Classification Study of Sound and Image Events Using Event Detection Systems

DR. G.Murugaboopathy, R.Rajan

Associate Proffessor
Dept of IT
Kalasalingam University
Rajapalayam, Tamilnadu.India

Research Scholar of Sankara university
Dept of IT
Kanchipuram, Tamilnadu,India

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 challenges that have been conducted in neighboring areas. Then we give detail on the experimental design of the tasks we designed, the approach to evaluation, and the data which we collected for the

tasks. We also consider some practicalities in the conduct of the challenge. In Section V we give the results of each task in the challenge, results which were first presented at the IEEE WASPAA 2013 conference [9]. We discuss issues emerging from the results such as the level of task difficulty, and in particular we compare the "live" and "synthetic" variants of our event detection challenge. Finally we consider the outlook for machine listening in light of the challenge: the state of the art, future directions indicated, and the contribution that this challenge has made. We also reflect on the organizational structure of this and other challenges, in relation to issues such as reproducibility and sustainability.

II. BACKGROUND

In this section we will briefly overview the tasks of acoustic scene classification and detection of sound events within a scene, both of which have been studied in recent literature. We discuss their relation to other machine listening tasks, and outline standard approaches taken. We will then discuss recent evaluation campaigns in machine listening, which set the context for our own campaign.

Acoustic scene classification aims to characterize the acoustic environment of an audio stream by selecting a semantic label for it [10]. It can be considered as a machine-learning task within the widespread single-label classification paradigm, in which a set of class labels is provided and the system must select exactly one for any given input [11, Chapter 1]. It therefore has parallels with audio classification tasks such as music genre recognition [12] or speaker recognition [13], and with classification tasks in other time-based media such as video. When classifying time-based media, a key issue is how to analyze temporally- structured data to produce a single label representing the media object overall.

There are two main strategies found in the literature. One is to use a set of low-level features under a "bag-of-frames" approach, which treats the scene as a single object and aims at representing it as the long-term statistical distribution of some set of local spectral features. Prevailing among different features for the approach is the Mel-frequency Cepstral Coefficients (MFCCs) that have been found to perform quite well [10]. Foote [14] is an early example, comparing MFCC distributions via vector quantisation. Since then, the standard approach to compare distributions is by constructing a Gaussian Mixture Model (GMM) for each instance or for each class [10]. The other strategy is to use an intermediate representation prior to classification that models the scene using a set of higher level features that are usually captured by a vocabulary or dictionary of "acoustic atoms". These atoms usually represent acoustic events or streams within the scene which are not necessarily known a priori and therefore are learned in an unsupervised manner from the data. Sparsity or other constraints can be adopted to lead to more discriminative representations that subsequently ease the classification process. An example is the use of non-negative matrix factorization (NMF) to extract bases that are subsequently converted into MFCCs for compactness and used to classify a dataset of train station scenes [15]. Building upon this approach, the authors in [16] used shift-invariant probabilistic latent component analysis (SIPLCA) with temporal constrains via hidden Markov models (HMMs) that led to improvement in performance. In [17] a system is proposed that uses the matching pursuit algorithm to obtain an effective time-frequency feature selection that are afterwards used as supplement to MFCCs to perform environmental sound classification.

The goal of acoustic event detection is to label temporal regions within an audio recording, resulting in a symbolic description 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 annotation of time segments but focuses on speech "turns" rather than individual events [18].

The majority of work in event detection 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 detection is perhaps a more demanding task than scene classification, but at the same time heavily intertwined. For example, information from scene classification can provide supplementary contextual information for event detection [22]. Many proposed approaches can be found in the literature among which spectrogram 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-overlapping 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], birdsong [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 component-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 separation, and then to analyze the resulting signals individually [29], [21].

However, note that the computational equivalent of auditory 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 framewise 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.1 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 monophonic 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 minimize 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 evaluations. For this we used VirtualBox software which runs on all common operating systems, together with a disk image based on Xubuntu 12.10 Linux.2 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. 3 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 singlelabel 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, tubestation. 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 generalizability 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 covaried 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 (totalling around 50% of the recorded duration), and collated these segments 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-offrames" 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 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.5 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 under an open-source licence,6 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 Soundfield 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 Visualiser7 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 dataset8 and the development dataset,9 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 because 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 density10 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 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].

$$AEER = \frac{D + I + S}{N} \qquad (1)$$

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

$$P = \frac{c}{e}, \quad R = \frac{c}{r}, \quad F = \frac{2PR}{P + R}. \qquad (2)$$

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 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

$$F' = \frac{1}{K} \sum_{k} F_k \qquad (3)$$

no clear improvements. The baseline event detection system was made available to challenge participants under an open source license.

