Applied Acoustics 171 (2021) 107643

Contents lists available at ScienceDirect

Applied Acoustics

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

Detection and classification of human-produced nonverbal audio events

Philippe Chabot, Rachel E. Bouserhal, Patrick Cardinal, Jérémie Voix *

École de technologie supérieure, 1100 Notre-Dame St W, Montréal, Québec H3C 1K3, Canada

ARTICLE INFO

Article history:
Received 13 January 2020
Received in revised form 29 June 2020
Accepted 31 August 2020
Available online 1 October 2020

Keywords:
Hearable
Nonverbal
Classification
Audio event detection
Biosignals

ABSTRACT

Audio wearable devices, or hearables, are becoming an increasingly popular consumer product. Some of these hearables contain an in-ear microphone to capture audio signals inside the user's occluded earcanal. Mainly, the microphone is used to pick up speech in noisy environments, but it can also capture other signals, such as nonverbal events that could be used to interact with the device or a computer. Teeth or tongue clicking could be used to interact with a device in a discreet manner, and coughing or throat-clearing sounds could be used to monitor the health of a user. In this paper, 10 human produced nonverbal audio events are detected and classified in real-time with a classifier using the Bag-of-Audio-Words algorithm. To build this algorithm, different clustering and classification methods are compared. Mel-Frequency Cepstral Coefficient features are used alongside Auditory-inspired Amplitude Modulation features and Per-Channel Energy Normalization features. To combine the different features, concatenation performance at the input level and at the histogram level is compared. The real-time detector is built using the detection by classification technique, classifying on a 400 ms window with 75% overlap. The detector is tested in a controlled noisy environment on 10 subjects. The classifier had a sensitivity of 81.5% while the detector using the same classifier had a sensitivity of 69.9% in a quiet environment.

1. Introduction

The human body produces numerous audio signals, besides speech, that could be exploited for human–machine interfacing. These audio signals could be used to communicate silently with a device or to have a better understanding of the health and emotional state of a user. A click of the tongue could be used to stop or resume playback; excessive grinding of the teeth could indicate that someone is stressed or anxious [1], while swallowing saliva at a higher rate could indicate a user's level of emotional arousal [2]. Likewise, sickness could be detected from excessive coughing or clearing of the throat.

These nonverbal human-produced audio signals are subtle and, in some cases, inaudible. An efficient way to capture these signals is through the use of an occluding intra-aural device, equipped with an in-ear microphone (IEM). It is important that the intra-aural device create an acoustical seal within the user's earcanal. The seal ensures that external sounds are attenuated, while simultaneously inducing *the occlusion effect*, an amplification of low frequency vibrations produced inside the body and propagated through bone and tissue conduction [3]. These nonverbal signals are amplified due to the occlusion effect and can be picked up using an IEM.

In [4], Martin & Voix used an intra-aural device equipped with an IEM to detect heart beat signals and respiratory rate. Since heartbeat and breathing rate are well documented periodic signals, this was achieved using traditional signal processing techniques, such as envelope detection. However, such techniques are incapable of detecting aperiodic signals that have not been previously studied such as the clicking or grinding of teeth. Recent advancements in machine learning techniques for non-speech audio event detection and classification [5–7], now make the detection of such events possible.

For audio classification tasks, the most widely used machine learning algorithms are Convolutional Neural Networks (CNN) [8,5,9], Gaussian mixture models (GMM) [10,11], Support Vector Machine (SVM) [6] and Hidden Markov Models (HMM) [7]. However, in recent years, the Bag-of-Audio-Words (BoAW) or Bag-of-Features technique has been prominent in the audio classification and detection literature [12–14]. The Bag-of-Word technique, which inspired BoAW, has been used more for Natural Language Processing (NLP) tasks, but it has been successfully modified to work with audio signals for the classification of audio events such as walking, door closing and typing on a keyboard.

The features most often used for audio event classification are the Mel Frequency Cepstral Coefficients (MFCC) concatenated with their delta and acceleration. However, MFCCs are mainly used in speech applications and might not be the best way to represent

* Corresponding author.

https://doi.org/10.1016/j.apacoust.2020.107643
0003-682X/

nonverbal signals. As shown in [15], combining MFCC with other features, such as the auditory-inspired amplitude modulation features (AAMF) that were originally developed for whispered speech, increases the sensitivity of the classifier.

A recent technique, Per Channel Energy Normalization (PCEN), used for keyword spotting applications has proven effective in environments with a low signal-to-noise ratio (SNR) [16,17]. In the context of this work, PCEN can be useful for the classification of nonverbal audio events because of the low SNR of some of the more subtle events that are being classified such as blinking and closing the eyes forcefully.

An increasing number of studies in the literature are being conducted on wearables to try to analyze human produced audio signals. In [18], a wearer's voice and biometrics were used to authenticate the identity of the user. In [19], a wearable that detects and classifies human-made social-audio signals such as speech was presented. Others have worked on the detection of eating habits using audio events recorded by the microphone of a smartwatch [20]. Without necessarily using audio signals, intra-aural wearables are also being explored, such as a wearable that can detect eating activities using infrared proximity detectors inside the ear [21] and one detecting tongue movement using an in-ear sensor [22]. Algorithms that can classify the emotional state of a subject using heart rate and galvanic skin response already exist [23], however, their performance could be enhanced by adding nonverbal indicators of emotional states such as the unconscious or involuntary grinding of teeth or excessive salivating.

The classification and detection of human-produced nonverbal audio events could have many real-world uses. A user could wear an earpiece containing an IEM that is connected to a device that processes the audio signals. For example, by detecting and classifying non-voluntary events such as coughing, clearing of the throat or saliva noises, the health of the wearer could be monitored and by detecting voluntary events such as teeth-clicking, tongue-clicking or forceful blinking, users could interface with a computer relatively silently and without using their hands. Another application is to improve in-ear noise dosimetry. In a noisy workplace environment, where a worker's noise dose is monitored, the noise exposure is assessed through in-ear dosimetry [24]. However, wearer-induced disturbances (WID) such as human produced audio events could affect these measurements. Through the detection and classification of these events, both voluntary and involuntary sounds made by the worker could be removed from the in-ear noise dose calculation. In a real-world environment, this battery-powered portable device could detect in real-time these audio events. To achieve this, the algorithm needs to be as efficient and as small as possible to be able to run on a device having low-computational resources.

The primary goal of this paper is to further develop on [15] by using new classification algorithms as well as a new signal processing algorithm. Another goal is to build an audio event detector using the classifier that will be able to run in real-time on a device with low computational power. The BoAW algorithm is tested with different clustering and classification algorithms, namely GMM, K-Means, SVM and Random forest. As for the input vector, fusion techniques at different levels are tested using MFCC, AAMF and PCEN feature extraction techniques.

