DCASE 2017 CHALLENGE SETUP: TASKS, DATASETS AND BASELINE SYSTEM

Annamaria Mesaros[1], *Toni Heittola*[1], *Aleksandr Diment*[1], *Benjamin Elizalde*[2], *Ankit Shah*[2], *Emmanuel Vincent*[3], *Bhiksha Raj*[2], *Tuomas Virtanen*[1]*

[1] Tampere University of Technology, Laboratory of Signal Processing, Tampere, Finland
{annamaria.mesaros, toni.heittola, aleksandr.diment, tuomas.virtanen}@tut.fi
[2] Carnegie Mellon University, Department of Electrical and Computer Engineering,
& Department of Language Technologies Institute, Pittsburgh, USA
bmartin1@andrew.cmu.edu, aps1@andrew.cmu.edu, bhiksha@cs.cmu.edu
[3] Inria, F-54600 Villers-lès-Nancy, France, emmanuel.vincent@inria.fr

ABSTRACT

DCASE 2017 Challenge consists of four tasks: acoustic scene classification, detection of rare sound events, sound event detection in real-life audio, and large-scale weakly supervised sound event detection for smart cars. This paper presents the setup of these tasks: task definition, dataset, experimental setup, and baseline system results on the development dataset. The baseline systems for all tasks rely on the same implementation using multilayer perceptron and log mel-energies, but differ in the structure of the output layer and the decision making process, as well as the evaluation of system output using task specific metrics.

Index Terms— Sound scene analysis, Acoustic scene classification, Sound event detection, Audio tagging, Rare sound events, Weak Labels

1. INTRODUCTION

Sounds carry a large amount of information about our everyday environment and physical events that take place in it. Humans are very skilled in perceiving the general characteristics of the sound scene around them, whether it is a busy street, a quiet park or a quiet office environment, and recognizing individual sound sources in the scenes, such as cars passing by, birds, or footsteps. Developing computational methods to automatically extract this information has huge potential in several applications, for example searching for multimedia based on its audio content [1], making context-aware mobile devices [2], robots, cars, etc., and intelligent monitoring systems [3, 4] to recognize activities using acoustic information. However, a significant amount of research is still needed to reliably recognize sound scenes and individual sound sources in real-life soundscapes, where multiple sounds are present, often simultaneously, and distorted by the environment.

Building up on the success of the previous editions, DCASE 2017 Challenge supports the development of computational scene and event analysis methods by comparing different approaches using common publicly available datasets. The continuous effort in this direction will set another milestone of development, and anchor the current performance for further reference. The challenge consists of four tasks: acoustic scene classification, detection of rare sound events, sound event detection in real-life audio, and large-scale weakly supervised sound event detection for smart cars.

Acoustic scene classification is a prominent topic in environmental sound classification. It is defined as recognition of the environment in which a recording has been made, relying on the assumption that an acoustic scene, as a general characterization of a location or situation, is distinguishable from others based on its general acoustic properties. It has been present as a task in DCASE 2013 [5] and DCASE 2016 [6], and has been approached in a variety of ways. A review of the features and classifiers used for it is presented in [7], with features including the well-known mel-frequency cepstral coefficients [2, 8] or more specialized features such as histograms of sound events [9] or histogram of gradients learned from time-frequency representations [10], and acoustic models such as hidden Markov models (HMMs) [2], Gaussian mixture models (GMMs) [8] or support vector machines (SVMs) [10, 11]. More recently, the emergence of methods using deep learning is noticeable, with many of the submitted systems for DCASE 2016 being based on various types of deep neural networks (DNNs) [6].

Sound event detection is defined as recognition of individual sounds in audio, involving also estimation of onset and offset for distinct sound event instances, possibly for multiple sound classes. It assumes that similar sounds can be represented as a single class, and as such this class is sufficiently different from other sound classes to allow recognition. The most used features for sound event detection are mel-scale representations, namely cepstral coefficients or log energies [12, 13, 14], and they are used with various machine learning methods, including HMMs [12], non-negative matrix factorization (NMF) [15, 16], random forests [11], and DNNs [14, 17].

Sound event detection in real-life audio presents many difficulties for automatic methods, such as the inherent acoustic variability of the sounds belonging to the same sound event class, or other sounds overlapping with the sound event of interest. In some situations, the target sound events are very rare, imposing additional burden on detection systems to avoid false detections. DCASE 2017 Challenge addresses rare sound events and highly overlapping sounds through two separate tasks: detection of rare sound events, sound event detection in real-life audio.

