This is the post print version of the article, which has been published in Intelligent service robotics .2018, 11 (4), 335-346. http://doi.org/10.1007/s11370-018-0258-9.

Common Sounds in Bedrooms (CSIBE) Corpora for Sound Event Recognition of Domestic Robots

Csaba Kertész and Markku Turunen
University of Tampere, Kalevantie 4, 33100 Tampere, Finland
csaba.kertesz@ieee.org (C. Kertesz), markku.turunen@uta.fi (M. Turunen)

Abstract—Although sound event recognition attracted much attention in the scientific community, applications in the robotics domain have not been in the focus. New databases were published and classifiers were explored in this paper to guide the future practical developments of domestic robots. A corpus (CSIBE-RAW) was collected from the internet to build acoustic models to recognize 13 sound events and omit background noises. As a case study, CSIBE-RAW was rerecorded in four room settings (CSIBE-AIBO) to create reverberation-tolerant classifiers for a Sony ERS-7. After nine classifiers were reviewed, the convolutional neural network (CNN) achieved the best accuracy (95.07%) after multi-conditional learning and it was suitable for real-time classification on the robot. The effects of lossy audio codecs were studied, lossy encoder-tolerant audio statistics were specified for the feature vector and the Ogg Vorbis encoder with 128 kbit VBR was found superior to store big data and avoid any significant accuracy loss with the compression ratio 1:8.

Highlights:
- New sound event corpus for domestic robotics with background noise modeling
- A practical approach to deploy the convolutional neural network model to a robot
- Analysis about the lossy encoding effects on recognition accuracy

Keywords: Domestic Robots, Recognition, Indoor Audio Corpus, Sony AIBO, Deep Learning.

Introduction

The contextual interpretation of the environment involves the fusion of multiple cues for the human beings (Goldstein, 2010). The localization and object recognition have been traditional research fields in robotics to reach human-level performance in visual senses and hearing (speech recognition (Yamamoto et al., 2006), music annotation (Ness et al., 2011)). Sound event recognition is a relatively new research field in the last decade that shifted the interest from the anthropomorphic bias to a more natural point of view of the auditory scene. This paper introduces a new corpora for isolated sound event recognition for indoor robotic applications and some common problems are highlighted with possible solutions to build robust acoustic models.

The only sound event database for robotics was published in (Maxime et. al., 2014). The NAR Dataset was recorded with a Nao humanoid robot from Aldebaran Robotics in a kitchen and contained 22 sound events as well as 20 English words. The average signal-to-noise ratio (SNR) of the recordings was 15 dB because of the noisy fans inside the robot body and the SVM classifier achieved 91.5% accuracy after 10-fold cross-validation despite the challenging conditions. This result was reached with file-averaged feature vectors and the model was not evaluated with unseen data.

The Acoustic Event Dataset (AED) (Plinge et al., 2014) was recorded in a smart room environment and Gaussian mixture model (GMM) was trained with a 600 msec sliding data window. The classifier was tested with unseen data and it distinguished 11 events with 87% accuracy from their relative small database. The unknown events were not modeled in their system, but the silence was a separate class.

The IEEE DCASE Challenge was organized in 2013 to establish an international competition for identifying sound scenes and events. The DCASE-OL dataset with 16 events was dedicated for event detection in real office environment (Stowell et al., 2015). The best system in the challenge had 61% frame-based precision on unseen data without modeling the background sounds.

Beltrán et al. (2015) proposed a novel sound event recognition method with temporal histograms of Mel-based Multi-Band Spectral Entropy Signature coefficients and they reported better results than MFCC-based SVM classification with source separation (non-negative matrix factorization). The background noise was not modeled, but their approach can detect the mixture of two events without any source separation technique. Their CICESE corpus contained several reusable datasets, but most of them are incomplete and do not reflect the description in their published paper which makes any comparison hard with their results.

Unlike the previous works with event classes, the CHiME-Home database (Foster et al., 2015) was annotated on a higher granularity for speech, human activity, television and household appliances. 4-second long audio chunks were allowed to hold multiple labels and the GMM classifier obtained 89% accuracy after 10-fold cross-validation. Some event classes represented the background noise (television, household appliances), but GMM was not evaluated with unseen data.

To mention an outdoor example, Salamon et al (Salamon et al., 2014) invented a new taxonomy for urban sound classification and their dataset (UrbanSound8K) had 18.5 hours of audio with 10 events. Temporal statistics complemented the feature vectors and they found the 4 seconds-long sliding window optimal. The model performance was estimated to 69% with 10-fold cross-validation of random forest (RF) and SVM classifiers.

Usually, the unintended sound events and the background noises were not modeled in the past works (Beltrán el al, 2015; Salamon et al., 2014; Stowell et al., 2015) unlike in (Foster et al., 2015). The new corpora (CSIBE) in this paper are specialized to the indoor robotics applications and an event class is dedicated to represent the auditory background with appliance sounds, object and human related noises. The audio files were acquired from free sources on the internet to have a clear licensing situation for the further usage. Creating universal acoustic models is a challenge because of the different noise levels and microphone characteristics. To provide a practical example for robots, the collected sounds were recorded again by replaying through a high-quality speaker and capturing with the stereo microphones of a Sony ERS-7 robot dog. A baseline sound event recognition system was developed and these two datasets (CSIBE-RAW, CSIBE-AIBO) were evaluated with 10-fold cross-validation and unseen data. CSIBE-AIBO had to be stored in lossy audio format, therefore, the effect of Ogg Vorbis and MP3 codecs were examined before drawing the conclusion at the end of the paper.

CSIBE Corpora

The available sound event corpora contain events of specific scenarios (smart room (Plinge et al., 2014), office (Stowell et al., 2015), kitchen (Maxime, et. al, 2014), urban area (Salamon et al., 2014)) and they have been built without modeling the background noises. This paper proposes that modeling the audio events out of interest and sources in alternate locations are important to shorten the gap between research and practical applications. The authors could not find a free or commercial audio corpus for indoor applications which has a separate event class for *background noise* and the samples provide high intraclass variability. (The background noise is defined as a sound that should not take the attention of the robot.) Therefore, a corpus was assembled from free online databases, public research datasets, own recordings and it was contributed back to the scientific community on the internet (DOI: 10.5281/zenodo.1243714). The license of each sound sample was included in the package to enable the reuse with a clear legal status of the research data. The new Common Sounds In BEdrooms corpus (CSIBE) consists of typical sound events in bedrooms/living rooms as people interact with social robots in these places at home. The database have two parts:

- **CSIBE-RAW**: Human speech and other events were collected from the internet in this dataset, complemented with new recordings. The samples had excellent and clear sound quality, they were stored in mono WAV format with 16 bit depth, 44.1 kHz sampling rate. All files were labeled according to the sound event type.

