# TB or not TB: Cough Detection and Tuberculosis Classification for Pulmonary Health Estimation

ELLIOT SABA, University of Washington, USA
JOSH FROMM, University of Washington, USA
JIAYAO CLARA LU, University of Washington, USA
SHWETAK N. PATEL, University of Washington, USA

Cough sounds contain important information on the health of the pulmonary system largely untapped by medical professionals due to the difficulty of accurately extracting diagnostically relevant metrics. Medical professionals employ a variety of techniques to estimate pulmonary health; however automated, accurate, mobile cough detection and classification technologies hold the promise of enabling new healthcare applications and sensing modalities that extend beyond the domain of current pulmonary health estimation methods. We present a novel cough detection and classification system that builds upon recent advances in adversarial deep learning. Our system enables applications such as personal cough counting, stationary cough collection, and fast and inexpensive screening for tuberculosis. We achieve a cough detection rate of 83.8% with less than 10 false alarms per hour, and a tuberculosis classification true positive rate of 80.6% with a false positive rate of 40.4% on a unique clinical dataset, and measure run-time performance for deployment on resource-constrained devices.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**; • **Applied computing → Life and medical sciences**; • **Computing methodologies →** *Machine learning*;

Additional Key Words and Phrases: Mobile Health, Pulmonary Health Estimation, Machine Learning

**ACM Reference Format:**
Elliot Saba, Josh Fromm, Jiayao Clara Lu, and Shwetak N. Patel. 2018. TB or not TB: Cough Detection and Tuberculosis Classification for Pulmonary Health Estimation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 0, 0, Article 0 (May 2018), 17 pages. https://doi.org/0000001.0000001

## 1 INTRODUCTION

Pulmonary ailments account for four of the top ten causes for death worldwide and are especially prevalent in low-income countries [54]. It is estimated that over 1 billion people worldwide suffer from pulmonary disease [21]. Coughing is a symptom of many of these ailments, including (but not limited to) asthma, tuberculosis, cystic fibrosis, lower respiratory infection, chronic obstructive pulmonary disease, and over a hundred others [31]. Symptom tracking is an important part of the health care process at all stages, whether for screening the general public to find new cases, assessing new patients, or tracking long-term cases [24, 37]. Current systems to track pulmonary ailments through cough sounds include patient self-reporting, manual cough counting and analysis, and automated cough frequency trackers [24, 27, 47]. Self-reporting of cough frequency and cough characteristics has been shown to lack the accuracy necessary for usage in clinical situations [9]. Manual cough counting, due to the unpredictable and intermittent characteristic of coughs, can be a very time-intensive process,

Authors' addresses: Elliot Saba, University of Washington, Seattle, WA, 98195, USA, sabae@uw.edu; Josh Fromm, University of Washington, WA, 98195, USA, jwfromm@uw.edu; Jiayao Clara Lu, University of Washington, WA, 98195, USA, jiayao@uw.edu; Shwetak N. Patel, University of Washington, WA, 98195, USA, shwetak@cs.washington.edu.


2474-9567/2018/5-ART0 $15.00
https://doi.org/0000001.0000001

requiring a dedicated listener to record all cough events over a time duration large enough to gather cough data for diagnostic purposes. Automated cough frequency trackers of varying accuracies for different applications exist [6, 8–11, 13, 20, 33, 39, 40, 47, 53]; however recent advances in machine learning technologies open up new applications.

The advent of new embedded computing platforms including smartphones, single-board computers such as the Raspberry Pi [3], and smart speaker systems such as the Amazon Echo [2] give rise to new applications and constraints on machine learning platforms. Algorithms that would have insufficient recall for a clinical application may instead be used within a smart speaker system to passively listen to coughs within a doctor's office for epidemiological work. Algorithms that function well within the spacious computational confines of a dedicated server may not fit within the constrained memory and computational limits defined by the power budget of a smartphone. Here, we list a few potentially interesting applications that are enabled by developments in this area:

- *Mobile cough counter*: Implemented as a cough detection algorithm on a personal mobile computation device with a microphone such as a mobile phone or smartwatch, this application would enable ubiquitous, personal pulmonary health tracking through constant cough detection. This application requires efficient algorithms so as to have as little impact on power usage as possible, to lengthen device lifetime, and to coexist with other classification algorithms such as personal digital assistants.
- *Listening station*: Implemented as a cough detection algorithm on a stationary device such as a smart speaker, this application would enable nighttime cough tracking for sleep studies and population cough counting for epidemiological studies where the cough station is located in public settings such as doctor's office waiting rooms. The ability for all computation to be performed locally on the device is critical for user privacy in this application, so that no raw audio is ever stored or transmitted outside of the device.
- *Clinical tool*: Implemented as a cough classification algorithm on a device such as a smartphone, this application would enable cheap screening methodologies for pulmonary ailments. We draw a clear distinction between cough detection algorithms, which typically must run faster than real-time and consume minimal computational resources for power and computational budget reasons, and cough classification algorithms which typically are only run once a cough has been identified, either manually or through a previously run cough detection algorithm. As such, cough classification algorithms do not have the same computational budgets applied to them and can run much slower than real-time.