Sound recordings are shared on the Internet on a minute-by-minute basis. These recordings are predominantly videos and constitute the largest archive of sounds we've ever seen. Most of their acoustic content is untagged; hence automatic recognition of sounds within recordings can be achieved by sound event detection.

*AM, TH and TV received funding from the European Research Council under the ERC Grant Agreement 637422 EVERYSOUND.

However, most of the literature and the previous two iterations of DCASE focus on audio-only recordings and supervised approaches, by which the training and test data are annotated with strong labels (including precise timestamps). Collecting such annotations hardly scales to the number of web videos and sound classes. Therefore, we argue that there is a need for semi-supervised approaches that are trained and evaluated with weak labels (not including precise timestamps). Current literature has shown potential using unsupervised [18, 19, 20] and semi-supervised approaches [21, 22] some of them employing weak labels [23]. Success in this task would complement other modalities for video content analysis.

This paper presents in detail the DCASE 2017 Challenge tasks. For each task we provide the task definition, information about the dataset, the task setup and baseline system, and baseline results on the development dataset. The baseline systems for all tasks rely on the same implementation and use the same features and techniques; they differ in the way they handle and map the input data to target outputs, as this is application specific and was chosen according to the task.

2. CHALLENGE SETUP

The challenge provided the potential participants with four tasks, with publicly available datasets and a baseline system for each task. Challenge submission consisted in system output(s) formatted according to the requirements. In addition, participants were required to submit a technical report containing the description of the system(s) in sufficient detail, to allow the community to compare and understand all submissions. The timeline of the challenge is presented in Table 1, and the general organization of the datasets and baseline systems presented in detail in the following sections.

2.1. Datasets

A *development dataset* was provided for each task when the challenge was launched, consisting of predefined *training* and *test* sets (for some tasks in a cross-validation folds format) to be used during system development. A separate dataset, referred to as *evaluation dataset*, was kept for evaluation of the developed systems. The development datasets consist of audio material and associated reference annotations in a task-specific format, and an experimental setup for reporting system performance on the development dataset. The organizers' recommendation was to use the provided experimental setup, in order to allow a direct comparison between submissions. Access to the datasets was provided through the challenge website [1].

As general rules applicable for all tasks, participants were not allowed to use external data for system development, with datasets from a different task considered as external data. However, manipulation of the provided training and development data was allowed, for augmentation without use of external data (e.g. by mixing data sampled from a probability distribution function or using techniques such as pitch shifting or time stretching).

The evaluation datasets were provided as audio only, without reference annotations, shortly before the challenge submission deadline. Participants were required to run their systems on this data and submit the system outputs to the organizers for evaluation. Participants were not allowed to make subjective judgments of the evaluation data, nor to annotate it. The use of the evaluation dataset to train the submitted system was also forbidden. Reference annotations for the evaluation data were only available to the organizers, therefore they were responsible with performing the evaluation of the results according to the metrics for each task.

[1] http://www.cs.tut.fi/sgn/arg/dcase2017/

Table 1: Challenge timeline

Release of development datasets	21 Mar 2017
Release of evaluation datasets	30 June 2017
Challenge submission	31 July 2017
Publication of results	15 Sept 2017
DCASE 2017 Workshop	16-17 Nov 2017

2.2. Baseline system

A baseline system was provided, with a common implementation for all tasks. The system consists of a basic approach that was tailored to each task. Its purpose is to provide a comparison point for the participants while developing their systems. The performance of the baseline system on the development set is provided for each task. When run with the default parameters, the system downloads the needed dataset and outputs the task-specific results [24].

The implementation is based on a multilayer perceptron architecture (MLP) and uses log mel-band energies as features. The features are calculated in frames of 40 ms with a 50% overlap, using 40 mel bands covering the frequency range 0 to 22050 Hz. The feature vector was constructed using a 5-frame context, resulting in a feature vector length of 200. The MLP consists of two dense layers of 50 hidden units each, with 20% dropout. The network is trained using Adam algorithm for gradient-based optimization [25]; training is performed for maximum 200 epochs using a learning rate of 0.001, and uses early stopping criteria with monitoring started after 100 epochs and a 10 epoch patience. The output layer of the network is task specific, and will be described in the corresponding section. The network is trained using the aforementioned features, and the learning target is presented according to the implemented task. The baseline system also includes evaluation of the system outputs using a specific metric for each task.