Where F_K is the F-measure for events of class k . Matlab code for the metrics can be found online.

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 modelling 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 20i , 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 $\hat{V} \in \mathbb{R}^{\Omega \times T}$ the constant-Q spectrogram of a test recording (Ω: number of log-frequency bins; T: number of time frames), $W \in \mathbb{R}^{\Omega \times N}$ the pre-extracted dictionary and $H \in \mathbb{R}^{N \times T}$, the NMF model attempts to approximate V as a product of W and H. In the supervised case (when W is known and kept fixed), this involves simply estimating H iteratively until convergence using the following multiplicative update, ensuring a non-increasing divergence between $\hat{V}$ and WH [44]

$$H \leftarrow H \otimes \frac{W^T((WH)^{-1} \otimes \hat{V})}{W^T}. \qquad (4)$$

In order to detect sound events, we sum together the (non-binary) activations per class obtained from H . Finally, a threshold θ 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 θ is the same for all event classes; the optimal N and θ values were chosen empirically by maximizing the -measure for the two annotations on the development set. Smoothing on the activations was also tested with

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 intentions 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.

TABLE I
TIMELINE OF DCASE CHALLENGE ORGANIZATION. THE TIMELINE IS DIVIDED INTO MAIN PHASES, AND MILESTONES ARE HIGHLIGHTED

April 2012:	Challenge proposed to IEEE AASP
May 2012:	Challenge accepted by IEEE AASP
Jun 2012:	Initial recordings made, to test equipment and to produce examples for discussion
Jun 2012:	Organisers of IEEE WASPAA 2013 agree to host a challenge results special session
Jun–Dec 2012:	Roaming recordings for SC task
Jul 2012:	⇒ **Call for participation published** to various mailing lists; website established
Aug 2012:	Challenge publicised in IEEE SPS Newletter
Aug–Sep 2012:	Community discussion on dedicated challenge mailing list.
Sep 2012:	⇒ **Task design completed**
Sep–Oct 2012:	Office recordings for OL and OS tasks
Oct–Nov 2012:	Ground-truth annotation of OL recordings (external annotators: 2 x 35 hours)
Nov–Dec 2012:	Listening sessions to all recordings, to ensure data quality; error-checking and correction of manual annotations
Dec 2012:	⇒ **Public release of training/development datasets** (audio and annotations)
Dec 2012:	Online survey of potential participants (preferred tasks, programming languages, operating systems)
Jan 2013:	Create the Linux virtual machine disk image
Jan–Feb 2013:	Write paper (EUSIPCO 2013) introducing the tasks and baseline systems
Jan–Feb 2013:	Generating synthetic testing dataset for OS task
Jan 2013:	⇒ **Publication of finalised task specifications** (output formats, eval metrics etc.) as well as virtual machine
Feb 2013:	Publication of the scripts used to calculate the evaluation metrics
Feb 2013:	Publication of synthetic development dataset for OS task
Feb–Mar 2013:	Further community discussion; official confirmation that for OL/OS three separate evaluation metrics will be applied
Mar 2013:	Request for participants to email to confirm participation
Apr 2013:	⇒ **Deadline for participants to submit code**
Apr–May 2013:	Running all code submissions – team liaises with authors about software issues etc, compiles result statistics
May 2013:	Deadline for participants to submit extended abstracts for WASPAA 2013
May 2013:	Results released privately to each participating team
Jun 2013:	Write paper (WASPAA 2013) giving results of the challenge
Oct 2013:	⇒ **Results released publicly** at WASPAA 2013

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 ' ' (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.