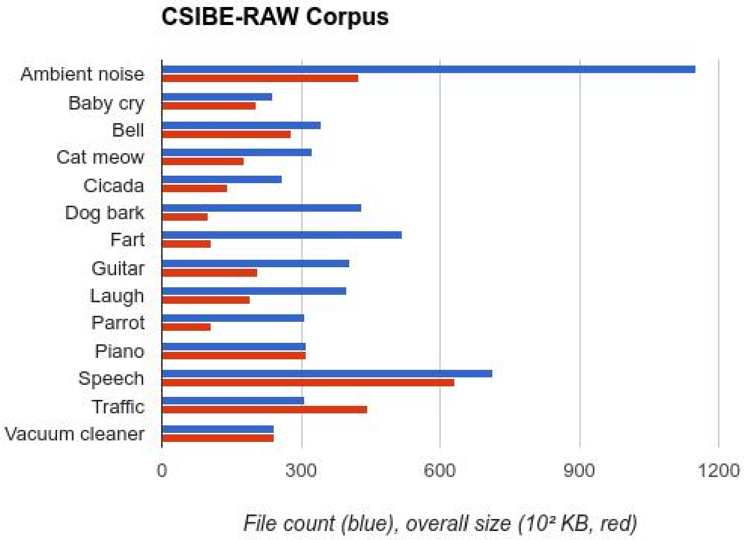

Fig. 1. Sample counts for each sound event in CSIBE-RAW.

- **CSIBE-AIBO**: The samples of CSIBE-RAW were played back through a speaker and recorded with the robot microphone. The details about the recording conditions are described in Chapter 2.2.

2.1 CSIBE-RAW

The core part of CSIBE was collected from public internet sources (freesound.org, AED (Plinge et al., 2014), DCASE-OL - IEEE DCASE Challenge (Stowell et al., 2015), NAR Dataset (Maxime, et. al., 2014)) and new recordings were done. The sample file counts of 14 sound events in the database were balanced around 300 except the ambient noise and speech (Fig. 1). The first was overrepresented to have a strong class for the uninteresting background noises, the latter was important for reliable speech detection in human-robot interactions. Further characteristics of CSIBE-RAW can be observed on Fig. 1. when the sample counts (blue lines) are compared to the sample size (red line). When those two lines are close to each other for a certain event (e.g piano, bell, vacuum cleaner, traffic), the average duration of the samples are close to 1 second, but short events (100-200 msec) cause a shorter red line (e.g flatulence, parrot, ambient noise) because the same amount of data is divided among more samples.

The overall size of CSIBE-RAW is comparable to the indoor sound event databases in the literature:

- AED for smart rooms has 11 events (350 MB), 213 samples.
- NAR (Maxime, et. al., 2014) for kitchen scene with Nao robot has 42 events, 831 samples (42 MB).
- DCASE-OL (Stowell et al., 2015) for office scene has 16 events (398 MB in 16 bit WAV), 1280 samples.

Although more events are in NAR (42) and DCASE-OL (16), CSIBE-RAW has a much larger sample set (5954 samples). Each event in CSIBE-RAW was recorded with multiple microphones and sound sources to ensure the high intraclass variability and the higher chance for interclass correlation. For example, male, female and children voices in various languages (English, Spanish, Hungarian, French and Japanese) are represented in the speech event and the ambient noise class includes a diverse range of sounds. Enough samples were available on the internet except flatulence and parrot events. The church bell and traffic noises are environmental sounds, but they were included in the database because they are audible inside, similar to cicada, that is a common insect in Asia with high pitch sound. The background noise includes ambient sounds which are not important for a robot or their short audition is not sufficient for a human without visual cue: door, drawer, keys, knock, pen, chair, cup, keyboard, breathing, throat, cough, microwave, steps and zip.

CSIBE-RAW was randomly split into a training (CR_T) and a validation set (CR_V) what is denoted in the following brackets (CR_T/CR_V) for each event: *ambient noise* (326/824 samples), *baby cry* (39/201 samples), *church bell* (70/273 samples), *cat meow* (62/263 samples), *cicada* (39/221 samples), *dog bark* (149/282 samples), *flatulence* (226/292 samples), *guitar* (109/296 samples), *laugh* (154/244 samples), *parrot* (111/198 samples), *piano* (57/253 samples), *speech* (125/588 samples), *traffic* (32/277 samples) and *vacuum cleaner* (32/211 samples), in overall 5954 samples. The audio data durations of the classes were balanced in CR_T thus there were many short flatulences in CR_T, but fewer samples of cat meows, traffic and vacuum cleaner.

One file in the database corresponds to one sound event sample where the silent audio chunks were removed from. In this way, the database is easier to process because there are no additional files for labeling.

2.2 CSIBE-AIBO

The standard datasets are often not suitable to develop practical acoustic models for robotics because the training data must incorporate microphone dynamics, various noise levels and reverberant conditions. CSIBE-AIBO attempts to step forward in this direction to show how a base model can be developed by rerecording CSIBE-RAW in multiple settings.

Sony ERS-7 has two “ears” with microphones and these devices feature 16 kHz sampling rate in 16 bit depth. This robot is an embedded platform, therefore, the recording quality is not so clear like a Zoom H1 or H2 recorder. CSIBE-RAW samples were played back with a high quality speaker (Audio Pro Addon One) in silent rooms from different locations relative to the robot:

- Reverberant room, speaker was 1 meter away, 30° counterclockwise to the head.
- Non-reverberant room, speaker was 1 meter away, 30° counterclockwise to the head.
- Reverberant room, speaker was 3 meters away, 1 meter high, 180° clockwise to the head.
- Non-reverberant room, speaker was 3 meters away, 1 meter high, 180° clockwise to the head.

CSIBE-RAW samples were recorded in these room settings with stereo microphones which resulted eight times more data compared to the mono CSIBE-RAW samples. The new sounds were affected by the low-end input quality, reverberation, the recording distance, the microphone displacement relative to the source direction and the microphone self-noise. CSIBE-AIBO was separated to training and validation sets using the same partitions as CSIBE-RAW. CA'_T contains the rerecorded samples of CR_T in the first setting, CA_T in all settings and CA'_V, CA_V were generated from CR_V respectively.

Transferring the audio data from the robot was a challenge because CSIBE-RAW has more than an hour of samples and this procedure had to be automated. The internal storage of the robot could not be used to store the rerecorded samples by reasons of being small (max. 128 MB) and slow read/write speeds. The other option was to transfer the recordings from AIBO via the built-in wireless card with low throughput (appr. 30 KB/sec). The authors found the best compromise with encoding the recorded audio in lossy format and sending the compressed data to a PC on the same WLAN. The SNR of the sound events in CSIBE-AIBO varied between 8.31-16.15 dB (average: 10.85 dB) which was a bit lower than the noisy NAR Dataset (15 dB). The encoding parameters were explored during the initial experimentation and they are discussed in Chapter 4.5 and 4.6.

