagner attack.

2.2. Input Transformations

We pick input transformations that are likely to occur in the real world when playing an audio file and recording it on a smart phone. The input transformations are as follows:

Mp3 compression - Mp3 compression is a popular format for storing audio files. It is done using the libmp3lame encoding library inside ffmpeg [22]. In our experiments, we compress the adversarial audio examples at three constant bit-rates–48kbps, 128kbps and 320kbps. The lower the bitrate, the higher the information loss will be.

Re-sampling - We are interested in the effects of removing high frequency content on adversarial examples. In our work, resampling serves as a low pass filter and is performed using the resampy[2] python library. Resampy uses a band limited sinc interpolation method for re-sampling [23]. We use the "kaiser best" configuration which is the high quality version of resampy. The adversarial audio files in our experiments have a sampling rate of 32kHz. We resample the audio files to 8kHz, 16kHz, and 20kHz and resample it back to 32kHz.

White noise addition - White noise is a standard digital distortion. It is added to the adversarial examples at a signal-to-noise ratio of 20dB, 40dB and 60dB.

Live reverberation - Using the live recording setting of the audio degradation toolbox [24], we obtain the impulse response for the "Great Hall"–one of the live rooms with a very long reverb. We applied said impulse response to add reverb to our adversarial audio files using convolution. After convolution, we eliminate the tail of the audio file in order to preserve its original length.

2.3. Datasest

We use the FSDKaggle2018 dataset [25] introduced for the DCASE 2018 challenge on general-purpose audio tagging. This dataset consists of 41 classes, ranging from urban sounds such as buses, keys jangling and fireworks to musical instruments such as cello, snare drum and Glockenspiel. We use the adversarial audio examples generated on this dataset in Subramanian et al. [18]. For the untargeted attacks we use 6 audio files per class making a total of 246 audio files per model per attack. For the targeted attack we use a subset of 6 classes and for each class we generate 5 targeted adversarial attacks to each of the other 5 classes. This makes 180 audio files per model per attack. Since the adversarial attack algorithms are not 100% effective the actual number of adversarial examples are a bit lower than the number indicated above.

[2] https://github.com/bmcfee/resampy

Model	Training	Test
VGG13	0.9714	0.8093
CRNN	0.9768	0.8437
GCNN	0.9803	0.8437
dense_mel	0.9876	0.89875
dense_wav	0.9698	0.86125

Table 1: Model performance on training and test data.

2.4. Models

We use the DenseNet models described by Jeong and Lim [26]. The DenseNet model concatenates the input to the output for each module. We use two versions of the architecture, the first version uses log mel spectrogram as input (dense_mel) to the model and the second one uses raw audio (dense_wav) as the input to the model.

We use three models from Iqbal et al. [27], the VGG13, CRNN and GCNN networks. VGG13 is a convolutional neural network inspired by the VGG13 architecture. CRNN is a convolutional recurrent neural network that uses a bidirectional RNN after the convolutional layers. The GCNN is a gated convolutional neural network where the gated component is inspired from Long-Short Term Memory (LSTMs). The input to all three models is a log mel spectrogram.

Table 1 shows the training and test accuracy for each of the models on the FSDKaggle2018 dataset [25]. We pick these models because they were the top submissions for the DCASE 2018 challenge on "General purpose audio-tagging".

2.5. Experiment and metrics

The experimental setup has three sets of labels. First is the ground truth, which is the label associated with the audio file from the FSDKaggle2018 dataset [25]. The second is the adversarial label, generated by applying adversarial attacks on the audio file from the aforementioned dataset. The third is the transformed label, which is generated by running each audio file through each of the input transformations separately. Once these transformed inputs are run through the model, it generates the transformed label. Our task is to verify how effective a defense and input transformation is against an adversarial attack. We compare how many transformed labels are ground truth, adversarial, or different from both.

A good defense would convert a lot of the transformed labels to the ground truth; however, if an adversarial attack is robust, a lot of the transformed labels will remain the adversarial labels. We use signal-to-noise ratio and output confidence values generated from Subramanian et al. [18] to explain the results.

