Monitoring Activities of Daily Living Using Audio Analysis and a raspberryPI: A Use Case on Bathroom Activity Monitoring

Georgios Siantikos[1], Theodoros Giannakopoulos[1], and Stasinos Konstantopoulos[1]

Institute of Informatics and Telecommunications,
NCSR Demokritos,
Athens , Greece
{dickos, tyianak, konstant}@iit.demokritos.gr

Abstract. A framework that utilizes audio information for recognition of activities of daily living (ADLs) in the context of a health monitoring environment is presented in this chapter. We propose integrating a Raspberry PI single-board PC that is used both as an audio acquisition and analysis unit. So Raspberry PI captures audio samples from the attached microphone device and executes a set of real-time feature extraction and classification procedures, in order to provide continuous and online audio event recognition to the end user. Furthermore, a practical workflow is presented, that helps the technicians that setup the device to perform a fast, user-friendly and robust tuning and calibration procedure. As a result, the technician is capable of "training" the device without any need for prior knowledge of machine learning techniques. The proposed system has been evaluated against a particular scenario that is rather important in the context of any healthcare monitoring system for the elder: In particular, we have focused on the "bathroom scenario" according to which, a Raspberry PI device equipped with a single microphone is used to monitor bathroom activity on a 24/7 basis in a privacy-aware manner, since no audio data is stored or transmitted. The presented experimental results prove that the proposed framework can be successfully used for audio event recognition tasks.

Keywords: audio analysis, activities of daily living, health monitoring, remote monitoring, audio sensors, raspberryPI, audio event recognition

1 INTRODUCTION

Although fully autonomous artificial intelligence is actively researched and advanced, the current state of the art (and at the level of maturity required for commodity electronics) has machine learning methods rely on delicate training and configuration sessions in order to adapt to different environments. When embedding machine learning methods in commodity electronics this is typically worked around by uploading the signal and receiving analysis results from remote centralized services. Recent examples include voice-operated personal assistant applications and companion, toy, and “pet” applications.

This model, however, suffers from its obvious privacy implications. These implications are further exacerbated in the telemedicine domain for two reasons: the data collected by the remote service is not only more sensitive, but the users might also not be able to make informed decisions or might not be offered reasonable alternatives. Home monitoring for the elderly is a prime example: the increase in life expectancy and in the need for long-term care creates a pressure to seek alternatives to institutional healthcare for the aged population. Advancements in robotics and automation and in artificial intelligence and intelligent monitoring are explored as a way to prolong independent living at home while providing guarantees of safety and adequate medical monitoring ([Barger et al., 2005], [Hagler et al., 2010], [MOTS et al., 2002]). The users of such solutions, however, might be suffering from mild cognitive impairment or be unable to afford conventional monitoring, which makes ethically questionable any consent they provide to upload and analyse raw content of their activities of daily living (ADL) in order to extract medical monitoring information. Several methods have been used to detect activities of daily living in real home environments, focusing on elderly population ([Vacher et al., 2013], [Vacher et al., 2010], [Costa et al., 2009], [Botia et al., 2012]) and a wide range of modalities.

In this paper, we present an audio analysis system (Section 2) that explores the integration of the audio sensor and the processing unit as Raspberry PI[1] device. Such a unit is able to execute signal processing and machine learning algorithms in order to eliminate the need to provide raw content: the only information that leaves the confines of the integrated unit is an abstract ADL log. Although such information still needs to be managed in full accordance to guidelines pertaining private data, the level of obtrusiveness is greatly reduced by the assurance that no unwarranted analysis or recording can conceivably be done.

Our system is designed to satisfy two key requirements: that the analysis algorithms are computationally efficient so that they can be implemented for the Raspberry PI device; and that they can be tuned and configured for different acoustic environments by technicians without machine learning expertise. In order to evaluate the proposed approach on these requirements, we motivate and present a use case based on bathroom usage (Section 3) and draw conclusions (Section 3).

