# Towards Device-Agnostic Mobile Cough Detection with Convolutional Neural Networks

Filipe Barata*, Kevin Kipfer†, Maurice Weber‡, Peter Tinschert§, Elgar Fleisch*§, Tobias Kowatsch§
*Center for Digital Health Interventions, Department of Management, Technology and Economics, ETH Zurich
†Department of Computer Science, ETH Zurich
‡Department of Mathematics, ETH Zurich
Zürich, Switzerland
Email: fbarata@ethz.ch, kipfer.kevin.j@gmail.com, webermau@student.ethz.ch, efleisch@ethz.ch
§Center for Digital Health Interventions, Institute of Technology Management, University of St. Gallen
St. Gallen, Switzerland
Email: peter.tinschert@unisg.ch, tobias.kowatsch@unisg.ch

***Abstract*—Ubiquitous mobile devices have the potential to reduce the financial burden of healthcare systems by providing scalable and cost-efficient health monitoring applications. Coughing is a symptom associated with prevalent pulmonary diseases, and bears great potential for being exploited by monitoring applications. Prior research has shown the feasibility of cough detection by smartphone-based audio recordings, but it is still open as to whether current detection models generalize well to a variety of mobile devices to ensure scalability. We first conducted a lab study with 43 subjects and recorded 6737 cough samples and 8854 control sounds by 5 different recording devices. We then reimplemented two approaches from prior work and investigated their performance in two different scenarios across devices. We propose an efficient convolutional neural network architecture and an ensemble based classifier to reduce the cross-device discrepancy. Our approach produced mean accuracies in the range [85.9%, 90.9%], showing consistency across devices (SD = [1.5%, 2.7%]) and outperforming prior learning algorithms. Thus, our proposal is a step towards cost-efficient, ubiquitous, scalable and device-agnostic cough detection.**

***Index Terms*—Cough monitoring; mobile sensing; machine learning; convolutional neural network**

## I. Introduction

Coughing is the most common complaint as to why individuals seek medical advice [1], [2]. It is mostly known to be a prominent symptom of the common cold, but it is also associated with many respiratory diseases including chronic obstructive pulmonary disease (COPD), asthma, tuberculosis, gastro-oesophageal reflux, chronic bronchitis and cystic fibrosis. Cough is defined by a three-phase expulsive motor act, starting with an inspiratory effort, followed by a forced expiratory effort against a closed glottis and ending by an opening of the glottis and rapid expiratory airflow [1]. Its characteristic sounds, however, are generated by rapid changes in airflow caused by the contractions of muscles in the chest wall, abdomen, diaphragm and larynx [1].

Finding an objective measure of cough frequency in patients is an undertaking that began in the 1950s and continues today. This is motivated by the eminent need to assess the severity of cough and the effectiveness of treatment in patients with respiratory conditions, and the often found unreliable assessment of cough based on patient self-reports [3]. To avoid inaccuracies of patient self-reports and to reduce the patient's burden of data collection, automatic cough detection systems have been proposed to count coughs from audio recordings. Cough monitoring, or more specifically, cough detection from audio recordings has been thoroughly investigated in literature as outlined in the related work section. No cough monitor, however, is currently the gold standard [1].

The challenge of designing an automated cough detection system is manifold. Particularly challenging is the rare occurrence of cough, which demands high specificity from the cough detection system to avoid false alarms from other similar and more frequently occurring respiratory sounds such as throat-clearing, laughter or speech. Moreover, a sensitivity beyond 90% has been identified to satisfy expert requirements [4]. Another challenge is that a cough detection system must be capable of operating continuously over a prolonged time frame in order to capture a representative cough frequency with respect to a respiratory disorder, i.e. multiple hours, a night or even a whole day. Therefore, either the cough detection system should be connected to the power supply or the battery life of the system should be sufficiently long. Since most cough detection systems are envisioned to be operating in the field under any circumstances, they operate on limited battery life and therefore energy efficiency is a critical design criterion [5]. Finally, cough is associated with the most prevalent chronic diseases defining a need for a scalable and cost-efficient cough detection system. In asthma, a chronic disorder involving the airways and the lungs, coughing is considered an important symptom because it predicts disease severity [6], indicates worse prognosis [7], and in particular overnight it has been identified as a potential biomarker for asthma control [8]. Asthma is estimated to affect 334 million people worldwide with an estimated increase of asthma patients by 100 million in 2025 [9]. It has a significant economic impact due to its prevalence among younger working age groups and loss of productivity. The current annual costs of healthcare and productivity loss are estimated to reach EUR 33.9 billion in the EU [10].

978-1-5386-9138-0/19/$31.00 ©2019 IEEE

Ubiquitous available ICT, such as smartphones or wearable devices can be used to unobtrusively monitor patients' health condition as these devices are omnipresent and almost always carried by individuals [11]. Moreover, low-cost smartphones have sensors able to achieve clinical accuracies, e.g. the prediction of gait speed in 6-min walk tests, a standard assessment for COPD or congestive heart failure [12]. Also, the feasibility of cough detection by smartphone-based audio recordings has been studied in [5], [13]. However, it is still open as to whether these detection models generalize well to a variety of mobile devices to ensure population-wide scalability. In addition, the varying characteristics of different devices has been identified in literature as a limiting factor in audio based machine learning applications [14], [15]. Ultimately, device-agnostic smartphone based cough detection would not only guarantee scalability and cost-efficiency, but would also provide a reproducible measurement of one of the notorious symptoms of human disease.

Against this background, we address the following research question in this paper: *To which extent can machine learning based approaches enable device-agnostic mobile cough detection?*

As a primary contribution to the existing body of research, this work investigates the feasibility of employing machine learning to predict coughs from audio recordings and evaluates the predictive power thereof across different devices. It is a first step towards a scalable cost-efficient case-specific cough detection system on the smartphone. We first implement two methods for smartphone cough detection known from literature [13] and [5]. Further, we propose a convolutional neural network (CNN) architecture and exploit an ensemble implementation to further boost the performance in a device-agnostic preserving manner.

The remainder of this paper is structured as follows. The following section discusses related work on cough detection approaches. To answer the research question and to compare the efficiency of our approach with prior work, we describe in Section 3 a lab experiment in which we have created a corresponding dataset with five devices and 43 participants. We further describe the methodology, where we reimplement two approaches from prior work and propose an efficient CNN architecture and its ensemble. In Section 4 we present our results. Section 5 discusses our findings. Finally, Section 6 summarizes our work.

## II. Related Work

Cough detection or more particularly the monitoring of coughing has been under research and development since the 1950s [25]. Table I gives an overview of related work about automated cough detection algorithms.

