This is the published version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: http://openaccess.city.ac.uk/13650/

Link to published version: http://dx.doi.org/10.1109/TMM.2015.2428998

Copyright and reuse: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.
Detection and Classification of Acoustic Scenes and Events

Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Member, IEEE, Mathieu Lagrange, and Mark D. Plumbley, Fellow, IEEE

Abstract—For intelligent systems to make best use of the audio modality, it is important that they can recognize not just speech and music, which have been researched as specific tasks, but also general sounds in everyday environments. To stimulate research in this field we conducted a public research challenge: the IEEE Audio and Acoustic Signal Processing Technical Committee challenge on Detection and Classification of Acoustic Scenes and Events (DCASE). In this paper, we report on the state of the art in automatically classifying audio scenes, and automatically detecting and classifying audio events. We survey prior work as well as the state of the art represented by the submissions to the challenge from various research groups. We also provide detail on the organization of the challenge, so that our experience as challenge hosts may be useful to those organizing challenges in similar domains. We created new audio datasets and baseline systems for the challenge; these, as well as some submitted systems, are publicly available under open licenses, to serve as benchmarks for further research in general-purpose machine listening.

Index Terms—Audio databases, event detection, machine intelligence, pattern recognition.

I. INTRODUCTION

EVER since advances in automatic speech recognition (ASR) were consolidated into working industrial systems [1], the prospect of algorithms that can describe, catalogue and interpret all manner of sounds has seemed close at hand. Within ASR, researchers continue to advance recognition quality, in challenging audio conditions such as distant speech against noisy backgrounds [2]. Elsewhere, advances in Music Information Retrieval (MIR) have brought us systems that can transcribe the notes and chords in music [3], or identify the track title and artist from a low-quality sound snippet [4]. However, speech and music are just two of the many types of sound that can be heard in a typical indoor or outdoor environment. Increasingly, machines deployed in diverse environments can hear—whether they be mobile phones, hearing aids or autonomous robots—but can they make sense of what they hear?

Sound is often a useful complement to modalities such as video, carrying information not otherwise present such as information from speech and birdsong. Sound can also be more convenient to collect, e.g. on a mobile phone. Information gathered from a semantic audio analysis can be useful for further processing such as robot navigation, user alerts, or analyzing and predicting patterns of events [5]. Beyond listening devices, the same technologies have applications in cataloguing/searching audio archives, whose digital collections have grown enormously in recent decades [6]. Audio archives often contain a rich diversity of speech, music, animal sound, urban soundscapes, ethnographic recordings and more, yet their accessibility currently lags behind that of text archives.

In order to stimulate research in machine listening for general audio environments, in 2012–2013 we organized a research challenge under the auspices of the IEEE Audio and Acoustic Signal Processing Technical Committee: the challenge on Detection and Classification of Acoustic Scenes and Events (DCASE). This challenge focused on two concrete but relatively general types of task that a general machine listening system would carry out: recognizing the general environment type (the acoustic "scene"), and detecting and classifying events occurring within a scene.

These tasks which we describe as "machine listening" tasks can also be considered to come under the umbrella of computational auditory scene analysis (CASA) [7]. This nomenclature refers back to Bregman's influential work on human "auditory scene analysis" capabilities [8], and thus CASA is often taken to imply an approach which aims either to parallel the stages of processing in human audition, and/or to mimic the observed phenomena of human audition (which may include illusions such as the "missing fundamental") [7, Chapter 1]. These human-centric aims do not directly reflect our goal here, which is to develop systems that can extract semantic information about the environment around them from audio data.

The purpose of this paper is to give a complete description of the challenge, for two purposes: firstly to acquaint the reader with the state of the art in machine listening, and secondly to provide guidance and lessons learnt for the benefit of people.
running research challenges in future. In the following, we first
give some research background in the topic, and previous chal-
lenge that have been conducted in neighboring areas. Then
we give detail on the experimental design of the tasks we de-
derived from the results such as the level of task difficulty,
and in particular we compare the “live” and “synthetic” vari-
ants of our event detection challenge. Finally we consider the
outlook for machine listening in light of the challenge: the state
and in particular we compare the “live” and “synthetic” vari-
\[17\] a system is proposed to use the matching pursuit
algorithm to obtain an effective time-frequency feature se-
lection that are afterwards used as supplement to MFCCs to
perform environmental sound classification.

The goal of acoustic event detection is to label temporal re-
gions within an audio recording, resulting in a symbolic de-
scription such that each annotation gives the start time, end
time and label for a single instance of a specific event type. It is
related in spirit to automatic music transcription [3], and also to
speaker diarization, which similarly recovers a structured anno-
tation of time segments but focusses on speech “turns” rather
than individual events [18]. The majority of work in event de-
tection treats the sound signal as monophonic, with only one
event detectable at a time [19], [20]. In general audio scenes,
events may well co-occur, and so polyphonic event detection
(allowing for overlapping event regions) is desirable. However,
salient events may occur relatively sparsely and there is value
even in monophonic detection. There has been some work on
extending systems to polyphonic detection [21]. Event detec-
tion is perhaps a more demanding task than scene classifica-
tion, but at the same time heavily intertwined. For example, in-
formation from scene classification can provide supplementary
contextual information for event detection [22]. Many proposed
approaches can be found in the literature among which spectro-
gram factorization techniques tend to be a regular choice. In [23]
a probabilistic latent semantic analysis (PLSA) system, a closely
related approach to NMF, was proposed to detect overlapping
sound events. In [20] a convolutive NMF algorithm applied on
a Mel-frequency spectrum was tested on detecting non-over-
lapping sound events. Finally, a number of proposed systems
focus on the detection and classification of specific sound events
from environmental audio scenes such as speech [24], bird-
song [25], musical instrument and other harmonic sounds [26],
pornographic sounds [27] or hazardous events [28].