3. RESULTS AND DISCUSSION

Table 2 provides a reference for how the defenses affect the ground truth audio data before an adversarial attack is performed. Table 3 compares the effectiveness of mp3 compression, white noise addition, re-sampling and live reverb averaged across all of the models and adversarial attack algorithms. In general, the numbers look as we expect: the more distortion we add to the adversarial example, the more likely it will stop being an adversarial example. The best defense against the adversaries on average is adding noise at 20dB. As we raise the volume of noise to 40dB, the number of audio examples that are destroyed lowers; however, the number of adversarial examples that are classified as a different label is lowered as well. Live reverb is the second most successful defense against the adversaries. These simple input transformations can defend against adversaries, showing that there is a need to create more powerful attacks against sound event classification.

Transform	GT	Diff
mp3 48k	93.98	6.02
mp3 128k	99.92	0.08
mp3 320k	100	0
noise 20dB	86.67	13.33
noise 40dB	97.89	2.11
noise 60dB	99.84	0.16
sr 8kHz	55.36	44.63
sr 16kHz	85.77	14.23
sr 20kHz	91.87	8.13
live reverb	82.85	17.15

Table 2: Summary of defenses on ground truth data.

Transform	Adv	GT	Diff
mp3 48k	45.02	50.23	4.75
mp3 128k	83.66	15.71	0.63
mp3 320k	91.55	8.26	0.17
noise 20dB	3.40	73.72	22.87
noise 40dB	31.68	63.69	4.61
noise 60dB	81.00	17.58	1.41
sr 8kHz	27.93	41.71	30.34
sr 16kHz	42.36	47.85	9.78
sr 20kHz	47.90	44.89	7.20
live reverb	4.09	67.07	28.83

Table 3: Summary of performance given as a percentage of adversarial examples that remain adversarial, that go back to ground truth and that change completely.

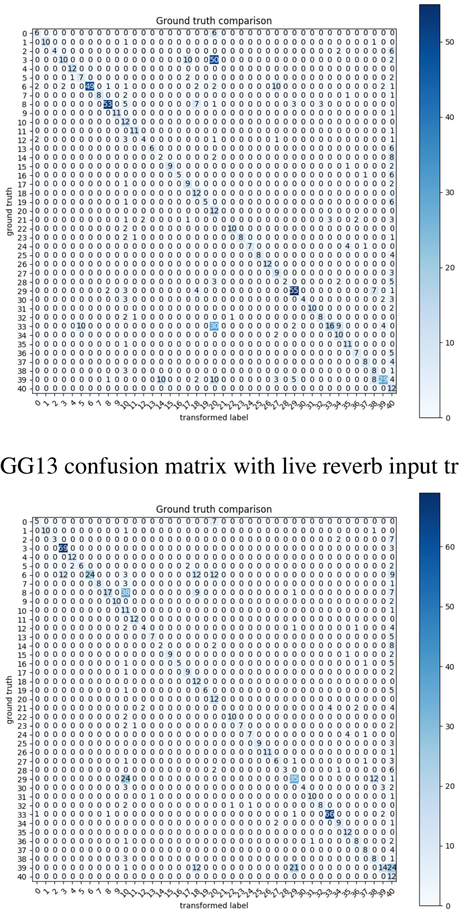
The next table, Table 4, shows how each model behaves in response to these input transformations on the adversarial attacks. We do not show all the data in the interest of space; instead, we show the best two defenses for each model. Across all the models, adding white noise is an effective defense against adversarial attacks. Besides adding white noise, reverb is one of the better defenses. In general, the defenses are more successful for the DenseNet models than for the other three models. One of the reasons for this could be because the adversarial attacks for the DenseNet model are optimized on the output probabilities; whereas the other three models are optimized on the output scores. This means that optimizing an adversarial attack for the DenseNet models only needs to maximize the relative score of the desired class.

Between the DenseNet models the raw audio configuration of the DenseNet behaves differently. Resampling at 20kHz successfully eliminates adversarial examples at 97.18%. For the mel spectrogram configuration of the DenseNet resampling at 20kHz is only successful for 72.36% of the cases. This strongly suggests that the model is sensitive to different changes between the mel spectrogram and raw audio inputs. We speculate that the perceptual weights introduced by the mel spectrogram make that version of the DenseNet give less importance to higher frequencies. This means that losing frequency above 20kHz would impact the mel spectrogram model less adversely than the raw audio model.

