UNSUPERVISED CONTRASTIVE LEARNING OF SOUND EVENT REPRESENTATIONS

Eduardo Fonseca[1*], *Diego Ortego*[2*], *Kevin McGuinness*[2], *Noel E. O'Connor*[2], *Xavier Serra*[1]

[1]Music Technology Group, Universitat Pompeu Fabra, Barcelona {eduardo.fonseca}@upf.edu
[2] Insight Centre for Data Analytics, Dublin City University (DCU) {diego.ortego}@insight-centre.org

ABSTRACT

Self-supervised representation learning can mitigate the limitations in recognition tasks with few manually labeled data but abundant unlabeled data—a common scenario in sound event research. In this work, we explore unsupervised contrastive learning as a way to learn sound event representations. To this end, we propose to use the pretext task of contrasting differently augmented views of sound events. The views are computed primarily via mixing of training examples with unrelated backgrounds, followed by other data augmentations. We analyze the main components of our method via ablation experiments. We evaluate the learned representations using linear evaluation, and in two in-domain downstream sound event classification tasks, namely, using limited manually labeled data, and using noisy labeled data. Our results suggest that unsupervised contrastive pre-training can mitigate the impact of data scarcity and increase robustness against noisy labels, outperforming supervised baselines.

Index Terms— Contrastive learning, sound event classification, audio representation learning, self-supervision

1. INTRODUCTION

Sound event recognition (SER) has been traditionally framed as a supervised learning problem, that is, relying on annotated datasets. The two largest labeled SER datasets, AudioSet [1] and the recently released FSD50K [2], are instrumental resources for SER, yet they have several shortcomings. AudioSet provides a massive amount of content but the official release does not include waveforms, and the labelling in some classes is less precise.[1] By contrast, FSD50K consists of open-licensed audio curated with a more thorough labeling process, but the data amount is more limited. Common to both datasets is the tremendous effort needed to collect the human annotations.

Alternatives to conventional supervised learning include the paradigms of semi-supervised learning [3], few-shot learning [4], learning from noisy labels [5, 6], or self-supervision [7]. Among these, self-supervision is appealing as it is the only one able to leverage large amounts of unlabeled data without external supervision. Sources of unlabeled data for SER include websites such as Flickr or Freesound, which host substantial amounts of open-licensed audio(visual) material with high diversity of everyday sounds. While content in these websites is typically accompanied by metadata, it is usually too sparse for a meaningful mapping to a sound event label set (in the case of Flickr [8]), or sometimes the user-provided metadata can be an underrepresentation of the actual acoustic content (in the case of a portion of Freesound). In these cases, self-supervision allows leveraging this unlabeled audio content in order to learn useful audio representations without prior manual labelling or metadata.

Self-supervised learning is a learning paradigm in which representations are learned by training networks on *auxiliary* or *pretext* tasks that do not require explicit labels. The central idea is that by solving the pretext task, the network is able to learn useful low-dimensional representations from large amounts of unlabeled data. These representations can then be used for downstream tasks, for example where only few data, or poorly labeled data, are available.

Self-supervised learning has shown great promise in computer vision [9, 10], and interesting results in the audio domain [7, 11]. One of the first works in self-supervised sound event representation learning is [7], adopting a triplet loss-based training by creating anchor-positive pairs via simple audio transformations, e.g., adding noise or mixing examples. In [12], the pretext task consists of predicting the long-term temporal structure of continuous recordings captured with an acoustic sensor network. Recently, [11] proposes two pretext tasks, namely, estimating the time distance between pairs of audio segments, and reconstructing a spectrogram patch from past and future patches. In the last few years, self-supervised learning methods using *contrastive* losses have gained increasing attention, not only for images [9, 10], but also for speech [13, 14, 15], and sound events [16]. In the context of contrastive learning, a recent trend is to learn representations by contrasting different versions or *views* of the same data example, computed via data augmentation. After achieving state-of-the-art in image recognition [9], this approach has been successfully applied for speech recognition [14, 15].

