

# Evaluation in a Real Environment of a Trainable Cough Monitoring App for Smartphones

Carlos Hoyos-Barceló<sup>1,2</sup>, José Ramón Garmendia-Leiza<sup>3</sup>, María Dolores Aguilar-García<sup>4</sup>, Jesús Monge-Álvarez<sup>1</sup>, Diego Asay Pérez-Alonso<sup>2</sup>, Carlos Alberola-López<sup>2</sup>, Pablo Casaseca-de-la-Higuera<sup>1,2</sup> [0000-0003-1565-0842]

<sup>1</sup>School of Computing, Engineering, and Physical Sciences. University of the West of Scotland, Paisley, PA1 2BE, UK

<sup>2</sup>Laboratorio de Procesado de Imagen. ETSI Telecomunicación. Universidad de Valladolid. Valladolid, 47002, España

<sup>3</sup>SACYL. Gerencia de Asistencia Sanitaria de Palencia. Complejo Asistencial Universitario. Palencia, 34005, España.

<sup>4</sup>SACYL. Gerencia de Asistencia Sanitaria de Palencia. CS Villamuriel de Cerrato. Villamuriel (Palencia), 34190, España.  
Carlos.hoyos@uws.ac.uk

**Abstract.** This paper presents SmartCough, an m-health app for Android smartphones that monitors cough trends in patients with respiratory diseases. The app is designed to be battery-efficient, fast, and robust against noise. It relies on efficiently-implemented machine learning algorithms that have been validated in laboratory conditions. Since these conditions are rarely met in a real situation where the user carries the phone inside their pocket or bag, the app features a self-training module that allows easy adaptation to new environments. In this paper, we have evaluated the app with real patients in an outdoor setting to test the performance in real environments that are hostile to cough detection. Our results show that the average sensitivity obtained in laboratory conditions drops significantly (down to 60%) when the baseline configuration is employed. By activating the built-in self-training module, the median sensitivity raises to 85.87% after a small training step, with a bounded false positive rate. The achieved performance is analogous to the one obtained in laboratory conditions, making the app suitable for use in real life scenarios.

**Keywords:** Cough Detection, m-health, optimization, android.

## 1 Introduction

The emerging technology field of *mobile-Health* (m-health) advocates for the use of smartphones to deliver ubiquitous telemedicine solutions at reduced production costs [1], taking advantage of the huge investments that have already been done in infrastructures to make this technology globally available. Modern smartphones are wearable devices packed with sensors that can run general-purpose programs, and which most users already carry with themselves for most of the day. This makes them the perfect

choice of platform for the development of health monitors for ambulatory tracking of medical conditions even in places with low technology penetration.

We present SmartCough, a cough monitoring app for the Android platform that detects cough in real time and generates trend statistics in the form of short and long-term charts. These charts can be used to reveal symptoms such as a sudden increase in coughing rate, which might be an indication of an impending COPD exacerbation. As most exacerbations are caused by bacterial or viral infection [2], early detection of the symptom would allow the patients to treat the disease before their condition becomes critical, avoiding hospitalisation [3]. The cost of patients with mild to severe cases of COPD were estimated in 2015 to be in order of €96.4 billion per annum in the EU, plus another €283 billion in opportunity costs [4], so a potential reduction in hospitalisation rates and bed-days could turn into huge savings for the national health services. Other uses for the app would be as a support tool to track the response of a patient to medication, and as an aid in the diagnosis of patients with unknown respiratory diseases.

The engineering challenges for a product for real-life deployment are different than those to prove the feasibility of a system in a research lab. For example, the limited CPU power and battery may cause a correct but unoptimised app to be unable to process cough events at the time they occur or will deplete the battery of the device too quickly for practical use. That is why in previous work, we presented an optimised machine learning engine for cough detection that could perform in real-time even on low-end devices [5]. In [6] we proposed additional optimisations to make the app battery-friendly in order to support over three days of continuous operation.

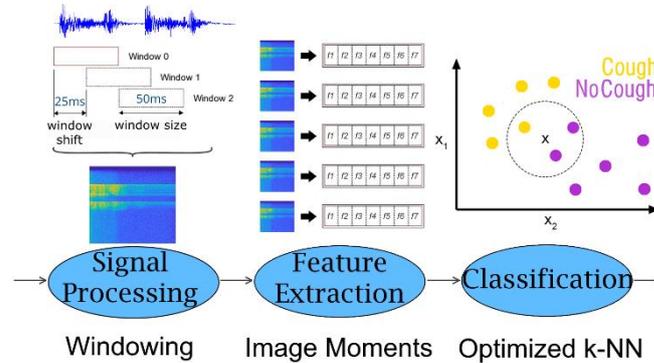
Another problem resides in the fact that scientific research is typically evaluated on unrealistic databases, as the set of samples used for training and testing cannot possibly contain every event that can happen in real life [7], so the systems trained this way rely on optimistic assumptions and may behave poorly outside of that controlled environment (e.g. lab conditions). We previously evaluated the algorithms implemented in Smartcough on a synthetic database [8] and achieved good performance in a variety of noise conditions. After that, we trained the system with a database of real patients recorded in a clinical environment, also with good evaluation results [9]. However, these results may not translate so well on a real-life trial with different hardware, the device carried inside a pocket—which significantly weakens the signal—and unexpected sources of noise such as walking nearby to a construction area. Solving this requires a system able to adapt to data and conditions that it was not trained to handle.