Most of the approaches listed rely on feature extraction. In conventional machine learning, those features are typically handcrafted with the goal of reducing the data size, which is required to describe the prediction problem at hand, in our case predicting cough from sensor data. Subsequently, the computed features are used as input for the algorithm to train and enable the learning of a pattern represented in the data, such as coughs. This has distinct advantages. First, a condensed representation of the data requires less memory, less computation power and ultimately reduce the risk of overfitting the model to the training samples resulting in poor generalization to new unseen samples. However, this comes all with the risk of losing valuable information and limiting the representation power of the learning algorithm.

By contrast, deep learning, a specific class of machine learning shifts the complexity of handcrafted feature engineering towards model optimization, since convergence is not given. This paradigm shift helps taking advantage of additional available computation power and data [26]. Deep learning architectures have come to bolster the state-of-the-art of various domains, such as speech recognition [27] and image classification [28]. They are based on artificial neural networks [29] and can be characterized as a model consisting of a cascade of multiple layers of nonlinear information processing [30]. Even though some of the most prominent cough detection algorithms, which have been employed to conventional condenser microphone signals [17], [21]–[23] exploit neural network architectures, they heavily focus on the engineering of features and still have rather shallow architectures consisting of 2-4 hidden layers. This may be explained by the difficulty of collecting large amounts of data from real subjects resulting in fewer data samples, which limited scaling to a deeper network. Other approaches have focused on computationally efficient solutions providing cough detection algorithms for the smartphone [5], [13]. They all employed conventional machine learning algorithms, such as random forests and $k$-nearest neighbors. Even if the provided sensitivity and specificity values are comparable or even exceed other approaches their efforts were not investigated across devices. There is only one approach, which used two smartphones for the recordings [5]. The devices were placed in different locations, i.e. on the table or inside the bag/pocket, and evaluated separately with respect to their position. The large corpus of research in this field, as well as the repeatedly reported high-performance values over different datasets, establish the proof of concept for cough detection. However, it is still open as to whether these detection models generalize well to a variety of mobile devices to ensure population-wide scalability.

## III. Method

### A. Data Collection

We first created a labeled audio corpus for training and evaluation of our models. The aim of this audio corpus was to record various voluntary cough sounds, but also some other sounds identified in the literature as typical examples of sounds that are confused with cough [24], [31]. Voluntary coughs have been used in previous literature to show the feasibility of cough monitors [18], [19], [24], [32] or objectively evaluate the performance of several sensors for cough detection [31]. The lab setting followed a similar set-up as in [31], where a person is sitting (in a quiet environment) at a table on which the recording device is placed. Consistent with these approaches

| Author | Recording Device | Algorithm | Subjects (Coughs) | Cough Type | Sensitivity | Specificity |
|---|---|---|---|---|---|---|
| Coyle et al. 2005 (LifeShirt) [16] | Contact Mic. + Sensor Array | *no details available* | 8 (3645) | Reflex | 78.1% | 99.6% |
| Barry et al. 2006 (HACC) [17] | Lapel Mic. | PNN | 15 (2000) | Reflex | 80% | 96% |
| Birring et al. 2008 (LCM) [3] | Lapel Mic. | HMM | 15 (1836) | Reflex | 91% | 99% |
| Vizel et al. 2010 (PulmoTrack) [18] | Piezoelectric Belt + Lapel & 2 Contact Mic. | *no details available* | 12 (*no details available*) | Voluntary | 96% | 94% |
| Drugman et al. 2011 [19] | Contact Mic. | ANN | 22 (2304) | Voluntary | 94.7% | 95% |
| Larson et al. 2011 [13] | Smartphone Built-In Mic. | RF | 17 (2558) | Reflex | 92% | 99.5% |
| McGuiness et al. 2012 (VitaloJak) [20] | Piezo Sensor | Median Frequency Threshold | 10 (*no details available*) | Reflex | 97.5% | 97.7% |
| Swarnkar et al. 2013 [21] | Matched Pair Low-Noise Mic. | NN | 3 (342) | Reflex | 93.44% | 94.52% |
| Liu et al. 2014 [22] | Lapel Mic. | GMM-HMM & GMM-RBM | 20 ($>$ 2549) | Reflex | 90.1% | 88.6% |
| Amrulloh et al. 2015 [23] | Low-Noise Mic. | TDNN | 24 (2090) | Reflex | 93% | 98% |
| Amoh et al. 2016 [24] | Wearable Sensor / Contact Mic. | CNN & RNN | 14 (627) | Voluntary | 87.7% | 92.7% |
| Monge-Alvarez et al. 2018 [5] | (2x) Smartphone Built-In Mic. | $k$-NN | 13 (*no details available*) | Reflex | 88.51% | 99.7% |

TABLE I: Overview over prior literature on automated cough detection algorithms. Note: Microphone Mic., Probabilistic Neural Network PNN, Hidden Markov Models HMM, Artificial Neural Network ANN, Random Forest RF, Neural Network NN, Gaussian Mixture Models GMM, Restricted Boltzmann Machine RBM, Time Delay Neural Network TDNN, Convolutional Neural Network CNN, Recurrent Neural Network RNN and $k$-Nearest Neighbor $k$-NN.

we conducted a lab study to investigate the feasibility of generalizing cough detection to a variety of mobile devices. In addition, two recording distances were used, to mitigate the influence of a possible distance bias. The devices were placed on the table, once directly in front of the chair, and once shifted to the left by 1m, with a chair-table distance of 15cm. The data collected in a lab setting followed the study protocol outlined in [33]. Participants were instructed to voluntarily cough 20 times at two different distances while being audio recorded by five different devices: HTC M8, Samsung S6, Apple iPhone 4, Google Nexus 7 tablet and one studio microphone Røde NT1000). Also, different control sounds such as laughter, throat clearing, forced expiration, and speech were recorded in the same manner. The mobile devices were set up to record using the standard audio recording application to increase external validity of the audio recordings. They were also connected to an audio interface (Focusrite Scarlett 18i20), which again was connected to a computer to control the recording by means of Audacity, an audio software for multi-track recording and editing as can be seen Figure 1. The studio microphone was connected to the audio interface too, but the device-specific driver defined its settings. The devices were turned on to record all the sounds at the preset sampling frequency and bit rate, namely 44.1 kHz with 16 bits, respectively.

