Pattern Recognition Letters 164 (2022) 40–45

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

An Open-Set Recognition and Few-Shot Learning Dataset for Audio Event Classification in Domestic Environments

Javier Naranjo-Alcazar[a,b], Sergi Perez-Castanos[a], Pedro Zuccarello[a], Ana M. Torres[c], Jose J. Lopez[d], Francesc J. Ferri[b], Maximo Cobos[b,*]

[a] Visualfy, Benisanó 46181, Spain
[b] Computer Science Department, Universitat de València, Burjassot 46100, Spain
[c] Dpt. Ingeniería eléctrica, electrónica, automática y comunicaciones, Universidad de Castilla-La Mancha, Cuenca 16002, Spain
[d] iTEAM Institute, Universitat Politècnica de València, Valencia 46022, Spain

ARTICLE INFO

Article history:
Received 16 April 2022
Revised 6 July 2022
Accepted 19 October 2022
Available online 22 October 2022

Edited by Maria De Marsico

Keywords:
Audio Dataset
Classification
Few-Shot Learning
Machine Listening
Open-set Recognition
Sound Processing

ABSTRACT

The problem of training with a small set of positive samples is known as few-shot learning (FSL). It is widely known that traditional deep learning algorithms usually show very good performance when trained with large datasets. However, in many applications, it is not possible to obtain such a high number of samples. This paper deals with the application of FSL to the detection of specific and intentional acoustic events given by different types of sound alarms, such as door bells or fire alarms, using a limited number of samples. These sounds typically occur in domestic environments where many events corresponding to a wide variety of sound classes take place. Therefore, the detection of such alarms in a practical scenario can be considered an open-set recognition (OSR) problem. To address the lack of a dedicated public dataset for audio FSL, researchers usually make modifications on other available datasets. This paper is aimed at providing the audio recognition community with a carefully annotated dataset[1] for FSL in an OSR context comprised of 1360 clips from 34 classes divided into *pattern sounds* and *unwanted sounds*. To facilitate and promote research on this area, results with state-of-the-art baseline systems based on transfer learning are also presented.

1. Introduction

[1] The automatic classification of audio clips is a research area that has grown significantly very recently [1–3]. The research interest in these algorithms is motivated by their numerous applications, such as audio-based surveillance, hearing aids, home assistants or ambient assisted living, among others. In contrast to most deep learning methods, few-shot learning (FSL) tackles the problem of learning with few samples per class. FSL approaches gained focus when trying to address intra-class classification in the context of face recognition problems [4], including applications such as access control and identity verification [5–7]. In order to tackle this problem, loss functions such as ring loss [8] or center loss [9] have been proposed, together with different embeddings from network architectures such as siamese [10,11] and triplet [12,13]. These loss functions are aimed at solving convergence issues, which also require careful training procedures to appropriately choose the pairs or triplets used. Another practical issue arising in many real-world intelligent audio applications is open-set recognition (OSR) [14]. This problem occurs when a system has to face unfamiliar situations for which it has not been trained. A system prepared for OSR should be capable of correctly classifying examples corresponding to classes seen during the training stage while rejecting examples corresponding to new, previously unseen classes. OSR has been addressed in the past by applying modifications to classical machine learning algorithms such as support vector machines [15,16] or nearest neighbour classification [17]. In the last years, deep learning solutions for OSR have also started to emerge, such as OpenMax [18], deep open classifier (DOC) [19] or competitive overcomplete output layer (COOL) [20].

The problems of FSL and OSR appear frequently in smart acoustic applications. For example, a given user may be exposed to several alerts or beeps at home, emitted by different domestic ap-

* Corresponding author.:
E-mail addresses: javier.naranjo@visualfy.com, janal2@alumni.uv.es (J. Naranjo-Alcazar), sergi.perez@visualfy.com (S. Perez-Castanos), pedro.zuccarello@visualfy.com (P. Zuccarello), ana.torres@uclm.es (A.M. Torres), jjlopez@dcom.upv.es (J.J. Lopez), francesc.ferri@uv.es (F.J. Ferri), maximo.cobos@uv.es (M. Cobos).

[1] https://zenodo.org/record/3689288.

https://doi.org/10.1016/j.patrec.2022.10.019

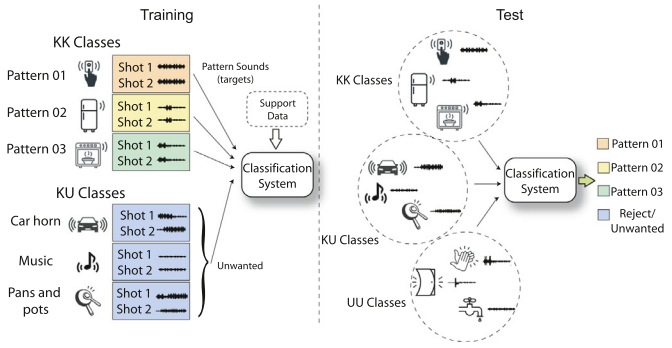

Fig. 1. Training and test under FSL and OSR conditions. In the training stage (left), only a few examples (shots) are available for each class, where some classes are targets to be recognized (KK classes) and others are unwanted classes to be rejected (KU classes). In the test stage (right) the system receives as input examples from the target classes and also from unwanted ones, where new classes different from the ones seen during training (UU classes) are also present.

pliances (e.g. oven, refrigerator). A smart system to differentiate between both alerts should not classify both sounds into a single "alarm" class, but should be capable of identifying correctly those specific *pattern sounds*. However, only a limited number of examples recorded by the user may be available for training. In addition, the system should neglect or discard the variety of possible sounds appearing in a domestic environment. Therefore, there is a need to design machine learning systems trained with a small number of audio examples capable of both identifying the classes of interest (FSL) while rejecting the sounds coming from other unexpected sources (OSR).