TABLE II
SUMMARY OF SUBMITTED SCENE CLASSIFICATION SYSTEMS

Participants	Code	Method	Lang
Chum et al. [46]	CHR	Various features at 2 frame sizes, classified either: (a) per-frame SVM + majority voting; (b) HMM	Matlab
Elizalde [47]	ELF	Concatenation of 4 different mono mixdowns; "i-vector" analysis of MFCCs, classified by pLDA	Matlab
Geiger et al. [48]	GSR	Diverse features, classified within 4-second windows using SVM, then majority voting	Weka/ HTK
Krijnders and ten Holt [49]	KH	"Cochleogram" representation, analysed for tonelikeness in each t-f bin, classified by SVM	Python
Li et al. [50]	LTT	Wavelets, MFCCs and others, classified in 5-second windows by treebagger, majority voting	Matlab
Nam et al. [51]	NHL	Feature learning by sparse RBM, then event detection and max-pooling, classified by SVM	Matlab
Nogueira et al. [52]	NR1	MFCCs + MFCC temporal modulations + event density estimation + binaural modelling features, feature selection, classified by SVM	Matlab
Olivetti [53]	OE	Normalised compression distance (Vorbis), Euclidean embedding, classified by Random Forest	Python
Patil and Elhilali [54]	PE	Auditory representation analysed for spectrotemporal modulations, classified within one-second windows using SVM, then weighted combination of decision probabilities	Matlab
Rakotomamonjy and Gasso [55]	RG	Computer vision features (histogram of oriented gradient) applied to constant-Q spectrogram, classified by SVM	Matlab
Roma et al. [56]	RNH	Recurrence Quantification Analysis applied to MFCC time-series, classified by SVM	Matlab
Baseline		MFCCs, classified with a bag-of-frames approach	Python
Majority Vote	MV	Majority voting of all submissions	Python

TABLE III
SUMMARY OF SUBMITTED EVENT DETECTION SYSTEMS

| Participants	Code	Method	Lang
Chauhan et al. [57]	CPS	Feature extraction - Segmentation - Likelihood ratio test classification	Matlab
Diment et al. [58]	DHV	MFCCs (features) - HMMs (detection)	Matlab
Gemmeke et al. [59]	GVV	NMF (detection) - HMMs (postprocessing)	Matlab
Niessen et al. [60]	NVM	Hierarchical HMMs + Random Forests (classification) - Meta-classification	Matlab
Nogueira et al. [61]	NR2	MFCCs (features) - SVMs (classification)	Matlab
Schröder et al. [62]	SCS	Gabor filterbank features - HMMs (classification)	Matlab
Vuegen et al. [63]	VVK	MFCCs (features) - GMMs (detection)	Matlab
Baseline		NMF with pre-extracted bases (detection)	Matlab
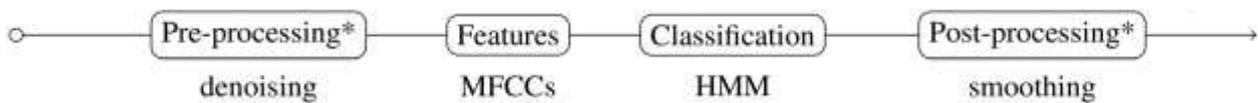
Fig. 1. Schematic of event detection systems (nodes with a * are not systematically used). Below, state-of-the-art design choices are given as examples.

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 unit tests 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 detection task. Variants for each system were allowed, which increased the total number of systems somewhat.

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. Firstly, 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.
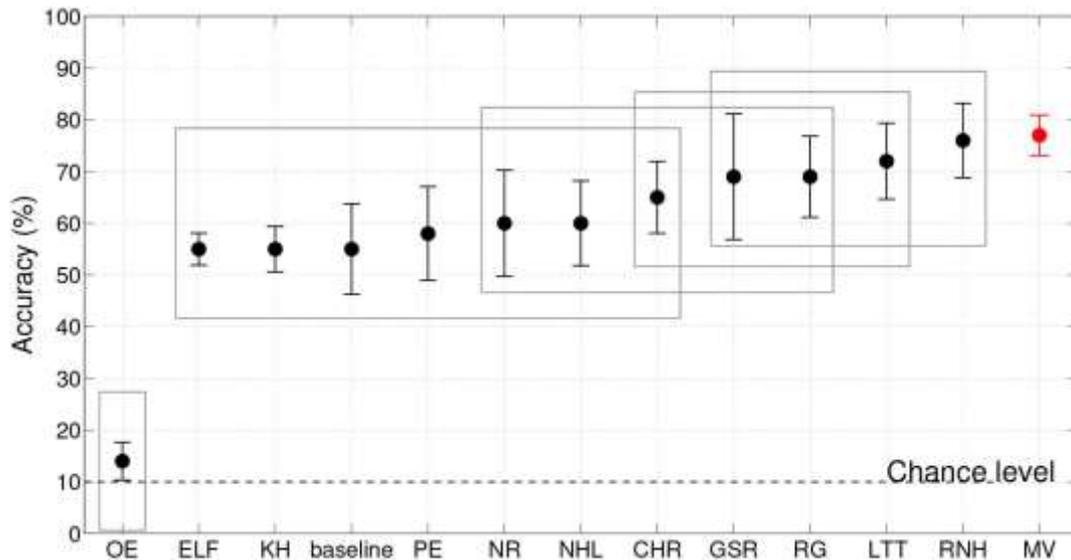