We employed second-hand devices, which range among the most popular models in Europe in 2016 covering the leading mobile operating systems Android and iOS. Also, the devices differ due to different microphone specifications, processing hardware, usage and service life. To investigate the recording quality of each device, we compared the spectra of the measured signals of the different devices to the spectrum of the signal measured by the studio microphone. In detail, this means we computed the power spectral density (PSD) [34] and used the PSD of the studio microphone as the reference. We computed the mean squared error (MSE) of the PSD of the reference measurement and the smartphone/tablet measurement over all measured coughs. We subsequently normalized the MSEs of each cough with respect to the coughing participant. The average including the standard deviation of the MSE for each measuring device can be seen in Figure 2. Higher means of the MSE for HTC M8 and Google Nexus 7 recordings indicate a higher deviance from the recordings of the studio microphone Røde NT1000 and therefore suggesting a lower quality in comparison to the recordings of Apple iPhone 4 and Samsung S6. The audio data were human annotated, labeling all acoustic events as one of the five categories cough, speech, throat clearing, laughter and forced expiration. All acoustic events were annotated by a single person. Calculating inter-rater reliability was considered unnecessary due to the highly standardized lab setting with virtually no interfering noises.

### *B. Evaluation*

In order to answer our research question we investigated two scenarios. First, we trained and tested our models on all devices, but tested on unseen subjects in our database. This emulates the scenario of deploying a mobile cough detection model on a known device. The evaluation is then conducted on each device separately, since our recordings consist of audio events recorded over 5 devices at the same time. In the second scenario we trained on samples of a subset of 4 devices, but tested on the remaining unseen device with unseen

Fig. 1: Data collection setup.

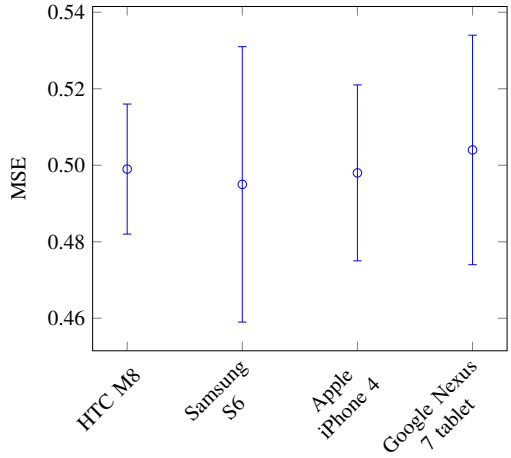


Fig. 2: PSD comparison of the different recording devices to the studio microphone. The mean and the standard deviation refer to the mean squared error (MSE) computed over the different cough recordings made by the smartphones/tablet and the recordings made by the studio microphone.

participants. Analogously, as in the first scenario we then evaluated it with respect to all devices individually. This scenario emulates the case of deploying a model on a device which is unknown to the model builder. In deep learning the split into two or three sets for training, validation and evaluation of the architectures is favored over other approaches, such as cross-validation, due to the long training phases of the models. This comes at the risk of overfitting to the specific dataset and the lack of generalizability to unseen samples. To mitigate that effect we split our sets into disjunct training, validation and test datasets containing different participants. The models were then trained iteratively on the training set, but tuned and selected based on the prediction results on the validation set. After the model was chosen, it was again trained on the merged training and validation set and its final prediction results were then reported on the unseen test set. The approaches proposed in this work were implemented using Tensorflow, an open-source software library for training neural networks in Python [35]. Also, TensorFlow-Slim [36], a lightweight library on top of Tensorflow for defining and training complex

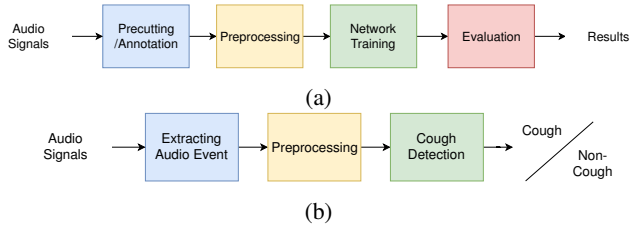


Fig. 3: (a) Machine learning pipeline: Describes the learning of a model from audio signals. (b) Cough detection pipeline: Describes the inference of cough from audio recordings.

models, was used. All the models were trained on the cluster infrastructure of ETH Zurich. The approaches of prior work were implemented in Python using the machine learning library Scikit-learn [37]. Finally, benchmarking of memory usage, floating-point operations and number of parameters was accomplished by using the Model Benchmark Tool provided by TensorFlow.

### C. Classifiers for Mobile Cough Detection

To investigate the research question, we use previous approaches which have been developed for mobile cough detection as baselines. We reimplemented the most cited [13] and the most recent approach [5] with regard to cough detection using a smartphone. These reimplementations are best efforts and were subject to further optimization. Also, we introduce an architecture inspired by recent research [38], where we developed an efficient CNN-based model. We further introduce a leave-one-device-out ensemble to foster device-agnostic classification performance. Detailed descriptions of preprocessing, network training, and implementation are provided in the following subsections. An overview of the corresponding blocks of the machine learning pipeline is shown in Figure 3a. In this work we focus on building models for cough detection. These models can be deployed in an automatic cough detection pipeline as shown in Figure 3b. Moreover, to make cough detection more efficient, most approaches introduce a step before the detection, where the stream of acoustic data is segmented into time frames and subsequently analyzed to determine whether the time frame at hand may contain any significant data. This decision is made by computing rather simple features such as the energy of the time frame and the comparison to a predefined threshold. In this manner time frames containing silence are discarded. The remaining time frames are processed further to serve as input for the prediction model and ultimately result in a cough/non-cough classification. We omit the description of such a step of extracting a relevant audio event since it has been sufficiently addressed in previous work [13], [24]. Thus, the preprocessing step in both pipelines in Figure 3a and 3b can be used interchangeably.

### D. Approaches from Prior Work

*1) Random Forest with PCA for Feature Selection:* For the first method, we implemented the approach outlined in [13].

In the preprocessing step, we first extracted a window of size 320 ms from the original raw audio signal and then employed min-max normalization in order to scale it into the range [-1,1]. We then generated a feature representation of the audio data and subsequently trained a random forest classifier on these features. Specifically, the model is generated by first running PCA with ten components on the vectorized magnitude spectrograms yielding the first ten features for the model. As in [13], we computed the residual error between the reconstructed and original vectorized spectrograms, and additionally the mean decibel energy for both the entire reconstructed spectrograms and for the FFT coefficients above 11kHz, and below 11kHz, yielding four more features. After the cough model is generated, we train a random forest classifier with 500 trees. We further optimized on the maximal number of levels in each decision tree and the maximal number of features resulting in a maximal depth of 14 and 10 features, respectively. The parameters, maximal number of levels (14), maximal number of features (10), number of trees (500), window size (320 ms) and the decision point in frequency (11 kHz) were result of optimization and differ from the approach in [13].