In this paper, we propose to learn sound event representations using the pretext task of contrasting differently augmented views of sound events. The different views are computed primarily via mixing of training examples with unrelated background examples, followed by other data augmentations. We evaluate the learned representations using linear evaluation, and in two downstream sound event classification tasks (in the same domain as the pretext task), namely, using limited manually annotated data, and using noisy labeled data. Our results suggest that unsupervised contrastive pre-training can mitigate the impact of data scarcity and increase robustness against noisy labels. Specifically, when training a linear classifier on top of the pre-trained embedding, we recover most of the supervised performance. In addition, fine-tuning a model initialized with the pre-trained weights outperforms supervised baselines. These results show that our method is able to learn useful audio representations even with more limited resources (data and compute) than other previous works (e.g., [7, 12, 11]). To our knowledge, this is the first work conducting contrastive sound event representation learning by maximizing the similarity between differently augmented views. Our contributions are *i)* a new framework for learning sound event representations via data-augmenting contrastive learning, and *ii)* an empirical evaluation of the proposed method through extensive experiments. Code is available.[2]

---

*Equal contribution.
[1]https://research.google.com/audioset/dataset/index.html
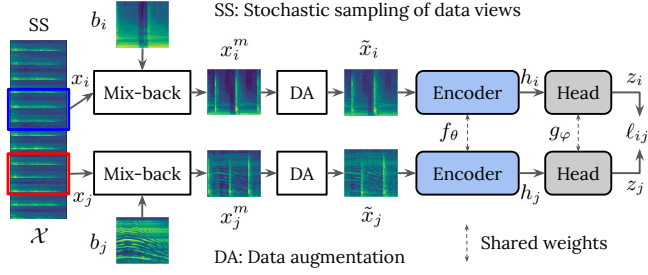
[2]https://github.com/edufonseca/uclser20

Fig. 1. Contrastive learning of sound event representations overview.

2. METHOD

We seek learning sound event representations from unlabeled data using self-supervision, i.e., exploiting a pretext task whose *labels* are inferred from an unlabeled dataset. To do so, we create pairs of correlated views (denoted as positive examples) via different augmentations of a single sound event example. Then, their corresponding embedding representations are compared using a contrastive loss [9, 17, 18] that pulls together representations of positive examples, while pushing apart those of negative ones (i.e., unrelated examples). Our hypothesis is that discriminative sound event representations can emerge by solving this task. Figure 1 illustrates the main components of the proposed method, which is inspired by the recent SimCLR [9]. Next we explain the main components. Implementation details and hyper-parameter choices can be inspected in the released code.[2]

Stochastic sampling of data views. The incoming training examples to our framework are log-mel spectrograms of audio clips. From each training example, $\mathcal{X}$, we sample two views, which we call time-frequency (TF) *patches*. These patches, $x_i \in \mathcal{X}$ and $x_j \in \mathcal{X}$ are selected randomly over the length of the clip spectrogam. Sec. 4.1 analyzes the benefits of this stochastic sampling over other alternatives.

Mix-back. The first operation that we apply to each incoming patch is what we call *mix-back*. It consists in *mix*ing the incoming patch x_i with a *back*ground patch, b_i, as follows:

$$x_i^m = (1 - \lambda)\, x_i + \lambda \left[E\left(x_i\right) / E\left(b_i\right) \right] b_i, \qquad (1)$$

where $\lambda \sim \mathcal{U}\left(0, \alpha\right)$, $\mathcal{U}$ is a uniform distribution, $\alpha \in [0,1]$ is the mixing hyper-parameter (typically small), and $E\left(\cdot\right)$ denotes the energy of a given patch. A similar approach is used in [7] to create examples for triplet loss-based training. The energy adjustment of Eq. 1 ensures that x_i is always dominant over b_i, even if $E\left(b_i\right) >> E\left(x_i\right)$, thereby preventing aggressive transformations that may make the pretext task too difficult. Before Eq. 1, patches are transformed to linear scale (inversion of the log in the log-mel) to allow energy-wise compensation, after which mix-back is applied, and then the output, x_i^m, is transformed back to log scale. Background patches b are randomly drawn from the training set (excluding the input clip $\mathcal{X}$), hence they are out-of-batch in the vast majority of cases. Recent work [19] shows that useful representations arise if data views share as little information as possible, while preserving relevant semantic information that keeps the predictive power for related downstream tasks. This is our motivation to use mix-back: *i)* shared information across positives is decreased by mixing x_i and x_j with different backgrounds, and *ii)* semantic information is preserved due to sound transparency (i.e., a mixture of two sound events inherits the classes of the constituents) and the fact that the positive patch is always

predominant in the mixture. Mix-back can be understood as a data augmentation, but we separate it from the others as it involves two input patches.