Fig. 2. Mean values and confidence intervals of the accuracy of methods for SC evaluated on the DCASE private dataset using stratified 5-fold cross-validation. The boxes enclose methods that cannot be judged to perform differently with a significance level of 95% using a sign test [65]. For example, GSR baseline, but we cannot confirm that CHR baseline. Figure is adapted from [64].

- **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.

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

Label	bus	busystreet	office	openairmarket	park	quietstreet	restaurant	supermarket	tube	tubestation
bus	81	3	0	4	1	0	0	4	6	2
busystreet	1	69	14	2	1	2	1	3	3	5
office	1	0	55	13	9	12	4	3	1	3
openairmarket	1	2	0	59	13	0	9	12	3	2
park	1	1	8	3	51	29	3	2	1	1
quietstreet	0	5	4	3	29	43	9	5	0	1
restaurant	1	1	0	16	5	0	53	21	2	3
supermarket	6	5	6	6	4	7	10	42	7	7
tube	7	7	1	1	2	2	5	3	44	28
tubestation	5	16	1	4	1	2	3	8	19	41

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 cochleogram [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 submission.
The strong performance of this meta-classifier is particularly notable given its simplicity—all systems are combined with equal weights. It suggests that for around 77% of soundscapes some algorithms make a correct decision, and the algorithms that make an incorrect classification do not all agree on one particular incorrect label. This allows to combine the decisions into a relatively robust meta-classifier. (Note that we did not test for significance of the comparison between MV and the other results, because the MV output is not independent of the output of the individual submissions.)

More sophisticated meta-classifiers could perhaps extend this performance further. Table IV shows a confusion matrix for the scene labels as round percentages of the sum of all confusion matrices for all submissions. Confusions were mostly concentrated over classes that share some acoustical properties such as park/quietstreet and tube/tubestation.

Our labels contained five indoor and five outdoor locations, and both types showed a similar level of difficulty for the algorithms.

B. OL/OS Results

Results for the event detection OL and OS tasks are summarized in Tables V and VI, respectively.13 The baseline was outperformed by most systems for both tasks. Results for the OL task indicate the high level of difficulty in recognising sound events (from many possible classes, with great variability) from noisy acoustic scenes. The best performance for the OL task using most types of metrics is achieved by the SCS submission, which used a Gabor filterbank feature extraction step with by 2-layer hidden Markov models (HMMs) for classifying events, followed by the NVM submission, which used a meta-classifier combining hierarchical HMMs and random forests. Results for each event class separately are visualized in Fig. 3, where it can be seen that most systems had solid detection rates for clearing throat, coughing, door knocks and speech, but had weak results for drawers, printers, keyboards and switches.

Many of the OL/OS methods employed a decomposition step, either explitcly (e.g. GVV) or implicitly (e.g. DHV), which is of interest from the perspective of CASA (see Section I). It should be noted that MFCCs were not proven as useful for the event detection tasks as with the scene classification tasks, with more rich and auditory model-based representations proving to be more useful (such as Gabor filterbanks and Mel-magnitude spectrograms). Again, contrary to the SVM-dominated scene classification task, variants of HMMs were proven to be both the most popular as well as reliable tools for event detection, due to their ability to model timeseries data. Of particular interest are submissions that were also submitted to the polyphonic OS task, where two systems experimented with multiple Viterbi passes (DHV, GVV) in order to handle overlapping events.

Regarding statistical significance tests in event detection systems, to the authors' knowledge no such tests have been attempted so far in the literature. As has been argued in [16] for the multi-pitch detection problem (which is structurally similar to sound event detection), indicators of statistical significance are not highly pertinent for multi-class detection problems: in practice, even a small performance difference can often yield statistical significance. Detailed system descriptions and detailed results per system can be found on the challenge website.