Recognition System

Features must be extracted from the audio to train a classifier. The most popular features in the literature are the Mel-frequency cepstrum coefficients (MFCC) (Beltrán el al., 2015; Mesaros et al., 2010; Salamon et al., 2014) which approximate the human auditory perception. In this paper, the audio data were framed by a sliding Hann-filtered window (32 msec) with 33% of overlap, fast Fourier analysis was performed to extract harmonic spectrum, spectral peaks and 26 MFCCs for each frame. The first MFCC coefficient was dropped, the remaining were added to the feature vector. There is no clear consensus in the literature about the ideal MFCC count, some earlier studies employed 13 MFCC components (Beltrán et al., 2015; Chu et al., 2009; Terence et al., 2013), but other works included 15 (Phan et al., 2015), 16 (Mesaros et al., 2010), 20 (Ruiz-Martinez et al., 2013), 26 (Salamon et al., 2014) and 40 (Nouza et al., 2013).

The feature extraction was done with the libxtract library (Bullock, 2007) in C++ and the implementation details of each audio statistic can be found in the github repository[1]. The following 23 statistics were calculated to complement 25 MFCC to 48 features:

- Audio data frames: standard deviation, maximum, min-max range, kurtosis, fundamental frequency, non-zero count, average deviation, variance and zero crossing rate.
- FFT spectrum: pitch of Harmonic Product Spectrum analysis, irregularity (Jensen, 1999), centroid, variance and standard deviation.
- Bark coefficients: loudness.
- Peak spectrum: standard deviation, partials count (non-zero component count) and centroid.
- Harmonic spectrum: arithmetic mean and tristimulus (Pollard and Jansson, 1982).
- MFCC frames: minimum, arithmetic mean and standard deviation.

These statistics were selected by sequential forward floating feature selection and an iterative examination of the feature importances with cross-validation in the initial experiments. Two features (spectral crest, variance of spectral harmonics) were removed since they were sensitive to lossy encoding. Eventually, the feature vector had small computational cost (1 msec) on the robot and it was robust to lossy audio codecs.

The temporal frame integration (superframes, bag-of-words) was tried without satisfactory results in the initial experiments, therefore, majority voting was used for temporal smoothing to do the classification of each sample file in the next chapters. When a label must be associated with a file, feature vectors are extracted, predicted with a classifier and the label with the most predictions is voted to be the final.

Experimental Results

4.1 Classifier Comparison

The classifiers were mostly implemented with the Machine Learning module of OpenCV (Bradski and Kaehler, 2008), except maximum entropy (ME) (Andrew and Gao, 2007; Tsuruoka et al., 2009), convolutional neural network with tiny-dnn[2] and SVM with linear kernel (King, 2009). The latter was chosen because of the SVM codes in OpenCV uses an old fork of libsvm with custom modifications and several users reported reduced performance with linear kernel behind libsvm. The linear SVM with Dual Coordinate Descent Method (Hsieh et al., 2008) in Dlib provided better

[1] https://github.com/jamiebullock/LibXtract

[2] https://github.com/tiny-dnn/tiny-dnn

accuracy for the authors with this dataset. The SVMs used C = 0.1 parameter while ME was regularized with an Orthant-Wise Limited-memory Quasi-Newton Optimizer (L1 = 0.00001). The hyperparameters for the decision tree and random forest were $Tree_{depth} = 20$, $Forest_{size} = 20$ and the minimal sample count for node split was set to 100.

Several earlier studies focused on deep neural networks for sound event recognition (Hertel et al., 2016; Choi et al., 2016; Cakir et al., 2016). Some explored the input features (Dennis, 2014; McLoughlin et al., 2015; Hertel et al., 2016) for the networks, some compared different network topologies (McLoughlin et al., 2015; Phan et al., 2016). This paper introduces a simple convolutional neural network without automatic feature extraction which can be deployed to embedded systems. The first two layers in the proposed neural network were fully-connected with 200 units, one convolutional layer had 9x1 kernel with stride 1 and the last fully-connected layer contained 100 neurons. The fully-connected layers had leaky rectified linear activation function and the convolutional layer had tanh. The CNN training was executed for 50 epochs with adaptive gradient method (adagrad) and batch size 64.

Each hyperparameter was specified with parameter search on preliminary data during the initial experimentation. The features were rescaled to [0, 1] for CNN, other classifiers got the data after standardization before training and prediction.

The previous datasets were evaluated with cross-validation (CV) (Beltrán et al., 2015; Stowell et al., 2015; Terence et al., 2013) what estimates the model accuracy, but the proper validation is done with unseen data. The 10-fold cross-validation was done with the training sets of CSIBE (CR_T, CA'_T) in this paper and the model generalization was explored by building models with these sets and evaluating them with CR_V or CA'_V.

Fig. 2 shows the CV of nine classifiers with the training set of CSIBE-RAW (CR_T) and the rerecorded version (CA'_T) from CSIBE-AIBO where all results were calculated with majority voting. Apart from the standard, frame-based evaluation, aggregated frames (Bergstra et al., 2006) were computed by replacing the original frames with temporally calculated mean and standard deviation of every 9 frames with 30% overlap ($CR_{T,a9}$, $CA'_{T,a9}$). This data aggregation results smaller training set size by negligible computational costs and improved accuracy is expected compared to the frame-based evaluation (Terence et al., 2013). As we can see, the aggregated frames enhanced performance (even columns are higher) except SVM_{RBF}, KNN and DT what was a bit surprising. CNN was the top performing classifier in all cases, nevertheless, KNN, DT, RF provided solid accuracies over 80%. Because of SVM_{RBF} and expectation-maximization (EM) with Gaussian mixtures did not reach the performance of other classifiers, they were left out from the further analysis.

SVM and EM can collapse if the training dataset is too big, the training time can increase rapidly and performance is degraded. These algorithms can perform better if the frame count is reduced with sliding window approaches (mean and standard deviation vectors or ARMA models) (Maxime, et. al., 2014; Terence et al., 2013). SVM_{RBF}, NB and EM on Fig. 2 had the biggest improvements among the other classifiers when the cross-validation was done with aggregated frames ($CR_{T,a9}$, $CA'_{T,a9}$) which reduced the training set sizes to 1/3. Although the aggregation yielded the least improvement for CNN, but its training time was reduced by 1/3 what is important to speed up the slow training of deep learning networks.