This paper is organized as follows. Section 2 presents background information, including previous work and the nature of the nonverbal events that are to be classified. The description of the classifiers and signal analysis techniques, as well as the fusion techniques used are presented in Section 3. Section 4 includes the methods in which the real-time detection system is implemented. The results are presented in Section 5 followed by the discussion and conclusions in Sections 6 and 7 respectively.

2. Background

The in-ear biosignal audio database (iBad) introduced in [4] was used to train and test the classifier. The audio signals were captured using a binaural intra-aural device equipped with IEM and outer-ear microphones (OEM) as shown in Fig. 1.

When properly placed, the intra-aural device creates an acoustical seal within the earcanal. This seal is important and useful, as it attenuates external sounds while inducing the occlusion effect. The occlusion effect amplifies nonverbal signals due to the amplification of the vibrations propagated through bone and tissue conduction, making it possible for the IEM to capture them. Therefore, it is assumed that the bandwidth of the signals captured is limited to 2 kHz which is the bandwidth of bone and tissue conduction [3,25]. For this reason, audio samples from the iBad database are downsampled to 8 kHz from the original recording of 48 kHz. This downsampling allows the feature extraction algorithms to perform faster and lowers the dimensionality of the feature vectors by removing unusable data. It is crucial that the classification computations be simplified given that the end goal is for the algorithm to run in real-time on a low-power computer. It was shown in previous work that this downsampling does not negatively affect the accuracy of the classifier [15].

For the purposes of this work, only the audio files containing the nonverbal audio events are used. In total there were 24 files containing on average 4 min of audio content from each participant. The participants were asked to perform approximately 10 s of each of the following events: move their body and head, clear their throat, talk, swallow their saliva, click their tongue, grind their teeth, click their teeth softly and then loudly, close their eyes softly and forcefully, blink their eyes softly and forcefully and yawn. Of these events, 10 were chosen for this study as shown in Table 1.

Audio samples each containing a single iteration of an event were extracted from the audio files post hoc. The length of the audio sample was set to 400 ms to accommodate for the length of some of these events such as talking, clearing the throat and saliva-swallowing noise. The open-source software Audacity [26] was used to perform manual segmentation. Each of the events was segmented, tagged and extracted. The complete list of events is presented in Table 1.

A recent paper on this subject compared multiple classifiers, input vectors, and augmentation with noise and binaural data [15]. For the classifiers, SVM, MLP, and GMM classifiers were tested. For the features, MFCC and AAMF were tested using three different feature vector aggregators. Factory noise was added to the samples to make the classifier more robust to noisy signals. A summary of the results is presented in Table 2 and Table 3. In addition, it was shown that using samples from signals captured in both ears simultaneously served as a type of data augmentation because it increased the sensitivity of the classifier. This is because

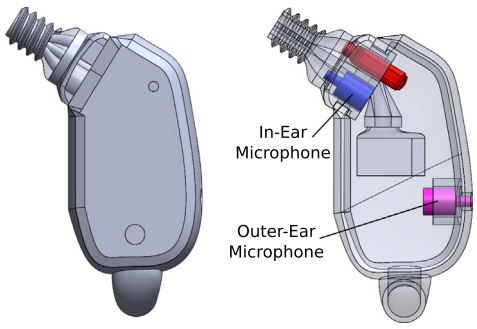

Fig. 1. The occluding intra-aural device featuring an IEM and an OEM, used to collect the data for the database.

Table 1
The total of 400 ms samples for each class.

Event	Number of samples
Clicking of teeth (ct)	560
Tongue clicking (cl)	364
Blinking forcefully (bf)	207
Closing the eyes (ce)	286
Closing the eyes forcefully (cef)	329
Grinding the teeth (gt)	170
Clearing the throat (clt)	163
Saliva noise (sn)	213
Coughing (c)	219
Talking (t)	526

Table 2
Sensitivity of each classifier using the different feature vector aggregators.

	Feature Vector Aggregators		
Classifier	Full	Contextual	Frame
GMM	0.692	0.755	0.739
MLP	0.592	0.498	0.340
SVM	0.633	0.579	0.409

Table 3
Sensitivity of the GMM classifier with MFCC and MFCC+AAMF contextual features using Clean, Noisy and Clean & Noisy datasets for training and testing.

		Test dataset	
Features	Train Dataset	Clean	Noisy
MFCC	Clean	0.754	0.243
MFCC	Clean&Noisy	0.731	0.705
MFCC+AAMF	Clean	0.755	0.329
MFCC+AAMF	Clean & Noisy	0.735	0.728

even though they were recorded simultaneously, the signals captured from each ear differed slightly due to changes in earcanal geometry and the way the intra-aural device was placed in the earcanal. The results showed that the GMM classifier with the contextual input vector and the combination of MFCC and AAMF performed best, both in a noisy and clean environment. The confusion matrix of this classifier is shown in Fig. 2.

Fig. 2 shows that the most confused events were clicking of the teeth softly (*cts*) and clicking of the teeth loudly (*ctl*). Further investigation into this discrepancy revealed that the participants' interpretation of these classes was too broad. The main difference between the two classes was the intensity of the impulses. As seen in Fig. 3, the RMS value of the signal generated by clicking of the teeth varies greatly. When placing an upper threshold at an RMS value of 0.945, 14% of *cts* are misclassified as *ctl* and 29.1% of *ctl* are misclassified as *cts*. These values match those calculated from the confusion matrix found in Fig. 2 by calculating on the samples predicted as either *ctl* or *cts*, which are respectively 14.4% $(0.126/(0.744+0.126))$ and 29.8%. Therefore, to resolve participant variability and inconsistency, clicking of the teeth softly (*cts*) and clicking of the teeth loudly (*ctl*) are grouped into one event simply dubbed clicking of the teeth (*ct*).

Certain nonverbal events used in the current work can be categorized by the nature and source of the captured signal. Some audio events are short impulses produced with the mouth such as clicking of the teeth and tongue, while other events come from eye movements, and thus are of very low amplitude, such as blinking forcefully, closing the eyes and closing the eyes forcefully.

The various audio events shown in this paper have seldom been investigated in the literature. The nature of the events being classified differ greatly from one another. This becomes clear when looking at the spectrograms in Fig. 4. Four different wavelet spectrograms are presented for four randomly selected samples from different events: Clearing of the throat (*clt*), Saliva noise (*sn*), Coughing (*c*) and Clicking of the teeth (*ct*). The spectrogram of the cough signal shows harmonics at the very end of the event as some people tend to voice the end of their cough. The saliva noise event contains low frequency signals with some sporadic high frequency occurrences. Clearing of the throat contains more high frequency content, and some harmonics can be detected as often people voice this event as well. Clicking of the teeth is a clear impulse containing a large range of frequencies at a precise moment in time. This large variation in amplitude and spectral content requires a feature extraction tool that can account for such wide differences while properly representing each signal.