However, for the OL task there was a large enough number of participants that we were able to examine statistically whether the different metrics tended to rank systems in the same order. We applied the Kruskal-Wallis test, a nonparametric test for comparing whether multiple groups of data (here, the evaluation ranks for each system) exhibit differences in distribution [66, Section 6.2.2]. This found a significant pattern of agreement among the OL rankings judged across all evaluation measures $H=88.4, p=3.3\times10^{-14}$). 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.

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

| | Evaluation Method | | | | | | | | | |
| | Event-Based | | | | Class-Wise Event-Based | | | | Frame-Based | |
System	F (%)	F_{offset} (%)	AEER	$AEER_{offset}$	F (%)	F_{offset} (%)	AEER	$AEER_{offset}$	F (%)	AEER
CPS	2.23	1.65	2.285	2.301	0.65	0.49	1.872	1.891	3.82	2.116
DHV	26.67	22.43	2.519	2.676	30.72	25.29	2.182	2.370	26.0	3.128
GVV	15.52	13.46	1.779	1.831	13.21	12.03	1.556	**1.606**	31.94	1.084
NVM_1	32.57	24.95	1.864	2.095	29.37	21.80	1.639	1.899	40.85	1.115
NVM_2	34.16	26.28	1.852	2.095	33.05	24.88	1.602	1.877	42.76	1.102
NVM_3	34.51	27.01	1.827	2.052	33.52	24.65	1.575	1.846	45.50	1.212
NVM_4	30.47	24.68	1.906	2.083	28.17	21.62	1.650	1.849	42.86	1.360
NR2	19.21	15.26	3.076	3.244	21.54	17.64	2.857	3.010	34.66	1.885
SCS_1	39.47	36.74	1.669	1.749	36.33	34.20	1.579	1.677	53.02	1.167
SCS_2	**45.17**	**41.06**	**1.601**	**1.727**	**41.51**	**38.32**	**1.511**	1.646	**61.52**	1.016
VVK	30.77	25.40	2.054	2.224	24.55	20.36	1.762	1.949	43.42	**1.001**
Baseline	7.38	1.58	5.900	6.318	9.00	1.86	5.960	6.462	10.72	2.590

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

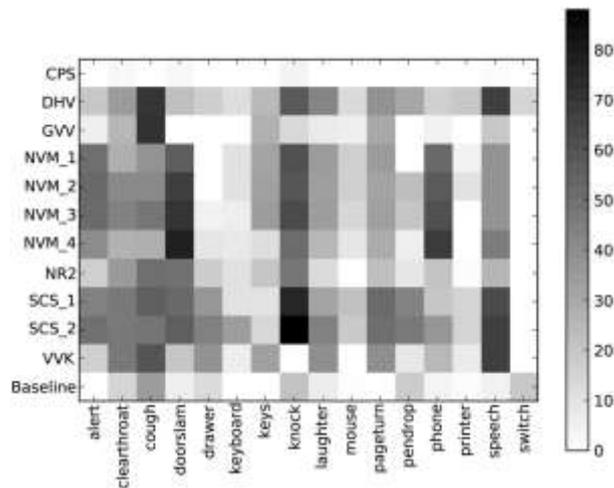| | Event-Based | | | | Class-Wise Event-Based | | | | Frame-Based | |
System	F (%)	F_offset (%)	AEER	AEER_offset	F (%)	F_offset (%)	AEER	AEER_offset	F (%)	AEER
DHV	16.06%	11.12%	3.52	3.76	**18.69%**	**12.18%**	2.90	3.16	18.68%	7.98
GVV	**17.00%**	**13.60%**	**1.68**	**1.77**	14.16%	11.40%	**1.22**	**1.30**	**21.28%**	**1.32**
VVK	13.78%	9.16%	**1.68**	1.80	10.51%	7.48%	1.38	1.49	13.51%	1.89
Baseline	7.75%	0.57%	6.26	6.87	9.47%	0.33%	5.33	5.95	12.76%	2.80

Fig. 3. Event detection (OL) results in class-wise (%) for each event class separately.

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.

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 overfit to the test data, and ensuring that results are reproducible in the sense of empirically verifying that the submitted software can be run by a third party.

An interesting point of comparison is provided by similar challenges run via the Kaggle website such as the 2013 SABIOD machine listening challenges.