The baseline system was implemented using Python, using Keras for machine learning. It has all needed functionality for dataset handling, storing and accessing features and models, and evaluating the results, and allows straightforward adaptation and modification of the various involved steps. Participants were allowed and encouraged to build their system on top of the given baseline system.

3. TASK 1: ACOUSTIC SCENE CLASSIFICATION

The goal of acoustic scene classification is to classify a test recording into one of the provided predefined classes that characterizes the environment in which it was recorded for example "park", "home", "office", as illustrated in Fig. 1.

The dataset provided for this task is TUT Acoustic Scenes 2017, which consists of TUT Acoustic Scenes 2016 [26] as the development set, and a newly recorded evaluation set. The main difference is that for this edition of the challenge, the original recordings of 3-5 minutes length were split into 10 s long segments which were provided in individual files and considered as independent. Shorter audio segments provide less information to the system for the decision making process, thus increasing the task difficulty from the

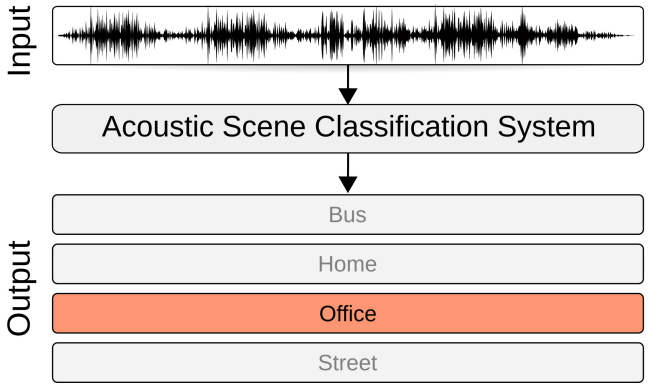

Figure 1: A schematic illustration of the acoustic scene classification addressed in Task 1.

Table 2: Class-wise accuracy of the baseline system for Task 1.

Acoustic scene	Development set Acc. (%)	Evaluation set Acc. (%)
Beach	75.3	40.7
Bus	71.8	38.9
Cafe/Restaurant	57.7	43.5
Car	97.1	64.8
City center	90.7	79.6
Forest path	79.5	85.2
Grocery store	58.7	49.1
Home	68.6	76.9
Library	57.1	30.6
Metro station	91.7	93.5
Office	99.7	73.1
Park	70.2	32.4
Residential area	64.1	77.8
Train	58.0	72.2
Tram	81.7	57.4
Overall	**74.8**	**61.0**

previous edition. This length is regarded as challenging for both human and machine recognition, based on the study in [2]. A detailed description of the data recording and annotation procedure can be found in [26].

The acoustic scene classes considered in this task were: bus, cafe/restaurant, car, city center, forest path, grocery store, home, lakeside beach, library, metro station, office, residential area, train, tram, and urban park. A cross-validation setup containing four folds was provided, splitting the available audio material in the development set such that all segments obtained from the same original recording are included to one side of the learning algorithm, either training or test. For each class, the development set contains 312 segments of 10 seconds (52 minutes of audio material).

For this task, the baseline system was tailored to a multi-class single label classification setup, with the network output layer consisting of softmax type neurons representing the 15 classes. The classification decision was based on the output of the neurons, which can be active only one at a time. Frame-based decisions were combined using majority voting to obtain a single label per classified segment. The system performance was measured using accuracy, defined as the ratio between the number of correct system outputs and the total number of outputs [27]. The system was trained

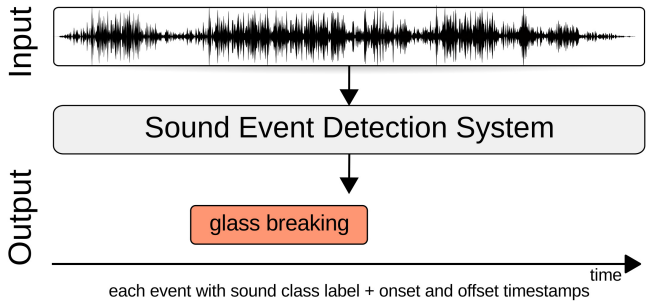

Figure 2: A schematic illustration of the detection of rare sound events addressed in Task 2.

and tested using the provided four fold cross-validation setup, obtaining an average classification accuracy of 73.8% on the development set and 61.0% on the evaluation set. Class-wise accuracy is presented in Table 2. Ranking of the systems submitted for the challenge is done using classification accuracy.