In the recent literature for audio detection and classification, deep learning algorithms are often used, such as in [8,5,9] or [27]

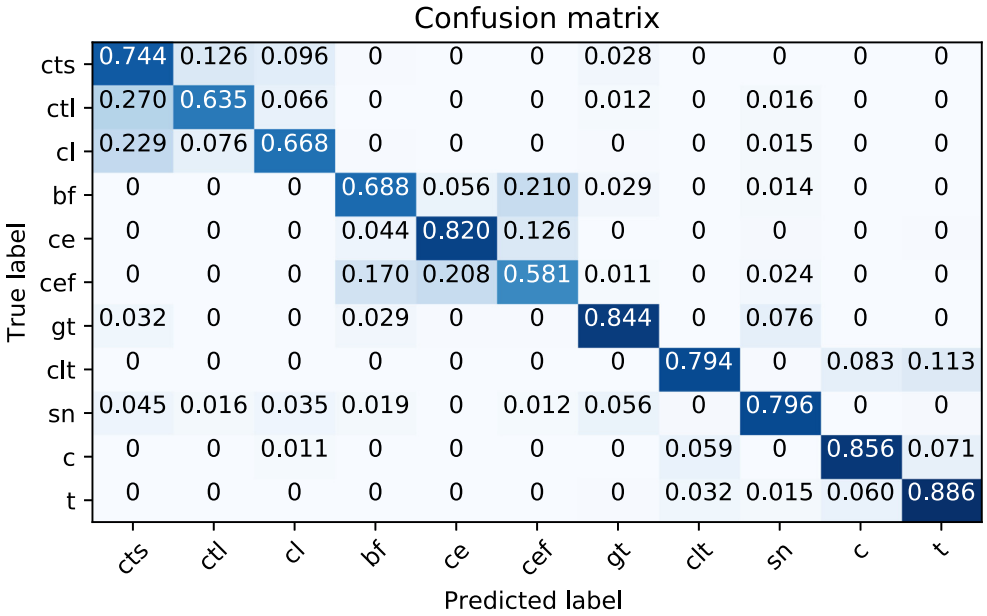

Fig. 2. Confusion matrix of the best result found in previous work on this database [15].

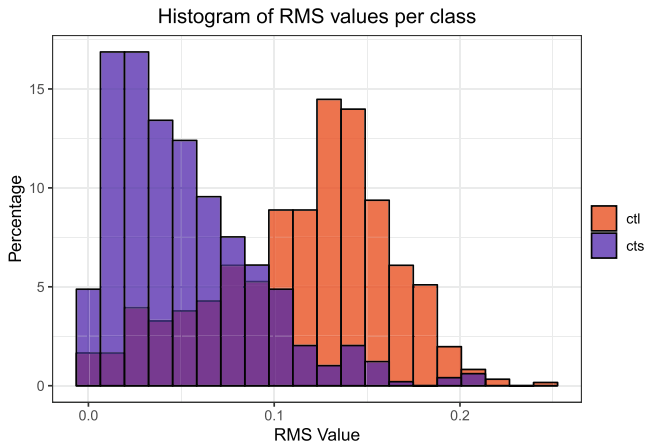

Fig. 3. Comparison of RMS values of the samples from the *cts* and *ctl* classes.

where bird sounds are detected using PCEN features with a CNN algorithm using context-adaptive neural network. However, due to the computation limitation of the low-power computer used in this paper, deep learning algorithms were not chosen. Instead, another lighter algorithm, BoAW was chosen due to its presence in the recent non-deep learning state of the art as seen in [12–14].

3. Methodology

In this section, the methodology used to create an audio event classifier is shown. In Section 3.1, the various signal processing methods used to extract the feature vectors are described. In Section 3.2 the BoAW algorithm is presented as well as the variations that were tested.

3.1. Signal processing methods

Three signal processing methods are used and compared in this paper: MFCC, AAMF and PCEN. MFCC is widely used in the literature in audio event classification while the other two are less common.

PyHTK, a python script, is used to extract the MFCC features as computed by HTK [28]. HTK is a well-known program that extracts MFCC from audio files used mainly for speaker verification tasks [29]. For the 400 ms sample, frames of 50 ms with 50% overlap are calculated given that these parameters resulted in the best performance in [15]. Twelve coefficients plus their delta and acceleration followed by the Zero Crossing Rate (ZCR) of the frame are concatenated to form the MFCC input vector.

AAMFs were introduced for speaker verification using whispered speech [30]. AAMFs analyze the modulation of the signal on small windowed frames of the spectrogram. They give information on the modulation bands for each acoustical frequency for each context of 200 ms as presented in [30]. Since they have a high dimensionality of 216, Principal Component Analysis (PCA) is applied to reduce the dimensionality to 40 as done in [30].

In the literature, PCEN has been used for far-field keyword spotting [17]. One of the main drawbacks of the log compression used in MFCC is that it uses a substantial amount of dynamic range for low amplitude signals. Low signal amplitudes that are close to zero are not very useful for classification because they are usually the least interesting part of the signal. Instead of using log compression, PCEN uses dynamic compression. This compression technique stabilizes signal levels for each channel of the MelSpectrogram and gives more headroom to high amplitude signals. This is interesting for the signals used in this work because of the significant difference in amplitude between some classes like coughing and blinking forcefully as discussed in Section 2. To extract the PCEN, the formula found in [17] is used:

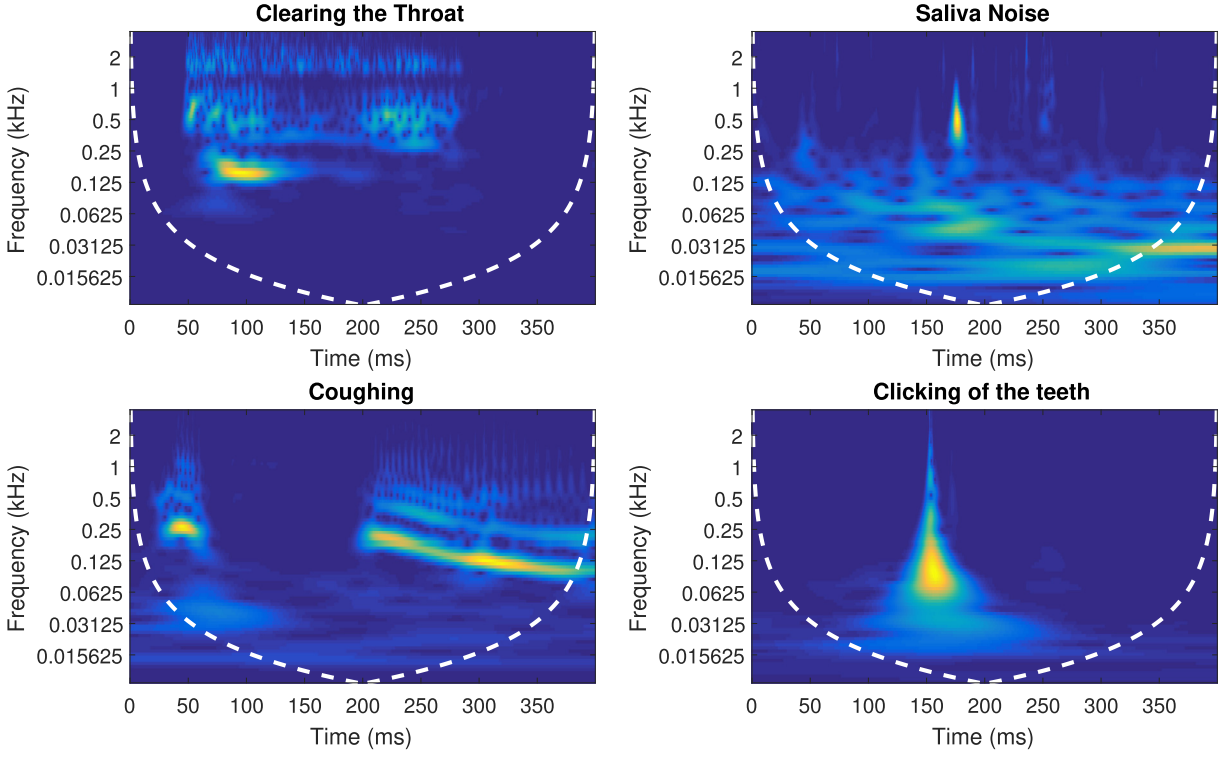

Fig. 4. Wavelet spectrogram of four different audio events.

$$\mathrm{PCEN}(t,f) = \left(\frac{E(t,f)}{(\epsilon + M(t,f))^{\alpha}} + \delta\right)^{r} - \delta^{r} \quad (1)$$

where $E(t,f)$ is the MelSpectrogram of the signal and $M(t,f)$ represents the first order IIR filter seen in [17] in the Laplace domain:

$$M(t,f) = (1-s)M(t-1,f) + sE(t,f) \quad (2)$$

The following parameters were chosen empirically and found to give the best results for the classification problem at hand: $s = 0.025,\ \alpha = 0.75,\ \delta = 2$ and $r = 0.05$.

3.2. Machine learning methods

This paper mainly uses the BoAW algorithm, also called Bag-of-Features or Bag-of-Frames. This technique is derived from the Bag-of-Words technique that is mainly used in text-based applications where first, a dictionary is built containing words that could appear in the text. In audio-based applications, a clustering technique must be used to create the "audio words". Clustering is used on frames of 50 ms with 25 ms overlap to find the "audio words" in the 400 ms sample. The frame size and overlap percentage chosen were found to provide the best results through experimental testing. After creating the dictionary, frames of the audio sample are clustered into the "audio words" contained in the dictionary. A histogram is then built containing the frequency of occurrences of each "audio word" in the sample. The histogram is then used as input for a classification algorithm. Multiple algorithms were tested for both the clustering and classifying task.

As found in Plinge et al. [13], using a super-dictionary can improve the accuracy of the classifier. The super-dictionary is built by creating a dictionary of size N for each of the C classes, and then clustering each class separately instead of creating a general dictionary by disregarding the labels and using the clustering algorithm on the whole dataset.

Another modification made to the classic BoAW method is that instead of fusing the different features at the input level, the fusion occurs at the histogram level. For each feature used, one dictionary is created and one histogram is extracted, and the histograms are subsequently concatenated. For the N features used, N different dictionaries are created. This modification makes it possible to apply different lengths of frames and overlap to the different features versus concatenating the features into one vector, which requires that all feature extraction techniques output the same number of frames to concatenate the frames of each technique. When the fusion is at the histogram level, the number of frames can differ between each feature extraction technique. This also helps by giving a smaller feature vector to the clustering algorithm used to create the dictionary. The structure of this classification technique can be seen in Fig. 5.

For the purposes of this work, the GMM clustering method with a covariance matrix was chosen as it has already shown good performance in a previous work [15] and is comparable to the K-Means method using the Lloyd's algorithm [31], which is a clustering algorithm that is often used for BoAW. These clustering methods are used to find the audio-words of the dictionary in an unsupervised way.

For the classification task, the two algorithms compared were the SVM and the Random Forest. The SVM is programmed using the Scikit-Learn library on Python using the OneVSAll and SVC modules [32]. Different SVM kernels were tested and the best results were obtained with a polynomial kernel of second degree with constant $C = 0.01$. The Random Forest is programmed using the Scikit-Learn library with 500 decision trees in total. These hyper-parameters were chosen empirically using the following validation technique.

3.3. Validation of the classifier

Performance is calculated using a 10-fold cross validation. First, all the classes are divided equally between the folds. Then, all the participants are separated in such a way that the participants are not found in two different folds so that the algorithm does not test on a participant it was trained on.

For this classifier, sensitivity and precision are chosen as performance metrics with their standard deviation. The formula to calculate the sensitivity (SN) and precision (PREC) are as follows:

$$\mathrm{SN} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad (3)$$

$$\mathrm{PREC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad (4)$$

where TP represents the true positive rate, FP the false positive rate and FN the false negative rate. These metrics are chosen because they do not contain the true negative rate (TN), which would not be representative since the number of TN is, on average, ten times greater than the number of TP.

4. Real time implementation of detection system

The proposed BoAW classification algorithm is implemented to detect and classify the non-verbal events in a continuous audio stream in real-time. The detection by classification technique is used to detect the events of interest. This technique continuously classifies the input audio stream to detect the occurrence of an event. To do so, frames of audio data of 100 ms are recorded and concatenated with the previous chunk to create a complete 400 ms sample. Each new frame of 100 ms is concatenated at the end of the sample and the oldest 100 ms is removed to keep the sample to a constant length of 400 ms, creating a first-in first-out buffer. This sample is then classified and compared to the three previous samples. If two samples or more out of the four are of the same class, then they are considered part of this class and a detection occurs. Otherwise, no audio event is detected. Due to the low computational power of the device used, only the channel with the better-fitted earpiece is used for classification since the noise isolation is greater and the occlusion effect is more accentuated, increasing the SNR of the audio event.

The best performances found during the classification study as discussed in Section 5 are used for real-time detection. Therefore, the BoAW detector is used with the MFCC and PCEN features. The two types of features are computed separately even if they both calculate the same MelSpectrgoram to more easily change individual parameters. The fusion of the two features is done at the histogram level and the clustering and classification algorithm used in the BoAW are the GMM and SVM. The GMM contains 15 Gaussians per class and the SVM has a constant of $C = 0.01$.