Stochastic Data Augmentation. We adopt data augmentation techniques (DAs) directly computable over TF patches (rather than waveforms), and that are simple for on-the-fly computation, thus favouring speed rather than acoustical/mathematical correctness. We consider DAs both from computer vision and audio literature: random resized cropping (RRC), random time/frequency shifts, compression, specAugment [20], Gaussian noise addition, and Gaussian blurring. All augmentations are stochastic as their hyper-parameters are randomly sampled from a distribution for each patch. The final augmentation policy adopted consists of sequentially applying RRC, compression and Gaussian noise addition (see Sec. 4.3 for other policies). These DAs transform x_i^m into the input patch $\tilde{x}_i$ for the encoder network.

Encoder Network. We use a CNN based network f_θ to extract the embedding $h_i = f_\theta\left(\tilde{x}_i\right)$ from the augmented patch $\tilde{x}_i$, where h_i is the embedding right before the final fully-connected classification layer and θ are its parameters. Once the contrastive learning is over and the encoder is trained, the representation h_i can be used for downstream tasks.

Projection Head. Following [9], a simple projection network g_φ with parameters φ maps h_i to the final L2-normalized low-dimensional representation z_i where the contrastive loss is applied. Our head consists of an MLP with one hidden layer, batch-normalization, and a ReLU non-linearity. Note that the projection head is only used during contrastive learning, i.e., once the training is over, only the trained encoder is used for downstream tasks.

Contrastive Loss. The contrastive loss adopted for a positive pair of examples, x_i and x_j, is the *NT-Xent* loss [9]:

$$\ell_{ij} = -\log \frac{\exp\left(z_i \cdot z_j / \tau\right)}{\sum_{v=1}^{2N} \mathbb{1}_{v \neq i} \exp\left(z_i \cdot z_v / \tau\right)}, \qquad (2)$$

where z_j is the representation for the patch x_j, $\tau > 0$ is a temperature scaling, $\mathbb{1}_{v \neq i} \in \{0, 1\}$ is an indicator function that returns 1 if $v \neq i$, and N is the batch size. Note that the generation of two views for each example extends the mini-batch from N to $2N$. Therefore, every patch has a single positive pair and $2N - 2$ negative pairs. Minimizing Eq. 2 during training adjusts the parameters θ and φ to maximize the numerator (i.e., maximize agreement between positives) while simultaneously minimizing the denominator, thus forcing similar views to neighboring representations and dissimilar ones to non-neighboring ones. The DA operations over each view of one example are two different instantiations of the same family of transformations. However, the embeddings z_i and z_j are obtained with a single instantiation of the encoder and projection head (see *shared weights* in Figure 1).

3. EXPERIMENTAL SETUP

3.1. Dataset

We use the FSDnoisy18k dataset [21], containing 42.5h of Freesound audio distributed across 20 classes drawn from the AudioSet Ontology [1]. The dataset includes a small *clean* training set (1,772 clips / 2.4h), a larger *noisy* train set (15,813 clips / 38.8h), and a test set (947

clips/ 1.4h). Labels in the clean and test sets are manually-labelled with *Freesound Annotator* [22], whereas labels in the noisy set are inferred automatically from metadata, hence featuring real-world label noise. The dataset is singly- and weakly-labeled, and clips are of variable-length in range [0.3, 30]s. We avoid larger vocabulary datasets, e.g., AudioSet [1], due to the computationally intensive contrastive learning experiments conducted—intractable under our compute resources. FSDnoisy18k has a smaller vocabulary, while featuring a relatively large amount of per-class training data compared to other datasets ([23, 24, 25, 21, 26]). The limited dataset scope implies, however, that the learned representations are not transferable to unrelated datasets or downstream tasks, forcing to conduct an *in-domain* evaluation. FSDnoisy18k allows evaluation of learned representations on two real-world scenarios: using a small clean set, and a larger noisy set.