These challenges 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 some 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.

VII. CONCLUSION

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.

REFERENCES

[1] L. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

[2] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The PASCAL CHiME speech separation and recognition challenge," Comput. Speech Language, vol. 27, no. 3, pp. 621–633, 2012.

[3] E. Benetos and S. Dixon, "Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model," J. Acoust. Soc. America, vol. 133, pp. 1727–1741, 2013.

[4] A. Wang, "An industrial strength audio search algorithm," in Proc. 4th Int. Conf. Music Inf. Retrieval, Oct. 2003, pp. 7–13.

[5] J.-M. Valin, F. Michaud, B. Hadjou, and J. Rouat, "Localization of simultaneous moving sound sources for mobile robot using a frequencydomain steered beamformer approach," in Proc. 2004 IEEE Int. Conf. Robot. Automat., Apr.–May 2004, vol. 1, pp. 1033–1038.

[6] R. Ranft, "Natural sound archives: Past, present and future," Anais da Academia Brasileira de Ciências, vol. 76, no. 2, pp. 456–460, 2004.

[7] D. L. Wang and G. J. Brown, Computational Auditory Scene Analysis: Principles, Algorithms, and Applications. Piscataway, NJ, USA: IEEE Press, 2006.

[8] A. S. Bregman, Auditory Scene Analysis: The Perceptual Organization of Sound. Cambridge, MA, USA: MIT Press, 1994.

[9] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in Proc. IEEE Workshop Appl. Signal Process. Audio Acoust., Oct. 2013, pp. 1–4.

[10] 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," J. Acoust. Soc. America, vol. 122, no. 2, pp. 881–891, 2007.

[11] I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2005.

[12] N. Scaringella, G. Zoia, and D. Mlynek, "Automatic genre classification of music content: A survey," IEEE Signal Process. Mag., vol. 23, no. 2, pp. 133–141, Mar. 2006.

[13] J. J. Wolf, "Efficient acoustic parameters for speaker recognition," J. Acoust. Soc. America, vol. 51, pp. 2044–2056, Jun. 1972.

[14] J. Foote, "Content-based retrieval of music and audio," in Proc. SPIE, Multimedia Storage Archiving Syst. II, 1997, vol. 3229, pp. 138–147.

[15] B. Cauchi, "Non-negative matrix factorisation applied to auditory scenes classification," M.S. thesis, ATIAM , ParisTech, Paris, France, Aug. 2011.

[16] E. Benetos, "Automatic transcription of polyphonic music exploiting temporal evolution," Ph.D. dissertation, School of Electron. Eng. And Comput. Sci., Queen Mary University of London, London, U.K., Dec. 2012.

[17] S. Chu, S. Narayanan, and C.-C. Jay Kuo, "Environmental sound recognition with time-frequency audio features," IEEE Trans. Audio, Speech Language Process., vol. 17, no. 6, pp. 1142–1158, Aug. 2009.

[18] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," IEEE Trans. Audio, Speech, Language Process., vol. 14, no. 5, pp. 1557–1565, Sep. 2006.

[19] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in Proc. Eur. Signal Process. Conf., Aug. 2010, pp. 1267–1271.

[20] C. V. Cotton and D. P. W. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in Proc. IEEE Workshop Appl. of Signal Process. Audio Acoust., Oct. 2011, pp. 69–72.

[21] T. Heittola, A. Mesaros, T. Virtanen, and A. Eronen, "Sound event detection in multisource environments using source separation," in Proc. Workshop Mach. Listening Multisource Environ., 2011, pp. 36–40.

[22] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," EURASIP J. Audio, Speech, Music Process., vol. 2013, no. 1, 2013, Art. ID 1.

[23] A. Mesaros, T. Heittola, and A. Klapuri, "Latent semantic analysis in sound event detection," in Proc. Eur. Signal Process. Conf., 2011, pp. 1307–1311.

[24] J. P. Barker, M. P. Cooke, and D. P. W. Ellis, "Decoding speech in the presence of other sources," Speech Commun., vol. 45, no. 1, pp. 5–25, 2005.

[25] F. Briggs and B. Lakshminarayanan et al., "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," J. Acoust. Soc. America, vol. 131, pp. 4640–4650, 2012.

[26] D. Giannoulis, A. Klapuri, and M. D. Plumbley, "Recognition of harmonic sounds in polyphonic audio using a missing feature approach," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., May 2013, pp. 8658–8662.