4.1. Testing of the detection system

The detection algorithm is tested on audio captured from a device similar to the one depicted in Fig. 1 worn by participants in various noise environments. Ten participants were asked to produce the events of clicking of the teeth (ct), clicking of the tongue (cl), blinking forcefully (bf), grinding of the teeth (gt) and clearing the throat (clt) 5 times each and the events closing the eyes (ce), closing the eyes forcefully (cef), saliva noise (sn) and talking (t) for 5 s each. The recording took place in an audiometric booth where the participants were asked to repeat the test 3 times under three different noise conditions. In the first condition, the participants performed the events in a quiet environment. For the second and third conditions, the participants were exposed to 70 dBA of

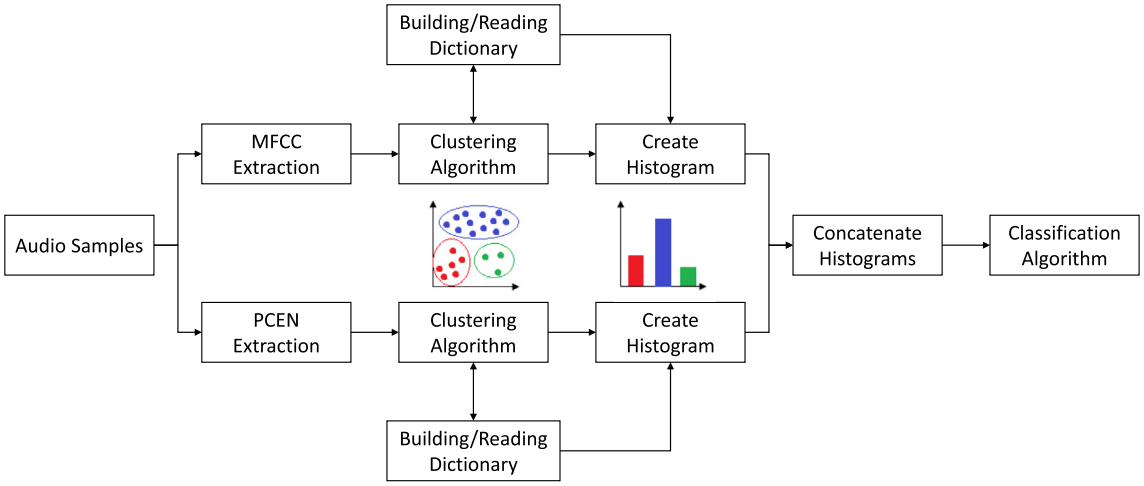

Fig. 5. Diagram of the BoAW algorithm using fusion at the histogram level with MFCC and PCEN features.

factory noise and babble noise consecutively from the NOISEX-92 database [33]. The two noisy environments were chosen to assess the performance of the detector in two of the potential use cases of this technology. The first relates to industrial workers in a noisy setting and the second to the consumer market, such as a cocktail party setting. Before the recording took place, the fit of the in-ear device was verified to ensure that the intra-aural device was well-fitted and creating an acoustical seal attenuating the ambient noise.

For the tests where the recording was made in a noisy environment, it is beneficial to denoise the IEM to reduce the degradation by noise on the captured signals. An adaptive filtering technique developed for intra-aural devices similar to that in Fig. 1 is used, since it has shown good results when denoising IEM speech using the relationship between the OEM and the IEM [34].

It is important to note that the IEM used for the test differed from the one used to record the training data. The IEM used for the test has a different frequency response containing a gain of up to 24 dB at 3 kHz, which the IEM that was used for the training did not have. This caused a decrease in the performance of the detector. To improve its performance, a notch filter at 3 kHz is used on the incoming audio to reduce the effect of the frequency boost caused by the microphone.

The detection algorithm is applied to the audio tracks and the results are manually compared to the events produced in the audio and compiled into a confusion matrix. An unknown class (*?*) is also created to represent the occurrences in which the detector does not find two of the same class in the last 3 classifications.

The detection algorithm is made to run in real-time on a mini-computer that contains 2 GB of RAM and an Atom Z3735F CPU at 1.33 GHz. The microphones are interfaced with the device using the CY8CKIT-059 DSP (Cypress Semiconductor Corp, San Jose, USA).

5. Results

5.1. Classification performance

The results of the comparison between the two fusion techniques is presented in Table 4. The fusion at the histogram level has a sensitivity of 81.3% and a precision of 82.6% compared to 80.2% and 81.3% respectively for the fusion at the input vector level.

The results of using different features when fusing at the histogram level can be seen in Table 5. For this test, four cases were tested: MFCC, MFCC & AAMF, MFCC & PCEN and MFCC & AAMF & PCEN. Fusion without the MFCC features are not tested given that previous attempts already showed poor results. The algorithms used for the clustering and classifying tasks of this test are the GMM and SVM.

Table 4
Comparison of the type of features fusion for the BoAW technique with MFCC and AAMF features.

Technique	Sensitivity	Precision
Fusion at Input Vector Level	80.2 ± 5.1	81.3 ± 5.0
Fusion at Histogram Level	81.3 ± 5.2	82.6 ± 4.8

The best results are achieved using the MFCC & PCEN features with a sensitivity of 81.5% followed by the MFCC & AAMF at 81.3%, the MFCC & AAMF & PCEN at 80.7% and the worst performance is achieved using only the MFCC features at 78.4% sensitivity. The precision follows the same trend as the sensitivity. Their percentages are presented in Table 5.

In Table 6, the results of the comparison between the clustering and classifying algorithm can be seen. Using the K-Means algorithm for the clustering method decreases the accuracy to 78% when using the SVM classifier and 78.8% when using the Random Forest classifier. This can be due to the fact that k-means gives spherical clusters while we previously found in [15] that the GMM with diagonal covariance matrices performed better than spherical covariance matrices. When using the GMM clustering algorithm, both the SVM and Random Forest classifiers achieve a sensitivity of 81.5%.

In Figs. 6 and 7, the confusion matrices of the best results using the SVM and Random Forest classifying algorithms are presented. Both classifiers achieve an average sensitivity of 81.5% but the sensitivity of certain classes such as *cef* and *sn* varied significantly.

Table 5
Comparison of the features used with the BoAW algorithm with the histogram level fusion using the GMM & SVM algorithms.

Features used	Sensitivity	Precision
MFCC	78.4 ± 5.4	80.1 ± 5.1
MFCC AAMF	81.3 ± 5.2	82.6 ± 4.8
MFCC PCEN	81.5 ± 4.4	83.0 ± 4.6
MFCC AAMF PCEN	80.7 ± 4.5	82.0 ± 4.6

Table 6
Comparison of the BOAW clustering and classification method with the MFCC and PCEN features.

Clustering	Classification	Sensitivity	Precision
GMM	SVM	81.5 ± 4.4	83.0 ± 4.6
GMM	Random Forest	81.5 ± 5.5	82.8 ± 4.9
K-Means	SVM	78.0 ± 4.6	79.1 ± 4.7
K-Means	Random Forest	78.8 ± 5.2	79.5 ± 4.6

5.2. Real-time detection

In Table 7 the sensitivity (SN) and precision (PREC) can be seen for every class in the different noise conditions. The average sensitivity over all the classes is 69.9% in quiet. It decreased to 63.5% and 55.6% for the factory and babble environments respectively. As for precision, the average for each class was 78.7%, 72.9% and 64.2% for silent, factory and babble noise respectively.