3.2. Learning Pipeline

Our pipeline consists of two stages: *i)* unsupervised contrastive learning of a low-dimensional representation (Sec. 4), and *ii)* evaluation of the representation using supervised tasks (Sec. 5). In both stages, incoming audio is transformed to 96-band log-mel spectrograms, and to deal with variable-length clips, we use TF patches of 1s (shorter clips are replicated; longer clips are trimmed in several patches). We use three networks: ResNet-18 [27], and a VGG-like and a CRNN similar to those in [2], commonly used for SER tasks (details can be inspected in the code[2]). Models are always trained using SGD with momentum 0.9 and weight decay 10^{-4}, using a batch size of 128 and shuffling examples between epochs. We always train on the noisy set and validate on the clean set, except in one of the downstream tasks in Sec. 5.2. For the contrastive learning experiments (Sec. 4), we follow the approach of Sec. 2. Models are trained for 500 epochs, with initial learning rate of 0.03, divided by 10 in epochs 325 and 425. For the supervised learning experiments (Sec. 5), during training we randomly sample a single patch when clips are longer than 1s. Models are trained for 200 epochs to minimize categorical cross-entropy, reducing the learning rate in epochs 80 and 160. The initial learning rate is 0.1 when training from scratch, and 0.01 in case of using unsupervised pre-trained weights in order to constrain the learning process.

3.3. Evaluation

To quantify the quality of the learned representations, we follow three approaches: a variation of the standard k-Nearest Neighbour (**kNN**) **evaluation** in [28], and the standard approaches of using **linear probes** [9] and **fine-tuning a model end-to-end**. For the **kNN evaluation**, we estimate the representation z for each validation or test patch and compare it against every other patch in the given set via cosine similarity. The prediction for every patch is, then, obtained by majority voting rule across the k neighbouring labels, where $k = 200$ as in [28, 29]. In turn, clip-level predictions are obtained by majority voting of patch-level ones. This evaluation is used for fast contrastive learning experimentation as no additional training is involved. Unlike kNN evaluation, **linear probes** and **end-to-end fine-tuning** procedures involve further training and passing patches through an entire model to produce prediction probabilities. The former involves training an additional linear classifier on top of the pre-trained unsupervised embedding, whereas the latter fine-tunes the model on a given downstream task after initializing it with the pre-trained weights. For both procedures, patch-level predictions are averaged per-class across all patches in a clip to obtain clip-

Table 1. kNN val accuracy for several ways of sampling TF patches.

Sampling method	kNN	Sampling method	kNN
Sampling at random	**70.1**	$d = 125$	67.9
$d = 0$ (same patch)	51.1	$d = 200$	69.9
$d = 25$	61.5	$d = 300$	68.5
$d = 75$	65.1	$d = 400$	69.7

level predictions. Common to all evaluation methods, once clip-level predictions are gathered for a given set, overall accuracies are computed on a per-class basis, then averaged with equal weight across all classes to yield the performance shown in Secs. 4 and 5. For linear probes and end-to-end fine-tuning, we report test accuracy provided by the best validation accuracy model. However, learning curves for the contrastive learning experiments using kNN eval were found to be relatively noisy, such that *best* accuracy is not always representative of the overall quality of the training process. Therefore, we decided to use the *average* validation accuracy across the last 50 epochs, as top performing model checkpoints appear at the end of the training. Finally, results by the three methods above are compared against supervised baselines, where metrics are computed following the same procedure as in end-to-end fine-tuning. Hereafter, we shall refer to validation as *val*.

4. ABLATION STUDY

We report ablation experiments to study some blocks of our proposed method. For each block, we always start from the best configuration.

4.1. Sampling TF Patches

Table 1 shows results when patches are randomly sampled along the clip, and when patches are sampled deterministically, separated by a sampling distance d (time frames). Stochastically sampling patches provides superior performance. We study the impact of progressively increasing d (might be bounded due to clip length) between a first randomly sampled patch x_i and a second patch x_j. The higher the distance, the better the representation learned. In particular, for $d < 101$ both patches overlap and these cases underperform no overlapping ones, where performance saturates. This aligns with recent results in computer vision, where increasing the distance between image crops is beneficial only up to some values, after which performance decreases due to little semantic content shared between views [19].