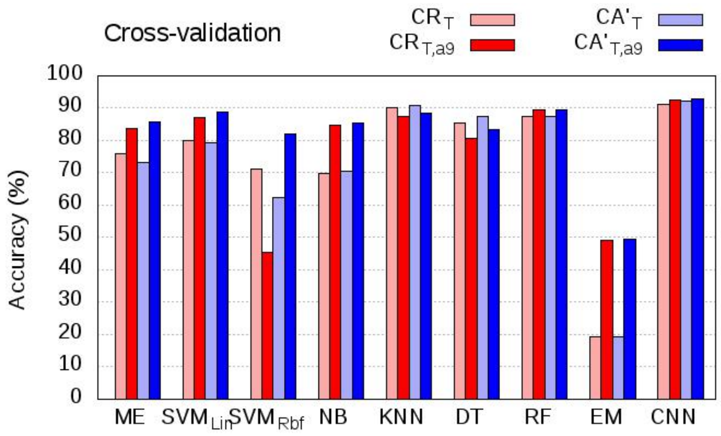

Fig. 2. Cross-validation results for several classifiers over the training set (CR$_T$) of CSIBE-RAW (red) and CA'$_T$ of CSIBE-AIBO (blue). The lighter colors (odd columns) were the frame-based votes and the aggregated frames were used in the darker (even columns), but the final results were calculated with majority voting. Classifiers: maximum entropy (ME), support vector machine with linear kernel (SVM$_{Lin}$), support vector machine with radial basis function (SVM$_{Rbf}$), naïve Bayes (NB), k-nearest neighbors (KNN), decision tree (DT), random forest (RF), expectation-maximization (EM) algorithm and convolutional neural network (CNN).

The past studies used various methods to check their datasets and a natural point is how the baseline system (BS) in this paper relates to the literature. Two public corpora were tested by BS with the same preprocessing steps of the original studies, but distinct classifier implementations and audio statistics. The dataset A of CICESE was cross-validated with a hidden Markov model (HMM) to 98% (F-score) in (Beltrán et al., 2015) while BS reached 99.2% accuracy with KNN, 98.8% with DT and 98.3% with RF. The file averaged NAR dataset was cross-validated to 88.4% with KNN and 91.5% with SVM in (Maxime, et. al., 2014) although BS had 91.41% with KNN and 92.1% with SVM$_{Lin}$. These results showed that the recognition system of this paper provides the same or slightly better performance compared to (Beltrán et al., 2015) and (Maxime, et. al., 2014).

4.2 Model Evaluation

The cross-validation estimates the model accuracy with unseen data, but it can lead to misunderstandings about the generalization power. Fig. 3 shows the model evaluations for seven classifiers which were selected after the cross-validation in Chapter 4.1. Models were built with the aggregated frames of the previous training sets (CR$_{T,a9}$, CA'$_{T,a9}$) and they were evaluated with the aggregated frames of their validation sets (CR$_{V,a9}$, CA'$_{V,a9}$) to identify any difference in the performance with unseen samples. The aggregated frame-based accuracies (lighter columns) varied between 70%-90%, the majority voting enhanced these results up to 90%-96% (darker columns). All classifiers were satisfactory after majority voting, but CNN was again the best in every situation. The actual model accuracies were underestimated by the cross-validation because all CR$_{mv}$ and CA'$_{mv}$ evaluations in Fig. 3 were over 90% while almost none of the cross-validations of the same classifiers reached 90% in Fig. 2.

CSIBE-RAW contains the collected samples from the internet and CSIBE-AIBO the rerecorded versions. The original sounds achieved higher frame-based results in all cases (CR$_{cf}$ vs. CA'$_{cf}$ in Fig. 3), but the majority voting turned this into the other direction and every CA'$_{mv}$ was higher than CR$_{mv}$. The classifiers delivered accuracies over 90% with majority voting regardless of the datasets (CR vs. CA'). The next subchapter will examine why the multi-conditional learning is needed for real-world applications.

4.3 Multi-conditional Learning (MCL)

The room reverberation and the microphone dynamics alter the feature vector and may raise the recognition error significantly. In MCL, the classifier is trained with samples with different distortions and the built model will be robust to these conditions. Training and validation sets were constructed in Chapter 2.2 for MCL, CA'_T contained rerecorded samples from one room setting and this training set represents the learning in one condition. However, the same sample files rerecorded in all four settings were included in CA_T to provide a training set with multiple reverberant and SNR conditions.

In (Mesaros et al., 2010), the isolated sound events were recognized with a GMM-HMM model and the system had 53% accuracy for clean samples, 47% for 10 dB SNR, 38% for 5 dB SNR and it decreased to 28% for 0 dB SNR. Ruiz-Martinez et al implemented SVM for environmental sounds (Ruiz-Martinez et al., 2013) and their model achieved 89% accuracy for clean samples, 85% for 10 dB SNR, 79% for 5 dB and 71% for 0 dB. These earlier works had considerable loss in the performance by adding artificial noise what can be handled with some solutions. The model adaption, signal enhancement and feature compensation can make the system more robust against noise (Dennis, 2014), but this paper uses the multi-conditional learning to build models that are tolerant for lower SNR, reverberant and non-reverberant conditions. On one hand, this learning method requires large training set, on the other hand, the authors wanted to avoid synthetic training data what often is not applicable outside the laboratory environment. The CSIBE-AIBO corpus was built according to these principles (Chapter 2.2) although the rerecording in four settings was a time consuming process.

The aforementioned problems were analyzed in Table I how DT, RF, SVM_{Lin} and CNN classifiers performed in model evaluation when the training and the validation sets were varied. The first two rows represent the baseline and come from Fig. 3. The model of the first row was built and evaluated on the clean samples of CSIBE-RAW, all accuracies were over 90%. Similar results were achieved with rerecorded sets in the second row because the recording environment altered both the training and the validation sets.

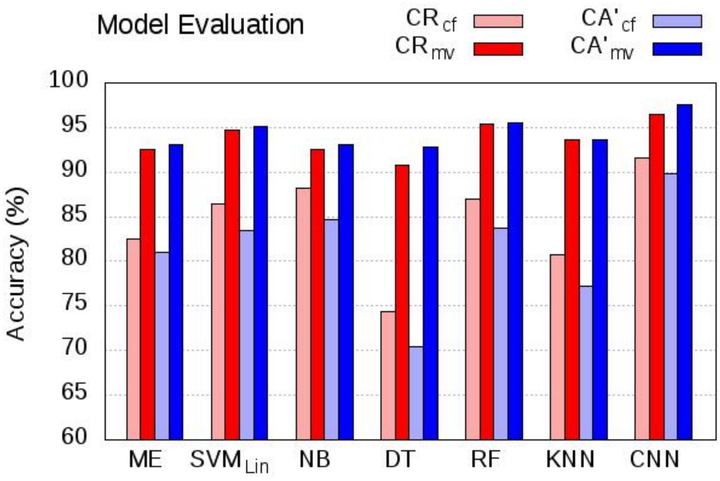

Fig. 3. Model evaluation of seven classifiers over the validation set (CR_V) of CSIBE-RAW (red) and CA'_V of CSIBE-AIBO (blue). The lighter colors (odd columns) were the aggregated frame-based evaluations and majority voting was used in the darker (even columns) in addition.

Table I. Model Evaluation without Multi-conditional Learning

Decision tree, random forest, linear svm and convolutional neural network performances with different training and validation sets. All results were calculated with aggregated frames and majority voting. The first two rows contain accuracies from Fig. 3.