Fig. 8 shows the confusion matrix containing the results of the performance of the detector combining every noisy environment.

6. Discussion

6.1. Classification performance

When comparing the results, it is clear that the fusion of different features is better when done at the histogram level than at the input level. One of the reasons for the increase in sensitivity and precision might be that the SVM algorithm is better at handling

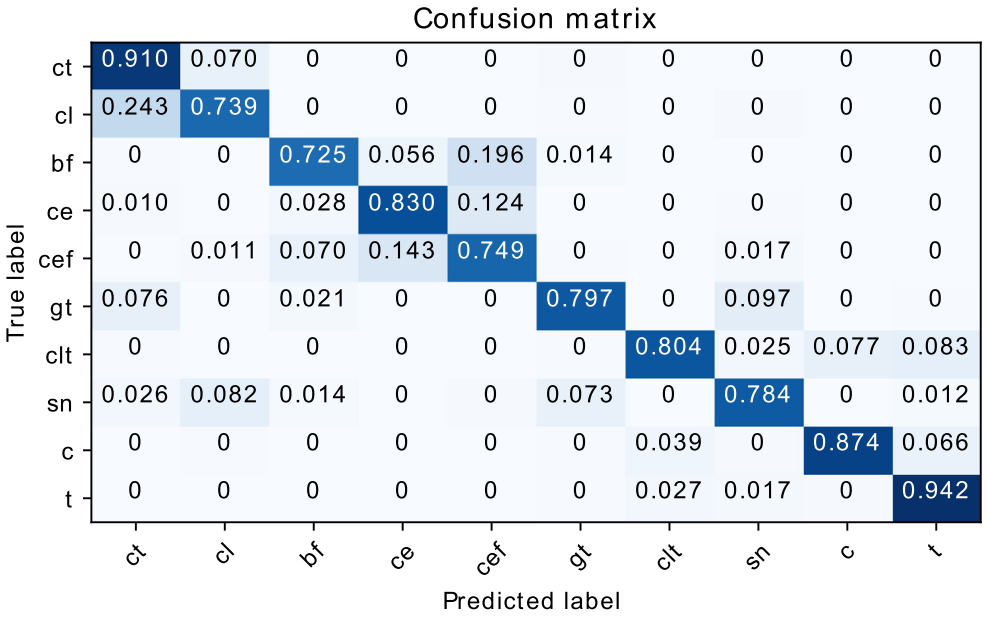

Fig. 6. Confusion matrix of the best results with 81.5% performance across all the classes with the SVM classifying algorithm.

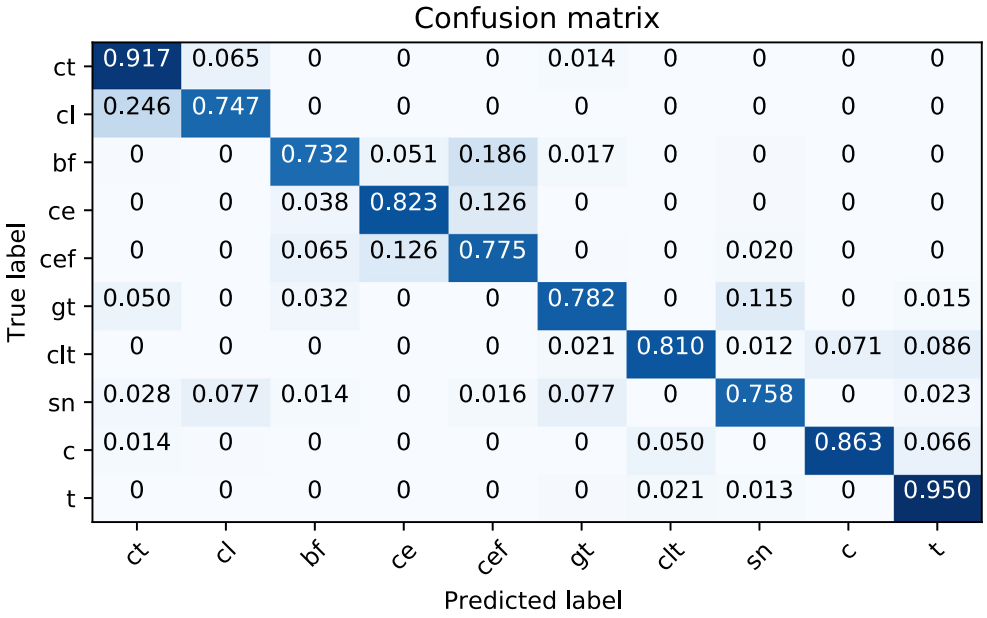

Fig. 7. Confusion matrix of the best results with 81.5% performance across all the classes with the Random Forest classifying algorithm.

Table 7
Sensitivity and precision for each classes under silent, factory and babble environments.

	Quiet		Factory		Babble	
Events	SN	PREC	SN	PREC	SN	PREC
ct	0.767	0.667	0.480	0.774	0.196	0.556
cl	0.833	0.641	0.612	0.556	0.686	0.479
bf	0.386	1	0.160	0.800	0.220	1
ce	0.434	0.657	0.348	0.889	0.240	0.75
cef	0.774	0.414	0.88	0.396	0.660	0.407
gt	0.182	1	0.422	0.679	0.212	0.647
clt	0.764	1	0.700	0.946	0.660	0.440
sn	0.911	0.708	0.900	0.556	0.940	0.528
c	0.926	1	0.843	0.977	0.750	1
t	1	0.786	1	0.718	1	0.613
AVG	0.699	0.787	0.635	0.729	0.556	0.642

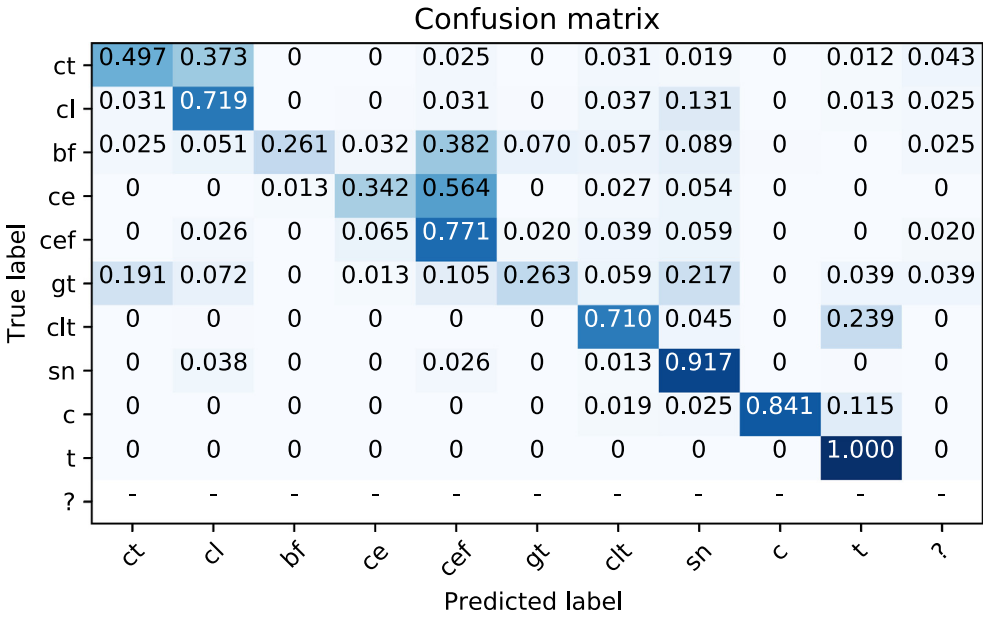

Fig. 8. Confusion matrix of the real-time detection algorithm.