4.2. Mix-back

Table 2 (left side) shows results for mix-back for best $\alpha \in \{0.02, 0.05, 0.1, 0.2\}$. It can be seen that using mix-back helps considerably, and adjusting the energy is also beneficial. The latter means that the foreground patch is always dominant over the background patch, thus preventing potentially aggressive transforms. The optimal α values are small, which indicates that a light mixture is preferred. We observe that lightly adding background from another patch (i.e., mix-back) is much more beneficial than adding artificial white noise (see first and second rows from Table 2 (right side)). These results suggest that mixing with natural backgrounds from real audio signals is suitable for contrastive representation learning.

4.3. Data Augmentation

Table 2 (right side) lists results for several DA policies. Each row represents the best result after sweeping the corresponding DA parameters. We started by exploring DAs applied individually (from bottom to top). The top DA (applied individually) is random resized

Table 2. kNN val accuracy for several mix-back and data augmentation (DA) settings.

Mix-back setting (α)	kNN	DA policy	kNN
w/ E adjustment (0.05)	**70.1**	RRC + comp + noise	**70.1**
w/o E adjustment (0.02)	66.2	RRC + comp	69.6
w/o mix-back	63.3	RRC + specAugment	70.0
		RRC	69.0
		specAugment [20]	68.0
		w/o DA	60.1

Table 3. Test accuracy for linear probes evaluation (second column), and for two downstream sound event classification tasks: a larger noisy set and a small clean set for training. *This is also the supervised baseline to compare with linear evaluation. p-t = pre-trained.

Model	Linear	Larger noisy set		Small clean set	
(weights in M)	-	random*	p-t	random	p-t
ResNet-18 (11)	**74.3**	65.4	**78.2**	56.5	**77.9**
VGG-like (0.3)	70.0	70.6	**72.8**	61.1	**72.3**
CRNN (1)	64.4	72.0	**74.2**	58.7	**69.1**

cropping (RRC). In particular, the optimal RRC found applies a mild cropping (instead of a harsh one), which can be seen as a small stretch in time and frequency (which also involves a small frequency transposition). This RRC slightly outperforms specAugment [20], which has been successfully used for contrastive learning of speech embeddings [14], and which also works well for sound events. Then, we explored DA compositions based on the mentioned RRC. We found that, to a lesser extent, compression (via spectrogram image contrast adjustment) and Gaussian noise addition also improve the learned representation. We adopt this DA policy for all experiments reported in Secs. 4 and 5 (top row in Table 2). A subsequent more thorough DAs exploration revealed promising results by composing RRC and specAugment, yielding almost top results (70.0). It seems possible that a more exhaustive (and costly) exploration of the DA compositions may lead to better results, e.g., complementing RRC and specAugment with other soft DAs such as compression or Gaussian noise addition. Nonetheless, this should be done carefully. For example, in our experiments we have seen that the ordering of the DAs matter, and joining individually-tuned DAs can be suboptimal as different DAs in a composition affect each other.

4.4. Encoder and Temperature

For encoder architectures, we explore ResNet-18, VGG-like and CRNN, obtaining 70.1, 67.7 and 67.1 val kNN accuracies, respectively. This results show that higher capacity (ResNet-18) is better for contrastive learning, which accords with [9]. We also experiment with the temperature τ of the contrastive loss function, which has been found to be a relevant parameter [10]. In particular we sweep $\tau \in \{0.1, 0.2, 0.3, 0.4\}$, obtaining the kNN accuracy of 68.9, 70.1, 68.9 and 67, respectively. Results show the framework's sensitivity to τ, which we also find to be highly dependent on the projection head configuration.

4.5. Discussion

In general, we observe that the framework is sensitive to hyperparameter changes, and that the various settings of each block affect each other, thus requiring extensive experimentation for appropriate tuning. From Secs. 4.1 and 4.2 we observe that overlapping patches when drawing positives is detrimental, and lightly adding natural backgrounds is beneficial. This could indicate that the original positive examples sometimes share time-frequency patterns that could be used to lower the loss of Eq. 2, but that hinder the learning of useful representations. These undesired patterns are denoted as *shortcuts* in computer vision [30]. Examples of shortcuts in audio self-supervised learning include e.g., recording gear, room acoustics or background noise. FSDnoisy18k is based on Freesound audio, which in turn is composed of audio contributed by users. Clips coming from the same user are likely to share some of these patterns, which has been shown to have an impact in supervised sound event tagging [2]. We hypothesize that this could be a source of shortcuts in our setting, which is being mitigated by stochastic sampling of positive patches

and mix-back. We intend to further investigate this in future work. Finally, previous works report advantages of using larger batch sizes as a way to provide more negative examples, which facilitates convergence (e.g., sizes up to 8k are used in [9]). Our compute resources do not support such large sizes, hence all our experiments are conducted with a batch size of 128 (commonly used for supervised learning). It is conceivable that using larger batches yields even better results than those reported here.