[27] D. Stowell and M. D. Plumbley, "An open dataset for research on audio field recording archives: Freefield1010," in Proc. Audio Eng. Soc. 53r Conf. Semantic Audio (AES53), 2014, pp. 1–7.

[28] 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. Audio, Speech, Language Process., vol. 14, no. 1, pp. 321–329, Jan. 2006.

[29] K. Ellis, E. Coviello, and G. Lanckriet, "Semantic annotation and retrieval of music using a bag of systems representation," in Proc. 12th Int. Conf. Music Inf. Retrieval, Oct. 2011, pp. 723–728.

[30] T. Fawcett, "An introduction to ROC analysis," Pattern Recog. Lett., vol. 27, no. 8, pp. 861–874, 2006.

[31] J.-J. Aucouturier and F. Pachet, "Improving timbre similarity: How high's the sky?," J. Negative Results Speech Audio Sci., vol. 1, no. 1, pp. 1–13, 2004.

[32] D. Giannoulis, D. Stowell, E. Benetos, M. Rossignol, M. Lagrange, and M. D. Plumbley, "A database and challenge for acoustic scene classification and event detection," in Proc. Eur. Signal Process. Conf., 2013, pp. 1–5.

[33] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "CLEAR evaluation of acoustic event detection and classification systems," in Proc CLEAR, 2007, pp. 311–322.

[34] R. Kompass, "A generalized divergence measure for nonnegative matrix factorization," Neural Comput., vol. 19, no. 3, pp. 780–791, 2007.

[35] C. Schörkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in Proc. Sound Music Comput. Conf., Jul. 2010, pp. 3–64.

[36] M. Chum, A. Habshush, A. Rahman, and C. Sang, "IEEE AASP scene classification challenge using hidden Markov models and frame based classification," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs. qmul.ac.uk/sceneseventschallenge/abstracts/SC/CHR.pdf

[37] B. Elizalde, H. Lei, F. G. , and N. Peters, "An I-vector based approach for audio scene detection," Tech. Rep., 2013 [Online]. Available: http:/ /c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/SC/ELF.pdf

[38] J. T. Geiger, B. Schuller, and G. Rigoll, "Recognising acoustic scenes with large-scale audio feature extraction and SVM," 2013 [Online]. Available: http://c4dm.qmul.ac.uk/sceneseventschallenge/ abstracts/SC/GSR.pdf

[39] J. D. Krijnders and G. A. ten Holt, "A tone-fit feature representation for scene classification," 2013 [Online]. Available: http://c4dm.eecs.qmul. ac.uk/sceneseventschallenge/abstracts/SC/KH.pdf

[40] M. J. Kim and H. Kim, "Automatic extraction of pornographic contents using radon transform based audio features," in Proc. 9th Int.Workshop Content-Based Multimedia Indexing, Jun. 2011, pp. 205–210.

[41] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "An adaptive framework for acoustic monitoring of potential hazards," EURASIP J. Audio, Speech Music Process., vol. 2009, 2009, Art. ID 13.

[42] H. G. Okuno, T. Ogata, and K. Komatani, "Computational auditory scene analysis and its application to robot audition: Five years experience," in Proc. 2nd Int. Conf. Informat. Res. Develop. Knowledge Soc. Infrastructure, Jan. 2007, pp. 69–76.

[43] D. Stowell and M. D. Plumbley, "Segregating event streams and noise with a Markov renewal process model," J. Mach. Learning Res., vol. 14, pp. 1891–1916, 2013.

[44] E. Vincent, S. Araki, F. Theis, G. Nolte, P. Bofill, H. Sawada, A. Ozerov, V. Gowreesunker, D. Lutter, and N. Duong, "The signal separation evaluation campaign (2007–2010): Achievements and remaining challenges," Signal Process., vol. 92, no. 8, pp. 1928–1936, 2012.

[45] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME' speech separation and recognition challenge: An overview of challenge systems and outcomes," in Proc. IEEE Workshop Automat. Speech Recog. Understand., Dec. 2013, pp. 162–167.

[46] J. S. Downie, A. F. Ehmann, M. Bay, and M. C. Jones, "The music information retrieval evaluation eXchange: Some observations and insights," in Advances in Music Information Retrieval, ser. Studies in Computational Intelligence. New York, NY, USA: Springer, 2010, vol. 274, pp. 93–115.