Training Set	Validation Set	DT	RF	SVM_{Lin}	CNN
$CR_{T,a9}$	$CR_{V,a9}$	90.78	95.41	94.69	**96.45**
$CA'_{T,a9}$	$CA'_{V,a9}$	92.83	95.57	95.07	**97.54**
$CR_{T,a9}$	$CA'_{V,a9}$	65.36	74.84	79.20	**79.88**
$CR_{T,a9}$	$CA_{V,a9}$	38.36	44.28	**53.22**	52.02
$CA'_{T,a9}$	$CA_{V,a9}$	62.30	62.21	64.62	**66.49**

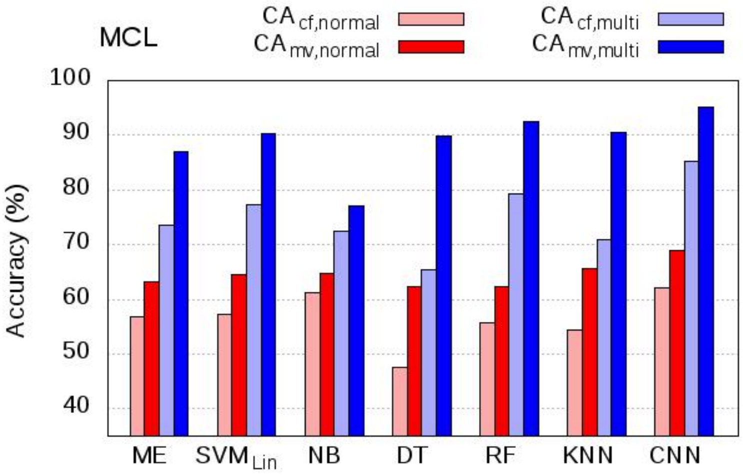

Fig. 4. Model evaluation with and without multi-conditional training. The validation set is $CA_{V,a9}$ of CSIBE-AIBO in all cases, but the training set is $CA'_{T,a9}$ for the red columns and $CA_{T,a9}$ for blue columns. The lighter colors show the aggregated frame-based evaluation and the darker represent the results after majority voting.

When the clean training set ($CR_{T,a9}$) was tested against the rerecorded validation set in one room setting ($CA'_{V,a9}$), the accuracies were decreased by 15-25% in the third row compared to the first. Evaluating $CR_{T,a9}$ with the rerecorded validation set in all four settings (fourth row), the performance was dropped even more (41-52%) compared to the first row. These results confirmed that models built on the original samples in CSIBE-RAW were not successful to generalize the recording conditions of CSIBE-AIBO. The last row in Table I contains the results for rerecorded training set in one setting ($CA'_{T,a9}$) and rerecorded validation set in four room settings ($CA_{V,a9}$). Despite the both sets were affected by the robot microphone and the reverberation, the accuracies were as low as 62-64% since $CA_{V,a9}$ was altered by all four settings.

To summarize the findings:

- CNN delivers the best performance almost every time without MCL (except the fourth row in Table I).
- The more challenges the validation set have (lower SNR, reverberation), the more the accuracies decrease.
- Deep neural networks cannot handle these problems with hand-crafted features. (Moving the feature extraction to autoencoders can be a solution if large amount of training data and GPU power are available.)

Table II. Classifier Properties

The accuracies are $CA_{MV,MULTI}$ results from Fig. 4. The training times are Single-threaded for SVM, DT and RF, but multithreaded for CNN on CPU. prediction time per feature vector is presented for SVM and CNN because they fit in the memory of the robot.

	SVM_{Lin}	DT	RF	KNN	CNN
Accuracy (%)	90.2	89.9	92.6	90.5	95.07
Training time (sec)	56	13	213	-^{*1}	3359
Memory usage (MB)	0.364	64.5	1136.7	63.6	3.5
Prediction time (msec)	0.36	-^{*2}	-^{*2}	-^{*2}	6

^{*1} KNN classifier does not have training step.

^{*2} DT, RF and KNN classifiers do not fit in the 64 MB RAM of the robot.

Table III. Confusion Matrix of CNN Model.

The rows show the real events and the columns how they were classified. (Ambient noise: AN; Baby cry: BC; Bell: B; Cat meow: CM; Cicada: C; Dog bark: DB; Flatulence: F; Guitar: G; Laugh: L; Parrot: PA; Piano: PI; Speech: S; Traffic: T; Vacuum cleaner: V)

%	AN	BC	B	CM	C	DB	F	G	L	PA	PI	S	T	V
AN	**89**						1		1			3		
BC		**92**		1						4				
B			**97**								1			
CM		3		**94**										
C					**100**									
DB						**98**								
F	5						**86**		1			3		
G								**99**						
L	1								**91**			6		
PA										**97**				
PI								3			**94**	1		
S												**98**		
T													**97**	
V														**99**

As a consequence, when big data is not accessible, multi-conditional learning is required to solve these obstacles. McLoughlin et al trained deep neural networks with spectrogram image features (McLoughlin et al., 2015) from different noise conditions and the models delivered similar accuracies for the clean and 20dB SNR testing samples though 1-6% decreases were for 10dB. In (Terence et al., 2013), when the MFCC features were trained to GMM, the model had 67.40% accuracy without MCL and 95.12% with MCL. Dennis had two systems based on SVM and HMM (Dennis, 2014) and dropped 20-90% accuracy without MCL, but the degradation was reduced to 2-30% under 0-20 dB SNR conditions with MCL. According to these earlier works, multi-conditional learning is an effective method to deal with different SNRs.

$CA_{cf,normal}$ and $CA_{cf,multi}$ (red columns) in Fig. 4 present seven classifiers trained with the aggregated frames of $CA'_{T,a9}$ and evaluated on $CA_{V,a9}$, similar to the fifth row in Table I. All classifiers delivered low accuracies (62-66%) after majority voting, none of them could generalize to the three unknown rerecording settings in the validation set $CA_{V,a9}$. Once the training set comprehended the rerecorded samples of CR_T in all four room settings ($CA_{T,a9}$), the multi-conditional learning improved the results ($CA_{cf,multi}$) by 24-30% and achieved 87-95% accuracies (dark blue columns) except naïve Bayes which had 77.10% after majority voting. As it happened in Fig. 2 and 3, CNN outperformed other algorithms again. The top-5 classifiers were picked for further analysis to select a final model for real-time usage on the robot.