*2) $k$-nearest Neighbor with Local Hu Moments as Robust Features:* For the second method, we implemented the approach outlined in [5]. In the preprocessing step a Kaiser window is used to separate the signal into different frames. From each frame the PSD is then estimated and normalized. Subsequently, the logarithm of the spectral energies for every window in a series of bands defined by a filterbank in the Mel scale is computed. The resulting spectral energy matrix is then subdivided into block matrices, from which the first Hu moment is then computed. This causes a reduction in the size of the column dimension in comparison to the the original spectral energy matrix, which is determined by the size of the block matrices. Finally, the discrete cosine transform is computed for each row and the first and last coefficient are discarded, resulting in the final feature representation. All feature sets were normalized to have zero-mean and unitary standard deviation. These features are then fed into a $k$-nearest neighbor classifier ($k$ equals 3), with standardized Euclidean distance as distance metric and with the inverse of the distance as weighting function. The parameters, window length (400 ms), sampling frequency (22.05 kHz), number of neighbors (3) were result of optimization and differ from the approach in [5].

### *E. Approach Proposed in this Work*

*1) Preprocessing:* Cough can be quantified in many different ways. The most intuitive method of counting cough is counting the characteristic explosive sounds [1]. Although obstructed airflow can produce noises with frequencies up to 20 kHz, cough characteristic and most energetic sounds are associated with frequencies below 10 kHz [39]. We, therefore, reduce the size of the data and optimize on the training time of our algorithms by downsampling the signals to 22.05 kHz after applying an anti-aliasing filter. Furthermore, in order to focus on the classification of the explosive characteristic, we computed the maximum amplitude of the extracted acoustic events in the time frames and subsequently extracted 325 ms around the maximum yielding 650 ms of the signal. The time frame of 650 ms corresponds approximately to the average total length of a cough [39] and was the result of hyperparameter optimization after employing grid search on the value range $[0.05s, 1s]$. We thus argue that 650 ms are enough to capture the explosive phase of a cough reflex. The extracted signal was then standardized by employing min-max normalization. Subsequently, a mel-scaled spectrogram was computed with 16 bands, 112 samples between successive frames and a 2048 point FFT. Mel-scaled spectrograms are visual representations of sound with respect to frequency and time. The frequency domain, however, is mel-scaled to represent the human perception of tone. Consistent with prior work we use mel-scaled spectrograms as inputs for our CNN architectures. Mel-spectrograms have been investigated thoroughly in literature and especially in conjunction with CNNs, they have been reported to perform best in comparison to other time-frequency representations [40].

*2) CNN Architecture:* Deep Learning methods have improved the state-of-the-art in various machine learning domains [26]. Especially their good generalization behavior in practice is among the main reasons for their adoption [41]. In recent years, CNNs have been one of the most frequently used architectures in the context of computer vision [42]. CNNs have proven effective in image classification, but have also shown promising results for audio applications, in particular in conjunction with spectrograms for audio event detection (e.g. environmental sounds of birds, violins, airplanes or footsteps) [43]. Typical CNN architectures consist of alternating convolution and max-pooling layers followed by a small number of fully connected layers [44]. At the heart of CNNs lies the convolutional layer, which exploits the translation invariance property of convolutions, i.e. it can abstract from the position where the object to be identified is located in the picture (or mel-spectrogram as in our application).

We therefore exploit the generalization behavior of deep learning, in particular CNNs to reduce the device-specific differences in our recordings. Among the most prominent CNN architectures is VGG [38], which was able to significantly improve image recognition accuracies by pushing depth to 19 layers and increasing the amount of convolutional layers. In our experiments, we exploited the same approach and optimized by iterating over the total amount of convolutional layers. Best results were achieved with a depth of 5 convolutional layers. Furthermore, as mentioned in the introduction, energy efficiency is a critical design criterion. We therefore developed our CNN cough detection architecture with two additional modifications in order to reduce the computational expenses of the CNN architecture. First, we introduced max global pooling, which combines the maximum of each feature map directly into the output layers, replacing the expensive fully connected layers. This operation, which may be seen as a simplification, brings the advantage that it does not introduce

new parameters. In our experiments we used max global pooling, which uses maximization instead of averaging [45], which ultimately resulted in a better performance on the validation set. Second, we replaced the regular convolutions with depthwise separable convolutions [46]. As a consequence, the number of calculations is reduced in comparison to a regular convolution and thus the efficiency of the architecture is improved. To illustrate this reduction we contemplate the following example, assume we have an image of $12 \times 12$ pixels, which we want to apply 64 $3 \times 3$ convolutional kernels with a stride of 1 and zero padding. The outcome is a new image of $10 \times 10$ pixels with 64 channels equivalent to moving a $3 \times 3$ kernel $10 \times 10$ times over 64 channels resulting in a total of $3 \times 3 \times 10 \times 10 \times 64 = 57.6k$ calculations. In contrast, a depthwise separable convolution introduces an additional $1 \times 1$ kernel separating the convolution into a depthwise $3 \times 3 \times 1 \times 1$ and a pointwise $1 \times 1 \times 64$ convolution. This results in $3 \times 3 \times 10 \times 10 = 0.9k$ plus $10 \times 10 \times 64 = 6.4k$ calculations yielding a total of $7.3k$ calculations. Further parameters which have been subject to hyper-parameter optimization on the validation set are the number of channels and the size of the convolutional filters. The optimum was reached for 64 channels. We further used dropout with a rate of 50%, which is a regularization technique to prevent overfitting [47]. The resulting parameters such as filter sizes and the whole architecture can be inspected in Figure 4b.

*3) Network Training:* In our work, the networks are trained using the stochastic optimization method Adam, which has been successfully applied in practice [48]. For our architectures, the learning rate lied in the range of [0.001, 0.007] and the batch size was set to 64. The learning rate and batch size were found by running the model several times with different configurations to find the optimal training hyper-parameters. Furthermore, weight initialization was accomplished by employing Xavier initialization [49] throughout the different architectures. We further applied techniques which have been reported to have a positive effect on stochastic gradient descent such as gradient clipping [50] and adding gradient noise [51]. We added gradient noise with a variance of 0.009.