4. TASK 2: DETECTION OF RARE SOUND EVENTS

Task 2 focused on the detection of rare sound events, as illustrated in Fig. 2. The audio material used in this task consists of artificially created mixtures, allowing the creation of many examples at different event-to-background ratios. Here, "rare" refers to target sound events occurring at most once within a half-minute recording. For each of the three target sound event classes, a separate system is to be developed to detect the temporal occurrences of these events.

The provided dataset consists of source files for creating mixtures of rare sound events with background audio, as well as a set of readily generated mixtures and the so-called recipes according to which the mixtures were created. Additionally, a software package was provided, which performs further generation of additional mixture recipes and generates the audio mixtures.

The background recordings originate from the TUT Acoustic Scenes 2016 development dataset [26], with the exception of segments naturally containing the target class events and interference from mobile phone, which were removed. The rare sound events are of the following classes: baby cry (106 training, 42 test instances, mean duration 2.25 s), glass break (96 training, 43 test instances, mean duration 1.16 s) and gun shot (134 training, 53 test, mean duration 1.32 s). The recordings were downloaded from freesound.org through the API with python wrapper[2]. In the "source" part of this dataset, these recordings were presented in their original form, and were accompanied by the added annotations of the temporal occurrences of isolated events.

We isolated the target sound events from the full-length recordings acquired from freesound.org that may consist of the actual event, silence regions and background noise in the following manner. First, a semi-supervised segmentation [28] was performed using an SVM trained to classify high-energy and low-energy frames. The active segments were then obtained with a dynamic threshold computed as a weighted average of top 10% and lower 10% of the onset probability values over all the analysis frames of a recording. Thereupon, a human annotator listened to each obtained segment, and segments containing irrelevant events were discarded (baby coughs, unrealistically sounding gun shots such as laser guns

[2] https://github.com/xavierfav/freesound-python-tools

Table 3: Baseline system results for Task 2, event-based metrics.

	Development set		Evaluation set	
Event Class	**ER**	F-score (%)	**ER**	F-score (%)
Baby cry	0.67	72.0	0.80	66.8
Glass break	0.22	88.5	0.38	79.1
Gun shot	0.69	57.4	0.72	46.5
Average	**0.53**	72.7	**0.63**	64.1

etc.). In the process of such screening, additional manual refinement of the timing of the events was performed with a step of 100 ms to eliminate pauses before and after the event, while not introducing any abrupt jumps at the boundaries.

The mixture generation procedure had the following parameters. For each target class in both training and test sets, there were 500 mixtures. The event presence rate was 0.5 (250 mixtures with target event present and 250 "mixtures" of only background). The event-to-background ratios (EBR) were -6, 0 and 6 dB. The EBR was defined as a ratio of average RMSE values calculated over the duration of the event and the corresponding background segment on which the event will be mixed, respectively. The background instance, the event instance, the event timing in the mixture, its presence flag and the EBR value were all selected randomly and uniformly. The data required to perform the generation of the exact mixtures (the filenames of the background and sound event, if present, the timing and the amplitude scaling factor of the event) was encoded in the so-called recipes. The recipes were generated randomly, but with a fixed seed of the random generator, allowing reproducibility.

The mixtures were generated by summing the backgrounds with the corresponding target event signals according to the recipes, with downsampling to 44100 Hz prior to summation in the case of a higher sampling rate. The resulting signals were scaled with a global empirical factor of 0.2, preserving the dynamics while avoiding clipping. The files were then saved in 24 bit format in order to avoid adding quantization noise.

The dataset is accompanied by a software package, which, given the default parameters, produces exactly the same mixture recipes and audio mixture files as in this dataset. It also allows for tuning the parameters in order to obtain larger and more challenging training datasets: number of mixtures, EBR values and event presence probabilities are adjustable.

The information needed to perform the split into training and test sets in terms of underlying source data was provided. The split of backgrounds was done in terms of recording location ID, according to the first fold of the DCASE 2016 task 1 setup, yielding 844 training and 277 test files. The sound events were split in terms of freesound.org user names. The ratio of target event examples was set to 0.71:0.29, and the split was performed in such a way that the isolated event counts are of a similar ratio. The resulting unique event counts are therefore the following:

- baby cry: 106 training, 42 test;
- glass break: 96 training, 43 test;
- gun shot: 134 training 53 test.