5. EVALUATION OF LEARNED REPRESENTATIONS

5.1. Baseline Systems and Linear Evaluation

Table 3 presents the test accuracies for the linear probes evaluation and the supervised baselines (second and third columns). The supervised CRNN and VGG-like perform similarly, while ResNet-18 performs worse, similarly as in [2] with FSD50K. In our case, this could be due to the capacity of the model (the largest by far), which may lead to overfitting of the smaller dataset (and the noisy labels). In linear evaluation of the contrastive weights, however, ResNet-18 is the top performing system, which accords with the kNN evaluation (Sec. 4.4) and with findings in [9]. Thus, with ResNet-18 the supervised baseline is exceeded by a considerable margin, whereas with VGG-like and CRNN most of the supervised performance is recovered (99% and 89%, respectively).

5.2. In-Domain Downstream Prediction Tasks

We conduct an in-domain evaluation (see Sec. 3.1), similarly to [12], for the two downstream sound event classification tasks posed by FSDnoisy18k: training on the larger set of noisy labels, and on the small set of clean data (in this case, 15% of the clean set is kept for validation). For each task, we compare *i)* a supervised baseline trained from scratch, with *ii)* fine-tuning the network initialized with unsupervised pre-trained weights—these correspond respectively to the columns *random* and *p-t* (pre-trained) in Table 3. These experiments aim at measuring benefits with respect to training from scratch in noisy- and small-data regimes. Table 3 shows that unsupervised contrastive pre-training brings great benefits, achieving better results than training from scratch in both tasks and across all network architectures considered. For both tasks, using ResNet-18 yields top accuracy in the pre-trained setup, and the lowest accuracy when trained from scratch. This suggests that the performance attainable with ResNet-18 supervised from scratch is limited, potentially by limited data and/or label quality. In contrast, unsupervised contrastive pre-training seems to alleviate these problems, leveraging ResNet-18's capacity and yielding superior performance. Greater improvements are observed in the "smaller clean" task, where, interestingly, the pre-trained performance shows little degradation with respect to that of the "larger noisy" task (despite having far fewer examples, see Sec. 3). Specifically, for ResNet-18, pre-trained performance decreases from 78.2 to 77.9, while training from scratch yields a substantial accuracy drop (65.4 to

56.5). A possible explanation for the similar pre-trained performance across tasks may be that, in the "smaller clean" task, the pre-trained model is fine-tuned with unseen clean data (albeit small). However, in the "larger noisy" task, the model is fine-tuned with the same data previously used for unsupervised contrastive learning (and the supervision provided is affected by label noise).

6. CONCLUSION

We present a framework for unsupervised contrastive learning of sound event representations, based on maximizing the similarity between differently augmented views of the same log-mel spectrogram. Via ablation experiments, we show that appropriately tuning the compound of positive patch sampling, mix-back, and data augmentation is vital for successful representation learning. The evaluation on in-domain sound event classification tasks suggests that unsupervised contrastive pre-training can mitigate the impact of data scarcity, and increase robustness against noisy labels as recently found in supervised image classification [31]. Future work includes exploring time-domain DAs, and using larger-vocabulary datasets to learn general-purpose representations transferable to out-of-domain tasks.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] 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 *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2017.

[2] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: an Open Dataset of Human-Labeled Sound Events," *arXiv preprint arXiv:2010.00475*, 2020.

[3] B. Elizalde, A. Shah, S. Dalmia, M.H. Lee, R. Badlani, A. Kumar, B. Raj, and I. Lane, "An approach for self-training audio event detectors using web data," in *Eur. Signal Process. Conf. (EUSIPCO)*, 2017.

[4] K.-H. Cheng, S.-Y. Chou, and Y.-H. Yang, "Multi-label few-shot learning for sound event recognition," in *Int. Workshop on Multim. Signal Process. (MMSP)*, 2019.