A diagram of the conditions under which training and testing are performed within a FSL+OSR context is shown in Fig. 1. The FSL condition is reflected by the small number of examples (shots) available during the training stage. On the other hand, the OSR condition is accounted by letting the system learn from examples corresponding to unwanted (non-target) classes. Since the number of examples is clearly insufficient, usually some meta-learning strategy and support data is needed to let the system learn to discriminate among data and exploit better the information provided by the available shots. In the test stage, the system is confronted towards examples pertaining either to target classes or to unwanted ones. Such unwanted examples might belong to the group of unwanted classes seen during the training stage, but they may also belong to new unseen classes, which makes the problem even more challenging. The classification system should be capable of identifying the target classes and to reject the unwanted ones. Following the OSR nomenclature (cf. Sec. 3), the involved groups of classes are denoted as KK, KU and UU in Fig. 1.

The dataset presented in this paper is aimed at facilitating research on FSL for audio event classification. A domestic environment is considered, where a particular sound must be identified from a set of *pattern sounds*, all belonging to a general "audio alarm" class. The challenge lies in detecting the target pattern by using only a reduced number of examples. To account for openness conditions, the dataset provides as well a folder of *unwanted sounds* containing audio samples from different subclasses which are not considered to be audio alarms or pattern sounds. An optimal FSL+OSR system would be able to correctly identify all the instances belonging to the different pattern sounds by using only a few training examples, while rejecting all the examples pertaining to the general unwanted class. A preliminary version of this dataset has already been used in a previous work [21]. Moreover, as one of the main motivations of this paper is to facilitate open research in the field of audio-oriented FSL and OSR, the dataset is accompanied by two baseline systems based on transfer learning.

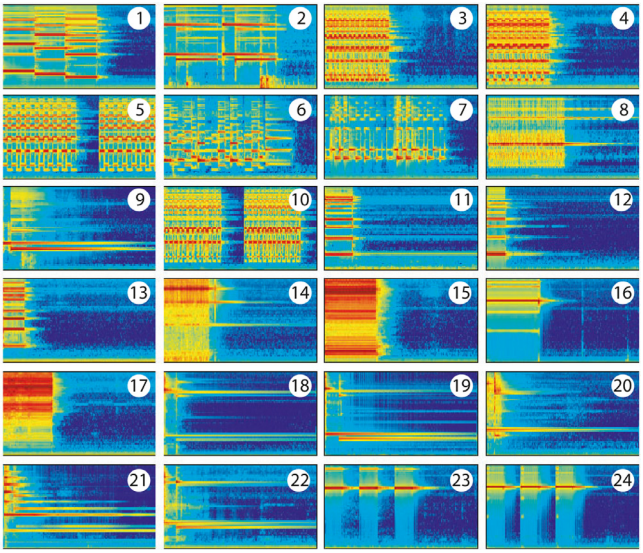

Fig. 2. Example spectrograms from the pattern sounds category.

2. Dataset

The dataset is divided into 34 taxonomic classes. These 34 classes are classified into one of two main sub-categories: *pattern sounds* and *unwanted*. The dataset is completely balanced, as every class contains exactly the same number of audio examples.

- *Pattern sounds category*: comprises 24 classes, each one being a different type of audio alarm (e.g. fire alarms or door bells). Each *pattern sound* class has 40 audio clips.
- *Unwanted category*: it is comprised of a total of 10 different classes, each one representing everyday domestic audio sources: *car horn, clapping, cough, door slam, engine, keyboard tapping, music, pots and pans, steps and water falling*. Each of these *unwanted* classes has 40 audio clips.

Moreover, a k-fold configuration is provided in order to check the generalisation of the results. The number of folds (k) for cross-validation depends on the number of shots used for learning. That means, when training with 4 shots, the number of folds is $k = 10$. For 2 shots, $k = 20$. Consequently, there are 40 folds for 1 shot. All the audio sequences have a duration of 4 seconds and have been recorded using a single audio channel with a sample rate of 16 kHz and 16 bits per sample. All the audios were obtained in a controlled low-noise scenario. The events were recorded individually and trimmed to the desired length. The dataset annotations and configuration files were manually generated by the authors. The dataset along with other detailed information is publicly available.[2] Examples corresponding to the same pattern sound class are expected to share similar characteristics, while those from unwanted classes tend to show higher variability, as they come from more general sound events. For illustrative purposes, the log-Mel spectrograms corresponding to examples of the *pattern sounds* classes are represented in Fig. 2.