The baseline system follows the common implementation with the following specifics. For each of the target classes, there is a separate binary classifier with one output neuron with sigmoid activation, indicating the activity of the target class. The performance of the baseline system is evaluated using event-based error rate and

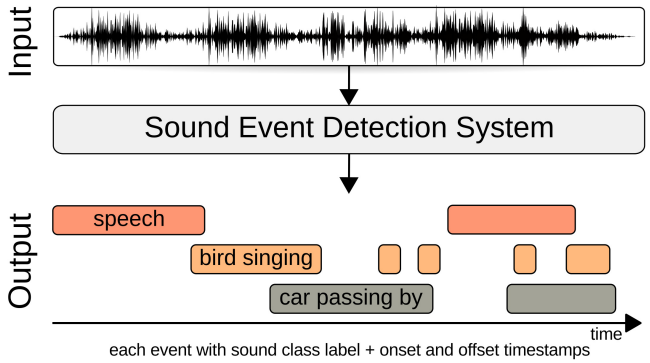

Figure 3: A schematic illustration of sound event detection in real-life audio addressed in Task 3.

event-based F-score as metrics using development dataset mixtures (provided training and test sets). Both metrics are calculated as defined in [27], using a collar of 500 ms and taking into account only the sound event onset. The performance of the baseline system is reflected in Table 3. The primary evaluation score for this task is the event-based error rate, and ranking of the systems submitted for the challenge is done using the average event-based error rate over the three classes.

5. TASK 3: SOUND EVENT DETECTION IN REAL-LIFE AUDIO

Task 3 evaluated the performance of sound event detection systems in multisource conditions similar to our everyday life, where the sound sources are rarely heard in isolation. A number of predefined sound event classes were selected, and systems are meant to detect the presence of these sounds, providing labels and timestamps to segments of the test audio, as illustrated in Fig. 3. In this task, there is no control over the number of overlapping sound events at each time, not in the training, nor in the test audio data.

The dataset used for this task is a subset of TUT Acoustic Scenes 2017, and is referred to as TUT Sound Events 2017. It consists of recordings of street acoustic scenes (city center and residential area) with various levels of traffic and other activity. The length of the audio is 3-5 minutes. The street acoustic scene was selected as representing an environment of interest for detection of sound events related to human activities and hazard situations.

Individual sound events in each recording were annotated by the same person using freely chosen labels for sounds, according to the annotation procedure described in [26]. Nouns were used to characterize the sound source, and verbs to characterize the sound production mechanism, using a noun-verb pair whenever this was possible. The annotator was instructed to annotate all audible sound events, decide the start time and end time of the sounds as he sees fit, and choose event labels freely.

The target sound event classes were selected so as to represent common sounds related to human presence and traffic. The selected sound classes for the task are: brakes squeaking, car, children, large vehicle, people speaking, and people walking. Mapping of the raw labels was performed, merging sounds into classes described by their source, for example "car passing by", "car engine running", "car idling", etc into "car", sounds produced by buses and trucks into "large vehicle", "children yelling" and "children talking" into

Table 4: Event instances per class in Task 3

Event label	Dev. set	Eval. set
brakes squeaking	52	23
car	304	106
children	44	15
large vehicle	61	24
people speaking	89	37
people walking	109	42
total	659	247

Table 5: Baseline system results for Task 3, segment-based metrics.

	Development set		Evaluation set	
	ER	F-score (%)	**ER**	F-score (%)
Overall	**0.69**	56.7	0.93	42.8
Class-wise performance				
brakes squeaking	0.98	4.1	0.92	16.5
car	0.57	74.1	0.76	61.5
children	1.35	0.0	2.66	0.0
large vehicle	0.90	50.8	1.44	42.7
people speaking	1.25	18.5	1.29	8.6
people walking	0.84	55.6	1.44	33.5

"children", etc. Due to the high level of subjectivity inherent to the annotation process, a verification of the reference annotation was done using these mapped classes. Three persons (other than the annotator) listened to each audio segment annotated as belonging to one of these classes, marking agreement about the presence of the indicated sound within the segment. Event instances that were confirmed by at least one person were kept, resulting in elimination of about 10% of the original event instances.

Partitioning of data into development and evaluation datasets was done based on the amount of examples available for each sound event class. Because the event instances belonging to different classes are distributed unevenly within the recordings, the partitioning of individual classes can be controlled only to a certain extent, but so that the majority of events are in the development set. A cross-validation setup provided in order to make results reported with this dataset uniform. The setup consists of four folds containing training and test subsets, and is made so that each recording is used exactly once as test data. While creating the cross-validation folds, the only condition imposed was that the test subset does not contain classes which are unavailable in the training subset. The number of instances for each event class in the development set is presented in Table 4.