higher dimensionality input than the GMM algorithm. By reducing the dimensionality at the GMM level, by having separate clustering algorithms for each feature, their outputs are concatenated into one larger histogram. The SVM might lose a bit of sensitivity because of the increase in dimensionality, but the increase in sensitivity due to the lower dimensionality at the GMM level is greater and results in an overall greater performance.

When looking at the performance of the various features used, an increase in both sensitivity and precision by adding a new feature to the MFCC can be noted. Due to the increase in sensitivity, it can be said that the information contained in the PCEN and AAMF features are complementary to the MFCC features when trying to classify the nonverbal events of interest. However, when combining all the features, there is a decrease in sensitivity and precision compared to the use of only two features. The information added by the PCEN and AAMF features might be redundant and the increase in performance due to the new information might not be sufficient to counteract the loss of performance caused by the increase in dimensionality.

When comparing the clustering algorithms, the GMM is superior to the K-Means regardless of the classification algorithm used. The Random Forest algorithm performs better than the SVM when used with the K-Means algorithm. When using the GMM clustering technique, the Random Forest algorithm has the same sensitivity as the SVM, 81.5%, but a slightly higher precision. When comparing their confusion matrices, the only classes with a difference in sensitivity of more than 1% are closing the eyes forcefully (*cef*), grinding the teeth (*gt*), saliva noise (*sn*) and coughing (*c*), *cef* being the only class that has better sensitivity with the Random Forest algorithm.

It is hard to compare the performance with the newest literature since the audio signal taken from IEM microphones is very different from what is being captured outside the ear. The signal is limited in frequency to only 4 kHz since the vibrations are propagated through bone and tissue conduction.

6.2. Real-time detection

When in a quiet environment, the performance in terms of sensitivity of the detector compared to the classifier decreases from 81.5% to 69.9%. The difference in performance might be due to the fact that detection with classification is a harder task than classification alone, as seen in [35]. Another problem might be that the IEM in the detection tests used was different than the ones used to build the training database and have slightly different frequency responses despite the compensation filters used. For some classes such as blinking forcefully (*bf*), closing the eyes (*ce*) and grinding the teeth (*gt*), the sensitivity decreased significantly, a difference

of more than 40%. These are the classes whose audio signals had the lowest energy. Therefore, small disturbances due to background noise or due to the different frequency response of the IEM had a greater impact on their detection.

The clicking of the teeth (*ct*) class was often classified as clicking of the tongue (*cl*) since both are impulsive sounds, which is why (*cl*) has a good sensitivity but poorer precision. The same thing happened between the classes blinking forcefully (*bf*), closing the eyes (*ce*) and closing the eyes forcefully (*cef*) with the first two being confused with the latter, which gave *cef* a precision of around 40% in every noise condition.

The considerable decrease in performance between the factory environment and the babble environment might be due to the frequency content of the babble sounds being closer to that of the events than to the factory sounds. The denoising algorithm might also have more difficulty with babble noise since it was built for denoising speech and might not perform as well with speech-like noise. The denoising algorithm itself might have affected the integrity of the audio signal, decreasing the performance of the detector.

It takes 96 ms for the mini computer used to extract the MFCC and PCEN features and classify a 400 ms audio buffer using this BoAW detection algorithm which makes it functional in real-time since we are using a sliding window of 100 ms.

To improve the detector, a bigger database should be built containing audio from various IEMs so the detector does not overfit on the frequency response of the microphone. Other features could also be found to better represent the various nonverbal events present in the database. The performance in terms of speed of the algorithm could be improved by programming the detector in a more efficient language such as C but for this prototype, a Python script was sufficient.

7. Conclusions

Classification of the nonverbal events achieved a performance of 81.5% in sensitivity and 83% in precision using the BoAW algorithm with GMM and SVM techniques. The MFCC and PCEN features were also concatenated at the histogram level, which improved performance compared to fusion at the feature level. The detector built using the classifier had an average sensitivity and precision of 69.9% and 78.7% respectively, over all the classes in a silent environment. In noisy environments it had a sensitivity of 63.4% for factory noise and 55.5% for babble noise. Its precision also decreased to 72.9% for factory noise and 64.2% for babble noise. This nonverbal event detector could be implemented in a hearable device to monitor the users' health, remove wearer-induced disturbances from noise dose calculation and allow the user to interact in a silent and subtle way with the wearable device or a remote computer.

Declaration of Competing Interest

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

References

[1] Sutin AR, Terracciano A, Ferrucci L, Costa PT. Teeth grinding: Is emotional stability related to bruxism?. J Res Personal 2010;44(3):402–5. https://doi.org/10.1016/j.jrp.2010.03.006. URL:https://linkinghub.elsevier.com/retrieve/pii/S0092656610000504.

[2] Cuevas JL, Cook EW, Richter JE, McCutcheon M, Taub E. Spontaneous swallowing rate and emotional state: possible mechanism for stress-related gastrointestinal disorders. Diges Dis Sci 1995;40(2):282–6. https://doi.org/10.1007/BF02065410. URL:http://link.springer.com/10.1007/BF02065410.

[3] Brummund MK, Sgard F, Petit Y, Laville F. Three-dimensional finite element modeling of the human external ear: simulation study of the bone conduction occlusion effect. J Acoust Soc Am 2014;135(3):1433–44. https://doi.org/10.1121/1.4864484. URL:https://asa.scitation.org/doi/abs/10.1121/1.4864484.

[4] Martin A, Voix J. In-ear audio wearable: measurement of heart and breathing rates for health and safety monitoring. IEEE Trans Biomed Eng 2017. https://doi.org/10.1109/TBME.2017.2720463. 1–1, URL:http://ieeexplore.ieee.org/document/7959201/.

[5] Phan H, Hertel L, Maass M, Mertins A. Robust audio event recognition with 1-Max pooling convolutional neural networks. arXiv:1604.06338 [cs]00008 arXiv: 1604.06338 (Apr. 2016). http://arxiv.org/abs/1604.06338.

