LEARNING WITH OUT-OF-DISTRIBUTION DATA FOR AUDIO CLASSIFICATION

Turab Iqbal, Yin Cao, Qiuqiang Kong, Mark D. Plumbley, Wenwu Wang

Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK

ABSTRACT

In supervised machine learning, the assumption that training data is labelled correctly is not always satisfied. In this paper, we investigate an instance of labelling error for classification tasks in which the dataset is corrupted with out-of-distribution (OOD) instances: data that does not belong to any of the target classes, but is labelled as such. We show that detecting and relabelling certain OOD instances, rather than discarding them, can have a positive effect on learning. The proposed method uses an auxiliary classifier, trained on data that is known to be in-distribution, for detection and relabelling. The amount of data required for this is shown to be small. Experiments are carried out on the FSDnoisy18k audio dataset, where OOD instances are very prevalent. The proposed method is shown to improve the performance of convolutional neural networks by a significant margin. Comparisons with other noise-robust techniques are similarly encouraging.

Index Terms— Audio classification, out-of-distribution, convolutional neural network, pseudo-labelling

1. INTRODUCTION

Supervised learning refers to the use of labelled training data, the availability of which provides a tremendous advantage in many applications of machine learning. In practice, labels are not always correct [1], prompting additional efforts to carefully verify them. This can be prohibitively costly when scaling to large datasets, which often results in limited data for training. In order to utilise much larger datasets, there has been interest in learning methods that do not rely on clean data [2, 3, 4, 5]. To this end, this paper investigates a case of labelling error for audio classification in which the dataset is corrupted with out-of-distribution (OOD) instances: data that does not belong to any of the target classes, but is labelled as such.

Large amounts of annotated data are available to use when considering the world wide web [6, 7]. However, due to the uncontrolled/miscellaneous nature of these sources of data, irrelevant (OOD) instances are likely to be encountered when curating the data. For example, Freesound Annotator [6] is a platform of datasets comprised of over 260 K audio samples annotated by the public, where the authors of this platform have observed a considerable number of OOD instances [2]. OOD corruption can occur for a number of reasons, such as uncertainty in the sound (e.g. being unable to discriminate between clarinet sounds and flute sounds) and uncertainty in the label semantics (e.g. 'keyboard' could refer to keyboard instruments or it could refer to computer keyboards).

In this paper, it is argued that certain OOD instances, when labelled appropriately, can be beneficial for learning, and that this depends on their likeness to the in-distribution (ID) data. Using a continuous label space, one can even assign 'soft' labels to these instances to reflect uncertainty in what the most appropriate target class is. By considering the new labels as the correct labels, OOD corruption can be framed in terms of label noise [1]; for each instance, a (pseudo-)correct label exists, but the label assigned by the annotator may be incorrect.

Considering the problem in terms of label noise allows the incorporation of methods developed for label noise. In particular, this paper proposes a pseudo-labelling method for the OOD training data. There are two main stages: (1) OOD detection and (2) relabelling. To detect and relabel the relevant instances, an auxiliary classifier trained on a much smaller dataset of manually-verified ID examples is used. The original ground truth of the training data is also exploited. Requiring a small amount of verified data is not unreasonable, as the cost of doing so is relatively low. Convolutional neural networks are used as baselines to assess the proposed method and compare it to alternatives methods. Experiments are carried out on the FSDnoisy18k dataset [2], which is a large audio dataset with a substantial number of OOD training examples.

1.1. Related Work

The OOD detection methods most relevant to our work are those that use a discriminative neural network model to directly detect OOD data. Hendryks and Gimpel [8] proposed using the maximum of the softmax output as a measure of confidence, where a low value indicates that an instance is OOD. Using this idea as a baseline, several improvements have been proposed, including different confidence measures [9] and changes to the learning dynamics during training [10]. Our paper adapts ODIN [9], which applies input pre-processing and logit scaling to influence the softmax output. There are two key differences, however. First, the method is modified to detect OOD instances that are *similar* to the ID data. Second, our paper is concerned with OOD *training* data rather than *test* data. As a result, we also exploit the ground truth labels.

Pseudo-labelling is a method that has been proposed in the past for the noisy label problem, with a number of works using neural network models. Reed et al. [4] proposed a method called *bootstrapping*, which adaptively relabels the training data as a combination of the observed label and the current classifier prediction. Li et al. [5] proposed a similar method, but instead of using predictions during training, they used the prediction of an auxiliary classifier trained on a verified dataset. This is the approach that is adopted in this paper. However, our proposed method also includes a detection stage to only relabel a subset of the training examples.