3. Experimental setup

The aim of the experiments is to test the performance of the baseline system over the proposed dataset considering both OSR and FSL conditions. The evaluation under open-set conditions is

[2] https://zenodo.org/record/3689288.

Table 1
Number of classes of each configuration and resulting openness.

Pattern Sounds	$\|KK\|$	$\|KU\|$	$\|UU\|$	$\|C_{TR}\|$	$\|C_{TE}\|$	O^*
Full set	24	10	0	34	34	0
		5	5	29	34	0.04
		0	10	24	34	0.09
Trios	3	10	0	13	13	0
		5	5	8	13	0.13
		0	10	3	13	0.39

based on the concept of *openness* [22]. For this purpose, the *pattern sounds* and *unwanted* categories detailed in Sect. 2 are further subdivided as follows:

- *Known Known* (KK) classes: are the classes whose audios have been used for training/validation labeled as positive events to be recognized. In the context of this work, KK classes would match the *pattern sounds* category.
- *Known Unknown* (KU) classes: are the classes whose audios have been used for training/validation, but labeled as unwanted categories so that they are not classified as positive events during testing. In this work, KU classes would be represented by a subset of the *unwanted* classes.
- *Unknown Unknown* (UU) classes: as in the previous case, UU classes are a subset of the *unwanted* group. The difference between KU and UU is that the audios in UU classes are not used for training/validation; instead, they are only used in the testing phase. It is expected that audios in UU subset will be classified as unwanted by the system after the training/validation stage has been finished.

The openness, O^*, can be calculated using the formula [23]:

$$O^* = 1 - \sqrt{\frac{2 \times |C_{TR}|}{|C_{TR}| + |C_{TE}|}}, \tag{1}$$

where C_{TR} is the set of classes used during training, $C_{TR} = \text{KK} \cup \text{KU}$, and C_{TE} corresponds to the set of classes used in testing phase, $C_{TE} = C_{TR} \cup \text{UU}$. Openness values are bounded to the range $0 \leq O^* < 1$. When $C_{TR} = C_{TE}$, it reaches its minimum value ($O^* = 0$), meaning that, during testing, the algorithm is not required to face events that belong to classes unseen during training. On the contrary, as the difference between $|C_{TE}|$ and $|C_{TR}|$ becomes larger, with $|C_{TE}| > |C_{TR}|$, the openness tends to approach to its maximum value: $O^* \to 1$. This means that, during testing, the system needs to reject events belonging to classes unseen during training.

In a first batch of experiments, all 24 *pattern sounds* classes have been used together as KK classes. In a second batch, *pattern sounds* have been selected in 8 groups of 3 classes each (8 trios, as later identified in Section 5), therefore, only 3 classes per run have been used as KK. The particular classes in each trio have been selected to cover different everyday situations ranging from very different sounds as (1,9,17) to more similar ones as (4,5,16). This second batch reflects a more realistic scenario where the number of classes in the union of KU and UU subsets (KU ∪ UU) outnumbers the classes in the KK group. Besides, the experimental setup was designed to have several degrees of freedom taking into account the number of positive audio samples used for training (also called shots) and different values of openness. Experiments with one, two and four shots have been carried out. In order to obtain different values of openness, the ratio given by the number of KU classes and the number of UU classes has been set to 10/0, 5/5 and 0/10. This results in $O^* \in \{0,\ 0.04,\ 0.09\}$ for the first batch of experiments and $O^* \in \{0,\ 0.13,\ 0.39\}$ for the second. Table 1 summarizes the details related to the two types of experiments described above. Note that, in all cases, we have a completely balanced classification problem with $|KK|$ classes, with a reject option.

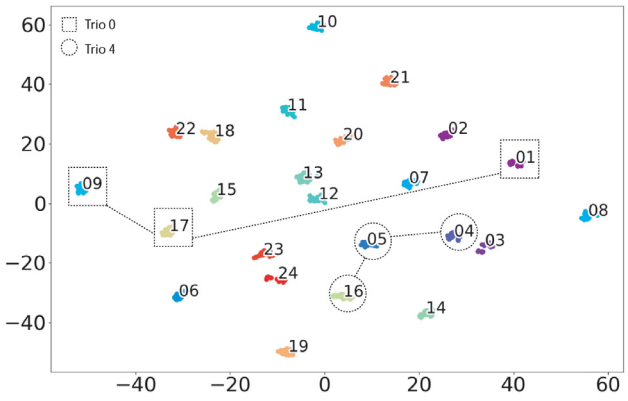

Fig. 3. t-SNE mapping from L^3-net representation of 24 KK categories.

4. Baseline systems

Due to the lack of reference approaches aimed at simultaneously dealing with FSL and OSR, we propose here two simple system baselines. The FSL problem is addressed by making use of the embeddings extracted from different well-known pre-trained networks, following a transfer learning approach. The OSR problem is tackled by including sigmoid-based activations at the output layer to allow a threshold-based rejection of unwanted classes.