The issue of polyphony is pertinent to both of the above tasks,
since audio scenes are polyphonic (multi-source) in general. As
with music, it is possible to perform some analysis on the audio
signal as a whole without considering polyphony, though it is
likely that some benefit can be obtained from considering the
component sources that make up the signal. Such a compo-
nent-wise analysis is analogous to the auditory streaming that
occurs in Bregman’s model of human audition [8]. In speech
recognition applications it can often be assumed that there is
one dominant source that should be the focus for analysis [24],
but this is not the case for general audio scenes. One strategy
to handle polyphonic signals is to perform audio source separa-
tion, and then to analyze the resulting signals individually [29],
[21]. However, note that the computational equivalent of au-
ditory streaming does not necessarily require a reconstruction
of the individual audio signals—Bregman does not claim that
human listeners do this—but could work with some mid-level
representation such as a multisource probabilistic model [30]. Source-separation for general-purpose audio is still a long way from being a solved problem [31]. For example, the evaluation used in recent challenges for “speech recognition in multisource environments” did not require submitted algorithms to perform audio source-separation: evaluation was performed on speech transcription output. Submitted algorithms generally did not involve a source-separation step, many used spatial or spectral noise suppression in order to focus on one source rather than separating all sources [32].

In machine listening, public evaluation and benchmarking of systems serves a valuable role. It enables objective comparison among various proposed systems, and can also be used for studying performance improvements throughout the years. Many such challenges have been centred on speech. For example, the DARPA EARS Rich Transcription evaluations (2002–2009) focussed on speaker-diarization tasks, applied to broadcast news as well as recordings of meetings [18]. The MIREX challenges (2005–present) evaluated MIR systems for their performance on specific musical tasks such as melody transcription or rhythm tracking [33]. The SISEC challenges (2007–present) focussed on audio source separation algorithms, both for speech mixtures and for music [31]. The CHiME challenges (2011, 2013) focussed on speech recognition in noisy multi-source sound environments [2]. None of the aforementioned challenges directly relates to the general-purpose machine listening tasks we consider here. Some of them use broadly similar task outlines (e.g. classification, diarization), but often use domain-specific evaluation measures (e.g. speech transcription accuracy, audio separation quality). They also attract contributions specialised to the particular audio domain.

For the present purposes, the most closely-related challenge took place in 2006 and 2007, as part of the CLEAR evaluations conducted during the CHIL project [34]. Several tasks on audio-only, video-only or multimodal tracking and event detection were proposed, among them an evaluation on “Acoustic Event Detection and Classification”. The datasets were recorded during several interactive seminars and contain events related to seminars (speech, applause, chair moving, etc). From the datasets created for the evaluations, the “FBK-Irst database of isolated meeting-room acoustic events” has widely been used in the event detection literature; however, the aforementioned dataset contains only non-overlapping events. The CLEAR evaluations, although promising and innovative at the time, were discontinued with the end of the CHIL project.

One further related challenge in audiovisual research is TRECVID Multimedia Event Detection, where the focus is on audiovisual, multi-modal event detection in video recordings [35]. Some researchers have used the audio extracted from the audiovisual TRECVID data in order to evaluate their systems; however a dataset explicitly developed for audio challenges would offer a much better evaluation framework since it would be much more varied with respect to audio.

### III. The Challenge

In the present section we describe the evaluation design for our challenge tasks. Before this, we describe the requirements gathering process that we conducted, and the considerations that fed into our final designs.

#### A. Requirements Gathering

As described above, the tasks considered in this challenge relate to those explored in previous experimental studies, and to some degree to those explored in previous evaluation campaigns. There is therefore a body of literature from which to draw potential task designs. Importantly, however, the task designs were developed through a period of community discussion, primarily via a public email list. This was crucial to ensure that the designs had broad relevance to current research, and did not unfairly penalise potential participants. An example of the latter is in the choice of evaluation measures for event detection: there was a debate about which evaluation measures were most appropriate, as well as issues such as the appropriate level of granularity in frame-wise evaluation. It was this discussion that led to the decision to report three different evaluation measures for event detection (see Section III-C3). Other issues discussed included annotation data formats, the nature of synthetic sequences, and the use of other existing datasets.

Our motivation was to design challenge tasks to reflect useful general-purpose inferences that could be made in an everyday audio environment, pertinent to a broad range of machine listening applications. Our focus was on everyday sounds beyond speech and music, since the latter are already well-studied. We also wished to design tasks for which performance could be improved without necessarily being overly reliant on other processing components such as high-quality source separation or ASR. We decided to design challenge tasks separately for scene classification and for event detection and classification, using data relating to urban and office environments.

Many applications of machine listening relate to processing embodied in a fixed hardware setup, such as a mobile phone or a robot. This differs from applications such as audio archive analysis, for which a system must be robust to signal modifications induced by variation of microphones and preprocessing across the dataset [36]. For embodied machine listening, aspects such as the microphone frequency response will be constant factors rather than random factors. We chose to design our tasks each with a fixed configuration of recording equipment.

One pertinent question was whether existing data could be used for our evaluation, or whether it would be important to create new datasets. Previous studies have used relatively small datasets; further, some of these are not publicly available. Alternatively, online archives such as Freesound hold a large amount of soundscape data. However, these vary widely in recording conditions, recording quality and file format [6], [37], and so were unsuitable for our experimental aim to evaluate systems run with consistent audio front-end. Thus it was important to make new recordings. This gave us various advantages: as well as allowing us to control conditions such as the balance of sound types, it also meant that we were able to create private testing data unseen by all participants, to ensure that there was no inadvertent overfitting to the particulars of the task data. Conversely, it meant we could release the public data under a liberal open-content license, as a resource for the research community even beyond our immediate focus.

Given that everyday sound environments are polyphonic—multiple sound events can occur at the same

---

time—with varying degrees of density, and given that general audio source separation is still a difficult problem, it was important to design event detection task(s) so that we could explore the effect of polyphony on event detection systems. Such systems might be designed with a simplifying monophonic assumption; with source separation used to feed multiple monophonie analyses; or with full polyphonic inference. There is little data available to suggest how these different strategies perform as the event density varies. In order to have experimental control over the event density, we chose two parallel approaches to creating event detection audio data. In one, we made live recordings of scripted monophonic event sequences in controlled environments. In the other, we made live recordings of individual events, and synthetically combined these (along with ambient background recordings) into synthetic mixtures with parametrically controlled polyphony. We describe these approaches further in Section III-C.