2 PROPOSED METHOD

2.1 Overall architecture

The main part of the whole system is a microphone-equipped Raspberry PI single-board PC that is used for all data acquisition and processing. Its small-

[1] Please cf. `https://www.raspberrypi.org`

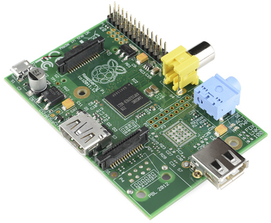


Fig. 1. The Raspberry PI Device

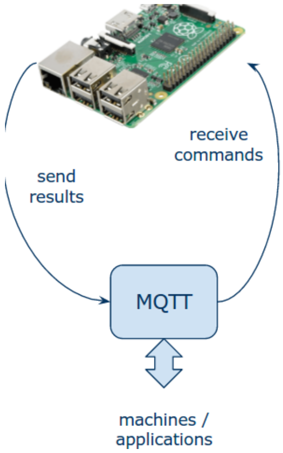
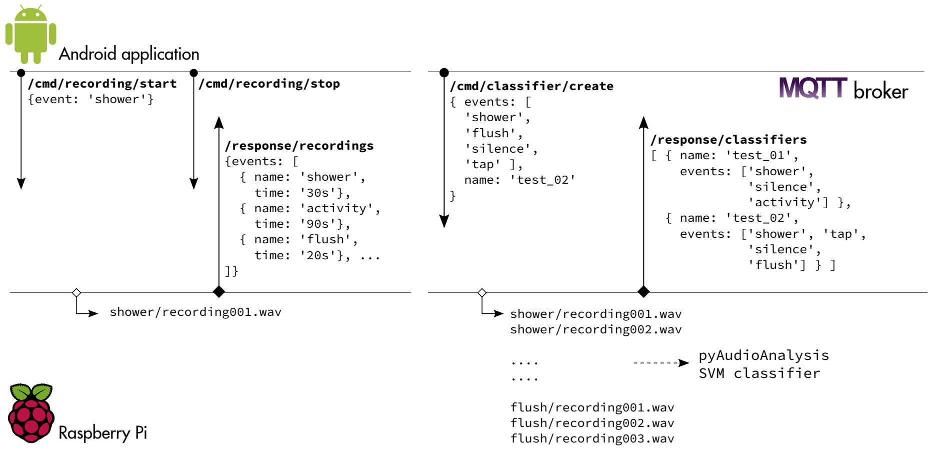


Fig. 2. Mqtt-based messaging communication

form factor, low energy consumption and low overall cost make it ideal for installing it in any room/area we want to monitor and its processing power is enough for running our algorithms in real time. In our experiments we used a Raspberry PI model B with a Wolfson audio card.

Communication to/from the PC is made using the MQTT machine-to-machine communication protocol. MQTT is a lightweight messaging protocol that implements the brokered publish/subscribe pattern, created widely used in IoT applications. Without going into technical details, the main idea is that when connected to a specified MQTT broker, various machines/applications can send messages under a certain topic and others can listen to these when ”subscribed” to these topics. In our use case, it is used both for sending commands to the Raspberry PI (for example to start/stop recording) and for remotely receiving the processing results.

For this purpose, two MQTT clients were implemented: The first is installed in the Raspberry PI and is subscribed to a ”command” topic in order to receive requests for collecting training data, building audio classes models and finally



```
/cmd/recording/start
{event: 'shower'}
```

```
/cmd/recording/stop
```

```
/response/recordings
{events: [
  { name: 'shower',
    time: '30s'},
  { name: 'activity',
    time: '90s'},
  { name: 'flush',
    time: '20s'}, ...
]}
```

```
shower/recording001.wav
```

```
/cmd/classifier/create
{ events: [
  'shower',
  'flush',
  'silence',
  'tap' ],
  name: 'test_02'
}
```

```
/response/classifiers
[ { name: 'test_01',
    events: ['shower',
             'silence',
             'activity'] },
  { name: 'test_02',
    events: ['shower', 'tap',
             'silence',
             'flush'] } ]
```

```
shower/recording001.wav
shower/recording002.wav

....
....

flush/recording001.wav
flush/recording002.wav
flush/recording003.wav
```