*4) Ensemble Method:* In machine learning, ensemble methods use the power of different models in order to improve the predictive performance over one individual model [52]. Among the most popular methods lies bootstrap aggregating (bagging) [53], where each model of the ensemble is trained on a randomly drawn subset of the training set. This promotion of model variance has been shown to often improve performance of amalgamated models such as CNNs [54]. In particular, it has been shown empirically that by averaging the outputs of multiple estimators, a better estimate with less generalization error is obtained [55]. We hope to exploit this property of ensembles in a device-agnostic manner by employing bagging to our CNN architecture, not only to train on a subset of the training set, but also on a subset of the devices used for the recording. If an ensemble learning algorithm is sensitive to perturbation on training samples, then the individual models are dissimilar, and thus combining them will help improve the generalization performance [56]. Analogously, if constraining the training on randomly choosing samples of a subset of the recording devices introduces enough variance, then we may be able to reduce the generalization error across devices. Regardless of the number of devices, we propose the bagging algorithm as in Algorithm 1, where the inner loop corresponds to the common bagging algorithm. Since for our recordings we had a set of five devices ($K = 5$) at our disposal, we optimized on the amount of training data available to train each of our models ($I$ = CNN architecture) choosing the other parameters accordingly as $n = 4$ or 3, $\mu = |S_{\{D_k\}}|$ and $T = 1$, resulting in an ensemble of 5 or 4 models each trained on samples of incongruent sets of 4 or 3 devices with respect to our two scenarios. Figure 5 depicts the case of the ensemble consisting of 5 models.

**Algorithm 1** Device-agnostic bagging

**Input:** $I$ (an inducer, responsible for creating the model),
$T$ (the number of iterations),
$S = S_1, \dots S_K$ (the training set composed of recordings of different devices $K$),
$\mu$ (the subsample size)
**Output:** : $M_{t,k}; t = 1, \dots, T; k = 1, \dots, \binom{K}{n}$

$k \longleftarrow 1$
**repeat**
  $D_k \longleftarrow$ Sample one set of $n$ devices of $\binom{K}{n}$ combinations without replacement.
  $S_{\{D_k\}} \longleftarrow$ Fetch training samples related to this sample of devices $D_k$
  $t \longleftarrow 1$
  **repeat**
    $S_{k,t} \longleftarrow$ Sample $\mu$ instances from $S_{\{D_k\}}$ with replacement.
    Build classifier $M_{k,t}$ using $I$ on $S_{k,t}$
    $t++$
  **until** $t > T$
  $k++$
**until** $k > \binom{K}{n}$

## IV. Results

### A. Dataset

Overall, 43 healthy participants (31 female, 12 male) were recruited. Their age ranged from 18 to 45 with a mean value of 26 (SD: 6). This resulted in a total of 6737 cough, 3985 laugher, 3695 throat clearing, 731 speech and 443 forced expiration audio signals, which represents a vast number of cough samples and participants in comparison to former studies (see Table I). Further, training, validation and test set were composed in the following way. Out of 43 participants, we drew 11 at random, their audio samples were then included in the test set. From the remaining 32 participants, 5 participants were drawn at random and included in the validation set. The result fulfilled roughly the ratio of a 60/15/25 data split.


(a) VGG-19 architecture [38]

(b) Proposed CNN architecture

Fig. 4: CNN architectures: The annotations "dep sep conv $1 \times 7, 16$" represent a depthwise separable convolutional layer with a $1 \times 7$ convolutional filter and 16 channels and "fc 2" a fully connected layer with two outputs, respectively. Max pooling is abbreviated as "max pool".

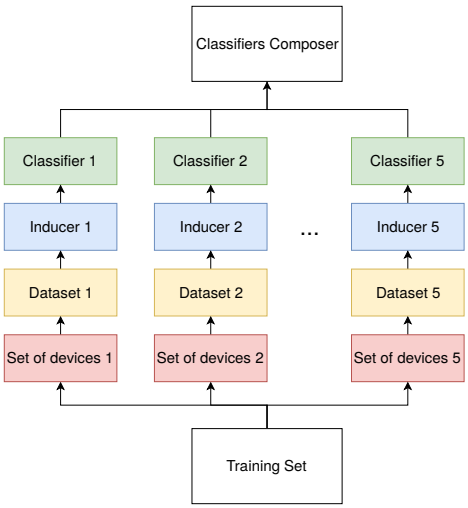


Fig. 5: Proposed bagging algorithm for five devices ($K = 5$) and parameters $n = 4$, $\mu = |S_{\{D_k\}}|$ and $T = 1$.

### B. Performance Results

The results of our evaluations are listed in Tables II - III. Table II gives an overview of the device-specific accuracies for the different devices. Table III describes the results across devices and shows the mean and the standard deviation of the following metrics, sensitivity (SENS), specificity (SPEC), accuracy (ACC), Matthews correlation coefficient (MCC), precision (PPV) and negative predictive value (NPV):

$$\begin{aligned}
\text{SENS} &= \frac{TP}{TP+FN}, \\
\text{SPEC} &= \frac{TN}{TN+FP}, \\
\text{ACC} &= \frac{TP+TN}{TP+TN+FP+FN}, \\
\text{MCC} &= \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \\
\text{PPV} &= \frac{TP}{TP+FP}, \\
\text{NPV} &= \frac{TN}{TN+FN},
\end{aligned}$$

where $TP, FP, TN$ and $FN$ denote the number of true and false positives and true and false negatives, respectively.

In Scenario I, the training was done on all devices and tested on unseen participants. We observe that our proposed ensemble architecture achieves not only the best mean accuracy of 90.9% when evaluated on all devices but also when evaluated on a per device basis. Only with comparable results to the single CNN architecture in terms of NPV and SENS. The two approaches from prior work did not attain the same levels of performance with respect to our metrics. The random forest based classifier outperforms the $k$-NN based classifier in terms of our metrics achieving an overall mean accuracy of 83.5%. The biggest discrepancy lies in the MCC values, which is a balanced metric used to measure the quality of a binary classification, with $k$-NN having the lowest value with 46.8% and the ensemble having the highest value with 81.7%. Further, the standard deviation of the SENS, ACC, MCC and NPV values is reduced for the CNN based architectures in comparison to the other approaches. We also observe that the quality of the devices' recordings is reflected in the results, meaning the accuracy values on HTC M8 and Nexus 7 tablet recordings are lower in comparison to the accuracy values of the other higher quality devices with respect to our reference (column accuracy means of Table II for Scenario I from left to right: 86.7%, 85.7%, 86.2%, 80.5%, 81.2%). Figure 6 concludes the results section of the first evaluation, showing receiver operating characteristic curves for all four approaches on all devices. The greatest area under the curve is again achieved by the ensemble (0.96), followed by the single CNN (0.95), the random forest (0.91) and the $k$-NN classifier (0.79).