In December 2012 we conducted a survey of potential participants to characterise their preferred software platforms. This indicated that most participants wished to use Matlab, Python, R or C/C++ to create their submissions. However, all of these frameworks come in multiple versions across multiple operating systems, and it can be difficult to ensure that code running on one system will run correctly on another. To minimise the risk of such issues, we created and published a Linux virtual machine which participants could use during development, and which would also be the environment used to run the submission. For this we used VirtualBox software which runs on all common operating systems, together with a disk image based on Xubuntu 12.10 Linux. The disk image was augmented by adding the public datasets into the home folder, and also by installing Python, R and C/C++, as well as some common audio-processing toolboxes for each environment. The resulting disk image is available online from our research repository. Due to software licensing constraints we could not include Matlab in the disk image, and so we handled Matlab-based submissions separately from the virtual machine.

We next describe the finalised design and data collection for the scene classification task, and for the event detection tasks.

B. Scene Classification Task (SC)

Acoustic scene classification can be considered as a single-label classification task (see Section II). Alternative designs are possible, such as classification with hierarchical labels [38], unsupervised clustering of audio scenes, or multi-label “auto-tagging” [39]. However, single-label classification is the design most commonly seen in prior literature in acoustic scene recognition [14]–[17], [10], and also lends itself to clear evaluation measures. We therefore designed the SC task as a train/test classification task, of similar design to previous audio classification evaluations [33].

We created datasets across a pre-selected list of scene types, representing an equal balance of indoor/outdoor scenes in the London area: bus, busystreet, office, openairmarket, park, quietstreet, restaurant, supermarket, tube, tubeestation. The limitation to the London area was a pragmatic choice, known to participants. We made sure to sample across a wide range of central and outer London locations, in order to maximise generalisability given practical constraints. To enable participants to further explore whether machine recognition could benefit from the stereo field information available to human listeners [7, Chapter 5], we recorded in binaural stereo format using Soundman OKM II in-ear microphones.

For each scene type, three different recordists (DG, DS, EB) visited a wide variety of locations in Greater London over a period of months (Summer and Autumn 2012), and in each scene recorded a few minutes of audio. We ensured that no systematic variations in the recordings co-varied with scene type: all recordings were made in moderate weather conditions, and varying times of day, week and year, and each recordist recorded each scene type.

We then reviewed the recordings to select 30-second segments that were free of issues such as mobile phone interference or microphone handling noise (totaling around 50% of the recorded duration), and collated these into two separate datasets: one for public release, and one private set for evaluating submissions. The duration of 30 seconds is comparable with that of other datasets in this topic, and was judged to be long enough to contain sufficient information in principle to distinguish the classes. The segments are stored as 30-second WAV files (16 bit, stereo, 44.1 kHz), with scene labels given in the filenames. Each dataset contains 10 examples each from 10 scene types, totalling 50 minutes of audio per dataset. The public dataset is published online under a Creative Commons CC-BY licence.

For the SC task, systems were evaluated with 5-fold stratified cross validation. Our datasets were constructed to contain a balance of class labels, and so classification accuracy was an appropriate evaluation measure [40]. The raw classification (identification) accuracy and standard deviation were computed for each algorithm, as well as a confusion matrix so that algorithm performance could be inspected in more detail.

1) Baseline System for Scene Classification: The “bag-of-frames”MFCC + GMM approach to audio classification (see Section II) is relatively simple, and has been criticized for the assumptions it incurs [41]. However, it is quite widely applicable in a variety of audio classification tasks. Aucouturier and Pachet [10] specifically claim that the MFCC + GMM approach is sufficient for recognizing urban soundscapes but not for polyphonic music (due to the importance of temporal structure in music). It has been widely used for scene classification among other recognition tasks, and has served as a basis for further modifications [17]. The model is therefore an ideal baseline for the Scene Classification task.

Code for the bag-of-frames model has previously been made available for Matlab. However, for maximum reproducibility we wished to provide simple and readable code in a widely-used programming language. The Python language is very widely used, freely available on all common platforms, and is notable for its emphasis on producing code that is readable by others. Hence we created a Python script embodying the MFCC + GMM classification workflow, publicly available

---

2[Online]. Available: http://c4dm.eecs.qmul.ac.uk/xdrr/handle/123456789/32
3[Online]. Available: http://c4dm.eecs.qmul.ac.uk/xdrr/handle/123456789/29
under an open-source licence, and designed for simplicity and ease of adaptation [42].

C. Event Detection Tasks (OL, OS)

For the Event Detection tasks, we addressed the problem of detecting acoustic scenes in an office environment, making use of existing office infrastructure within Queen Mary University of London, and also providing a continuation of the CLEAR evaluations [43], which also addressed the task of event detection in an office environment. In order to encourage wide participation, and also to explore the challenge of polyphonic audio scenes, we designed two subtasks: event detection of non-overlapping sounds (Event Detection - Office Live) and event detection of overlapping sounds (Event Detection - Office Synthetic). In both cases, systems are required to detect predominant events in the presence of background noise.

1) Recorded Dataset (OL): After a consultation period with members of the acoustic signal processing community, for the Event Detection - Office Live (OL) task we created recordings of office scenes, consisting of the following 16 classes: door knock, door slam, speech, laughter, clearing throat, coughing, drawer; printer; keyboard click, mouse click, object (pen, pencil, marker) on table surface, switch, keys (put on table), phone ringing, short alert (beep) sound, page turning.