4.1. L^3-net

L^3-net[24] is a neural network trained with two specific partitions of Audioset[25] from subsets corresponding to environmental and music videos. The parameters of the embedding were set as follows: *content_type* = "music" [24], *input_repr* = "mel256", *embedding_size* = 512 and *hop_size* = 0.5. The selection of the content type might be explained by the fact that most alarm sounds show a harmonic-like spectrum, which may resemble more to music sources than to environmental sounds. For the computation of the L^3-net embeddings, each audio clip is divided into 1-second segments with a hop size of 0.5 seconds. Taking into account the 1 second analysis window used by L^3-net, the above parameters lead to an embedding matrix of size 512×7[24]. We summarize this output by averaging across the temporal dimension, resulting in a 512×1 column-vector representation. For visualization purposes, a t-SNE mapping of such representation for the KK classes is shown in Fig. 3. Note that it captures faithfully the similarity existing among examples of the same pattern sound class, leading to visibly condensed clusters.

4.2. YAMNet

The implementation of this system follows the same procedure as the one using L^3-net embeddings. In this case, the audio preprocessing is based on log-Mel spectrograms using 64 frequency bands and a frame size of 0.96 s with 50% overlap. For 4 second audio clips, the extracted audio embeddings have a shape of 1024×8. As with L^3-net, the mean across the temporal axis is computed to flatten such output. YAMNet has also been trained using Audioset.

4.3. System classifier

For the classification task, a multi-layer perceptron with two fully-connected hidden layers with 512 and 128 units respectively

Table 2
Baseline system average accuracies (%) and corresponding standard deviations (not shown for ACC_w) with 24 KK classes using L^3-net network. Shots indicates the number of training examples per class.

Shots	Openness coefficient									
	$O^*=0$			$O^*=0.04$				$O^*=0.09$		
	ACC_{KK}	ACC_{KU}	ACC_w	ACC_{KK}	ACC_{KUU}	ACC_{UU}	ACC_w	ACC_{KK}	ACC_{UU}	ACC_w
1	13.8±12.9	99.8±1.0	56.8	57.7±8.4	90.4±5.4	84.8±9.8	74.1	60.1±7.8	39.6±13.4	49.9
2	81.1±5.5	99.4±0.8	**90.3**	83.2±4.8	90.2±5.1	82.5±9.6	**86.7**	83.3±5.6	33.3±11.6	58.3
4	94.8±2.2	99.6±0.4	**97.2**	94.3±2.2	88.3±5.7	79.4±9.5	91.3	94.8±2.4	26.1±10.1	60.5

was implemented as in [24]. This neural network is fed with either YAMNet or L^3-net embeddings independently. All activation units are ReLUs. The output layer has 24 or 3 units (each one corresponding to a class of *pattern sounds*) with sigmoid activation function. Output targets corresponding to different *unwanted sound* audio clip subcategories are set to zero vector of the appropriate size. This indicates the absence of any *pattern sounds* category. Adam optimizer [26] was used. The loss function during training was binary cross-entropy and the evaluation metric was categorical accuracy. At test time, an audio clip is classified as known, or *pattern sound*, when the corresponding output probability ranks the highest and above a threshold with value 0.5. In the case where this threshold is exceeded by more than one class, the system predicts the class having the highest detection probability. The code for replicating the results is fully available[3].

5. Results

The aim of the experiments is to test the capability of the baseline systems to correctly classify the examples corresponding to the set of target pattern sounds (KK classes) while successfully rejecting any sound pertaining to an unwanted class, regardless of whether it belongs to a KU class or a UU class.

Following the criteria of Task 1C of DCASE-2019 [27], the ACC_w measure is used,

$$O^*=0 \quad \text{(without UU)}: \quad ACC_w = wACC_{KK} + (1-w)ACC_{KU}, \tag{2a}$$

$$O^*\neq 0 \quad \text{(with KU and UU)}: \quad ACC_w = wACC_{KK} + (1-w)ACC_{KUU}, \tag{2b}$$

$$O^*\neq 0 \quad \text{(with only UU)}: \quad ACC_w = wACC_{KK} + (1-w)ACC_{UU}, \tag{2c}$$

where w is an arbitrary weight that allows to balance the importance of the accuracy relative to target and unwanted classes. In the above equations, ACC_{KK} is the multiclass accuracy over test examples exclusively from target (KK) classes. Correspondingly, ACC_{KU} and ACC_{UU} denote the same accuracy when considering test data either from KU or UU classes and considering two output labels only: pattern and unwanted. Finally, when the openness is such that there are both KU and UU classes, then the rejection capability is measured by the ACC_{KUU}, which is the mean of ACC_{KU} and ACC_{UU}. In the present work w has been given a fixed value of $w=0.5$. Note that the formulas in Eq. (2) take into account accuracies of all the categories, KK, KU and UU. Therefore, it is a convenient way of analyzing the trade-off between correct prediction and rejection.

Results are presented following k-fold cross-validation as indicated in Sec. 2 repeated 5 times. All the tables show the mean accuracy and standard deviation across all runs and folds. Best performance between the two proposed baseline systems is highlighted using bold typeface.

[3] https://github.com/Machine-Listeners-Valencia/fsl_osr_dataset_baseline.