Fig. 3. From left to right: User initiates an event recording and the corresponding file is created. When user stops, a response is returned with information about the events recorded so far. When a reasonable amount of data is gathered, an SVM classifier for the desired events can be created using the pyAudioAnalysis library. In this case, the response contains information about the classifiers available for future use. Reproduced from [**?**]

use them for real-time classification. The second one is bundled in an Android application and is used for sending remotely the corresponding commands and listening to the classification results. The system is designed with ease of use in mind and the only set-up needed is connecting the two clients to the same broker. By having a dedicated broker this step can be performed automatically, making the whole system plug-and-play.

2.2 System Calibration

Once setup, the system has to go through a training phase in order to be used for real-life scenarios. This includes recording, feature extraction, manual annotation of the recorded events and classifier tuning / training. Figure 3 shows the proposed calibration procedure. During this phase, the various events are recorded using the Android application as a remote controller of the Raspberry PI device that makes the actual recording and further processing. An audio file is created on user's demand and the user/technician is informed about the categories and durations of already recorded data. He then provides the current recording's label (e.g. ”door bell”).

When a reasonably large amount of data is gathered (typically about 1-2 minutes of recordings for each category), the technician uses the mobile application to trigger the training process (that is also executed on the Raspberry PI device). After this process, the Raspberry PI is ready to monitor and recognize sound in the ”learned” environment. This conceptual sequence of steps for the calibration procedure is also visualized in Figure 4.

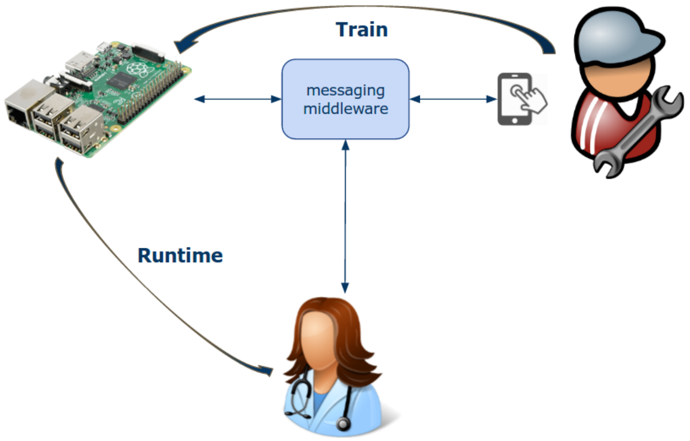


Fig. 4. MQTT-based calibration procedure

2.3 Audio event recognition

Audio features In total, 34 audio features are extracted on a short-term basis. This process results in a sequence of 34-dimensional short-term feature vectors. In addition, the processing of the feature sequence on a mid-term basis is adopted. According to that the audio signal is first divided into mid-term windows (segments). For each segment, the short-term processing stage is carried out and the feature sequence from each mid-term segment, is used for computing feature statistics (e.g. the average value of the ZCR). Therefore, each mid-term segment is represented by a set of statistics. In this Section we provide a brief description of the adopted audio features. For detailed description the reader can refer to the related bibliography [Giannakopoulos and Pikrakis, 2014] [Theodoridis and Koutroumbas, 2008], [Hyoung-Gook et al., 2005]. The time-domain features (features 1 - 3) are directly extracted from the raw signal samples, while the frequency-domain features (features 4-34, apart from the MFCCs) are based on the magnitude of the Discrete Fourier Transform (DFT). The cepstral domain (e.g. used by the MFCCs) results after applying the Inverse DFT on the logarithmic spectrum. The complete list of features is presented in Table 1.