The baseline system was tailored to a multi-class multi-label classification setup, with the network output layer containing sigmoid units that can be active at the same time. This way, multiple output units can indicate activity of overlapping sound classes. The results are evaluated using segment-based error rate and segment-based F-score as metrics, using a segment length of one second. The four cross-validation folds are treated as a single experiment: the metrics are calculated by accumulating error counts (insertions, deletions, substitutions) over all folds [27], not by averaging the individual folds nor the individual class performance. This method of calculating performance gives equal weight to each individual sound instance in each segment, as opposed to being influenced by class balance and error types [29]. The system trained and tested using the provided cross-validation setup obtained an overall error rate of 0.69 and an overall F-score of 56.7% on the development set, as shown in Table 5. On the evaluation dataset, the system obtained an error rate of 0.93 and an F-score of 42.8. For completeness, individual class performance is presented along the overall performance. The primary evaluation score for this task is the overall segment-based error rate, and ranking of the systems submitted for the challenge is also done using the same metric, calculated on the evaluation dataset.

6. TASK 4: LARGE-SCALE WEAKLY SUPERVISED SOUND EVENT DETECTION FOR SMART CARS

Task 4 evaluated systems for the large-scale detection of sound events using weakly labeled audio recordings. The audio comes from YouTube video excerpts related to the topic of transportation and warnings. The topic was chosen due to its industry relevance and the under use of audio in this context. The results will help define new grounds for large-scale sound event detection and show the benefit of audio for self-driving cars, smart cities and related areas. The task consisted of detecting sound events within 10-second clips and it was divided into two subtasks:

- Subtask A: Without timestamps (same as audio tagging, Fig 4)
- Subtask B: With timestamps (similar to Task 3, Fig 3)

The task employed a subset of AudioSet [30]. AudioSet consists of an ontology of 632 sound event classes and a collection of 2 million human-labeled 10-second sound clips drawn from YouTube videos. The ontology is specified as a hierarchical graph of event categories, covering a wide range of human and animal sounds, musical instruments and genres, and common everyday environmental sounds. To collect the dataset, Google worked with human annotators who listened, analyzed, and verified the sounds they heard within the YouTube 10-second clips. To facilitate faster accumulation of examples for all classes, Google relied on available YouTube metadata and content-based search to nominate candidate video segments that were likely to contain the target sound. Note that AudioSet does not come with precise time boundaries for each sound class within the 10-second clips and thus annotations are considered weak labels. Also, one clip may correspond to more than one sound event class. Task 4 relied on a subset of 17 sound events divided into two categories: *Warning* and *Vehicle*.

- *Warning sounds:* Train horn, Air horn Truck horn, Car alarm, Reversing beeps, Ambulance (siren), Police car (siren), Fire engine fire truck (siren), Civil defense siren, Screaming.
- *Vehicle sounds:* Bicycle, Skateboard, Car, Car passing by, Bus, Truck, Motorcycle, Train.

For both subtasks, the data was divided in two main partitions: development and evaluation. The development data was itself divided into training and test. Training had 51,172 clips, which are class-unbalanced and had at least 30 clips per sound event. Test had 488 clips, with at least 30 clips per class. A 10-second clip may have corresponded to more than one sound event class. The evaluation set had 1,103 clips, with at least 60 clips per sound event. The sets had weak labels denoting the presence of a given sound event within the audio, but with no timestamp annotations. For test and evaluation, strong labels (timestamp annotations) were provided for the purpose of evaluating performance on Subtask B.

The task rules did not allow the use of external data, such as other datasets. Similarly, it was not allowed to use other elements of the video from which the 10-sec clip was extracted, such as the

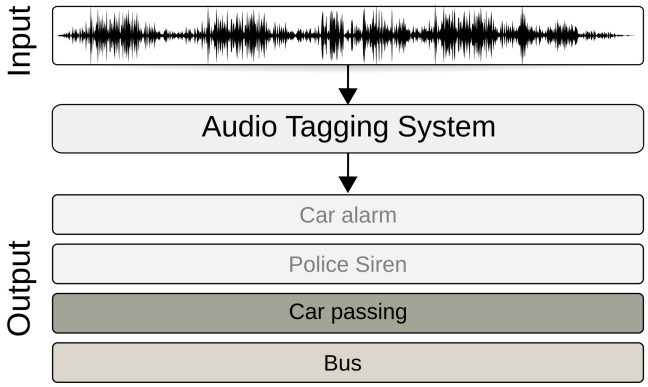

Figure 4: A schematic illustration of audio tagging addressed in Subtask A of Task 4.

rest of the video soundtrack, the video frames and the metadata (e.g. text, views, likes). Moreover, participants were not allowed to use the embeddings provided by AudioSet or other features that used external data indirectly, such as the ones derived from Transfer Learning. Additionally, only weak labels and none of the strong labels (timestamps) could be used for training the submitted system.