2. BACKGROUND

A single-label classifier is any function $f : \mathcal{X} \to \mathcal{Y}$ that maps an instance, $x \in \mathcal{X}$, to a label, $y \in \mathcal{Y}$, where $\mathcal{Y} \coloneqq \{1, ..., K\}$ and K is the number of classes. The instance x is said to belong to the class corresponding to y. A supervised learning algorithm is said to train a classifier using *examples* (training data) of the form $(x, y) \in \mathcal{X} \times \mathcal{Y}$ in order to be able to classify future instances (test data). In a standard learning problem, the training set and test set are assumed to be sampled from the same distribution D over $\mathcal{X} \times \mathcal{Y}$ [11]. This is no longer the case when examples are labelled incorrectly. In the problem that we are studying, the training set contains instances for which the marginal probability density function, $p(x)$, of D vanishes. Such instances are known as out-of-distribution instances.

The distribution D can determine whether an instance is OOD, but it cannot describe the OOD data itself nor its relation to the ID data. This is limiting, as OOD instances can possess properties that overlap with the ID data. For example, clarinets and flutes sound similar regardless of whether one of them is OOD. In this sense, OOD instances can be 'near' or 'far' from the ID data. We argue that this information, which is a manifestation of data uncertainty [12], can be considered as further knowledge for a learning algorithm to benefit from. Similar to knowledge distillation [13], we use pseudo-labels to convey this knowledge. The pseudo-labels are generally soft labels such that $y \in \{z \in \mathbb{R}_+^K : \|z\|_1 = 1\}$. This conveys the uncertainty in assigning an OOD instance to any one class.

From another perspective, assigning soft labels to OOD instances has already been shown to improve training. Mixup [14] is a data augmentation technique that linearly combines instances and their labels, such that $\hat{x} = \alpha x_1 + (1 - \alpha)x_2$ and $\hat{y} = \alpha y_1 + (1 - \alpha)y_2$, where y_1 and y_2 are understood to be one-hot vectors. When $y_1 \neq y_2$, $\hat{y}$ does not correspond to any one class, meaning that $\hat{x}$ is OOD [15]. Despite using OOD instances as additional training data, mixup has been shown to be an effective form of data augmentation [14, 15, 16]. This is further motivation for our method. In our problem, however, the pseudo-correct label $\hat{y}$ is *not* known and not just the result of a linear combination. To estimate the pseudo-label, we use two sources of information: the ground truth label and the prediction of an auxiliary classifier.

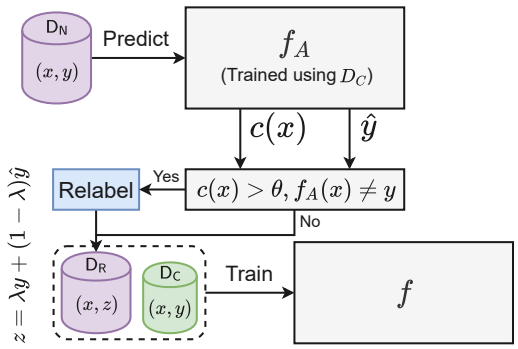

Fig. 1. An illustration of how OOD instances are detected and relabelled using the auxiliary classifier, f_A, and then used to train the primary classifier, f. D_C, D_N, and D_R denote the datasets defined in Section 3.

3. PROPOSED METHOD

To detect and relabel OOD instances, another training set, $\mathsf{D}_C \coloneqq (\mathsf{X}_C, \mathsf{Y}_C)$, consisting of only clean (verified) examples is used. That is, D_C only contains ID instances that are labelled correctly. In contrast, let $\mathsf{D}_N \coloneqq (\mathsf{X}_N, \mathsf{Y}_N)$ denote the dataset consisting of noisy (unverified) examples. The general steps of the proposed method are as follows (cf. Figure 1):

1. Train an auxiliary classifier, f_A, using D_C.
2. Detect the relevant OOD instances in D_N using f_A.
3. Relabel the detected instances using f_A. Let D_R denote the new dataset after relabelling.
4. Train the primary classifier, f, using $\mathsf{D}_C \cup \mathsf{D}_R$.

In the final step, both verified and unverified examples are used to train the classifier, but with the changes made in step 3. The two steps that deserve the most attention are steps 2 and 3, which are developed in the following subsections.

3.1. Out-of-Distribution Detection

As outlined in Section 1.1, there has been previous work on OOD detection based on estimating confidence values for the predictions of the classifier. Instances for which the confidence, $c(x) \in [0, 1]$, of the classifier is lower than a threshold, $\tau \in (0, 1)$, are considered OOD. Although these algorithms have been shown to be effective in various test cases [8], they have been more successful in detecting instances that are *far* from the ID data, while OOD instances that are *near* are typically not detected [9]. This is not appropriate, as our relabelling method relies on the OOD instances to be near.