Our contribution in this paper is to introduce a set of novel cough detection and classification algorithms that pay special attention to both the runtime requirements as well as the amount of data required to train the models. Specialized datasets are expensive to collect and curate, inspiring us to introduce an application of adversarial networks to integrate disparate datasets into a whole with minimal biases introduced. The two models constructed within this paper are a cough detection model (detecting cough events within an audio recording) and a cough classification model (classifying pre-segmented cough sounds into *tuberculosis* vs. *control* coughs). To address the *mobile cough counter* and *listening station* applications listed above, we achieve a cough detection rate of 83.8% with less than 10 false alarms per hour. To address the *Clinical tool* application, our cough classification model is applied to the task of classifying tuberculosis coughs and achieves a true positive rate of 80.6% with a false positive rate of 40.4% on the clinical dataset. Our models are explicitly designed to fit within embedded device profiles, enabling new sensing modalities such as smartphone and smartwatch continuous classification, as well as preserving user privacy by performing all computation upon local devices without the need for raw audio to be sent to a server for further processing.

## 2 BACKGROUND AND RELATED WORK

Cough sounds have long been known to be of diagnostic interest to pulmonologists. In 1996, Korpás et al. performed basic signal processing operations on captured "tussiphonogram" signals, inspecting the time domain waveforms and frequency domain periodograms in a clinical setting [31]. Of particular interest is the conclusion that the cough and the results of spirometry tests contain separate pieces of information; for administering bronchodilating drugs to patients would alter the spirometry results whereas it would leave the cough analysis results relatively unchanged.

In 2014, Spinou and Birring gave an overview of cough measurements and monitoring, with an explicit emphasis on the tools available to medical professionals that wish to investigate cough sounds and their frequency of occurrence [47]. Spinou and Birring note that there are dimensions upon which information from coughs can be extracted beyond just cough frequency alone; however, all of the available systems that are capable of extracting this information are at best semi-automatic, in the end requiring a human to perform the final classification. We note that their analysis included only systems that can take ambulatory recordings and extract cough sounds from them; they did not include systems intended to classify pre-segmented cough sounds.

Drugman et al. investigated the sensors most likely to be able to extract information from coughs, evaluating contact and non-contact microphones, electrocardiography sensors, chest belts, accelerometers and thermistors placed under the patient's nose [20]. Their analysis determined that, given the choice of a single sensor, non-contact microphones contained significantly more information about the cough than any other sensor.

### 2.1 Cough Detection Work

Barry et al. used Linear Predictive Coding (LPC) [43] and Mel-Frequency Cepstral Coefficients (MFCCs) [19] to model the sound of coughs and used a Probabilistic Neural Network to classify time windows as containing a cough or not [9]. This system, published as the *Hull Automatic Cough Counter*, requires a human to compute the final cough count after being presented with the windows of time containing cough sounds; doing so also allows humans to handle false positives and multiple coughs overlapping into a single classification. An hour of recorded audio could then be reviewed in 1 minute 35 seconds on average, substantially increasing the amount of data health care professionals could process.

Matos et al. fed MFCCs and derivatives of the same into a Hidden Markov Model (HMM) to determine the location of coughs in recordings [39]. Birring et al. built off of this work by designing the *Leicester Cough Monitor*, an ambulatory system that validated the system in a more naturalistic setting [13]. Using a portable recorder and a microphone that the user clips to their shirt, users are able to go about their lives while gathering a 6-hour audio recording of their daily activities including cough sounds. These coughs are automatically analyzed by an offline processing algorithm, and the results are reviewed by a human operator. Then, a second algorithm improves upon the original algorithm's detections using guidance from the human operator.

Barton et al. used a system published as *VitaloJAK*, a two-channel recording device that uses both a contact microphone on the chest and a non-contact microphone to simplify the signal processing challenge of disambiguating cough sounds from other acoustic events [10, 11]. The system calculates various running statistics on the short-time spectrum of the signals, and captures regions of the signal that have high energy and high spectral center of gravity. Once these periods of high energy/spectral center of gravity are determined, the coughs are counted manually. This system, similar to that of Barry et al. [9], is primarily used as a kind of data compression system, heavily reducing the amount of data that a human operator must classify in order to obtain cough counts.

Amrulloh et al. used a wide variety of signal features (MFCCs, Formant Frequency, Shannon Entropy, Zero-Crossing Rate, and Non-Gaussianity) fed into a Time-Delay Neural Network to segment cough sounds [8]. The authors expend much effort to accurately find not only the start times of coughs, but the end times as well, giving the ability to measure cough frequency and cough duration.

| *Publication* | *Results* | *Automatic* | *Dataset Size* | *Methods* |
|---|---|---|---|---|
| **Barry et al. [9]** | Sensitivity: **80%**<br>Specificity: **96%** | Partially | 75 | LPC/MFCC's with a PNN |
| **Matos et al. [39]**<br>**Birring et al. [13]** | Sensitivity: **91%**<br>Specificity: **99%** | Partially | 1834 | MFCC's with an HMM, followed by human-guided detection algorithms |
| **Monge-Alvarez [41]** | Sensitivity: **88%**<br>Specificity: **96%** | Fully | N/A | Hu moments with a KNN classifier |
| **Amoh [7]** | Sensitivity: **82%**<br>Specificity: **93%** | Fully | 627 | STFT visually classified using CNNs |
| **Larson et al. [33]** | Sensitivity: **92%**<br>Specificity: **99%** | Fully | 2558 | Eigenvector features with a random forests classifier |

Table 1. Selective summary of previous cough detection work showing the results, whether the method was semi- or fully automatic, the size of the dataset in number of coughs, and the methods employed.