5.1. Large number of target classes

The results obtained by the two baseline systems for the first batch of experiments are shown in Tables 2 and 3. In this first batch, the KK set comprises the 24 pattern sound classes. As indicated in Table 1, three values of openness are considered: $O^* \in \{0, 0.04, 0.09\}$. As expected, the results exhibit the difficulties encountered in FSL and OSR conditions. On the one hand, the lack of a large number of training examples affects considerably the classification performance, as evidenced, for example, by the low ACC_w values achieved by the L^3-net system with only one shot. On the other hand, as the openness value increases, the accuracy for KK classes remains similar whereas the accuracy of KU-UU classes decreases.

Low values in ACC_{UU} and/or ACC_{KU} indicate that the system is misclassifying unwanted events as *pattern sounds*, meaning that false positives are observed in the KK categories. As expected, the problems arising from UU classes are more evident under higher openness conditions. By letting the system learn from a set of unwanted sounds, the rejection capabilities are considerably increased. This is evidenced by the higher values in ACC_{UU} for $O^*=0.04$ with respect to the ones for $O^*=0.09$, independently of the baseline system used. Note, however, that the use of unwanted sounds for training the classifier may also have an impact in the accuracy achieved for the target pattern sounds. As shown in both tables, at $O^*=0$, the accuracy for the KK classes is worse than for higher openness. This is because the use of *KU* classes to train the system makes the underlying classification boundaries more restrictive, and the system is more prone to miss target instances.

In general terms, YAMNet shows a greater weighted accuracy regarding known and unknown situations when $O^* \in \{0.04, 0.09\}$. Thus, YAMNet could be understood as a more discriminative extractor when unknown situations are present. However, the most significant phenomenon can be seen when $O^*=0$ and the number of shots is equal to 1. A huge improvement in ACC_{KK} is observed with respect to L^3-net, leading to a better trade-off in ACC_w. The improvement of this feature extractor is nearly of 25 percentage points (see Table 2). The difference between 1 shot and 2 shots with $O^*=0$ using L^3-net is more than 30 percentage points, while for YAMNet is only of 8 percentage points. Therefore, YAMNet seems to be a more robust solution.

5.2. Small number of target classes

Tables 4 and 5 show the results for the second batch of experiments that consider only KK sets comprised of 3 pattern sound classes, considering 8 different and disjoint trios. As indicated in Table 1, the three values of openness in this case are: $O^* \in \{0, 0.13, 0.39\}$. Again, the general tendency is confirmed, where a lower number of shots or a higher openness level leads always to a decrease in performance. However, in this case, it can be observed that the particularities of the target classes can be also an important factor affecting the overall performance of the system. For example, with $O^*=0.39$, very low values for ACC_{UU} are ob-

Table 3
Baseline system average accuracies (%) and corresponding standard deviations (not shown for ACC_w) with 24 KK classes using YAMNet network. Shots indicates the number of training examples per class.

Shots	Openness coefficient									
	$O^* = 0$			$O^* = 0.04$				$O^* = 0.09$		
	ACC_{KK}	ACC_{KU}	ACC_w	ACC_{KK}	ACC_{KUU}	ACC_{UU}	ACC_w	ACC_{KK}	ACC_{UU}	ACC_w
1	64.4±3.7	95.8±2.6	**80.1**	65.6±3.3	91.0±4.2	89.4±5.7	**78.3**	66.9±3.2	47.3±13.1	**57.1**
2	78.8±2.3	97.6±1.9	88.2	79.3±2.3	91.8±4.2	87.6±6.2	85.6	80.4±2.3	41.7±11.7	**61.1**
4	90.8±1.7	99.1±0.9	94.9	91.0±1.7	92.8±2.8	87.4±4.9	**91.9**	92.0±1.6	36.5±8.6	**64.3**

Table 4
Baseline L^3-net system average accuracies (%) and corresponding standard deviations (not shown for ACC_w) for the second batch of experiments using trios (only 3 KK classes).