To rectify this, our proposed detection algorithm exploits the availability of labels in the training data. When there is a *mismatch* between the label and the classifier prediction, i.e. $f_A(x) \neq y$ for a training example (x, y), and the confidence of the classifier is *above* a threshold, i.e. $c(x) > \theta$ for $\theta \in (0, 1)$, the instance x is considered as an OOD instance that should

be relabelled. Denoting S_θ as the set containing these OOD instances, we have

$$S_\theta \coloneqq \{x : c(x) > \theta \text{ and } f_A(x) \neq y\}. \quad (1)$$

Selecting instances for which there is a mismatch indicates that the observed label, y, may be erroneous. By additionally ensuring that $c(x)$ is high, it only includes instances that are believed to be similar to one of the target classes.

One could also incorporate the mismatch condition for low-confidence OOD instances by defining

$$S_\tau \coloneqq \{x : c(x) < \tau \text{ and } f_A(x) \neq y\}. \quad (2)$$

This is intended to detect far-away OOD instances, as with the OOD detection algorithms in previous work [8, 9, 10], but the additional mismatch condition can reduce the number of ID instances that are incorrectly detected, since a mismatch is less likely for ID instances. The instances in S_τ can then be dealt with separately, e.g. by discarding them. This is explored as an additional step in the experiments presented later, but is not the main focus of this paper.

3.2. Pseudo-labelling

As explained in Section 2, OOD instances do not belong to any of the target classes, but they may share properties with one or more of them. As such, we propose to relabel the instances in S_θ as a convex combination of the observed label, y, and the prediction, $\hat{y}$, given by the auxiliary classifier.

$$z = \lambda y + (1 - \lambda)\hat{y}. \quad (3)$$

In (3), y is understood to be a one-hot vector, while $\hat{y}$ is a vector of probabilities, i.e. $\hat{y} \in \{z \in \mathbb{R}_+^K : \|z\|_1 = 1\}$, so that $f_A(x) = \arg\max \hat{y}_i$. The parameter $\lambda \in (0, 1)$ determines the weight to apply to the observed label and the prediction. $\hat{y}$ is used because the auxiliary classifier is confident in the prediction, so it is likely to be an optimal label for learning. On the other hand, there is a chance that it is wrong or suboptimal relative to the observed label. The weight, λ, can be interpreted as the prior belief that this is the case.

This method of relabelling has also been proposed in the past for noisy labels [5], where the authors found that it was an effective approach for several noisy datasets. For the selection of the parameter λ, they used a heuristic that derived the weight as a function of the performance of the auxiliary classifier on clean data and noisy data.

4. EXPERIMENTS

In this section, experimental results are presented to evaluate the proposed method[1]. The experiments were carried out using the FSDnoisy18k dataset [2], which is a crowdsourced audio classification dataset created as part of Freesound Annotator [6]. It contains $18\,532$ audio clips across 20 classes, totalling 42.5 hours of audio. The clip durations range from $300\,\mathrm{ms}$ to $30\,\mathrm{s}$. The dataset is divided into a training set containing $17\,585$ clips and a test set containing 947 clips. The test set has been manually verified so that all of the audio clips are ID. In contrast, only $10\,\%$ of the training set is verified: this is the data that is used to train the auxiliary classifier. It has been estimated [2] that $45\,\%$ of the unverified labels are incorrect, and that $84\,\%$ of the incorrect labels are OOD.

4.1. Baseline System

The evaluated systems are based on two baseline convolutional neural networks (CNNs) [17, 18]. We used two baselines to investigate two different settings: using a randomly-initialised model and using a pre-trained model. The two baselines are *VGG9* (randomly initialised) and *DenseNet-201* (pre-trained). VGG9 is based on VGG [17, 16] and contains 8 convolutional layers and 1 fully-connected layer. DenseNet-201 is DenseNet with 201 layers [18] and was pre-trained using ImageNet [19]. Although ImageNet is an image dataset, we found that it was surprisingly effective for pre-training. The architecture of each baseline was chosen independently based on performance. DenseNet-201 performed better than VGG9 when pre-training with ImageNet, while VGG9 performed better than DenseNet-201 when using randomly-initialised weights.

Features were extracted by converting the audio waveforms into logarithmic mel-spectrograms (log-mels) with a window length of 1024, a hop length of 512, and 64 bands. To ensure the neural networks received fixed-length inputs, we used a block-based approach as used in our previous work [16]. That is, the feature vectors were partitioned into blocks of length 128 and processed independently.