Classification As described in Section 2.3, the feature extraction process leads to a 68-dimensional feature vector for each 1-second audio segment, i.e. 2 statistics x 34 short-term features. Each unknown audio segment of fixed size (1 second in our case) is therefore represented by a 68-D feature vector. Each of

Table 1. Adopted short-term audio features

Index	Name	Description
1	Zero Crossing Rate	Rate of sign-changes of the frame
2	Energy	Sum of squares of the signal values, normalized by frame length
3	Entropy of Energy	Entropy of sub-frames' normalized energies. A measure of abrupt changes
4	Spectral Centroid	Spectrum's center of gravity
5	Spectral Spread	Spectrum's second central moment of the spectrum
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames
7	Spectral Flux	Squared difference between the normalized magnitudes of the spectra of the two successive frames
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients: a cepstral representation with mel-scaled frequency bands
22-33	Chroma Vector	A 12-element representation of the spectral energy in 12 equal-tempered pitch classes of western-type music
34	Chroma Deviation	Standard deviation of the 12 chroma coefficients.

these samples is classified using a Support Vector Machine with probabilistic output. We have selected to use probabilistic SVMs [Platt, 1999] due to their ability to generalize well especially in high dimensional classification problems [Chapelle et al., 1999]. The model is trained using a cross-validation procedure to select the optimal SVM parameter, namely the soft margin parameter C.

Audio analysis implementation Audio feature extraction and classification has been implemented using the pyAudioAnalysis library [Giannakopoulos, 15]. This is an open-source Python library that implements a wide range of audio analysis functionalities and can be used in several applications. Using pyAudioAnalysis one can classify an unknown audio segment to a set of predefined classes, segment an audio recording and classify homogeneous segments, remove silence areas from a speech recording, estimate the emotion of a speech segment, extract audio thumbnails from a music track, etc. In this work, pyAudioAnalysis has been used to extract audio features, to train the classification models and to perform cross validation experimentation in order to extract the respective performance measures. pyAudioAnalysis achieves 2× realtime performance on the Raspberry devices, which validates its usage in the context of the particular setup.

3 BATHROOM USE CASE AND EVALUATION

3.1 Use case and motivation

As discussed in the introduction, the motivating use case for our approach is medical monitoring. Specifically, we base our evaluation setup on allowing elderly people with mild cognitive impairment to maintain an independent life, at their own home, for longer than what is safely possible today.

In order to have a guideline about what information is used by medical doctors to assess such conditions, we use the interRAI Long-Term Care Facilities Assessment System (interRAI LTCF). interRAI LTCF enables comprehensive, standardized evaluation of the needs, strengths, and preferences of persons receiving care. interRAI has been analysed previously in order to identify assessment items, such as mood and ADL logs, that can be automatically recognized and are useful to medical personnel [RADIO Project, 2015]. Among the assessment items listed, we identified those regarding bathroom use as being closest to our concept: these items can be extracted by processing very sensitive content, and being able to provide guarantees about the management and processing of this content would have significant impact on the acceptance of any relevant solution by users.

In this context, we have recorded and manually annotated sounds from bathroom usage. In particular, the following audio classes are trained and evaluated by the proposed methodology:

- Silence - no sound
- Flushing water
- Shower
- Tap water
- Other activities

Note that the selected audio events are location-specific and therefore the adopted calibration workflow can be used during the installation phase, as described in Section 2.2.

3.2 Dataset

In order to train and evaluate the proposed event recognition methodology, we have recorded and manually annotated (using the mobile app described earlier in the paper) an audio dataset. The total duration of the dataset is almost 1 hour. The audio recordings and the respective ground truth is openly available at `https://iit.demokritos.gr/~tyianak/bathroomScenarioEventsNew.zip`

Two different bathroom locations have been used for recording / annotation. This gives us the opportunity to evaluate the performance of the proposed classification method when the respective models have been trained in a different setup. In particular, the complete dataset consists of the following parts:

- A large training dataset of more than 400 audio *segments* to be used for the scenario of the "static" training, according to which, the classifiers are trained beforehand. Note that this subset only consists of audio segments as it is only used for training.
- A subset of 4 audio (continuous) recordings and respective ground-truth annotations. The total duration of this set is 7 minutes. This dataset is used both for training and testing