4.4 Classifier Selection

The support vector machines (Salamon et al., 2014; Stowell et al., 2015; Terence et al., 2013) and KNN (Chmulik and Jarina, 2012; Plinge et al., 2014; Theodorou et al., 2014) have been widely implemented for sound event recognition while the decision tree-based classifiers have been received less attention (Delgado-Contreras et al., 2014; Phan et al., 2015; Salamon et al., 2014) and deep learning is the current mainstream (Hertel et al., 2016; Phan et al., 2016; Choi et al., 2016). To choice the final model, multiple aspects must be considered such as accuracy, training time, memory usage and prediction time. All classifiers in Table II show reasonable accuracies between 89.9%-95.07% which satisfied the first criterion. When the training set size is increased, the DT and RF models grow larger (Sebbanü et al., 2000; Sug, 2009). Although the RF model had the second best accuracy (92.6%) in Table II, but the memory consumption was over 1 GB after learning 165872 aggregated frames what was not acceptable for embedded systems. Similarly, the DT model (64 MB) also did not fit in the memory. In a previous work, KNN performed closely to SVM in (Maxime, et. al., 2014) and this classifier does not include a training phase, nevertheless, the training set must be cached in the memory and the bigger the training set, the longer the prediction time. Namely, KNN with CA_T can make one prediction in 23 msec on a high-end AMD FX 8350 desktop CPU which is not enough for real-time processing on the robot and the training set also does not fit in 64 MB RAM. Because of these reasons, DT, RF and KNN were not suitable for eventual tests on the robot.

CNN had the best accuracy (95.07%) after 1 hour-long training with moderate memory and CPU usage on the robot (Table II), therefore, this classifier was selected for onboard deployment. It is worth mentioning that SVM_{Lin} is a good alternative to CNN if some accuracy can be scarified for negligible memory usage (364 KB) and prediction time (0.36 msec).

The confusion matrix of the CNN model (accuracy: 95.07%, F-score: 95.54%, precision: 95.71%, recall: 95.47%) is shown in Table III where the cells were left blank if they contained less than 1%. The cicada samples were recognized all the time correctly because of the unique voice characteristics (high pitch) of this animal. Some sound events were challenging for the model because the laugh had similarities with the human speech (6% misclassification), the flatulences were short events and harder to distinguish from the ambient noise. In overall, the events were recognized with adequate accuracies (>85%).

The authors executed a preliminary test with the CNN model after multi-conditional learning. This model was deployed to the robot, feature vectors were generated directly from the microphone data. CNN predicted well live input, but the implementation details of a final recognition system on a Sony ERS-7 are out of scope in this paper.

4.5 Lossy Encoding Effects

As it was described in Chapter 2.2, CSIBE-AIBO was recorded with lossy Ogg encoder. This chapter explains the codec selection procedure and how the VBR settings were determinated. The target was to find a lossy codec which does not effect the classifier accuracy if either the training or the validation set is transcoded. *Full transcoding* denotes when both the training and the validation set are transcoded with the same lossy codec before the model building and evaluation processes. This is relevant for storing large audio databases in the fraction of the original disk space and using these big data to train and test deep neural networks without performance degradation. *Half transcoding* means transcoded training set and unaltered validation set. This evaluation step ensures that the DNN models built with full transcoding can be deployed on consumer devices where the model will recognize uncompressed audio from a real microphone.

Two popular lossy compression algorithms were examined, the Ogg Vorbis transcoding was implemented with libvorbis[3], the MP3 encoding with libmp3lame[4] and the MP3 decoding with LAME's mpglib version. The effects of lossy encoding has been studied in the literature for speech (Besacier et al., 2001; Pollak and Behunek, 2011; Sáenz-Lechón et al., 2008) and music (Uemura et al., 2014; Urbano et al., 2014) classifications, but this paper provides the first analysis for sound event recognition. The cited papers from the literature examined only the MP3 encoding while both Ogg and MP3 codecs are reviewed here. Some past works trained the acoustic model with uncompressed audio and the training set was transcoded with lossy codecs to check the effects (Besacier et al., 2001; Borsky et al., 2015), but half transcoding was employed in (Ng et al., 2004) and full transcoding in (Nouza et al., 2013; Uemura et al., 2014; Urbano et al., 2014). Minimum 32 kbit MP3 profile was sufficient to avoid performance decrease for speech recognition in (Besacier et al., 2001; Ng et al., 2004) and 64 kbit in (Sáenz-Lechón et al., 2008). Uemura et al (Uemura et al., 2014) found 32 kbit VBR enough for chord recognition while Urbano et al preferred at least 160 kbit CBR MP3 encoding (Urbano et al., 2014).

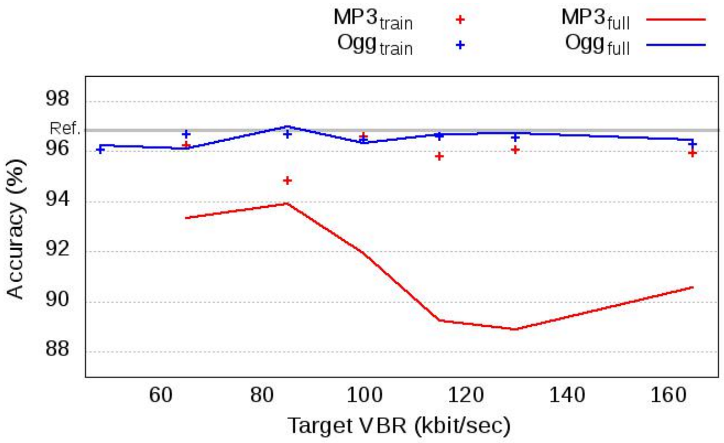

Fig. 5. CNN model performance when both the training ($CR_{T,a9}$) and the validation set ($CR_{V,a9}$) of CSIBE-RAW were transcoded with lossy codecs ($MP3_{full}$, Ogg_{full}) and when only the training set was transcoded ($MP3_{train}$, Ogg_{train}), but $CR_{V,a9}$ remained in wave format. The results are shown in the function of the target VBR bitrate. The gray reference line (Ref) shows the baseline CNN accuracy with the untouched training ($CR_{T,a9}$) and validation sets ($Cr_{v,a9}$).

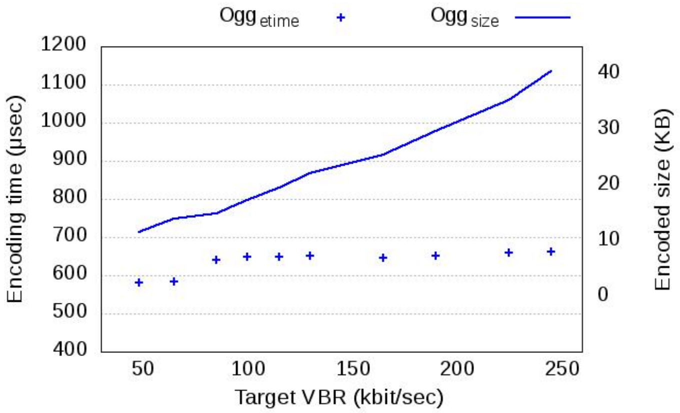

Fig. 6. Processing times on the robot and the encoded sizes of 3-seconds long, 16 kHz, stereo audio chunk with Ogg codec. The results are shown in the function of the target VBR bitrate. The crosses are related to the left scale, the lines to the right scale.