In Scenario II, the training was done on all devices except on the one annotated as column name and then evaluated on the recordings of the unseen device of unseen participants. Again the highest mean accuracy is achieved by the ensemble with 87.6%. The accuracy of the ensemble is highest on all single devices with exception of the HTC M8 recordings, where it is outperformed by the single CNN architecture. The ensemble further attains the best performance in the SPEC, ACC, MCC and PPV values. As in Scenario I, the random forest classifier

| Class. | Røde NT1000 (ACC %) | iPhone 4 (ACC %) | Samsung S6 (ACC %) | HTC M8 (ACC %) | Nexus 7 tablet (ACC %) |
|---|---|---|---|---|---|
| SCENARIO I: | | | | | |
| *Approaches from Prior Work* | | | | | |
| RF with PCA [13] | 88.0 | 84.8 | 87.2 | 79.5 | 78.2 |
| $k$-NN with Hu Moments [5] | 77.5 | 77.2 | 73.4 | 66.3 | 68.3 |
| *Approach Proposed in this Work* | | | | | |
| CNN with Mel Spec | 89.5 | 89.7 | 91.5 | 87.1 | 88.7 |
| Ensemble CNN | 91.7 | 90.9 | 92.8 | 89.2 | 89.7 |
| SCENARIO II: | | | | | |
| *Approaches from Prior Work* | | | | | |
| RF with PCA [13] | 85.0 | 84.7 | 85.3 | 75.6 | 73.1 |
| $k$-NN with Hu Moments [5] | 72.2 | 76.5 | 63.6 | 60.9 | 63.7 |
| *Approach Proposed in this Work* | | | | | |
| CNN with Mel Spec | 83.6 | 85.3 | 89.3 | 86.2 | 84.6 |
| Ensemble CNN | 86.8 | 89.8 | 91.0 | 85.6 | 84.6 |

TABLE II: Device-Specific Classification Results

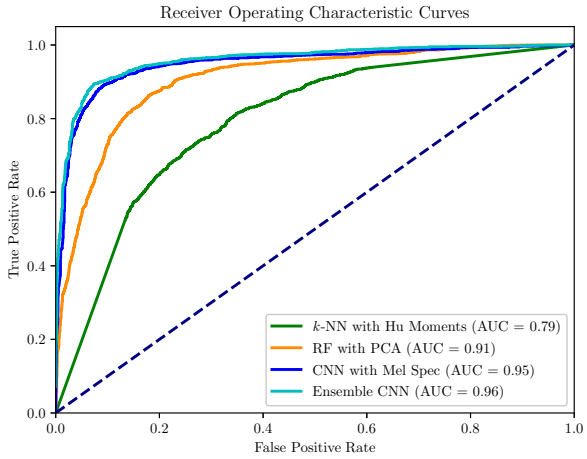


Fig. 6: Receiver operating characteristic curves with corresponding area under the curve values (AUC) for the four approaches investigated in Scenario I.

surpasses the $k$-NN classifier in terms of our metrics with a mean accuracy of 80.7 % and only remains second to the CNN architectures. We can further observe that the quality of the recordings is reflected in the results (column accuracy means of Table II for Scenario II from left to right: 81.9%, 84.1%, 82.3%, 77.1%, 76.5%). Finally, we can observe a drop in mean performance and an increase in standard deviation for Scenario II in comparison to Scenario I, in particular for the SENS, ACC, MCC and NPV values across all devices.

## V. Discussion

### A. Main Results

Given the dataset consisting of the same acoustic events recorded over five different recording devices, our results indicate that the difference in quality of the recordings matter and are the reason for the fluctuating performance values across devices. Those discrepancies are mitigated and can be made negligibly small at high performance values by our proposed architectures. However, we need to point out that those discrepancies are inevitably greater when the device is unknown to the model builder. Our efforts towards reducing the generalization error across devices by employing deep learning in form of a CNN architecture and its ensemble variation were successful and performed better in contrast to the established approaches in literature. The ensemble is especially remarkable, since the increase in performance is explained by the introduction of variance by alternating on the set of devices to which the model was trained. As shown in Figure 2 the devices differed in recording quality with respect to our reference.

### B. Practical Implications

In the development of the CNN architecture we strove for computational efficiency, resulting in a model with 17480 parameters where 1.232 MB memory and 10.74 million floating point operations are required for its execution. This raises the question of how to best employ the developed architectures into practice. Due to large computational power and memory requirements, deep learning architectures are typically deployed on cloud computing systems. In the case of smartphone-based cough detection, the smartphone continuously records audio signals and loads them into the buffer where it decides if the extracted audio event is relevant. If so, the data is transferred to the cloud in order to be classified. This brings the issue of additional privacy requirements, since audio signals may contain voice and thereby be privacy-sensitive information.

Continuous advances in mobile hardware make the deployment of deep learning models on the smartphone possible. For instance, the iPhone X is advertised to have 3GB RAM and on top of that a neural processing unit able to execute 600 billion floating-point operations per second. Not only efforts towards more powerful hardware but also towards achieving a more efficient deep neural network inference engine have been increasingly made [57], [58]. In a self-conducted experiment we deployed our model on a new Samsung Galaxy A3 (2017) smartphone with an Octa-core 1.6 GHz Cortex-A53 central processing unit and a battery capacity of 2350 mAh. For that purpose we developed an Android app, which continuously fills a 0.65s long audio buffer, from which a mel-scaled spectrogram is computed and classified with our pre-trained single CNN model using TensorFlow Lite, an open source deep learning framework for on-device inference [57]. When a cough is detected, the occurrence is counted and the total number is displayed. The pre-trained model was frozen in our Python development environment and converted to the TensorFlow Lite format before integration in the assets folder of the Android app. In this six hour long experiment, we compared the energy consumption of the developed app running our model in the background on a fully charged device in comparison to the energy consumption of the same device with same battery status without app. This experiment yielded a remaining percentage of battery charge of 75% and 95% for device with and without app, respectively. This results in an hourly average of 78.3 mA for the app