Trio	Shots	Openness coefficient									
		$O^* = 0$			$O^* = 0.13$				$O^* = 0.39$		
		ACC_{KK}	ACC_{KU}	ACC_w	ACC_{KK}	ACC_{KUU}	ACC_{UU}	ACC_w	ACC_{KK}	ACC_{UU}	ACC_w
0	1	65.1±16.1	99.4±1.1	82.3	85.9±13.4	97.7±4.6	98.4±4.1	**91.8**	100±0	18.6±8.9	**59.3**
	2	80.2±15.0	99.6±0.5	89.9	89.2±12.5	99.6±0.5	99.8±0.6	94.4	100±0	17.0±5.9	**58.5**
(1, 9, 17)	4	90.1±14.5	99.7±0.4	94.9	97.5±8.1	99.7±0.4	99.9±0.4	**98.6**	100±0	16.9±3.3	**58.5**
1	1	68.9±12.9	99.9±0.2	84.4	88.8±13.1	98.3±2.8	96.8±5.6	**93.5**	100±0	3.9±3.1	52.0
	2	84.7±16.5	99.9±0.3	92.3	89.0±14.5	98.7±2.4	97.6±4.7	**93.8**	100±0	3.6±2.6	51.8
(10, 12, 19)	4	88.0±15.6	99.9±0.4	93.9	96.2±9.6	96.7±3.1	93.8±5.8	**96.5**	100±0	3.8±3.5	51.9
2	1	55.5±18.6	99.9±1.0	77.7	78.4±13.4	99.8±0.9	99.7±1.7	89.1	98.6±2.4	14.8±12.1	**56.7**
	2	76.1±14.7	99.9±0.1	88.0	82.6±13.9	99.8±0.5	99.7±0.6	91.2	99.5±1.2	15.7±11.9	**57.6**
(2, 14, 22)	4	83.1±20.7	99.9±0.1	91.5	91.9±12.3	99.4±0.9	99.0±1.5	95.6	99.9±0.4	11.5±8.2	**55.7**
3	1	53.0±12.1	99.9±0.4	76.5	72.3±13.4	96.2±4.2	92.7±8.2	**84.3**	99.7±0.7	24.9±8.2	**62.3**
	2	64.6±16.1	99.9±0.3	82.2	78.4±13.7	95.7±4.6	91.6±8.7	**87.2**	99.8±0.5	23.3±6.1	**61.6**
(3, 6, 13)	4	77.4±19.0	99.8±0.9	88.6	90.3±11.4	92.0±3.2	84.8±6.0	91.1	99.8±0.4	24.5±6.0	**62.2**
4	1	71.7±15.2	100±0	**85.8**	88.5±10.1	99.3±1.3	98.6±2.5	**93.9**	99.8±0.8	2.4±2.4	**51.1**
	2	86.8±14.5	100±0	**93.4**	93.2±9.2	99.4±1.1	98.8±2.2	**96.3**	100±0.2	1.7±1.7	50.8
(4, 5, 16)	4	88.1±18.6	99.9±0.6	94.0	97.0±9.1	99.0±1.2	98.1±2.2	**98.0**	100±0	1.7±1.2	50.9
5	1	76.5±15.2	99.9±0.2	88.2	87.9±11.8	99.1±1.2	98.5±2.2	93.5	97.3±5.1	42.1±20.1	**69.7**
	2	85.1±15.4	99.9±0.1	92.5	93.4±7.7	98.8±1.2	97.8±2.3	96.1	99.1±2.6	39.1±19.8	**69.1**
(18, 21, 23)	4	89.3±16.4	100±0.1	94.6	97.2±8.1	98.3±1.2	96.8±2.1	97.7	99.9±0.3	34.3±20.2	**67.1**
6	1	87.0±13.5	99.7±0.5	93.4	96.0±7.8	99.3±0.8	99.4±0.6	**97.6**	100±0	30.9±11.6	**65.5**
	2	87.6±16.0	99.6±0.6	93.6	95.8±9.1	99.4±0.7	99.2±1.0	97.6	100±0	28.2±9.5	**64.1**
(8, 11, 24)	4	89.9±14.5	99.7±0.5	94.8	96.8±9.2	99.2±0.8	98.9±1.0	98.0	100±0	27.7±8.0	**63.9**
7	1	66.4±15.7	99.6±0.6	83.0	87.0±11.4	97.6±2.9	96.8±5.4	**92.3**	99.2±1.9	23.7±8.0	**61.5**
	2	82.1±13.7	99.5±0.7	90.8	90.0±9.8	98.6±1.7	98.4±3.0	94.3	99.8±0.6	24.0±6.7	61.9
(7, 15, 20)	4	83.7±15.3	99.5±0.9	91.6	94.4±10.1	98.5±1.5	98.1±2.7	96.5	100±0.2	24.2±5.3	62.1

Table 5
Baseline YAMNet system average accuracies (%) and corresponding standard deviations (not shown for ACC_w) for the second batch of experiments using trios (only 3 KK classes).