The evaluation metric of the two subtasks was different. For Subtask A, sound event detection without timestamps (audio tagging), we used F-score, precision and recall, where ranking of submitted systems was based on F-score. For Subtask B, sound event detection with timestamps, we used segment-based error rate (SBER) [27] and F-score, where ranking of submitted systems was based on SBER for segments of length one-second.

The baseline system shares the code base with other tasks, with detection decision based on the network output layer containing sigmoid units that can be active at the same time. The system also includes evaluation of the overall and class-wise results. The baseline was trained using the training set and tested using the test set and later also tested on the evaluation set. The results for the testing and evaluation sets are shown in Tables 6 for Subtask A and 7 for Subtask B.

7. CONCLUSIONS

The DCASE 2017 Challenge proposed four tasks relevant to current research in environmental sound classification. Compared to previous challenge, the current edition tackled two specific situations, namely detection of sound events that may appear very rarely, and the problem of using weak labels for training sound event detection systems. Of the established tasks, acoustic scene classification and sound event detection in real life audio were seen as important and yet to be solved research problems, worthy of inclusion in the ongoing work.

Through its public datasets and reporting of results, the challenge promotes open research and publications, disseminating the outcome to a large audience. The provided baseline system also offered a starting point for further development, along with a set comparison reference for each task.

8. ACKNOWLEDGMENT

Thanks to Rohan Badlani for his contribution to Task 4.

Table 6: Baseline system results for Task 4 - Subtask A, sound event detection without timestamps (audio tagging) based on micro-averaging.

	Development set (%)			Evaluation set (%)		
	F-score	Prec.	Rec.	**F-score**	Prec.	Rec.
Overall	**10.9**	7.9	17.6	18.18	15.0	23.07
Class-wise performance						
Train horn	0.0	0.0	0.0	14.1	100	7.6
Air horn, truck horn	0.0	0.0	0.0	0.0	0.0	0.0
Car alarm	0.0	0.0	0.0	0.0	0.0	0.0
Reversing beeps	0.0	0.0	0.0	0.0	0.0	0.0
Ambulance (siren)	0.0	0.0	0.0	0.0	0.0	0.0
Police car (siren)	35.8	29.1	46.6	38.8	32.6	47.8
Fire engine, fire truck (siren)	22.7	25.0	20.8	19.3	25.7	15.5
Civil defense siren	57.5	47.7	72.4	47.9	34.0	81.0
Screaming	0.0	0.0	0.0	0.0	0.0	0.0
Bicycle	0.0	0.0	0.0	4.2	100	2.1
Skateboard	0.0	0.0	0.0	0.0	0.0	0.0
Car	11.3	6.0	98.3	29.9	17.9	92.2
Car passing by	0.0	0.0	0.0	0.0	0.0	0.0
Bus	0.0	0.0	0.0	0.0	0.0	0.0
Truck	0.0	0.0	0.0	0.0	0.0	0.0
Motorcycle	0.0	0.0	0.0	14.2	100	7.6
Train	4.5	100	2.3	7.8	100	4.0

Table 7: Baseline system results for Task 4 - Subtask B, sound event detection with timestamps, based on segment-based error rate. The character [-] represents no prediction output by the system.

	Development set		Evaluation set	
	ER	F-score %	**ER**	F-score %
Overall	**1.02**	13.8	0.93	28.4
Class-wise performance				
Train horn	1.00	-	0.98	3.9
Air horn, truck horn	1.00	-	1.0	-
Car alarm	1.00	-	1.0	-
Reversing beeps	1.00	-	1.0	-
Ambulance (siren)	1.00	-	1.0	-
Police car (siren)	1.03	28.7	1.01	34
Fire engine, fire truck (siren)	1.02	8.4	0.98	16.5
Civil defense siren	0.69	58.2	0.64	67.4
Screaming	1.00	-	1.0	-
Bicycle	1.00	-	0.99	2.5
Skateboard	1.00	-	1.0	-
Car	5.9	21.1	1.75	46
Car passing by	1.00	-	1.0	-
Bus	1.00	-	1.0	-
Truck	1.00	-	1.0	-
Motorcycle	1.00	-	0.97	6.1
Train	1.00	0.5	0.99	1.8

9. REFERENCES

[1] M. Bugalho, J. Portelo, I. Trancoso, T. Pellegrini, and A. Abad, "Detecting audio events for semantic video search." in *Interspeech*, 2009, pp. 1151–1154.