3.3 Experimental Evaluation

Performance Measures Let CM be the confusion matrix, i.e. a $N_c \times N_c$ matrix (N_c is the total number of audio classes), whose rows and columns refer to the true (ground truth) and predicted class labels of the dataset, respectively. In other words, each element, $CM(i,j)$, stands for the number of samples of class i that were assigned to class j by the adopted classification method. The diagonal of the confusion matrix captures the correct classification decisions ($i = j$). CM is normalized row-wise, in order to discard the information that is related to the size of each class:

$$CM_n(i,j) = \frac{CM(i,j)}{\sum_{n=1}^{N_c} CM(i,n)} \tag{1}$$

Obviously, after the normalization process, the elements of each row sum to unity.

Three useful performance measures are then extracted from the confusion matrix. The first is the overall accuracy, Acc, of the classifier, which is defined as the fraction of samples of the dataset that have been correctly classified:

$$Acc = \frac{\sum_{m=1}^{N_c} CM(m,m)}{\sum_{m=1}^{N_c} \sum_{n=1}^{N_c} CM(m,n)} \tag{2}$$

Apart from the overall accuracy, we have adopted two class-specific measures that describe how well the classification algorithm performs on each class. The first of these measures is the class recall, $Re(i)$, which is defined as the proportion of data with true class label i that were correctly assigned to class i:

$$Re(i) = \frac{CM(i,i)}{\sum_{m=1}^{N_c} CM(i,m)} \tag{3}$$

where $\sum_{m=1}^{N_c} CM(i,m)$ is the total number of samples that are known to belong to class i. In addition, we use the class precision ($Pr(i)$), i.e. the fraction of samples that were correctly classified to class i if we take into account the total number of samples that were classified to that class:

$$Pr(i) = \frac{CM(i,i)}{\sum_{m=1}^{N_c} CM(m,i)} \tag{4}$$

Table 2. *1st experimental setup*: Single-recording training: Row-wise normalized confusion matrix, recall precision and F1 measures. Overall F1 measure: 68.1%

Confusion Matrix (%)

	Predicted				
True ⇓	Shower	Flush	Tap	Silence	Activity
Shower	89.4	1.8	3.0	0.1	5.8
Flush	7.2	70.7	0.4	2.1	19.6
Tap	5.7	4.0	85.8	0.8	3.6
Silence	1.4	4.0	0.0	58.0	36.5
Activity	13.0	11.6	2.6	31.2	41.6

Performance Measurements (%, per class)

Recall:	89.4	70.7	85.8	58.0	41.6
Precision:	76.6	76.8	93.5	62.9	38.8
F1:	82.5	73.6	89.5	60.3	40.2

Finally, the F_1-measure is also computed, which is the harmonic mean of the precision and recall values:

$$F_1(i) = \frac{2Re(i)Pr(i)}{Pr(i) + Re(i)} \tag{5}$$

Results Two categories of experiments have been conducted:

- 1st experimental setup: This is the proposed setup, according to which one or more sequences (i.e. the recordings of the second subset of the dataset described in Section 3.2) are used to train the classifiers, through the presented mobile interface. The rest of the recordings is used for evaluation.
- 2nd experimental setup: This setup is used for comparison. The idea here is to adopt a ”static” model trained from irrelevant data, i.e. segments that have not been recorded and annotated through the mobile interface.

Table 2 shows the (row-wise normalized) confusion matrix and the respective precision, recall and $F1$ measures for each audio class and for the *1st experimental setup*. This is the result of the evaluation process when only one recording is used during the training phase.

These results correspond to the most realistic and less demanding (in terms of calibration-training time). In addition, Table 3 demonstrates the ability of the classifiers to adopt to more data, if they can be available. In particular, 3 shows the same performance measures if a ”leave one out” process is used in the evaluation process, using the described dataset. That is, if three whole recordings are used for each training phase. Results indicate an almost 5% performance boosting. However, using three recordings instead of one means a 300% increase in the calibration time to be carried out by the technicians.