[6] Rabaoui A, Davy M, Rossignol S, Ellouze N. Using one-class SVMs and wavelets for audio surveillance. IEEE Trans Inf Forensics Secur 2008;3(4):763–75. https://doi.org/10.1109/TIFS.2008.2008216. 00082.

[7] Portelo J, Bugalho M, Trancoso I, Neto J, Abad A, Serralheiro A. Non-speech audio event detection. 2009 IEEE international conference on acoustics, speech and signal processing 2009:1973–6. https://doi.org/10.1109/ICASSP.2009.4959998. 00056.

[8] Salamon J, Bello JP. Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Process Lett 2017;24(3):279–83. https://doi.org/10.1109/LSP.2017.2657381.

[9] McLoughlin I, Zhang H, Xie Z, Song Y, Xiao W, Phan H. Continuous robust sound event classification using time-frequency features and deep learning. PLOS One 2017;12(9):. https://doi.org/10.1371/journal.pone.0182309. URL:http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0182309e0182309.

[10] Geiger JT, Helwani K. Improving event detection for audio surveillance using Gabor filterbank features. In: 2015 23rd European Signal Processing Conference (EUSIPCO); 2015. p. 714–8. doi: 10.1109/EUSIPCO.2015.7362476.

[11] Schröder J, Anemiiller J, Goetze S. Classification of human cough signals using spectro-temporal Gabor filterbank features. In: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2016. p. 6455–6459, 00005. URL:http://ieeexplore.ieee.org/abstract/document/7472920/.

[12] Pancoast S, Akbacak M. Bag-of-Audio-Words Approach for Multimedia Event Classification. INTERSPEECH 2012.

[13] Plinge A, Grzeszick R, Fink GA. A Bag-of-Features approach to acoustic event detection. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). p. 3704–8. https://doi.org/10.1109/ICASSP.2014.6854293.

[14] Schmitt M, Janott C, Pandit V, Qian K, Heiser C, Hemmert W, Schuller B. A Bag-of-Audio-Words approach for snore sounds' excitation localisation. In: Speech Communication; 12. ITG Symposium; 2016. p. 1–5.

[15] Bouserhal RE, Chabot P, Sarria-Paja M, Cardinal P, Voix J. Classification of nonverbal human produced audio events: a pilot study. Interspeech 2018, ISCA 2018:1512–6. https://doi.org/10.21437/Interspeech.2018-2299. URL:http://www.isca-speech.org/archive/Interspeech_2018/abstracts/2299.html.

[16] Arik SO, Kliegl M, Child R, Hestness J, Gibiansky A, Fougner C, et al. Convolutional recurrent neural networks for small-footprint keyword spotting. arXiv:1703.05390 [cs]ArXiv: 1703.05390 (Mar. 2017). http://arxiv.org/abs/1703.05390.

[17] Wang Y, Getreuer P, Hughes T, Lyon RF, Saurous RA. Trainable frontend for robust and far-field keyword spotting. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). p. 5670–4. https://doi.org/10.1109/ICASSP.2017.7953242.

[18] Peng G, Zhou G, Nguyen DT, Qi X, Yang Q, Wang S. Continuous authentication with touch behavioral biometrics and voice on wearable glasses. IEEE Trans Human-Machine Syst 2017;47(3):404–16. https://doi.org/10.1109/THMS.2016.2623562.

[19] Gao B, Woo WL. Wearable audio monitoring: content-based processing methodology and implementation. IEEE Trans Human-Machine Syst 2014;44(2):222–33. https://doi.org/10.1109/THMS.2014.2300698.

[20] Kalantarian H, Sarrafzadeh M. Audio-based detection and evaluation of eating behavior using the smartwatch platform. Comput Biol Med 2015;65:1–9. https://doi.org/10.1016/j.compbiomed.2015.07.013. URL:http://www.sciencedirect.com/science/article/pii/S0010482515002553.

[21] Bedri A, Verlekar A, Thomaz E, Avva V, Starner T. A wearable system for detecting eating activities with proximity sensors in the outer ear. In: Proceedings of the 2015 ACM international symposium on wearable computers, ISWC '15. New York, NY, USA: ACM: 2015. p. 91–2 [event-place: Osaka, Japan]. doi: 10.1145/2802083.2808411. http://doi.acm.org/10.1145/2802083.2808411.

[22] Taniguchi K, Kondo H, Kurosawa M, Nishikawa A. Earable TEMPO: a novel, hands-free input device that uses the movement of the tongue measured with a wearable ear sensor. Sensors 2018;18(3):733. https://doi.org/10.3390/s18030733. URL:https://www.mdpi.com/1424-8220/18/3/733.

[23] Swangnetr M, Kaber DB. Emotional state classification in patient-robot interaction using wavelet analysis and statistics-based feature selection. IEEE Trans Human-Machine Syst 2013;43(1):63–75. https://doi.org/10.1109/TSMCA.2012.2210408.

[24] Bonnet F, Voix J, Nélisse H. Effect of ear canal occlusion on loudness perception. Can Acoust 2016;44(3).

[25] Bouserhal RE, Falk TH, Voix J. On the potential for artificial bandwidth extension of bone and tissue conducted speech: A mutual information study. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2015. p. 5108–12.

[26] Team A. Audacity (r): Free audio editor and recorder [computer program]; 2014.

[27] Lostanlen V, Salamon J, Farnsworth A, Kelling S, Bello J. Robust sound event detection in bioacoustic sensor networks. PLOS One 2019;14:. https://doi.org/10.1371/journal.pone.0214168e0214168.
[28] Korinek D.HTK features in Python. original-date: 2015-12-18T12:00:42Z (9 2019). URL:https://github.com/danijel3/PyHTK.
[29] Ganchev T, Fakotakis N, George K. Comparative evaluation of various MFCC implementations on the speaker verification task. In: Proceedings of the SPECOM 1 (Jan. 2005).
[30] Sarria-Paja M, Falk TH. Fusion of auditory inspired amplitude modulation spectrum and cepstral features for whispered and normal speech speaker verification. Comput Speech Lang 2017;45:437–56.
[31] Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY. An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans Pattern Anal Mach Intell 2002;24(7):881–92. https://doi.org/10.1109/TPAMI.2002.1017616.
[32] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res 2011;12:2825–30.
[33] Varga A, Steeneken HJ. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. Speech Commun 1993;12(3):247–51.
[34] Bouserhal RE, Falk TH, Voix J. In-ear microphone speech quality enhancement via adaptive filtering and artificial bandwidth extension. J Acoust Soc Am 2017;141(3):1321–31.
[35] Phan H, Koch P, Katzberg F, Maass M, Mazur R, McLoughlin I, Mertins A. What makes audio event detection harder than classification? In: 2017 25th European signal processing conference (EUSIPCO); 2017. p. 2739–43, ISSN: 2076-1465. doi: 10.23919/EUSIPCO.2017.8081709.