[3] https://xiph.org/vorbis/
[4] http://lame.sourceforge.net

The lossy codec influence on CSIBE-RAW was investigated in Fig. 5. A gray line (Ref) shows the baseline performance of the convolutional neural network model with uncompressed data, $MP3_{train}$ and Ogg_{train} were obtained with half transcoding as well as $MP3_{full}$ and Ogg_{full} with full transcoding. The accuracies of $MP3_{train}$ and Ogg_{train} matched at 100 kbit VBR otherwise the MP3 encoding caused 0.2-1.2% loss against the Ogg Vorbis results. Ogg_{train} lost maximum 0.8% from the reference line even on lower bitrates, therefore, Ogg Vorbis is recommended for half transcoding on any bitrates.

The full transcoding with MP3 caused surprise because $MP3_{full}$ was lower by 2.7-7.9% from Ogg_{full}, especially on higher bitrates what contradicts the expectation of good quality over 128 kbit VBR. There might be some special encoding settings in LAME which can make some frequency bands sensitive to the MP3 format. Further investigation is needed later to answer this question.

Ogg_{full} (blue line in Fig. 5) delivered very similar accuracies compared to Ogg_{train} thus the same suggestion applies, Ogg Vorbis codec is advised for full transcoding and the accuracy did not decrease over 128 kbit VBR in comparison with the CNN model built from uncompressed audio.

Ogg Vorbis achieved minimal losses in accuracy in both full and half transcoding, therefore, this format is advised to store big audio databases for deep learning training in data centers.

4.6 Lossy Encoding on AIBO

The targets for the rerecorded samples on AIBO were the small compressed size and the short encoding speed. The small size saved wireless bandwidth and the encoding speed shortened the wait time when the samples were recorded again and collected for multi-conditional learning. Fig. 6 shows these variables in the function of different Ogg VBR settings. All processing times (blue crosses) were between 590-680 μsec, but the produced data size was 4 times bigger between 50 kbit and 250 kbit VBR profiles. To determinate the best compromise between the quality and encoding speed, the Ogg Vorbis codec performance can be compared on the Fig. 5 and Fig. 6. The higher variable bitrates (>128 kbit) of Ogg Vorbis increase the encoded data size (Ogg_{size} in Fig. 6) without offering additional performance (Ogg_{full} in Fig. 5). Therefore, 128 kbit VBR setting for Ogg Vorbis was the optimal selection to avoid any loss in accuracy caused by audio compression when CSIBE-RAW was rerecorded with the robot for CSIBE-AIBO. Once the audio was encoded with an average compression ratio 1:8, the data were transferred from the robot to a PC in a few hundred milliseconds via wireless network.

Conclusion

The paper described how the CSIBE corpora were created for non-overlapping sound event recognition in the robotics field. The samples were mainly gathered from free internet sources to build a redistributable CSIBE-RAW. This database contained 14 sound events where 13 events represented human speech, animal voices, musical instruments and household appliances. One special class modeled the ambient noises (e.g knock, drawer, keyboard, paper, breathing, steps) which are not important for a domestic robot. CSIBE-RAW was compared to the literature, its size (5954 sample files) was higher than the existing databases for indoor environment and the modeling of uninteresting events (ambient noise class) was also unique.

CSIBE-RAW was rerecorded with a stereo microphone of a robot in four room settings (CSIBE-AIBO) to train acoustic models which were tolerant to reverberant conditions and challenging SNR levels. Multiple experiments were carried out to find the optimal classifier and lossy encoding settings to deploy a real-time capable acoustic model on a Sony ERS-7 robot. The convolutional neural network was the appropriate classifier with multi-conditional learning to reach 95.07% accuracy with unseen audio data from CSIBE-AIBO.

Further contributions of the paper were the reported audio statistics in the recognition system which improved the standard MFCC results and they were robust against the lossy encodings hence the previously mentioned CNN model was built with compressed audio data. The lossy Ogg Vorbis and MP3 codecs were studied and the results suggested to select the Ogg Vorbis format with 128 kbit VBR profile.

Future work can include the introduction of new sound classes to the CSIBE corpora to recognize more environmental events, working out the details of the live sound event recognition on the robot and the investigation of the performance loss with MP3 codec with high VBR profiles.

The authors would like to emphasize that the lack of the classifier hyperparameters makes the reported performance measurements hard to interpret because direct comparisons will not be possible with new methods. For example, the NAR dataset evaluation (Maxime, et. al., 2014) involved SVM classifier, but it is unclear which kernel (linear, radial basis function or polynomial) and hyperparameters were used. The authors of this paper encourage the future works to present the classifier hyperparameters for reproducible research.

ACKNOWLEDGMENT

Nokia Foundation provided a grant (ID: 201510141) for two months in 2015 when the CSIBE-RAW dataset was collected, CSIBE-AIBO was recorded and the initial baseline system was implemented. We want to say special thanks to Toni Heittola and Annamaria Mesaros from Tampere University of Technology for their invaluable comments to overcome some problems.

REFERENCES

Andrew, G., Gao, J., 2007. Scalable training of L1-regularized log-linear models," in Proc. 24th Intl. Conf. on Machine learning, pp. 33-40.

Beltrán, J., Chávez, E., Favela, J., 2015. Scalable identification of mixed environmental sounds, recorded from heterogeneous sources, Journal of Pattern Recognition Letters, Vol. 68, pp. 153-160.

Besacier, L., Bergamini, C., Vaufreydaz, D., Castelli, E., 2001. The effect of speech and audio compression on speech recognition performance, in Proc. of 4th IEEE Multimedia Signal Processing, pp. 301-306.

Borsky, M., Pollak, P., Mizera, P., 2015. Advanced acoustic modelling techniques in MP3 speech recognition, EURASIP Journal on Audio, Speech, and Music Processing, Vol. 2015(1), pp. 1-7.

Bradski G. R., Kaehler, A., 2008. Learning OpenCV, 1st Edition, O'Reilly Media, 2008.

Bullock, J., 2007. LibXtract: A lightweight library for audio feature extraction, in Proc. Intl. Computer Music Conference.

Chmulik, M., Jarina, R., 2012. Bio-inspired optimization of acoustic features for generic sound recognition, in Proc. 19th Intl. Conf. on Systems, Signals and Image Processing (IWSSIP), pp. 629-632.

Chu, S., Narayanan, S., Kuo, C. C. J., 2009. Environmental Sound Recognition With Time–Frequency Audio Features, IEEE Transactions on Audio, Speech, and Language Processing, Vol. 17(6), pp. 1142-1158.

Delgado-Contreras, J. R., Garcia-Vazquez, J. P., Brena, R. F., 2014. Classification of environmental audio signals using statistical time and frequency features, in Proc. Intl. Conf. on Electronics, Communications and Computers (CONIELECOMP), pp. 212-216.