In the cases where the defenses are less effective, we want to know if the distribution of the adversarial examples on which the input transforms are effective or ineffective are similar. We pick the

Model	Best	Adv	GT	Diff
dense_mel	noise 20dB	1.52	97.86	0.61
	noise 40dB	6.25	92.97	0.76
dense_wav	sr 20kHz	1.46	97.55	0.97
	mp3 48k	1.58	97.18	1.22
VGG13	noise 20dB	4.34	58.74	38.89
	live	5.05	57.69	37.25
CRNN	noise 20dB	4.70	65.92	29.37
	noise 40dB	27.37	62.86	9.75
GCNN	noise 40dB	27.81	64.31	7.86
	live	5.98	57.74	36.26

Table 4: Top 2 defenses against the different model architectures averaged over all the adversarial attacks.



(a) VGG13 confusion matrix with live reverb input transform.

(b) VGG13 confusion matrix with noise 20dB input transform.

Figure 1: Comparison of two confusion matrices. The confusion matrix plots the ground truth against the transformed label.

Attack	SNR/Conf	Best	Adv	GT	Diff
deepfool	50/0.35	noise 20dB	4.14	77.23	18.61
		noise 40dB	16.09	76.17	7.72
		live	4.22	70.37	25.04
C&W untargeted	51.26/0.71	noise 20dB	5.88	74.83	19.28
		live	5.88	70.17	23.93
		noise 40dB	43.79	50.98	5.22
L-BFGS	56.25/0.98	noise 20dB	0.40	69.56	30.02
		noise 40dB	32.33	66.44	1.22
		live	1.63	62.77	35.59
C&W Targeted	50.49/0.97	noise 20dB	1.31	70.6	28.07
		noise 40dB	36.32	61.52	2.15
		live	3.46	60.93	35.60

Table 5: Table shows the top 3 defenses against input transforms for the different adversarial attacks averaged over all the models. The SNR in dB and label confidence (Conf) as probability of the adversarial examples are presented as averaged over all the models.

VGG13 model to compare since the success of noise at 20dB and live reverb is very close. We plot the confusion matrices for the two scenarios in figure 1. Noise at 20dB is an effective defense for label 3 (Bass drum) and 33 (Snare drum), but live reverb is not a good defense, live reverb is successful against label 8 (Clarinet) but, noise at 20dB is not as effective etc. Interestingly label 21 (Gunshot or gunfire) has 0% success for both defenses. Evidently, audio files from different labels seem to have disparate properties; therefore, applying each distortion type will affect the differing labels uniquely This would mean that while developing an adversarial attack that works in the real world, we need to come up with a solution that is robust to different types of distortion.

Table 5 shows the performance of the top 3 defenses for each adversarial attack algorithm. The targeted attacks seem more robust to these input transformations than the untargeted attacks but not by too much. We expect Carlini and Wagner to be more robust because it is a more powerful attack than Deepfool and L-BFGS as is shown in Subramanian et al. [18].

The fact that the SNR is very high for all of the attack algorithms is good from a real world perspective because that means that the noise added to make the audio files adversarial is less likely to be perceived. However, it is possible that the noise is being masked by the input transformations which makes the adversarial attacks not very robust. Given that the SNR is so high there is a lot of headroom to improve adversarial attack algorithms for sound event detection where we increase the amount of noise added without compromising too much on how perceivable the adversarial attacks are.

4. CONCLUSION

We show that simple input transformations such as mp3 compression, re-sampling, white noise addition and live reverb are effective defenses against popular adversarial attacks. White noise at 20dB is the most consistent method to defend against adversarial attacks with live reverb being a close second. The raw audio version of the DenseNet behaves differently with re-sampling at 20kHz, being the most effective defense. This suggests that different input representations affect the type of features that a deep learning model can learn by making the deep learning model focus on different frequency bands. Generally, we hope that these defenses give a sense of current weaknesses in research on adversarial attacks for audio.