Some studies have shown that adaptively training a system with input from the user significantly improves the performance, with over 30% better recall rates [10] [11]; however the training procedure for most machine learning engines is so computationally expensive [12] that available cough detectors pre-train the system once and leave it as is, forfeiting the use of this technique. We solved this problem by implementing an efficient training mode on our SmartCough app that lets users calibrate the cough detector to better handle their voice. The user only has to press a button and speak to the device, providing input that the app uses to learn in real-time. In this study we measure how the baseline app performs on a real noisy database, and how much our system improves when the accompanying training feature is used.

## 2 Materials and Methods

Our proposed cough detector is composed of three steps as shown in **Fig. 1**.

- *Signal processing*: the audio from the microphone is divided in 50 ms data frames with 25 ms overlap, and then their spectrogram is computed. For efficiency reasons, we apply a filter to detect and discard uneventful chunks of data; this filter is based on a minimum energy threshold and dynamically adjusts itself based on the energy of background noise, to quickly adapt to changing environments.
- *Feature Extraction*: for each spectrogram, we efficiently compute a set of features based on image moments that are used to identify cough windows [6].
- *Classification*: we use an efficient k-NN classifier to detect cough events [5]. The choice of k-NN allows the system to learn by adding new samples to our database.



**Fig. 1.** Diagram of our proposed detector

The testing database is composed of audio recordings of slightly over 24h from 20 patients with different respiratory conditions. We used a Sony Xperia Z2 android device to collect the data, storing the files in 16-bit WAV format at 44100 Hz. The patients were instructed to put the device inside their pockets or purses as they normally would, to get samples from real, noisy environments that are hostile to cough detection.

We used the software tool Praat [13] and created a subtitle file to manually label the coughs, which we used to set the ground-truth. If we could not determine with certainty if an audio segment should be marked as a cough or not, we removed it from the database and from further evaluation. We used this subtitle file and Praat to create separate audio clips for each sound event, then passed them to our app to test the cough detector—trained with its default database—and calculated the Sensitivity and the False Positive Rate (FPR), with the latter measured as False Positives per hour. The default database is composed of patient data from our previous indoor validation [9], augmented with a collection of SFX audio clips from Youtube from common sources of noise.

To test the effect of enabling the self-training module, we configured the app in training mode and then fed it a few seconds of random cough clips from each patient. As this training tends to make the detector overly sensitive to the voice of that patient,

we also trained the app with random sound samples from that patient that do not contain any cough. After that, we passed the rest of audio clips to calculate the new values of Sensitivity and the FPR. We trained the app for each patient individually, always reinstalling the app with default settings so that the different training tests were independent from each other.

### 3 Results and discussion

**Table 1** shows the system performance both in baseline mode, and after enabling the self-training module. The results in baseline mode reveal overall poor detection results and higher variability in sensitivity and FPR. However, both values greatly improve after using the real-time training feature that is built-in with the app. On **Fig 2**, we see that the system goes from an unaffordable sensitivity of 60% up to 85% after using the self-training feature, and how it removes outliers. A paired t-test run over the results before and after calibration revealed high significance in sensitivity improvements ( $p=2.7 \cdot 10^{-7}$ ). The statistical comparison for FPR values was close to significant ( $p=0.073$ ).

	Baseline		Trained	
	Sensitivity (%)	FP/h	Sensitivity (%)	FP/h
<b>Average</b>	61.56%	19	80.63%	16
<b>Median</b>	60.00%	15	85.87%	14.5
<b>STD</b>	18.36%	12	15.26%	5

**Table 1.** Performance of the Smartcough App before and after training.

When comparing SmartCough with other automatic cough monitors, Coughsense [3] achieves higher sensitivity (92%) but worse FPR (17 FP/h), whereas the Leicester Cough Monitor reports 71% sensitivity and 13 FP/h [14]. These proposals however, have not been implemented in Smartphones. Considering only smartphone-based cough-detectors, both ADAM [15] and SymDetector [16] reported sensitivities around 83%. ADAM's FPR was lower (11 FP/h) but it was only evaluated indoors, so signals were in principle less noisy than the ones employed in this work. The authors of both systems also reported battery depletion after less than a day of operation, which indicates that their choice of implementation is too resource-intensive for mobile devices.