[2] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321–329, Jan 2006.

[3] D. Stowell and D. Clayton, "Acoustic event detection for multiple overlapping similar sources," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustic (WASPAA)*, October 2015.

[4] S. Goetze, J. Schröder, S. Gerlach, D. Hollosi, J. Appell, and F. Wallhoff, "Acoustic monitoring and localization for social care," *Journal of Computing Science and Engineering*, vol. 6, no. 1, pp. 40–50, March 2012.

[5] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Trans. on Multimedia*, vol. 17, no. 10, pp. 1733–1746, October 2015.

[6] T. Virtanen, A. Mesaros, T. Heittola, M. Plumbley, P. Foster, E. Benetos, and M. Lagrange, *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*. Tampere University of Technology. Department of Signal Processing, 2016.

[7] D. Barchiesi, D. Giannoulis, D. Stowell, and M. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, May 2015.

[8] J.-J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.

[9] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Audio context recognition using audio event histograms," in *18th European Signal Processing Conference*, Aug 2010, pp. 1272–1276.

[10] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 1, pp. 142–153, Jan. 2015.

[11] B. Elizalde, A. Kumar, A. Shah, R. Badlani, E. Vincent, B. Raj, and I. Lane, "Experiments on the DCASE challenge 2016: Acoustic scene classification and sound event detection in real life recording," in *DCASE2016 Workshop on Detection and Classification of Acoustic Scenes and Events*, 2016.

[12] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real-life recordings," in *18th European Signal Processing Conference (EUSIPCO 2010)*, 2010, pp. 1267–1271.

[13] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22st ACM International Conference on Multimedia (ACM-MM'14)*, Nov. 2014.

[14] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, June 2017.

[15] J. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, and H. Van hamme, "An exemplar-based NMF approach to audio event detection," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct 2013, pp. 1–4.

[16] A. Mesaros, O. Dikmen, T. Heittola, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 151–155.

[17] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural network," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.

[18] B. Byun, I. Kim, S. M. Siniscalchi, and C.-H. Lee, "Consumer-level multimedia event detection through unsupervised audio signal modeling." in *INTERSPEECH*, 2012, pp. 2081–2084.

[19] B. Elizalde, G. Friedland, H. Lei, and A. Divakaran, "There is no data like less data: Percepts for video concept detection on consumer-produced media," in *Proceedings of the 2012 ACM international workshop on Audio and multimedia methods for large-scale video analysis*. ACM, 2012, pp. 27–32.

[20] Y. Xu, Q. Huang, W. Wang, P. Foster, S. Sigtia, P. J. Jackson, and M. D. Plumbley, "Unsupervised feature learning based on deep models for environmental audio tagging," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1230–1241, 2017.

[21] W. Han, E. Coutinho, H. Ruan, H. Li, B. Schuller, X. Yu, and X. Zhu, "Semi-supervised active learning for sound classification in hybrid learning environments," *PloS one*, vol. 11, no. 9, p. e0162075, 2016.

[22] A. Shah, R. Badlani, A. Kumar, B. Elizalde, and B. Raj, "An approach for self-training audio event detectors using web data," *arXiv preprint arXiv:1609.06026*, 2016.

[23] A. Kumar and B. Raj, "Weakly supervised scalable audio content analysis," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016.

[24] T. Heittola, A. Diment, and A. Mesaros, "DCASE2017 baseline system," https://github.com/TUT-ARG/DCASE2017-baseline-system, accessed June 2017.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[26] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*, 2016.

[27] ——, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/162

[28] T. Giannakopoulos, "pyaudioanalysis: An open-source python library for audio signal analysis," *PloS one*, vol. 10, no. 12, 2015.

[29] G. Forman and M. Scholz, "Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement," *SIGKDD Explor. Newsl.*, vol. 12, no. 1, pp. 49–57, nov 2010. [Online]. Available: http://doi.acm.org/10.1145/1882471.1882479

[30] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.