Trio	Shots	Openness coefficient									
		$O^* = 0$			$O^* = 0.13$				$O^* = 0.39$		
		ACC_{KK}	ACC_{KU}	ACC_w	ACC_{KK}	ACC_{KUU}	ACC_{UU}	ACC_w	ACC_{KK}	ACC_{UU}	ACC_w
0	1	83.8±9.4	97.3±3.3	**90.6**	87.0±8.7	92.3±4.5	90.6±5.6	89.6	94.0±7.1	17.3±13.0	55.6
	2	93.6±4.4	99.4±0.8	**96.5**	94.2±4.9	94.9±3.9	92.5±5.3	**94.5**	97.4±3.1	16.1±11.0	56.7
(1, 9, 17)	4	97.8±3.0	99.8±0.4	**98.8**	97.7±3.4	96.5±2.3	94.1±3.5	97.1	98.6±2.8	17.0±17.0	57.8
1	1	83.9±5.7	96.5±3.8	**90.2**	88.2±5.9	91.7±4.1	89.5±5.7	90.0	96.0±2.4	26.2±14.9	**61.1**
	2	92.8±4.8	99.4±1.1	**96.1**	92.6±5.9	91.6±4.7	87.8±6.8	92.1	97.2±2.5	25.4±16.6	**61.3**
(10, 12, 19)	4	96.5±2.7	99.8±0.3	**98.2**	96.4±2.3	95.1±3.3	91.2±5.8	95.7	98.0±2.2	21.6±14.4	**59.8**
2	1	96.9±3.7	99.9±0.1	**98.4**	97.7±4.8	97.7±3.0	95.9±3.7	**97.7**	98.7±5.5	11.0±7.2	54.8
	2	98.4±1.1	100±0	**99.2**	99.2±0.8	97.6±1.7	95.4±3.2	**98.4**	100±0	8.3±7.2	54.2
(2, 14, 22)	4	98.9±1.1	100±0	**99.5**	99.4±0.8	97.1±1.1	94.4±2.2	**98.2**	100±0	4.9±5.8	52.5
3	1	58.9±7.9	95.5±3.4	**77.2**	63.1±8.2	89.6±3.7	86.4±4.7	76.3	68.9±7.3	5.2±6.1	37.0
	2	70.8±7.0	98.3±1.5	**84.5**	73.6±6.1	91.7±3.4	88.3±4.5	82.7	79.0±6.4	7.5±8.2	43.3
(3, 6, 13)	4	85.9±5.2	99.6±0.6	**92.7**	86.8±4.7	95.9±2.7	93.1±4.2	**91.4**	94.1±4.3	8.8±5.2	51.5
4	1	71.1±8.2	97.6±3.0	83.7	75.1±8.1	92.4±4.4	90.0±5.3	83.7	82.1±8.2	12.4±12.5	47.3
	2	85.7±6.1	99.1±1.1	92.4	88.8±5.9	92.5±3.6	88.2±5.9	90.7	93.8±6.2	10.6±9.0	**52.2**
(4, 5, 16)	4	92.1±5.1	99.9±0.2	**96.0**	93.4±4.9	93.4±3.9	88.6±6.8	93.4	97.6±3.3	9.9±6.2	**53.8**
5	1	98.6±5.1	99.7±1.0	**99.2**	99.7±1.7	99.6±1.9	99.6±1.5	**99.6**	100±0	24.3±13.8	62.2
	2	99.6±2.1	100±0	**99.8**	99.9±0.5	99.9±0.2	99.9±0.2	**99.9**	100±0	20.9±12.7	60.4
(18, 21, 23)	4	100±0	100±0	**100**	100±0	100±0	100±0	**100**	100±0	21.3±15.4	60.8
6	1	94.2±7.8	99.6±1.4	**97.0**	94.6±6.1	98.1±2.6	96.8±3.5	96.4	96.4±3.2	14.6±7.1	55.5
	2	98.0±4.2	100±0	**99.0**	98.1±3.1	98.8±0.9	97.7±1.9	**98.4**	97.5±2.9	12.4±5.9	55.0
(8, 11, 24)	4	99.4±1.4	100±0	**99.7**	99.4±1.5	98.8±0.8	97.8±1.5	**99.1**	98.7±2.7	11.0±4.7	54.8
7	1	86.1±9.2	98.7±2.3	**92.4**	86.3±9.5	96.4±3.9	96.7±3.7	91.4	88.8±9.3	25.8±13.1	57.3
	2	93.2±4.5	99.6±0.7	**96.4**	93.4±4.2	98.0±2.4	98.3±1.9	**95.7**	94.6±3.4	31.8±11.4	**63.2**
(7, 15, 20)	4	96.0±2.4	99.7±0.6	**97.8**	96.0±2.3	99.3±0.8	99.6±0.4	**97.6**	96.2±2.5	35.9±9.7	**66.1**

tained in Table 4 for trios 1 and 4, considerably worse than for other trios in the dataset. The internal L^3-net representations of such target classes may probably lead to classification boundaries that are not discriminatory enough to reject successfully the unwanted sounds. Interestingly, the specific internal representations are also of high importance, as the same trios are not the ones with lowest performance in YAMNet (see Table 5). In any case, the differences between the two baseline systems are much more evident in this second batch of experiments than in the previous one. While $O^* = 0$ was the case that most favored the L^3-net baseline when $|KK| = 24$, with trios YAMNet seems to offer better performance for the same openness value. The tendency is also reversed for the highest level of openness ($O^* = 0.39$), as the L^3-net embeddings show now the best performance for most trios. Finally, note that the trio-wise results are quite balanced for $O^* = 0.13$, as both systems are similarly competitive. However, the winning system is again quite dependent on the actual trio.

6. Conclusions and future work

Few-shot learning (FSL) is a research area with increasing interest in the audio domain. However, the lack of public FSL audio datasets makes it necessary to manipulate other existing databases with the aim of adapting them properly to FSL research. Moreover, open-set recognition (OSR) can be an additional problem in practical FSL scenarios, where the models are likely to be tested with instances from unseen classes during training. This work presented a carefully designed audio dataset for FSL and OSR research, where target sounds are instances of classes corresponding to different audio patterns (fire alarms, doorbells, etc.). The dataset considers a domestic scenario where such audio pattern classes correspond to intentional sounds to be accurately detected in the presence of other unwanted sounds (coughs, door slams, etc.). Each class comes with different samples for FSL training, validation and testing, under different openness conditions. To facilitate the use of this dataset and promote algorithm development, we also provide results with a baseline system using transfer learning from pre-trained state-of-the-art convolutional neural networks. The results show that important trade-offs exist when both FSL and OSR conditions are considered, evidencing the need for novel learning architectures aimed at facing both types of problems. Future updates of this dataset will include more challenging acoustic conditions, such as different levels of noise, reverberation and overlapped events.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements and current affiliation of the authors