Models were trained using the categorical cross-entropy loss function with the Adam optimization algorithm [20]. Training was carried out for 40 epochs with a batch size of 128 and a learning rate of 0.0005, which was decayed by $10\,\%$ after every two epochs. The primary classifier, f, and the auxiliary classifier, f_A, were trained identically. However, f_A was specifically trained using the pre-trained DenseNet-201 model, regardless of the model used for f.

Unless otherwise stated, all hyperparameter values were selected by evaluating the models with a validation set, which contained 15 manually-verified examples from each class, and was sampled from the training set.

4.2. Evaluated Systems

Several systems were evaluated to assess the performance of the proposed method. Each system applies to both baselines. We evaluated a number of variations of the proposed method (starred below) as well as alternative methods proposed for noise robustness in general. The systems are as follows:

[1] Source code: `https://github.com/tqbl/ood_audio`

- *Clean*: The baseline trained with clean examples only.
- *Clean-DA*: *Clean* with data augmentation. The DenseNet variant of this system was used to train f_A.
- *Baseline*: The baseline trained with all of the examples.
- *OOD-R**: The method proposed in this paper.
- *OOD-RD**: Equivalent to *OOD-R*, except instances in S_τ (c.f. Section 3.1) are discarded.
- *All-R**: All the examples in D_N are relabelled.
- *Bootstrap*: Labels are updated dynamically using the bootstrapping method [4]. No OOD detection.
- $\mathcal{L}_q$ *Loss*: The baseline system with the $\mathcal{L}_q$ loss [21] (with $q = 0.7$) instead of the cross-entropy loss.
- *Noise Layer*: An additional linear layer maps the predictions to the same noisy space as the observed labels [3]. This layer is removed during inference.

The proposed system and its variations were configured with $\tau = 0.4$, $\theta = 0.55$, and $\lambda = 0.5$. The value of λ was selected based on the heuristic given in [5]. The values of θ and τ were selected by applying the detection algorithm on the validation set and selecting the threshold for which less than $5\,\%$ of the instances were (incorrectly) detected. This way, the selection of the thresholds is interpretable and independent of f_A. Instead of using the softmax output directly, ODIN [9] was used to compute $c(x)$; we found that ODIN detected more instances in D_N for a given number of incorrectly detected validation instances. Using ODIN, $25.7\,\%$ (resp. $13.8\,\%$) of D_N was detected as belonging to S_θ (resp. S_τ).

The purpose of *Clean-DA* is to compare data augmentation to using the noisy examples. We experimented with mixup [14] and SpecAugment (time/frequency masking) [22], and adopted the latter as it gave superior performance. The $\mathcal{L}_q$ loss is designed to be robust against incorrectly-labelled data, and is the approach taken by the authors of FSDnoisy18k [2]. The bootstrapping method is similar to the proposed method, except there is no OOD detection and no auxiliary classifier, as examples are relabelled *during* training as a combination of the ground truth and f's current prediction. The noise layer was proposed for class-conditional label noise [3], and was utilised by Singh et al. [23] for FSDnoisy18k.

To score the performance of the systems, average precision (AP) and accuracy were used as metrics. Both were computed as micro-averages and reported in percentages, where a higher percentage means higher performance. Five trials were carried out for each experiment to account for uncertainty.

4.3. Results

The results are presented in Table 1. When training with the clean examples only, data augmentation resulted in a noticeable improvement in performance. However, this improvement can be seen to be relatively small compared to using all of the examples for training. This shows that there is a benefit to training with the noisy examples in this dataset, despite a large percentage of them being OOD/incorrect. Among the alternative methods, the $\mathcal{L}_q$ loss and noise layer both resulted in large improvements, but only for the VGG baseline. Bootstrapping did not improve the performance for either baseline.

Table 1. Experimental results for all systems. AP and accuracy are reported with $68\,\%$ confidence intervals.