[5] E. Fonseca, S. Hershey, M. Plakal, D. P. W. Ellis, A. Jansen, and R. C. Moore, "Addressing missing labels in large-scale sound event recognition using a teacher-student framework with loss masking," *IEEE Signal Process. Lett.*, 2020.

[6] E. Fonseca, F. Font, and X. Serra, "Model-agnostic Approaches to Handling Noisy Labels When Training Sound Event Classifiers," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019.

[7] A. Jansen, M. Plakal, R. Pandya, D. P. W. Ellis, S. Hershey, J. Liu, R. C. Moore, and R. A. Saurous, "Unsupervised learning of semantic audio representations," in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2018.

[8] E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, and X. Serra, "Audio tagging with noisy labels and minimal supervision," in *Workshop on Detect. and Classif. of Acoust. Scenes and Events (DCASE)*, 2019.

[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *Int. Conf. on Mach. Learn. (ICML)*, 2020.

[10] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint: 2003.04297*, 2020.

[11] M. Tagliasacchi, B. Gfeller, F. de Chaumont Quitry, and D. Roblek, "Pre-training audio representations with self-supervision," *IEEE Signal Process. Lett.*, 2020.

[12] 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 (WASPAA)*, 2019.

[13] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint: 1807.03748*, 2018.

[14] A. Nandan and J. Vepa, "Language agnostic speech embeddings for emotion classification," 2020.

[15] E. Kharitonov, M. Rivière, G. Synnaeve, L. Wolf, P.-E. Mazaré, M. Douze, and E. Dupoux, "Data augmenting contrastive learning of speech representations in the time domain," *arXiv preprint: 2007.00991*, 2020.

[16] K. Shimada, Y. Koyama, and A. Inoue, "Metric learning with background noise class for few-shot detection of rare sound events," in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2020.

[17] A. Nandan and J. Vepa, "Language Agnostic Speech Embeddings for Emotion Classification," in *Int. Conf. on Mach. Learn. Workshops (ICMLW)*, 2020.

[18] T. Afouras, A. Owens, J.S. Chung, A. Zisserman, "Self-Supervised Learning of Audio-Visual Objects from Video," in *Eur. Conf. on Comput. Vision (ECCV)*, 2020.

[19] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What Makes for Good Views for Contrastive Learning?," *arXiv preprint: 2005.10243*, 2020.

[20] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *INTERSPEECH*, 2019.

[21] 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 *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2019.

[22] 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 *Int. Society for Music Inf. Retrieval Conf. (ISMIR 2017)*, 2017.

[23] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM International Conference on Multimedia*, 2014.

[24] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *ACM International Conference on Multimedia*, 2015.

[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," in *Workshop on Detect. and Classif. of Acoust. Scenes and Events (DCASE)*, 2018.

[26] M. Cartwright, A. E. Mendez, J. Cramer, V. Lostanlen, G. Dove, H.-H. Wu, J. Salamon, O. Nov, and J. Bello, "SONYC Urban Sound Tagging (SONYC-UST): A Multilabel Dataset from an Urban Acoustic Sensor Network," in *Workshop on Detect. and Classif. of Acoust. Scenes and Events (DCASE)*, 2019.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conf. on Comp. Vision and Pattern Recogn. (CVPR)*, 2016.

[28] Z. Wu, Y. Xiong, S.X. Yu, and D. Lin, "Unsupervised Feature Learning via Non-Parametric Instance Discrimination," in *IEEE Conf. on Comp. Vision and Pattern Recogn. (CVPR)*, 2018.

[29] J. Huang, Q. Dong, S. Gong, X. Zhu, "Unsupervised Deep Learning by Neighbourhood Discovery," in *Int. Conf. on Mach. Learn. (ICML)*, 2019.

[30] M. Minderer, O. Bachem, N. Houlsby, and M. Tschannen, "Automatic shortcut removal for self-supervised representation learning," *arXiv preprint: 2002.08822*, 2020.

[31] D. Hendrycks, K. Lee, and M. Mazeika, "Using Pre-Training Can Improve Model Robustness and Uncertainty," in *Int. Conf. on Mach. Learn. (ICML)*, 2019.