| Class. | SENS (mean ± SD %) | SPEC (mean ± SD %) | ACC (mean ± SD %) | MCC (mean ± SD %) | PPV (mean ± SD %) | NPV (mean ± SD %) |
|---|---|---|---|---|---|---|
| SCENARIO I: | | | | | | |
| *Approaches from Prior Work* | | | | | | |
| RF with PCA [13] | 86.7± 5.5 | 80.3±4.4 | 83.5±4.0 | 67.2±8.2 | 83.4±4.4 | 83.8±7.5 |
| $k$-NN with Hu Moments [5] | 81.2±7.3 | 65.6±3.3 | 72.5±4.6 | 46.8±10.1 | 65.9±3.3 | 80.7±8.6 |
| *Approach Proposed in this Work* | | | | | | |
| CNN with Mel Spec | 91.7±2.8 | 86.7±3.1 | 89.3±1.6 | 78.6±3.2 | 88.8±3.0 | 90.0±3.7 |
| Ensemble CNN | 91.7±3.1 | 90.1±3.5 | 90.9±1.5 | 81.7±3.0 | 92.0±3.2 | 89.5±4.2 |
| SCENARIO II: | | | | | | |
| *Approaches from Prior Work* | | | | | | |
| RF with PCA [13] | $83.3 \pm 8.0$ | $79.5 \pm 6.7$ | $80.7 \pm 5.3$ | $61.9 \pm 10.7$ | $83.0 \pm 7.8$ | $78.1 \pm 12.6$ |
| $k$-NN with Hu Moments [5] | $73.6 \pm 10.8$ | $62.3 \pm 4.3$ | $67.4 \pm 5.9$ | $35.4 \pm 13.8$ | $67.1 \pm 6.5$ | $67.9 \pm 16.0$ |
| *Approach Proposed in this Work* | | | | | | |
| CNN with Mel Spec | $86.7 \pm 5.9$ | $6.4 \pm 7.2$ | $85.8 \pm 2.2$ | $72.0 \pm 3.9$ | $88.8 \pm 6.9$ | $82.2 \pm 10.6$ |
| Ensemble CNN | $86.5 \pm 6.6$ | $90.5 \pm 4.8$ | $87.6 \pm 2.7$ | $75.5 \pm 5.4$ | $92.8 \pm 4.6$ | $81.2 \pm 10.2$ |

TABLE III: Classification Results across Devices

running in the background on the respective device. Even if this implementation can be made more efficient by introducing a step before classification discarding time frames containing silence, it shows the feasibility of using our model on-device over a prolonged period of time.

The results of this work demonstrate advantages in favor of the proposed CNN architectures, meaning higher prediction performance and smaller inter-device variability. We, therefore, argue that our architectures can enable device-agnostic mobile cough detection. The trade-off between the established approaches from literature, however, lies in the computational efficiency and thus energy consumption. The application of such a model, as shown in our experiment, may although depend on the battery life of the device. In stationary settings, where the person to be monitored lies flat (for example, in a hospital bed or overnight) a scenario could be envisioned where the smartphone is continuously plugged in. In more dynamic settings, where recharging of the device is not possible, a hybrid solution between the ensemble and the single CNN architecture may be considered assuming a drop in performance is tolerated. In conclusion, our proposed architectures fulfill the requirement of scalability by providing a high device-agnostic detection performance across devices. They may further find their way into practice and being deployed as a cloud- or client-based solution.

### C. Limitations

The obvious limitation of this study is that only voluntary coughs were used to examine the performance of the different approaches across devices. Even if voluntary coughs have already been used to show the feasibility of cough monitors [18], [19], [24], [32], reflex coughs would further allow a disease specific analysis of the performance values. Another limitation stems from the laboratory setting of the study, which limits the generalizability of the performance metrics. It is not clear whether the exact same performance patterns would emerge in settings that resemble a users everyday life. However, due to high performance values of our proposed architectures specifically in devices with lower quality recordings, a reasonable argument can be made that our architectures would still perform consistent across devices for reflex coughs in a different setting.

Finally, we would like to emphasize that our results can be enhanced by introducing a wider array of possible non-cough sounds, thereby increasing the amount of training data, yielding more powerful classifiers [59]. Moreover, transfer learning [60] and data augmentation [61] are further techniques, which can help improve the accuracy of CNN-based classifiers.

## VI. CONCLUSIONS & FUTURE WORK

This paper contributes to mobile cough detection by analyzing the performance of various mobile cough detection classifiers and their inter-device variability. We found that the proposed CNN architecture was less prone to variability across devices at higher performance values when compared to approaches from literature [5], [13]. We further bolstered performance by employing an altered version of bagging to the developed architecture. We, therefore, conclude that our architectures can enable device-agnostic cough detection by means of a smartphone, however the quality of the recording devices remains a limiting factor. Against the backdrop of these results, this work represents a very first step towards scalable, low-cost, ubiquitous, accurate and device-agnostic cough detection algorithms.

Our current work involves the collection of nocturnal data in a longitudinal field study with asthmatics and the assessment of the methods described in this paper with respect to asthmatic coughs. This is motivated by the number of nocturnal coughs which could be exploited as an objective assessment of asthma control [8], thus providing a scalable cost-efficient marker for asthma.

## REFERENCES


[1] A. Morice, G. Fontana, M. Belvisi, S. Birring, K. Chung, P. Dicpinigaitis, J. Kastelik, L. McGarvey, J. Smith, M. Tatar *et al.*, "Ers guidelines on the assessment of cough," *European respiratory journal*, vol. 29, no. 6, pp. 1256–1276, 2007.

[2] J. Smith and A. Woodcock, "Cough and its importance in copd," *International journal of chronic obstructive pulmonary disease*, vol. 1, no. 3, p. 305, 2006.

[3] S. Birring, T. Fleming, S. Matos, A. Raj, D. Evans, and I. Pavord, "The leicester cough monitor: preliminary validation of an automated cough detection system in chronic cough," *European Respiratory Journal*, vol. 31, no. 5, pp. 1013–1018, 2008.

[4] J. Smith and A. Woodcock, "New developments in the objective assessment of cough," *Lung*, vol. 186, no. 1, pp. 48–54, 2008.

[5] J. Monge-Alvarez, C. Hoyos-Barcelo, P. Lesso, and P. Casaseca-de-la Higuera, "Robust detection of audio-cough events using local hu moments," *IEEE Journal of Biomedical and Health Informatics*, 2018.

[6] R. de Marco, A. Marcon, D. Jarvis, S. Accordini, E. Almar, M. Bugiani, A. Carolei, L. Cazzoletti, A. Corsico, D. Gislason *et al.*, "Prognostic factors of asthma severity: a 9-year international prospective cohort study," *Journal of Allergy and Clinical Immunology*, vol. 117, no. 6, pp. 1249–1256, 2006.

[7] N. C. Thomson, R. Chaudhuri, C. M. Messow, M. Spears, W. MacNee, M. Connell, J. T. Murchison, M. Sproule, and C. McSharry, "Chronic cough and sputum production are associated with worse clinical outcomes in stable asthma," *Respiratory medicine*, vol. 107, no. 10, pp. 1501–1508, 2013.

[8] P. Tinschert, F. Rassouli, F. Barata, C. Steurer-Stey, E. Fleisch, M. A. Puhan, M. Brutsche, and T. Kowatsch, "Prevalence of nocturnal cough in asthma and its potential as a marker for asthma control (mac) in combination with sleep quality: protocol of a smartphone-based, multicentre, longitudinal observational study with two stages," *BMJ open*, vol. 9, no. 1, p. e026323, 2019.

[9] WHO, "World health organization. asthma." *Available: http://www.webcitation.org/6lwfJjosb [Accessed 2016-11-18].*, 2013.