Recordings were made in a number of office environments at Queen Mary University of London, using rooms of different size and with varying noise level or number of people in the room. We created three datasets: a training, a development, and a test dataset. Training recordings consisted of instantiations of individual events for every class. The development (validation) and test datasets consist of roughly 1 min long recordings of scripted every-day audio events. Scripts were created by random ordering of event types; we recruited a variety of participants to perform the scripts. For each script, multiple takes were used, and we selected the best take as the one having the least amount of unscripted background interference. Overall, the OL training dataset includes 24 recordings of individual sounds per class; the development dataset includes 3 recordings of scripted sequences; and the test set consists of 11 scripted recordings (the recording environments in the development and test datasets are non-overlapping).

Regarding equipment, recordings were made using a Sound-field microphone system, model SPS422B, able to capture 4-channel sound in B-format. The 4-channel recordings were converted to stereo (using the common “Blumlein pair” configuration). B-format recordings were stored along with the stereo recordings, with scope for future challenges to be extended to full B-format and take into account spatial information.

Given the inherent ambiguity in the annotation process (especially for annotating offsets), we created two sets of annotations. Annotators were trained to use Sonic Visualiser to use a combination of listening and inspecting waveforms/spectrograms to refine the onsets and offsets of each sound event. We then examined the two annotations per recording for consistency, and performed evaluations using an average of both annotations. The OL training dataset and the development dataset, both consisting of B-format recordings, stereo recordings, and annotations, were released under a Creative Commons license.

2) Synthetic Dataset (OS): We also decided that this challenge presented a good opportunity to study the relevance of considering artificial scenes built from a set of isolated events different from those of the training corpus. Though we admit that it is important to evaluate machine listening systems using real audio recordings, the potential gains from using artificial scenes as part of evaluation are numerous: ease of annotation, ability to generate many scenes with similar properties in order to gain better statistical significance, control of the complexity in terms of events overlap, strength of the background, etc. This will potentially help the designers of machine listening systems to better understand the behavior of those systems.

As in other domains, using synthetic data may lead to biased conclusions. It is for example well known that Independent Component Analysis (ICA) approaches in microphone arrays perform really well in separating the different sources within an additive non-convolutive mixtures but the input signal follows directly the mixture model assumed by those approaches. Special care was therefore taken in order to minimize the amount of artificial regularity induced by the generating system that could provide unrealistic benefits to some evaluated machine listening systems.

The scene synthesizer we considered here is able to create a large set of acoustic scenes from many recorded instances of individual events. The synthetic scenes are generated by randomly selecting, for each occurrence of each event we wish to include, one representative excerpt from the natural scenes, then mixing all those samples over a natural texture-like background with no distinctive sound events. The distribution of events in the scene is also random, following high-level directives that specify the desired density of events. The average Signal to Noise Ratio (SNR) of events over the background texture is also specified and is the same for all event types, unlike in the OL scenes. This is a deliberate decision taken to avoid issues with the annotation of potentially non perceptible events drowned in the background. In order to avoid issues with artificial spatialization, the recordings of individual events were mixed down to mono as an initial step.

The resulting development and testing datasets consist of 12 synthetic mono sequences with varying durations, with accompanying ground-truth annotations. Three subsets were generated with increasing levels of complexity in terms of event density: 4 recordings have a ‘low’ event density of 1.11, 4 recordings have a ‘medium’ event density of 1.27, and 4 recordings have a ‘high’ event density of 1.81. Three SNR levels of events over the background texture were used: −6 dB, 0 dB, and 6 dB.

3) Metrics: Following consultation with acoustic signal processing researchers, three types of evaluations were used for the OL and OS event detection tasks, namely frame-based, event-based, and class-wise event-based evaluations. Frame-based evaluation was performed using a 10 ms

[6][Online]. Available: http://code.soundsoftware.ac.uk/projects/smacpy

step and metrics were averaged over the duration of the recording. The main metric used for the frame-based evaluation was the acoustic event error rate (AEER) used in the CLEAR evaluations [43] 

\[ AEER = \frac{D + I + S}{N} \]  

where \( N \) is the number of events to detect for that specific frame, \( D \) is the number of deletions (missing events), \( I \) is the number of insertions (extra events), and \( S \) is the number of event substitutions, defined as \( S = \min(D, I) \). Additional metrics include the Precision, Recall, and F-measure (P-R-F). By denoting as \( r \), \( e \), and \( c \) the number of ground truth, estimated and correct events for a given 10 ms frame, the aforementioned metrics are defined as 

\[ P = \frac{r}{e}, \quad R = \frac{r}{c}, \quad F = \frac{2PR}{P + R}. \]  

For the event-based metrics, two types of evaluations took place, an onset-only and an onset-offset-based evaluation. For the onset-only evaluation, each event was considered to be correctly detected if the onset was within a 100 ms tolerance. This tolerance value was agreed upon during the community discussion via the challenge mailing list. It was argued that having a tolerance smaller than 100 ms would lead to poor results particularly in the case of ill-defined onsets and offsets for non-percussive events. For the onset-offset evaluation, each event was correctly detected if its onset was within a 100 ms tolerance and its offset was within 50% range of the ground truth event’s offset w.r.t. the duration of the event. Duplicate events were counted as false alarms. The AEER and P-R-F metrics for both the onset-only and the onset-offset cases were utilized.

Finally, in order to ensure that repetitive events did not dominate the evaluation of an algorithm, class-wise event-based evaluations were also performed. Compared with the event-based evaluation, the AEER and P-R-F metrics are computed for each class separately within a recording and then averaged across classes. For example, the class-wise F-measure is defined as 

\[ F' = \frac{1}{K} \sum_k F_k \]  

where \( F_k \) is the F-measure for events of class \( k \). Matlab code for the metrics can be found online.\(^1\)

4) Baseline System: We created a baseline system for both event detection tasks based on the non-negative matrix factorization (NMF) framework. NMF has been shown to be useful for modeling the underlying spectral characteristics of sources hidden in an acoustic scene [23], and can also support overlapping events, making it suitable for both the OL and OS tasks. We chose to design a supervised method for event detection, using a pre-trained dictionary of acoustic events [42].