**Fig. 2.** Sensitivity and FPR boxplots obtained before and after training.

## 4 Conclusions

After evaluating the Smartcough app over a real-life noisy database we found that even though the baseline performance of the detector is below acceptable levels, we achieve good results both in terms of sensitivity and false positive rate after calibrating the app through its self-training functionality. The achieved performance is comparable to that of other smartphone-based detectors that have only been evaluated over indoor-acquired databases, while surpassing them in terms of efficiency. Also, while it is possible to pre-train the default database with more data to improve the initial results, we have found that user personalisation is a superior approach that enables the system for use in real life scenarios.

## 5 Acknowledgement

The authors would like to acknowledge the clinical support provided by the Pneumology Service staff at Complejo Sanitario Asistencial de Palencia (Spain).

## 6 Conflict of Interest

We declare no conflicts of interest.

## 7 References

- [1] S. Adibi, "Mobile Health: A Technology Road Map," Springer, February 19, 2015, p. 593.

- [2] Balter MS, et al., "Canadian guidelines for the management of acute exacerbations of chronic bronchitis," *Can Respir*, vol. 10, no. 5, pp. 248-258, Jul-Aug 2003.
- [3] E. C. Larson, T. Lee, S. Liu, M. Rosenfeld and S. N. Patel, "Accurate and Privacy Preserving Cough Sensing using a Low-Cost Microphone," in *UbiComp '11*, pp. 375-384, Beijing, 2011.
- [4] European Respiratory Society (ERS), "The economic burden of lung disease," in *European Lung White Book*, <http://www.erswhitebook.org/chapters/the-economic-burden-of-lung-disease/>, 2015, Chapter 2. pp. 16-27.
- [5] C. Hoyos-Barceló, J. Monge-Álvarez, M. Zeeshan Shakir, J. Alcaraz-Calero and P. Casaseca-de-la-Higuera, "Efficient k-NN implementation for real-time detection of cough events in Smartphones," *Journal of Biomedical and Health Informatics (JBHI)*, vol. 22, no. 5, pp. 1662-1671, 2018.
- [6] C. Hoyos-Barceló, J. Monge-Álvarez, Z. Pervez, Luis M. San-José-Revuelta and P. Casaseca-de-la-Higuera, "Efficient Computation of Image Moments for Robust Cough Detection using Smartphones," *Computers in Biology and Medicine (CBM)*, vol. 100, Sept., pp. 176-185, 2018.
- [7] L. T. Nguyen, M. Zeng, P. Tague and J. Zhang, "I did not smoke 100 cigarettes today!: avoiding false positives in real-world activity recognition," in *UbiComp '15 Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Osaka, 2015.
- [8] J. Monge-Álvarez, C. Hoyos-Barceló, K. Dahal and P. Casaseca-de-la-Higuera, "Audio-cough event detection based on moment theory," *Applied Acoustics*, vol. 135, pp. 124-135, 2018.
- [9] J. Monge-Álvarez, C. Hoyos-Barceló, P. Lesso and P. Casaseca-de-la-Higuera, "Robust detection of audio-cough events using local Hu moments," *Journal of Biomedical and Health Informatics*, vol. 23, pp. 184-196, 2019.
- [10] K. Yatani and K. N. Truon, "BodyScope: A wearable acoustic sensor for Activity Recognition," in *UbiComp '12*, Pittsburgh, pp. 341-350, 2012.
- [11] V. Kõnönen, J. Mäntyjärvi, H. Similä, J. Pärkkä and M. Ermes, "Automatic feature selection for context recognition in mobile devices," *Pervasive and Mobile Computing*, vol. 6, pp. 181-197, 2010.
- [12] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez and S. Pastrana, "Power-aware anomaly detection in smartphones: An analysis of on-platform versus externalized operation," *Pervasive and Mobile Computing*, vol. 18, pp. 137-151, 2015.
- [13] P. Boersma and D. Weenink, "Praat: doing phonetics by Computer," Phonetic Sciences, University of Amsterdam, [Online]. Available: <http://www.fon.hum.uva.nl/praat/>.
- [14] S. Matos, S. S. Birring, I. D. Pavord and D. H. Evans, "Detection of Cough Signals in Continuous Audio Recordings Using Hidden Markov Models," *IEEE Transactions of Biomedical Engineering*, vol. LIII, no. 6, pp. 1078-1083, 2006.
- [15] M. Sterling, H. Rhee and M. Bocko, "Automated Cough Assessment on a Mobile Platform," *Journal of Medical Engineering*, vol. 2014, Article ID 951621, 9 pages, 2014.
- [16] X. Sun, Z. Lu, W. Hu and G. Cao, "SymDetector: Detecting Sound-Related Respiratory Symptoms Using Smartphones," in *UbiComp '15*, Osaka, pp. 97-108, 2015.