	AP		Accuracy	
System	VGG	DenseNet	VGG	DenseNet
Clean	$71.8_{\pm 0.21}$	$72.3_{\pm 0.47}$	$67.5_{\pm 0.23}$	$67.5_{\pm 0.23}$
Clean-DA	$74.6_{\pm 0.39}$	$75.7_{\pm 0.34}$	$66.4_{\pm 0.45}$	$69.4_{\pm 0.24}$
Baseline	$81.6_{\pm 0.55}$	$84.8_{\pm 0.25}$	$74.5_{\pm 0.71}$	$78.0_{\pm 0.36}$
OOD-R*	$\mathbf{86.0_{\pm 0.15}}$	$\mathbf{88.0_{\pm 0.21}}$	$\mathbf{77.9_{\pm 0.14}}$	$\mathbf{81.0_{\pm 0.50}}$
OOD-RD*	$\mathbf{86.1_{\pm 0.37}}$	$87.1_{\pm 0.36}$	$\mathbf{78.0_{\pm 0.56}}$	$80.3_{\pm 0.15}$
All-R*	$84.5_{\pm 0.55}$	$\mathbf{88.1_{\pm 0.19}}$	$77.1_{\pm 0.68}$	$\mathbf{81.4_{\pm 0.27}}$
Bootstrap	$81.2_{\pm 0.36}$	$84.9_{\pm 0.60}$	$74.8_{\pm 0.80}$	$78.7_{\pm 0.46}$
$\mathcal{L}_q$ Loss	$83.3_{\pm 0.48}$	$85.0_{\pm 0.72}$	$76.4_{\pm 0.56}$	$78.6_{\pm 0.64}$
Noise Layer	$83.0_{\pm 0.47}$	$84.7_{\pm 0.82}$	$76.2_{\pm 0.34}$	$78.0_{\pm 0.75}$

The proposed method, *OOD-R*, performed the best, with significant gains seen for both baselines. *OOD-RD*, which also discards examples in S_τ, did not perform better than *OOD-R*; discarding the examples actually worsened the performance for the DenseNet model, possibly due to removing ID examples. These results suggest that the neural networks are robust to far-away OOD instances being present in the training set, and that removing them should not be a priority. *All-R*, which relabels all of the examples in D_N, did not perform as well as *OOD-R* for the VGG model, which demonstrates the importance of the detection stage. On the other hand, there was no discernible difference between *All-R* and *OOD-R* for the DenseNet model. A possible reason is that the pre-trained DenseNet model is more robust to the label noise introduced when relabelling the low-confidence examples.

5. CONCLUSION

In this work, we investigated the problem of learning in the presence of out-of-distribution (OOD) data. We argued that OOD data can possess properties that are characteristic of the target classes, so that appropriately relabelling the instances to reflect this, rather than discarding them, can benefit training. Our proposed method involved training an auxiliary classifier on a smaller verified dataset, and using its predictions, along with the ground truth labels, to detect and relabel the relevant OOD instances. Using convolutional neural network baselines, experiments with the FSDnoisy18k dataset showed that our method substantially improves performance. The results also suggested that OOD instances that are very different from the target classes have little effect on performance when present in the training data. Future work includes investigating other detection and pseudo-labelling methods, including those that do not require any verified data.

6. REFERENCES

[1] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, 2014.

[2] 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 *International Conference on Acoustics, Speech and Signal Processing*, Brighton, UK, 2019, pp. 21–25.

[3] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," in *International Conference on Learning Representations*, San Diego, CA, 2015.

[4] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *International Conference on Learning Representations*, San Diego, CA, 2015.

[5] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L. Li, "Learning from noisy labels with distillation," in *International Conference on Computer Vision*, Venice, Italy, 2017, pp. 1928–1936.

[6] 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*, Suzhou, China, 2017, pp. 486–493.

[7] 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 *International Conference on Acoustics, Speech and Signal Processing*, New Orleans, LA, 2017, pp. 776–780.

[8] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations*, Toulon, France, 2017.

[9] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations*, Vancouver, Canada, 2018.

[10] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[11] B. van Rooyen and R. C. Williamson, "A theory of learning with corrupted labels," *Journal of Machine Learning Research*, vol. 18, no. 228, pp. 1–50, 2017.

[12] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems*, Montréal, Canada, 2018, pp. 7047–7058.

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[14] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, Vancouver, Canada, 2018.

[15] H. Guo, Y. Mao, and R. Zhang, "MixUp as locally linear out-of-manifold regularization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, HI, 2019, vol. 33, pp. 3714–3722.

[16] 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 2018 Workshop*, Woking, UK, 2018, pp. 212–216.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, San Diego, CA, 2015.

[18] G. Huang, Z. Liu, L. v. der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, 2017, pp. 2261–2269.

[19] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 248–255.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, San Diego, CA, 2015.

[21] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in Neural Information Processing Systems*, Montréal, Canada, 2018, pp. 8778–8788.

[22] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proceedings of Interspeech*, Graz, Austria, 2019, pp. 2613–2617.

[23] S. Singh, A. Pankajakshan, and E. Benetos, "Audio tagging using linear noise modelling layer," in *Detection and Classification of Acoustic Scenes and Events 2019 Workshop*, New York, NY, 2019, pp. 234–238.