The baseline method is based on NMF using the Kullback-Leibler divergence as a cost function [44]. As a time-frequency representation, we used the constant-Q transform with a log-frequency resolution of 60 bins per octave [45]. The training data is normalized to unity variance and NMF is used to learn a set of \( N \) bases for each class. The numbers of bases tested is 5, 8, 10, 12, 15, 20 and 20, the latter corresponding to learning individually one basis per training sample, for all 20 samples. Putting together the sets for all classes, we built a fixed dictionary of bases used subsequently to factorize the normalized input test data.

Formally, if we denote as \( \hat{V} \in \mathbb{R}^{G \times T} \) the constant-Q spectrogram of a test recording (\( \Omega \): number of log-frequency bins; \( T \): number of time frames), \( W \in \mathbb{R}^{N \times G \times N \times N} \) the pre-extracted dictionary and \( H \in \mathbb{R}^{N \times T} \), the NMF model attempts to approximate \( \hat{V} \) as a product of \( W \) and \( H \) [44].

\[ H \leftarrow H \otimes \frac{W^T((WH)^{-1} \otimes \hat{V})}{W^T}. \]  

In order to detect sound events, we sum together the (non-binary) activations per class obtained from \( H \). Finally, a threshold \( \theta \) is chosen to binarize the real-valued activations per class (i.e. for each row of \( H \)), in order to yield a sequence of estimated overlapped events. The value of \( \theta \) is the same for all event classes; the optimal \( N \) and \( \theta \) values were chosen empirically by maximizing the \( F \)-measure for the two annotations on the development set. Smoothing on the activations was also tested with no clear improvements. The baseline event detection system was made available to challenge participants under an open-source license.\(^2\)

D. Challenge organization

The full timeline for the challenge organization is given in Table I. Some of the items included of the timeline will be obvious to an outside observer. However, there are some aspects of the timeline and the workload which we believe merit emphasis, listed as follows.

- There were two periods which required the most time commitment from the organizing team: creating the datasets, and running the code submissions. In particular, as has been remarked by organizers of related challenges [33], no matter how many precautions are taken to ensure people submit code that will run on the organizers’ hardware (formal specifications, published virtual machine), it often requires many person-hours of attention before submitted code will run properly. This will be discussed further below. Recording the datasets also took significant time: this was not just the audio recording itself, but also the supervision of annotators, and the listening sessions and manual inspection to ensure data quality.

- We found it extremely useful to ask people to let us know of their intentions, in order to help us plan. In December 2012 we surveyed the community for indicative data about the level of interest in task participation, as well as the preferences for programming languages and operating systems. This information fed directly into our design of a Linux virtual machine for people to test their code. Then in March 2013 we asked participants to email us announcing their

\(^1\)[Online]. Available: https://code.soundsoftware.ac.uk/projects/aasp-dcase-metrics

\(^2\)[Online]. Available: http://code.soundsoftware.ac.uk/projects/d-case-event
intention to take part (with no commitment implied). This enabled us to plan resources, and to follow up on expected submissions that went astray. We received 20 notifications of intention to submit, some corresponding to multiple task submissions; of these only three did not eventually submit.

- One aspect of the timeline that could have been improved was the long wait between collating the results and releasing them publicly. It meant that participants could not compare and contrast results while their systems were “fresh in their minds”. However, this was due to our decision to co-ordinate with WASPAA 2013, which was an ideal forum for discussion of the challenge outcomes.

Regarding the execution of code submissions, our publication of a virtual machine as a standard platform certainly reduced the number of compatibility issues we had to deal with. However, there remained various software issues we encountered when running the code submissions, listed as follows:

- A frequent issue in Matlab submissions was opening the training annotation text files (for reading) using mode ‘r+’ (which is for reading and writing). This fails when files are read-only. We had set the test data as read-only, and in the specification we had stated that submissions must not write data in the test folder.

- One submission had been developed using Matlab on Windows; when we ran it using the same version of Matlab, but on Linux, it got rather poor results, which we initially attributed to overfitting. It later emerged that the poor performance was because a Matlab toolbox exhibited a bug only when running on Linux.

- On the virtual machine, there were occasional problems with version mismatch between Dynamic Link Libraries (DLLs). Such issues were reduced but not completely eliminated with the use of the virtual machine, often because participants did not fully test using the virtual machine, or occasionally added late modifications after testing.

- One submission output space-separated results rather than tab-separated. This was contrary to the published specifications but easy to miss in manual checking.

- Some submissions contained subtle bugs in data parsing. One submission accidentally ignored the last line of every text file it read, meaning that it output 19 decisions for each testing fold rather than 20. This wasn’t detected early on (because the output was correctly-formatted and could be scored), but only at the point where an overall confusion matrix was compiled. A different submission involved a script which parsed the text output from an executable. When the script failed to parse the text, it always decided on the last class in the list–failure was only detected in the large number of “tubestation” outputs.

Some of these issues (e.g. the data format issues) could have been prevented by providing unittests which participants must pass before submitting.

Earlier in the process, we also encountered a data issue: after we published the development datasets, a community member on the mailing list alerted us to a formatting error in some of the annotations (the text label *doorknock* was used in some places rather than the official label *knock*). Such issues occur despite the multiple steps of checking we performed before release. We updated the dataset to correct the issue, re-released it and confirmed this to the participants.

We required each submitted system to be accompanied by an extended abstract describing the system. We experienced no issues in publishing these abstracts; however in future evaluations we would consider explicit open-access licensing of the abstracts for greater clarity.

### IV. Submitted Systems

Overall, 11 systems were submitted to the scene classification (SC) task, 7 systems were submitted to the office live (OL) event detection task, and 3 systems to the office synthetic (OS) event...
The systems submitted for the scene classification task are listed in Table II, along with a short description of each system. Citations are to the extended abstracts giving further technical details about each submission. The methods for scene classification are discussed further in a tutorial article [64], while in Section V-A we will expand on some aspects of scene classification methods when considering which approaches led to strong performance.