Another area that we plan to explore is trying to explain the presence of adversarial attacks in sound event classification. We aim to combine research from interpretability and adversarial attacks in order to work towards explaining and interpreting deep learning.

5. REFERENCES

[1] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice

recognition," in {*USENIX*} *Security Symposium*, USA, 2018, pp. 49–64.

[2] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *IEEE Security and Privacy Workshops (SPW)*, USA, May 2018, pp. 1–7.

[3] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *arXiv:1607.02533*, 2016.

[4] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *International Conference on Learning Representations (ICLR)*, Canada, April 2018.

[5] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *arXiv:1711.01991*, 2017.

[6] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of JPG compression on adversarial images," Aug. 2016, arXiv: 1608.00853.

[7] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *International Conference on Machine Learning (ICML)*, Sweden, Jul 2018, pp. 284–293.

[8] K. Eykholt, I. Evtimov, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on machine learning models," in *arXiv:1707.08945*, 2017.

[9] H. Yakura and J. Sakuma, "Robust Audio Adversarial Example for a Physical Attack," *arXiv:1810.11793*, Oct. 2018.

[10] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *International Conference on Machine Learning (ICML)*, USA, Jun 2019, pp. 5231–5240.

[11] X. Liu, X. Zhang, K. Wan, Q. Zhu, and Y. Ding, "Towards Weighted-Sampling Audio Adversarial Example Attack," *arXiv:1901.10300*, Jan. 2019.

[12] T. Du, S. Ji, J. Li, Q. Gu, T. Wang, and R. Beyah, "SirenAttack: Generating Adversarial Audio for End-to-End Acoustic Systems," *arXiv:1901.07846*, Jan. 2019.

[13] K. Rajaratnam, K. Shah, and J. Kalita, "Isolated and Ensemble Audio Preprocessing Methods for Detecting Adversarial Examples against Automatic Speech Recognition," in *Conference on Computational Linguistics and Speech Processing (ROCLING)*, Taiwan, Oct. 2018, pp. 16–30.

[14] K. Rajaratnam and J. Kalita, "Noise flooding for detecting audio adversarial examples against automatic speech recognition," in *International Symposium on Signal Processing and Information Technology (ISSPIT)*. USA: IEEE, Dec 2018, pp. 197–201.

[15] Z. Yang, B. Li, P.-Y. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," in *International Conference on Learning Representations (ICLR)*, USA, 2019.

[16] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, S. Li, L. Chen, M. E. Kounavis, and D. H. Chau, "Shield: Fast, practical defense and vaccination for deep learning using jpeg compression," in *International Conference on Knowledge Discovery*, USA, 2018, pp. 196–204.

[17] M. Esmaeilpour, P. Cardinal, and A. L. Koerich, "A Robust Approach for Securing Audio Classification Against Adversarial Attacks," *arXiv:1904.10990*, Apr. 2019.

[18] V. Subramanian, E. Benetos, N. Xu, S. McDonald, and M. Sandler, "Adversarial attacks in sound event classification," in *arXiv:1907.02477*, 2019.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations (ICLR)*, Canada, 2014.

[20] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, USA, June 2016, pp. 2574–2582.

[21] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *arXiv:1608.04644*, 2016.

[22] F. Developers, "ffmpeg tool (version beld324)," http://ffmpeg.org, 2016.

[23] J. O. Smith, *Digital Audio Resampling Home Page*. https://ccrma.stanford.edu/ jos/resample/, January 28, 2002.

[24] M. Mauch and S. Ewert, "The audio degradation toolbox and its application to robustness evaluation," in *International Society for Music Information Retrieval Conference (ISMIR)*, Brazil, 2013.

[25] E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," *CoRR*, vol. abs/1807.09902, 2018. [Online]. Available: http://arxiv.org/abs/1807.09902

[26] I.-Y. Jeong and H. Lim, "Audio tagging system for dcase 2018: focusing on label noise, data augmentation and its efficient learning," DCASE2018 Challenge, Tech. Rep., 2018.

[27] T. Iqbal, Q. Kong, M. D. Plumbley, and W. Wang, "General-purpose audio tagging from noisy labels using convolutional neural networks," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, UK, 2018, pp. 212–216.