Finally, in Table 4 we present the comparison of the performances of the two experimental setups. The second experimental setup (i.e. the one based on large

Table 3. *1st experimental setup*: Three-recording training: Row-wise normalized confusion matrix, recall precision and F1 measures. Overall F1 measure: 73.8%

Confusion Matrix (%)

	Predicted				
True ⇓	Shower	Flush	Tap	Silence	Activity
Shower	85.6	1.6	2.6	0.2	10.0
Flush	5.7	86.5	0.0	1.5	6.3
Tap	5.2	3.7	85.5	0.7	4.9
Silence	0.1	3.4	0.0	72.5	24.0
Activity	5.8	9.7	1.5	29.0	53.9

Performance Measurements (%, per class)

Recall	85.6	86.5	85.5	72.5	53.9
Precision	83.6	82.5	95.4	69.8	54.4
F1:	84.6	84.5	90.2	71.1	54.2

pre-trained audio datasets) has been evaluated for three different classification approaches, namely: Hidden Markov Models, Convolutional Neural Nets and Support Vector Machines. On the other hand, only SVMs are used for the proposed approach (i.e. the 1st experimental setup), since it would not make sense to use the other two approaches which, by nature, require more training data. The results prove that, indeed, using location-specific audio data for training the models leads to better classification performance, compared to pretrained models, even if much larger datasets have been used.

Experimental Setup	Method	Performance F_1
1st	1 train	68
	N-1 train	74
2nd	HMM	57
	CNN	59
	SVM	58

Table 4. Performance results for both experimental setups. It is obvious that, regardless of the classification method, using a ”static” model that has been trained using external data leads to poor classification performance.

4 CONCLUSIONS

This paper has presented an architectural approach that employs a Raspberry PI device both as an audio acquisition and analysis unit, in the context of a health monitoring system. The overall goal of such system is to detect and recognize Activities of Daily Living (ADLs) in real living environments of elderly people. Both real-time audio feature extraction and classification methods have been implemented and integrated on the device. Apart from the audio analytics procedures implemented on Raspberry PI, we propose a workflow for a fast and easy-to-use calibration procedure. According to this, the implemented classifiers are actually trainned in the particular home's sound conditions, through a series of simple calibration steps, executed through an Android application, handled by the technician that installs the device. In that way, no requirements for knowledge on machine learning are needed.

Experimental evaluation has demonstrated a 70% classification performance even if a single recording (1 to 1.5 minute long) is used in the training process. In addition, experimental evaluation has shown that there is an actual need for a fast and easy retraining procedure. The complete software that was used for the experiments can be found in the project's Git repository [2] under an open-source license.

The proposed system architecture satisfies three vital requirements.

- First, by using computationally efficient algorithms, we manage to cover the needs of data acquisition and processing using only a low-spec'ed Raspberry PI device, while achieving a 2×realtime performance. This validates the system's suitability in the context of a *low-cost* health monitoring setup as it does not require a workstation or a PC (e.g. [Chen et al., 2005]), but a single Raspberry PI that serves both as an acquisition and an analysis module. In particular, the total cost of both the acquisition and analysis modules is less than 100$.
- In addition, the system achieves a satisfactory classification performance, given (a) the low-end hardware used and (b) the lack of demand for big training data. Although our method does not outperform (in terms of overall classification accuracy) other similar methods for ADL recognition in the context of a smart home environment, the significant differences in terms of overall cost and easiness of setup, can make the proposed approach preferable for real house applications. For instance, the approach in [Vuegen et al., 2013] achieves a 85% classification accuracy in a ADL recognition task, however the acquisition scenario requires multiple microphone sensors and therefore much higher cost.
- The setup procedure (i.e. configuration and calibration of the overall system) for different acoustic environments and target events can be performed by technicians without machine learning expertise. The need for a fast and easy procedure for training the audio classifiers, without any prior knowledge for machine learning methods has been met and the effectiveness of the procedure has been validated through experimental results.

Our ongoing and future research work focuses on the following directions:

- Extend the calibration procedure so that it also takes into account a "base dataset", i.e. an initial classification scheme that is tuned in the context of the annotation process and not re-trained from scratch.