The systems submitted for the event detection tasks are listed in Table III, along with a short description of each system. Citations are to the extended abstracts giving further technical details about each submission. Fig. 1 shows the processing chain adopted by the submitted algorithms. The main processing nodes are the feature computation and the classification for which a variety of implementations are considered. Optionally, the audio data can be pre-processed for example to reduce the influence of background noise, and the decisions given by the classifiers can be smoothed to reduce unrealistic transitions between events.

The system designs for each submission are now described.

- **CPS**: The CPS submission follows a scheme that combines segmentation, feature extraction, and classification. First, various frequency-based and time-based features are extracted. The audio stream is subsequently segmented using a speech segmenter that uses energy-based features. Each segment is then assigned to a class using a generalised likelihood ratio test classifier.

- **DHV**: The DHV submission was created for both the OL and OS tasks. It follows a generative classification scheme using HMMs with multiple Viterbi passes. Firstly,
MFCCs are extracted as features, and used as input to continuous-density HMMs (each state corresponds to an event class, including background noise). Polyphonic detection is achieved by performing consecutive passes of the Viterbi algorithm.

- **GVV**: The GVV submission uses a dictionary-based model using NMF. Firstly, a dictionary is created using samples from training set (called *exemplars*), using mel-magnitude spectrograms as time-frequency representations. The input spectrogram is projected onto the dictionary using NMF using the Kullback-Leibler divergence. The resulting event probability estimates are post-processed using an HMM containing a single state per event.

- **NVM**: The NVM submission follows a two-step classification scheme. At the first step, a large variety of audio features that capture temporal, spectral or auto-correlation properties of the signal are fed to two classifiers: a two-layer HMM and a random forest classifier. Another HMM is then used to combine the predictions.

- **NR2**: The NR2 submission follows a discriminative classification scheme implemented with support vector machines (SVMs). The classifier is fed with MFCCs that are computed using either the original signal or a noise-reduced one. The decisions coming from the classified versions are then combined and smoothed to reduce short transitions.

- **SCS**: The SCS submission follows a generative classification scheme with a 2-layer HMM decoding. The classifier is fed with 2 dimensional Gabor features (Time / Frequency) that allows percussive events to be nicely modelled. Before feature computation, the audio signal is enhanced using a noise suppression scheme that estimate the noise power spectral density and remove it in the spectral domain.

- **VVK**: The VVK submission follows a generative classification scheme with a GMM decoding. GMM models for each class of events and the background are first trained with MFCCs. The event models are next re-estimated in order to reduce the impact of background frames on the model likelihoods. At decoding the likelihoods are smoothed using a moving average filter and thresholded to produce the prediction.

- **Baseline**: A detailed description of the Baseline system is given in Section III-C.

V. RESULTS

A. SC Results

Fig. 2 shows the overall performance of submitted systems for the scene classification task. The baseline system achieved an accuracy of 55%; most systems were able to improve on this, although our significance tests were able to demonstrate a significant improvement over baseline only for the strongest four systems. The results indicate that level of difficulty for the task was appropriate: the leading systems were able to improve significantly upon the baseline, yet the task was far from trivial for any of the submitted systems. Also, the sizes of the error bars indicate that performance across the five folds was broadly consistent, indicating that the dataset was not overly heterogeneous. However, the statistical tests did not demonstrate significant differences between various systems (depicted by the large overlap of boxes in Fig. 2), which implies that a larger dataset may have enabled a more fine-grained ranking of systems. The results for this SC task are further analyzed in a tutorial article [64]. For that reason, here we discuss briefly the state of the art reflected in the SC task outcomes, allowing us to expand further on the OL/OS task outcomes in the next section.

The majority of the submitted systems used discriminative training, with many of the strong performers using an SVM as the final classifier. Further, most of the leading results were obtained by those who captured medium-range temporal information in the features used for classification. Four of the five highest-scoring systems did this: Roma *et al.* [56] captured temporal repetition and similarity using “recurrence quantification analysis”; Rakotomamonjy and Gasso [55] used gradient features from image-processing; Geiger *et al.* [48] extracted features from linear regression over time; Chum *et al.* [46] trained a HMM. Each of these is a generic statistical model for temporal evolution, whose fitted parameters can then be used as features for classification.
From the perspective of CASA, it is notable that none of the submitted systems used any kind of decomposition of each audio scene into auditory streams. We suggest that this is not due to any inherent difficulty in decomposing audio scenes, since automatic classification does not require “listening-quality” outputs from such preprocessing. Instead it seems likely that it is more difficult to design a classification workflow that makes use of structured scene analysis outputs, whose data may for example be sets of labelled intervals rather than time-series statistics. Two submissions made use of event detection as part of preprocessing, which does yield a structured parse of the audio scene [51], [52]. Those authors then used summary statistics from the density/strength of event detections as features. We propose that further refinement and development of this strategy may be a fruitful area for future work, perhaps via more sophisticated temporal summary statistics such as those noted above.

Also notable is that the submissions with the more perceptually-motivated features—auditory spectrogram [54] and cochleagram [49]—did not lead to the strongest results. Nor did the unsupervised feature learning of [51]. The various ways to approach audio feature design—perceptual, acoustic, statistical—each have their merits. Based on the present evaluation, we note only that the more sophisticated audio features did not yield a decisive advantage over simpler features.

We tested a simple majority-vote classifier from the pool of SC submissions, constructed by assigning to an audio recording the label that was most commonly returned by other methods. This attained a strong result, indicated as “MV” in the figure: 77% accuracy, slightly better than the leading individual results. Nor did the supervised feature learning of [51]. The various ways to approach audio feature design—perceptual, acoustic, statistical—each have their merits. Based on the present evaluation, we note only that the more sophisticated audio features did not yield a decisive advantage over simpler features.

Table IV shows a confusion matrix for the scene labels as rounded to the nearest integer.