[10] G. Gibson, R. Loddenkemper, Y. Sibille, and B. Lundbäck, *The European lung white book*. European Respiratory Society, 2013.

[11] S. Fox and M. Duggan, *Tracking for health*. Pew Research Center's Internet & American Life Project, 2013.

[12] J. Juen, Q. Cheng, and B. Schatz, "A natural walking monitor for pulmonary patients using mobile phones," *IEEE Journal of biomedical and health informatics*, vol. 19, no. 4, pp. 1399–1405, 2015.

[13] 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 *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 375–384.

[14] L. Li, Y. Chen, D. Wang, and T. F. Zheng, "A study on replay attack and anti-spoofing for automatic speaker verification," *arXiv preprint arXiv:1706.02101*, 2017.

[15] A. Deleforge and R. Horaud, "A latently constrained mixture model for audio source separation and localization," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2012, pp. 372–379.

[16] M. A. Coyle, D. B. Keenan, L. S. Henderson, M. L. Watkins, B. K. Haumann, D. W. Mayleben, and M. G. Wilson, "Evaluation of an ambulatory system for the quantification of cough frequency in patients with chronic obstructive pulmonary disease," *Cough*, vol. 1, no. 1, p. 3, 2005.

[17] S. J. Barry, A. D. Dane, A. H. Morice, and A. D. Walmsley, "The automatic recognition and counting of cough," *Cough*, vol. 2, no. 1, p. 8, 2006.

[18] E. Vizel, M. Yigla, Y. Goryachev, E. Dekel, V. Felis, H. Levi, I. Kroin, S. Godfrey, and N. Gavriely, "Validation of an ambulatory cough detection and counting application using voluntary cough under different conditions," *Cough*, vol. 6, no. 1, p. 3, 2010.

[19] T. Drugman, J. Urbain, and T. Dutoit, "Assessment of audio features for automatic cough detection," in *Signal Processing Conference, 2011 19th European*. IEEE, 2011, pp. 1289–1293.

[20] K. McGuinness, K. Holt, R. Dockry, and J. Smith, "P159 validation of the vitalojak 24 hour ambulatory cough monitor," *Thorax*, vol. 67, no. Suppl 2, pp. A131–A131, 2012.

[21] V. Swarnkar, U. Abeyratne, Y. Amrulloh, C. Hukins, R. Triasih, and A. Setyati, "Neural network based algorithm for automatic identification of cough sounds," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*. IEEE, 2013, pp. 1764–1767.

[22] J.-M. Liu, M. You, Z. Wang, G.-Z. Li, X. Xu, and Z. Qiu, "Cough detection using deep neural networks," in *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*. IEEE, 2014, pp. 560–563.

[23] Y. A. Amrulloh, U. R. Abeyratne, V. Swarnkar, R. Triasih, and A. Setyati, "Automatic cough segmentation from non-contact sound recordings in pediatric wards," *Biomedical Signal Processing and Control*, vol. 21, pp. 126–136, 2015.

[24] J. Amoh and K. Odame, "Deep neural networks for identifying cough sounds," *IEEE transactions on biomedical circuits and systems*, vol. 10, no. 5, pp. 1003–1011, 2016.

[25] H. A. Bickerman and S. E. Itkin, "The effect of a new bronchodilator aerosol on the air flow dynamics of the maximal voluntary cough of patients with bronchial asthma and pulmonary emphysema," *Journal of chronic diseases*, vol. 8, no. 5, pp. 629–636, 1958.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[27] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[29] M. vanGerven and S. Bohte, "Artificial neural networks as models of neural information processing: Editorial on the research topic artificial neural networks as models of neural information processing," 2017.

[30] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[31] T. Drugman, J. Urbain, N. Bauwens, R. Chessini, C. Valderrama, P. Lebecque, and T. Dutoit, "Objective study of sensor relevance for automatic cough detection," *IEEE journal of biomedical and health informatics*, vol. 17, no. 3, pp. 699–707, 2013.

[32] P. Casaseca-de-la Higuera, P. Lesso, B. McKinstry, H. Pinnock, R. Rabinovich, L. McCloughan, and J. Monge-Álvarez, "Effect of downsampling and compressive sensing on audio-based continuous cough monitoring," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. IEEE, 2015, pp. 6231–6235.

[33] I. Shih, T. Kowatsch, P. Tinschert, F. Barata, and M. Nißen, "Towards the design of a smartphone-based biofeedback breathing training: Identifying diaphragmatic breathing patterns from a smartphones microphone." Proc. of the 10th Mediterranean Conference on Information Systems (MCIS), 2016.

[34] P. Stoica, R. L. Moses *et al.*, *Spectral analysis of signals*. Pearson Prentice Hall Upper Saddle River, NJ, 2005, vol. 1.

[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[36] N. Silberman and S. Guadarrama, "Tf-slim: A high level library to define complex models in tensorflow."

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[39] J. Korpáš, J. Sadloňová, and M. Vrabec, "Analysis of the cough sound: an overview," *Pulmonary pharmacology*, vol. 9, no. 5-6, pp. 261–268, 1996.

[40] M. Huzaifah, "Comparison of time-frequency representations for environmental sound classification using convolutional neural networks," *arXiv preprint arXiv:1706.07156*, 2017.

[41] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 5947–5956.

[42] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[43] M. Meyer, L. Cavigelli, and L. Thiele, "Efficient convolutional neural network for audio event detection," *arXiv preprint arXiv:1709.09888*, 2017.

[44] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[45] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016, pp. 2921–2929.

[46] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *arXiv preprint*, pp. 1610–02 357, 2017.

[47] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[48] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[49] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[50] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[51] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, "Adding gradient noise improves learning for very deep networks," *arXiv preprint arXiv:1511.06807*, 2015.

[52] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[53] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, vol. 11, pp. 169–198, 1999.

[54] J. Guo and S. Gould, "Deep cnn ensemble with data augmentation for object detection," *arXiv preprint arXiv:1506.07224*, 2015.

[55] N. Ueda and R. Nakano, "Generalization error of ensemble estimators," in *Neural Networks, 1996., IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 90–95.

[56] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.

[57] Google Inc, "Tensorflow lite," https://www.tensorflow.org/lite, accessed: 2019-05-01.

[58] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, L. Jiao, L. Qendro, and F. Kawsar, "Deepx: A software accelerator for low-power deep learning inference on mobile devices," in *Information Processing in Sensor Networks (IPSN), 2016 15th ACM/IEEE International Conference on*. IEEE, 2016, pp. 1–12.

[59] M. Banko and E. Brill, "Scaling to very very large corpora for natural language disambiguation," in *Proceedings of the 39th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2001, pp. 26–33.

[60] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014, pp. 512–519.

[61] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.