[2] https://bitbucket.org/radioprojectanalysis/ict4awe2016

- Use long-term temporal knowledge to smooth the results of the classifier, based on prior knowledge regarding the events.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 643892. Please see `http://www.radio-project.eu` for more details.

References

Barger et al., 2005. Barger, T. S., Brown, D. E., and Alwan, M. (2005). Health-status monitoring through analysis of behavioral patterns. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(1):22–27.

Botia et al., 2012. Botia, J. A., Villa, A., and Palma, J. (2012). Ambient assisted living system for in-home monitoring of healthy independent elders. *Expert Systems with Applications*, 39(9):8136–8148.

Chapelle et al., 1999. Chapelle, O., Haffner, P., and Vapnik, V. N. (1999). Support vector machines for histogram-based image classification. *Neural Networks, IEEE Transactions on*, 10(5):1055–1064.

Chen et al., 2005. Chen, J., Kam, A. H., Zhang, J., Liu, N., and Shue, L. (2005). Bathroom activity monitoring based on sound. In *Pervasive Computing*, pages 47–61. Springer.

Costa et al., 2009. Costa, R., Carneiro, D., Novais, P., Lima, L., Machado, J., Marques, A., and Neves, J. (2009). Ambient assisted living. In *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, pages 86–94. Springer.

Giannakopoulos, 15 . Giannakopoulos, T. (2015–). pyAudioAnalysis: Python audio analysis library: Feature extraction, classification, segmentation and applications. [Online; accessed 2015-04-27].

Giannakopoulos and Pikrakis, 2014. Giannakopoulos, T. and Pikrakis, A. (2014). *Introduction to Audio Analysis: A MATLAB® Approach*. Academic Press.

Hagler et al., 2010. Hagler, S., Austin, D., Hayes, T. L., Kaye, J., and Pavel, M. (2010). Unobtrusive and ubiquitous in-home monitoring: a methodology for continuous assessment of gait velocity in elders. *Biomedical Engineering, IEEE Transactions on*, 57(4):813–820.

Hyoung-Gook et al., 2005. Hyoung-Gook, K., Nicolas, M., and Sikora, T. (2005). *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*. John Wiley & Sons.

Lane et al., 2015. Lane, N. D., Georgiev, P., and Qendro, L. (2015). Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 283–294. ACM.

MOTS et al., 2002. MOTS, T. M., Linda Fraas OTR, M., and Kathleen Stanton MS, R. (2002). Elder acceptance of health monitoring devices in the home. *Care Management Journals*, 3(2):91.

Platt, 1999. Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*. Citeseer.

RADIO Project, 2015. RADIO Project (2015). D2.2: Early detection methods and relevant system requirements. Available at `http://radio-project.eu/deliverables`.

Theodoridis and Koutroumbas, 2008. Theodoridis, S. and Koutroumbas, K. (2008). *Pattern Recognition, Fourth Edition*. Academic Press, Inc.

Vacher et al., 2010. Vacher, M., Portet, F., Fleury, A., and Noury, N. (2010). Challenges in the processing of audio channels for ambient assisted living. In *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, pages 330–337. IEEE.

Vacher et al., 2013. Vacher, M., Portet, F., Fleury, A., and Noury, N. (2013). Development of audio sensing technology for ambient assisted living: Applications and challenges. *Digital Advances in Medicine, E-Health, and Communication Technologies*, page 148.

Vuegen et al., 2013. Vuegen, L., Van Den Broeck, B., Karsmakers, P., Vanrumste, B., et al. (2013). Automatic monitoring of activities of daily living based on real-life acoustic sensor data: a preliminary study. In *Fourth workshop on speech and language processing for assistive technologies (SLPAT): Proceedings*, pages 113–118. Association for Computational Linguistics (ACL).

Wang et al., 2015. Wang, H.-H., Liu, J.-M., You, M., and Li, G.-Z. (2015). Audio signals encoding for cough classification using convolutional neural networks: A comparative study. In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, pages 442–445. IEEE.