This work was supported by the EU Horizon 2020 programme [grant No 779158]. Grants DIN2018-009982, PTQ-17-09106, RTI2018-097045-B-C21/C22 funded by MCIN/AEI/10.13039/501100011033, the latter also by “ERDF A way of making Europe”. Grants TED2021-131003B-C21/C22 funded by MCIN/AEI/10.13039/501100011033 and by the “EU Union NextGenerationEU/PRTR”. Grants AICO/2020/154 and AEST/2020/012, funded by GVA. The authors acknowledge also the Artemisa computer resources funded by the EU ERDF and Comunitat Valenciana, and the technical support of IFIC (CSIC-UV). Authors J. Naranjo, S. Perez and P. Zuccarello were working at Visualfy when this work was done, but they are now with the Instituto Tecnológico de Informática (ITI), Tyris AI and ITI, respectively.

References

[1] S.H. Bae, I. Choi, N.S. Kim, Acoustic scene classification using parallel combination of lstm and cnn, in: Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016), 2016, pp. 11–15.
[2] E. Cakır, T. Heittola, T. Virtanen, Domestic audio tagging with convolutional neural networks, IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2016).
[3] H. Phan, P. Koch, F. Katzberg, M. Maass, R. Mazur, A. Mertins, Audio scene classification with deep recurrent neural networks, in: Proc. Interspeech 2017, 2017, pp. 3043–3047.
[4] M. Wang, W. Deng, Deep face recognition: A survey, Neurocomputing.
[5] K. Chen, A. Salman, Extracting speaker-specific information with a regularized siamese deep network, in: Advances in Neural Information Processing Systems, 2011, pp. 298–306.
[6] R. Lu, K. Wu, Z. Duan, C. Zhang, Deep ranking: Triplet matchnet for music metric learning, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017, pp. 121–125.
[7] H. Bredin, Tristounet: triplet loss for speaker turn embedding, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017, pp. 5430–5434.
[8] Y. Zheng, D.K. Pal, M. Savvides, Ring loss: Convex feature normalization for face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5089–5097.
[9] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: European conference on computer vision, Springer, 2016, pp. 499–515.
[10] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a" siamese" time delay neural network, in: Advances in neural information processing systems, 1994, pp. 737–744.
[11] I. Melekhov, J. Kannala, E. Rahtu, Siamese network features for image matching, in: 2016 23rd International Conference on Pattern Recognition (ICPR), IEEE, 2016, pp. 378–383.
[12] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[13] E. Hoffer, N. Ailon, Deep metric learning using triplet network, in: International Workshop on Similarity-Based Pattern Recognition, Springer, 2015, pp. 84–92.
[14] D. Battaglino, L. Lepauloux, N. Evans, The open-set problem in acoustic scene classification, in: 2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC), IEEE, 2016, pp. 1–5.
[15] W.J. Scheirer, A. de Rezen, A. Sapkota, T.E. Boult, Toward open set recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (7) (2012) 1757–1772.
[16] W.J. Scheirer, L.P. Jain, T.E. Boult, Probability models for open set recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (11) (2014) 2317–2324.
[17] P.R.M. Júnior, R.M. De Souza, R.d.O. Werneck, B.V. Stein, D.V. Pazinato, W.R. de Almeida, O.A. Penatti, R.d.S. Torres, A. Rocha, Nearest neighbors distance ratio open-set classifier, Machine Learning 106 (3) (2017) 359–386.
[18] A. Bendale, T.E. Boult, Towards open set deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1563–1572.
[19] L. Shu, H. Xu, B. Liu, Doc: Deep open classification of text documents, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2911–2916.
[20] N. Kardan, K.O. Stanley, Mitigating fooling with competitive overcomplete output layer neural networks, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 518–525.
[21] J. Naranjo-Alcazar, S. Perez-Castanos, P. Zuccarello, F. Antonacci, M. Cobos, Open set audio classification using autoencoders trained on few data, Sensors 20 (13) (2020) 3741.
[22] W.J. Scheirer, L.P. Jain, T.E. Boult, Probability models for open set recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) 36.
[23] C. Geng, S.-j. Huang, S. Chen, Recent advances in open set recognition: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence.
[24] J. Cramer, H.-H. Wu, J. Salamon, J.P. Bello, Look, listen, and learn more: Design choices for deep audio embeddings, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 3852–3856.
[25] J.F. Gemmeke, D.P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R.C. Moore, M. Plakal, M. Ritter, Audio set: An ontology and human-labeled dataset for audio events, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017, pp. 776–780.
[26] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.
[27] A. Mesaros, T. Heittola, T. Virtanen, Acoustic scene classification in dcase 2019 challenge: Closed and open set classification and data mismatch setups, in: Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019), New York University, NY, USA, 2019, pp. 164–168.