### Table IV

<table>
<thead>
<tr>
<th>Label</th>
<th>bus</th>
<th>baysquare</th>
<th>office</th>
<th>openmarket</th>
<th>park</th>
<th>quietstreet</th>
<th>restaurant</th>
<th>supermarket</th>
<th>tube</th>
<th>tubestation</th>
</tr>
</thead>
<tbody>
<tr>
<td>bus</td>
<td>81</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>baysquare</td>
<td>1</td>
<td>0</td>
<td>55</td>
<td>13</td>
<td>9</td>
<td>12</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>office</td>
<td>1</td>
<td>0</td>
<td>59</td>
<td>13</td>
<td>0</td>
<td>12</td>
<td>9</td>
<td>12</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>openmarket</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>29</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>park</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>51</td>
<td>29</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>quietstreet</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>29</td>
<td>43</td>
<td>9</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>restaurant</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>16</td>
<td>5</td>
<td>0</td>
<td>53</td>
<td>21</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>supermarket</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>42</td>
<td>7</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>tube</td>
<td>7</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>44</td>
<td>28</td>
</tr>
<tr>
<td>tubestation</td>
<td>5</td>
<td>16</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>19</td>
<td>41</td>
</tr>
</tbody>
</table>

#### TABLE IV

**AGGREGATE CONFUSION MATRIX FOR SCENE CLASSIFICATION ACROSS ALL SUBMISSIONS. ROWS ARE GROUND TRUTH, COLUMNS THE INFERRED LABELS. VALUES ARE EXPRESSED AS PERCENTAGES ROUNDED TO THE NEAREST INTEGER**


13The original OS results published at the time of the challenge differ from the results published here due to a systematic fault affecting a subset of the labels in the original OS development and test datasets. This was found and fixed, and the three teams who submitted systems to the OS task were contacted and invited to revise their systems. The DHV system was re-trained on the corrected OS development data; the configurations of the other systems (GVV, VVK, and baseline) were not affected, and were left unchanged. All three systems were re-evaluated on the corrected test datasets to obtain the results here. The corrections to the data generally improved system performance, which is to be expected since they improved the correspondence between training and test.
TABLE V
RESULTS FOR THE PARTICIPATING SYSTEMS FOR THE OFFICE LIVE EVENT DETECTION TASK. THE STRONGEST PERFORMANCE ACCORDING TO EACH METRIC IS HIGHLIGHTED IN BOLD

<table>
<thead>
<tr>
<th>Evaluation Method</th>
<th>Event-Based</th>
<th>Class-Wise Event-Based</th>
<th>Frame-Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>F (%)</td>
<td>F offset (%)</td>
<td>AEER</td>
</tr>
<tr>
<td>CPS</td>
<td>2.33</td>
<td>1.65</td>
<td>7.78E</td>
</tr>
<tr>
<td>DBV</td>
<td>26.07</td>
<td>22.43</td>
<td>2.519</td>
</tr>
<tr>
<td>GVV</td>
<td>15.52</td>
<td>13.46</td>
<td>1.759</td>
</tr>
<tr>
<td>NVM_1</td>
<td>35.37</td>
<td>24.95</td>
<td>1.864</td>
</tr>
<tr>
<td>NVM_2</td>
<td>34.16</td>
<td>26.28</td>
<td>1.852</td>
</tr>
<tr>
<td>NVM_3</td>
<td>34.52</td>
<td>27.01</td>
<td>1.827</td>
</tr>
<tr>
<td>NVM_4</td>
<td>30.47</td>
<td>24.08</td>
<td>1.886</td>
</tr>
<tr>
<td>NR2</td>
<td>19.21</td>
<td>15.26</td>
<td>3.076</td>
</tr>
<tr>
<td>SCS_1</td>
<td>39.47</td>
<td>36.34</td>
<td>1.669</td>
</tr>
<tr>
<td>SCS_2</td>
<td>45.17</td>
<td>41.06</td>
<td>1.601</td>
</tr>
<tr>
<td>VVK</td>
<td>30.77</td>
<td>25.40</td>
<td>2.054</td>
</tr>
<tr>
<td>Baseline</td>
<td>7.38</td>
<td>1.58</td>
<td>5.990</td>
</tr>
</tbody>
</table>

TABLE VI
RESULTS FOR THE PARTICIPATING SYSTEMS FOR THE OFFICE SYNTHETIC EVENT DETECTION TASK. THE STRONGEST PERFORMANCE ACCORDING TO EACH METRIC IS HIGHLIGHTED IN BOLD. SEE FOOTNOTE 13

VI. REFLECTIONS AND RECOMMENDATIONS

Before concluding, we wish to draw some reflections out from the above results and from our experience of managing the DCASE challenge, and to offer some recommendations for future evaluation challenges. Our challenge comes in the context of a series of challenges coordinated by the IEEE AASP, such as challenges relating to distant and reverberant speech.15

Our design of the challenge involved participants submitting code, for the organisers to execute against private datasets. This design, in common with MIREX music audio challenges [33], incurs resource costs as the hosts must dedicate time to running the submissions. It also requires holding back some private data, which cannot immediately benefit the community as open data. However it has advantages such as ensuring participants do not

among the OL rankings judged across all evaluation measures ($H = 88.4, p = 3.3 \times 10^{-14}$). The only systematic deviation from consensus among the evaluation metrics was for submission GVV, which was ranked low (9th or 10th) on all F-measures but 3rd or better on all AEER measures.

For the OS task, the best performance across most of our evaluation metrics is achieved by the GVV system, which used NMF decomposition followed by HMM postprocessing. Overall rates for each system are broadly comparable with the OL task. It should also be noted that submitted systems performed better for signals with lower polyphony, with the exception of the DHV system, which had better performance with higher polyphony levels (5.57% frame-based F for low polyphony and 30.08% for high polyphony). As expected, the onset-offset evaluation produced worse results compared to onset-only evaluation for both tasks, although the performance difference is rather small (this may be explained by the percussive nature of most events).