Dennis, J., 2014. Sound Event Recognition in Unstructured Environments Using Spectrogram Image Processing, PhD Thesis, Nanyang Technological University.

Foster, P., Sigtia, S., Krstulovic, S., Barker, J., 2015. CHiME-Home: A Dataset for Sound Source Recognition in a Domestic Environment, in Proc. 11th IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).

Goldstein, E. B., 2010. Sensation and Perception, Wadsworth, p. 490.

Hsieh, C.-J., Chang, K.-W., Lin, C.-J., 2008. A Dual Coordinate Descent Method for Large-scale Linear SVM, in Proc. 25th Intl. Conf. on Machine Learning, pp. 408-415.

Jensen, K., 1999. Timbre Models of Musical Sounds, PhD. Dissertation, DIKU Report.

King, D. E., 2009. Dlib-ml: A Machine Learning Toolkit, Journal of Machine Learning Research, Vol. 10, pp. 1755-1758.

Maxime, J., Alameda-Pineda, X., Girin, L., Horaud, R., 2014. Sound representation and classification benchmark for domestic robots, in Proc. IEEE Intl. Conf. Robot. Autom. (ICRA).

McLoughlin, I., Zhang, H., Xie, Z., Song, Y., Xiao, W., 2015. Robust Sound Event Classification Using Deep Neural Networks, IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 23(3), pp.540-552.

Mesaros, A., Heittola, T., Eronen, A., Virtanen, T., 2010. Acoustic event detection in real life recordings, in Proc. EUSIPCO.

Ng, P. S., Sanches, I., 2004. The influence of audio compression on speech recognition systems, in Proc. 9th Conf. Speech and Computer, 2004.

Ness, S., Trail, S., Driessen, P., Schloss, A., Tzanetakis, G., 2011. Music Information Robotics: Coping Strategies for Musically Challenged Robots, in Proc. 12th International Society for Music Information Retrieval Conference (ISMIR), pp. 567-572.

Nouza, J., Cerva, P., Silovsky, J., 2013. Adding controlled amount of noise to improve recognition of compressed and spectrally distorted speech, in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8046-8050.

Phan, H., Maas, M., Mazur, R., Mertins, A., 2015. Random Regression Forests for Acoustic Event Detection and Classification, IEEE/ACM Transactions in Audio, Speech, and Language Processing, Vol.23(1), pp. 20-31.

Plinge, A., Grzeszick, R., Fink, G. A., 2014. A Bag-of-Features Approach to Acoustic Event Detection, in Proc IEEE Intl. Conference on Acoustics, Speech, and Signal Processing.

Pollak, P., Behunek, M., 2011. Accuracy of MP3 speech recognition under real-word conditions: Experimental study, in Proc. IEEE Signal Processing and Multimedia Applications (SIGMAP), pp. 1-6.

Pollard, H. F., Jansson, E. V., 1982. A Tristimulus Method for the Specification of Musical Timbre, Journal of Acustica, Vol. 51, pp. 162–71.

Ruiz-Martinez, C. A., Akhtar, M. T., Washizawa, Y., Escamilla-Hernandez, E., 2013. On investigating efficient methodology for Environmental Sound Recognition, in Proc Intl. Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), pp. 210-214.

Sáenz-Lechón, N., Osma-Ruiz, V., Godino-Llorente, J. I., 2008. Effects of audio compression in automatic detection of voice pathologies, IEEE Transactions on Biomedical Engineering, Vol. 55(12), pp. 2831-2835.

Salamon, J., Jakoby, C., Bello, J. P., 2014. A Dataset and Taxonomy for Urban Sound Research, in Proc. 22nd ACM International Conference on Multimedia, pp. 1041-1044.

Sebbanü, M., Nock, R., Chauchat, J., Rakotomalala, R., 2000. Impact of learning set quality and size on decision tree performances, Intl. Journal of Computers Systems and Signals, Vol 1(1), pp. 85-105.

Stowell, D., Stowell, D., Benetos, E., Lagrange, M., Plumbley, M. D., 2015. Detection and Classification of Acoustic Scenes and Events, IEEE Transactions on Multimedia, Vol. 17(10), pp. 1733-1746.

Sug, H., 2009. An effective sampling method for decision trees considering comprehensibility and accuracy, WSEAS Transactions on Computers, Vol. 8(4), pp. 631-640.

Terence, N. W. Z., Dat, T. H., Dennis, J., Siong, C. E., 2013. A robust sound event recognition framework under TV playing conditions, in Proc. Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp.1-5, 2013.

Theodorou, T., Mporas, I., Fakotakis, N., 2014. Audio Feature Selection for Recognition of Non-linguistic Vocalization Sounds, in Proc. Hellenic Conference on Artificial Intelligence, pp. 395-405.

Tsuruoka, Y., Tsujii, J., Ananiadou, S., 2009 . Stochastic Gradient Descent Training for L1-regularized Log-linear Models with Cumulative Penalty, in Proc. ACL-IJCNLP, pp. 477-485.

Uemura, A., Kazumasa, I., Katto, J., 2014. Effects of audio compression on chord recognition, in Proc. Intl. Conf. on Multimedia Modeling, pp. 345-352.

Urbano, J., Bogdanov, D., Herrera, P., Gómez, E., Serra, X., 2014. What is the Effect of Audio Quality on the Robustness of MFCCs and Chroma Features?, in Proc. 15th ISMIR Conference, pp. 573-578.

Yamamoto, S., Nakadai, K., Nakano, M., et al., 2006. Real-time robot audition system that recognizes simultaneous speech in the real world, in Proc. Intl. Conf. on Intelligent Robots and Systems (IROS), pp. 5333–5338.

Bergstra, J., Casagrande, N., Erhan, D., et al., 2006. Aggregate features and AdaBoost for music classification, Journal of Machine Learning, Vol. 65(2), pp 473–484.

Hertel, L., Phan, H., Mertins, A., 2016. Comparing Time and Frequency Domain for Audio Event Recognition Using Deep Learning, in Proc. IEEE Intl. Joint Conf. on Neural Networks (IJCNN 2016), arXiv:1603.05824.

Phan, H., Hertel, L., Maass, M., et al., 2016. Robust Audio Event Recognition with 1-Max Pooling Convolutional Neural Networks, in Proc. 17th Annual Conf. of the Intl. Speech Communication Association (INTERSPEECH 2016), arXiv:1604.06338.

Choi, I., Kwon, K., Hyun Bae, S., et al., 2016. DNN-Based Sound Event Detection with Exemplar-Based Approach for Noise Reduction, in Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE2016).

Cakir, E., Heittola, T., Huttunen, H., et al., 2016. Polyphonic sound event detection using multi label deep neural networks, in Proc. IEEE Intl. Joint Conf. on Neural Networks (IJCNN 2016).