[47] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofalo, D. Mostefa, and P. Soundararajan, "The CLEAR 2006 evaluation," in Multimodal Technologies for Perception of Humans. Berlin, Germany: Springer- Verlag, 2007, pp. 1–44.

[48] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVID," in Proc. 8th ACM Int. Workshop Multimedia Inf. Retrieval, 2006, pp. 321–330.

[49] B. L. Sturm, "A simple method to determine if a music information retrieval system is a 'horse'," IEEE Trans. Multimedia, vol. 16, no. 6, pp. 1636–1644, Oct. 2014.

[50] E. Olivetti, "The wonders of the normalized compression dissimilarity representation," Tech. Rep., 2013 [Online]. Available: http://c4dm. eecs.qmul.ac.uk/sceneseventchallenge/abstracts/SC/OE.pdf

[51] K. Patil and M. Elhilali, "Multiresolution auditory representations for scene classification," Tech. Rep., 2013 [Online]. Available: http:// c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/SC/PE.pdf

[52] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs.qmul.ac.uk/ sceneseventschallenge/abstracts/SC/RG.pdf

[53] G. Roma, W. Nogueira, and P. Herrera, "Recurrence quantification analysis features for auditory scene classification," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs.qmul.ac.uk/ sceneseventschallenge/abstracts/SC/RNH.pdf

[54] S. Chauhan, S. Phadke, and C. Sherland, "Event detection and classification," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs.qmul. ac.uk/sceneseventschallenge/abstracts/OL/CPS.pdf

[54] A. Diment, T. Heittola, and T. Virtanen, "Sound event detection for office live and office synthetic AASP challenge," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/ abstracts/OL/DHV.pdf

[55] J. F. Gemmeke, L. Vuegen, B. Vanrumste, and H. Van hamme, "An exemplar-based NMF approach for audio event detection," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs.qmul.ac.uk/ sceneseventschallenge/abstracts/OL/GVV.pdf

[56] W. Nogueira, G. Roma, and P. Herrera, "Automatic event classification using front end single channel noise reduction, MFCC features and a support vector machine classifier," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/ abstracts/OL/NVM.pdf

[57] M. E. Niessen, T. L. M. Van Kasteren, and A. Merentitis, "Hierarchical sound event detection," Tech. Rep., 2013 [Online]. Available: http:// c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/OL/NR2.pdf

[58] J. Schröder, B. Cauchi, M. R. Schädler, N. Moritz, K. Adiloglu, J. Anemüller, S. Doclo, B. Kollmeier, and S. Goetze, "Acoustic event detection using signal enhancement and spectro-temporal feature extraction," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs.qmul.ac. uk/sceneseventschallenge/abstracts/OL/SCS.pdf

[59] L. Vuegen, B. Van Den Broeck, P. Karsmakers, J. F. Gemmeke, B. Vanrumste, and H. Van hamme, "An MFCC-GMM approach for event detection and classification," Tech. Rep., 2013 [Online]. Available: http:/ /c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/OL/VVK.pdf

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

[61] J. D. Gibbons and S. Chakraborti, Nonparametric Statistical Inference, 5th ed. London, U.K.: Chapman & Hall, 2010.

[62] P. Sprent and N. C. Smeeton, Applied Nonparametric Statistical Methods, 3rd ed. Boca Raton, FL, USA: CRC Press, 2001. D. Li, J. Tam, and D. Toub, "Auditory scene classification using machine learning techniques," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/SC/LTT.pdf

[63] J. Nam, Z. Hyung, and K. Lee, "Acoustic scene classification using sparse feature learning and selective max-pooling by event detection," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs.qmul.ac.uk/ sceneseventschallenge/abstracts/SC/NHL.pdf

[64] W. Nogueira, G. Roma, and P. Herrera, "Sound scene identification based on MFCC, binaural features and a support vector machine classifier," Tech. Rep., 2013 [Online]. Available: http://c4dm.eecs.qmul.ac. uk/sceneseventschallenge/abstracts/SC/NR1.pdf

[65] E. Olivetti, "The wonders of the normalized compression dissimilarity representation," Tech. Rep., 2013 [Online]. Available: http://c4dm. eecs.qmul.ac.uk/sceneseventschallenge/abstracts/SC/OE.pdf