It is also instructive to look at the correlation between the ranking of systems that were both submitted to the Office Live and Office Synthetic challenges. It allows us to study the consistency of performance of the evaluated systems using natural and artificial data. The OL and OS tests are not independent, since they partly use the same audio source material, but comparing their outcomes gives us an indication of whether the synthesis procedure for OS had a strong impact on the eventual rankings. Let us consider results achieved using the Class-Wise Event-Based metrics as they are more resilient to discrepancies between datasets in terms of density of events within the scene. Apart from a slight permutation of GVV and VVK systems, the strong level of correlation (average of 90% over the 4 metrics in terms of the Spearman’s rank correlation coefficient) indicates that considering artificially synthesized sound scenes may have some meaning for this kind of task.

VI. REFLECTIONS AND RECOMMENDATIONS

Before concluding, we wish to draw some reflections out from the above results and from our experience of managing the DCASE challenge, and to offer some recommendations for future evaluation challenges. Our challenge comes in the context of a series of challenges coordinated by the IEEE AASP, such as challenges relating to distant and reverberant speech.15

Our design of the challenge involved participants submitting code, for the organisers to execute against private datasets. This design, in common with MIREX music audio challenges [33], incurs resource costs as the hosts must dedicate time to running the submissions. It also requires holding back some private data, which cannot immediately benefit the community as open data. However it has advantages such as ensuring participants do not

and the virtual machine, and infrastructure such as code- and annotators, programmers assisting with issues such as code
This includes staff and PhD students as core organisers, data available by a large research group (see Acknowledgments).

would not have been possible without the resources made
implications for hosting such challenges. Dataset collection
DCASE challenge, in particular to highlight the resource
obtained match the results that the submitters obtained on their
This reduced but by no means eliminated the time required to
formal task specifications, baseline code and a virtual machine.
Community involvement was crucial to the successful con-
VII. CONCLUSION
challenged centered around automatic classification of animal sounds. The mode of interaction in that case was not to submit code, but
to submit system output. Further, participants could iteratively modify their code and submit updated output, getting feedback
in the form of results on a validation dataset. This does carry
risk of overfitting to the specifics of the challenge, and
less direct reproducibility, although the winning submission
was required to be made open-source and confirmed by the
hosts. Relative to DCASE, the SABIOD challenges appeared
to encourage a greater amount of ad-hoc participation from
independent machine-learning professionals, perhaps due to the
immediate feedback loop made possible by the online system.
The workflows represented by the DCASE and the SABIOD
challenges each have their own strengths and weaknesses, and
we look forward to further refinements in public evaluation methodology.

We have enumerated the steps involved in running the
DCASE challenge, in particular to highlight the resource implications for hosting such challenges. Dataset collection and annotation was the main requirement on staff time. This challenge was not funded explicitly by any project, and so would not have been possible without the resources made available by a large research group (see Acknowledgments). This includes staff and PhD students as core organisers, data annotators, programmers assisting with issues such as code and the virtual machine, and infrastructure such as code- and data-hosting facilities.

In Section III-D we described various steps we took to ensure that the challenge would run smoothly, such as publishing formal task specifications, baseline code and a virtual machine. This reduced but by no means eliminated the time required to run and troubleshoot the code submissions received. A clear recommendation that emerges from this experience is that a formal test for the submitted code to be run at submission time would help greatly. This could be applied in the form of automated unit testing, or more simply by the challenge organizers running the submissions using public data and confirming that the results obtained match the results that the submitters obtained on their own system.

Community involvement was crucial to the successful conduct of this challenge, in particular for discussing the task specifications, but also for negotiating logistics of submission and discussing the final results. The support of the IEEE AASP Technical Committee and the IEEE WASPAA 2013 Conference Committee helped us to form this community.

With the DCASE challenge we aimed to frame a set of general-purpose machine listening tasks for everyday audio, in order to benchmark the state of the art, stimulate further work, and grow the research community in machine listening beyond the domains of speech and music. The challenge results illustrate tasks we designed for this had the right level of difficulty for this: none of the tasks was trivial for any submitted system, and a range of scores was achieved enabling comparison of the advantages and disadvantages of systems. The strong level of participation from a diverse set of researchers indicates that the tasks were pertinent to current research.

For the scene classification (SC) task, the leading systems attained results significantly above baseline and comparable to average results from human listeners. A strategy used by many of the strongest systems was to use feature representations which capture medium-scale temporal information about the sound scene. However there is still room for improvement beyond the highest-scoring system; we demonstrated this was possible with a simple majority-vote metaclassifier aggregating the submitted systems, illustrating that there is information yet present in the audio that can drive stronger performance in future. The best way to improve the SC task in future rounds would be through larger dataset sizes in order to draw stronger conclusions about the significance of differences between system performances.

For the event detection (OL/OS) tasks, the leading systems achieved relatively strong performance, although with substantial scope for improvement. This was particularly evident in the polyphonic OS task, indicating that polyphony in audio scenes remains a key difficulty for machine listening systems and more development is needed in this area. However, the class-wise analysis of results also indicates that some event types proved harder to detect than others, even in the monophonic OL task, indicating that the ability for one system to detect a wide range of sound types is also a key challenge. Future event detection challenges could be improved with further community attention to evaluation metrics and their relation to practical requirements. It may also be of value to evaluate systems explicitly regarding the correlation between their performance and the level of polyphony in a scene.

Regarding the community formed around this research topic, we were very encouraged by the strong level of participation, and by the decisions of various groups to publish their submitted systems as open-source code. These, alongside the resources which we published (open-source baseline systems; open datasets; virtual machine disk image) provide a rich resource for others who may wish to work in this area. The community has set a benchmark, establishing that leading techniques are able to extract substantial levels of semantic detail from everyday sound scenes, but with clear room for improvement in future.

ACKNOWLEDGMENT

The authors would like to thank the IEEE AASP Technical Committee for endorsing and supporting this work, as well as all challenge participants—not only for their submissions but their community participation in shaping the challenges, and their presentations at IEEE WASPAA 2013. They would also like to thank the IEEE WASPAA 2013 Conference Committee for their support in organizing the special session.


17For open dataset downloads see http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/
REFERENCES