Larson et al. used eigenvector decompositions of cough sounds fed into a random forest classifier to detect coughs [33]. Of particular note is the unusually high accuracy attained by the authors as well as the development of a "privacy-preserving" feature of the algorithm; the raw audio data, as part of the analysis process, is transformed through the eigenvector matrix built during training, destroying the intelligibility of any speech in the audio recording but retaining the cough sounds well enough to perform cough detection. This is a feature many medical professionals desire of any offline algorithm since it eliminates the privacy concerns of retaining large amounts of speech data of users that are not necessarily affiliated with the study.

## 2.2 Previous Cough Classification Work

Many groups have done research throughout the past few decades to develop automatic systems to classify various attributes of coughs such as whether the cough is wet/dry or the intensity of the cough. A “wet” cough refers to one that produces sputum (biological byproduct typically created due to the body’s immune response to an infection within the lungs) when the patient coughs. This sputum causes a "rattling" sound within the patients chest as they cough. The effect ranges from subtle to extremely prominent depending on a variety of factors, one of which is the severity of pulmonary disease causing the buildup of sputum within the lungs in the first place [31]. Its presence serves as an important diagnostic component when trying to quantify the severity of a pulmonary ailment within a patient. Tuberculosis, pneumonia, bronchitis, and other pulmonary ailments have prominent effects on the pulmonary system, and cough classification work is founded upon the hypothesis that the effects these diseases have upon the cough sound is identifiable.

Al-Khassaweneh and Abdelrahman used Wigner distribution functions and wavelet packet transforms to analyze the time-frequency energy distributions of cough sounds to detect asthma [5]. Their analysis showed that asthmatic patients tend to have coughs with different energy signatures than non-asthmatic patients. In particular, asthmatic coughs had distinguishably more energy, especially in the low frequency/long scale bins.

Subbaraj et al. employed a bandpass temporal energy estimate of cough recordings to classify the intensity of a cough [49]. They also investigated potential visualizations for their cough counting methods, mapping classified cough intensity versus time, to give a high level overview of a patient’s cough activity over long periods of time.

| *Publication* | *Task* | *Results* | *Automatic* | *Dataset Size* | *Methods* |
|---|---|---|---|---|---|
| **Al-Khassaweneh and Abdelrahman [5]** | Asthma | Sensitivity: **88%** | Fully | 24 | Spectral estimation with KNN [18] |
| **Subbaraj et al. [49]** | Intensity | Accuracy: **98%** | Partially | 215 | Temporal energy-based regression |
| **Swarnkar et al. [50]** | Wet vs. Dry | Sensitivity: **55%**<br>Specificity: **93%** | Fully | 536 | A variety of features with an LRM |
| **Abeyratne et al. [4]** | Pneumonia | Sensitivity: **80%**<br>Specificity: **73%** | Fully | 440 | A variety of features with an LRM |
| **Botha [14]** | Tuberculosis | Sensitivity: **82%**<br>Specificity: **95%** | Fully | 518 | Log-spectral bands with an LRM and clinical metrics |

Table 2. Summary of previous cough classification work showing what classification task was attempted, the results, whether the method was semi- or fully-automatic, the size of the dataset in number of coughs, and the methods employed.

Although the cough detection and wet/dry classification is not automatic and requires human intervention, the intensity regression is fully automatic.

Swarnkar et al. used a wide variety of signal processing methods to differentiate wet and dry coughs [50]. Their methods included analyzing spectral energy, temporal envelope, and time-independent waveform statistics such as kurtosis fed into a Logistic Regression Model. Of particular note is the inconsistency of the human ground-truth scoring of coughs as wet/dry. The authors employed two domain experts to perform the scoring, and the two experts agreed on only 80% of all cough events. This underscores the difficulty of such a classifier; even among human domain experts, what constitutes a wet cough versus a dry cough is not completely well-defined. The accepted approach in this and other works is to use only the events upon which a majority of expert annotators are in agreement and to ignore the others.

Abeyratne et al. analyzed cough sounds to diagnose asthma in pediatric patients [4]. Using a combination of time-series statistics (e.g., non-gaussianity score, kurtosis, etc.), formant-frequency tracking, general temporal-spectral energy-based features (e.g., MFCCs) and others, the authors built a logistic regression model to classify children as either pneumonia or non-pneumonia. Although their cough classification accuracies are not as high as many of the other references, their chosen standard to measure against was the World Health Organization guidelines for diagnosing pneumonia, which gives a sensitivity of 83% and a specificity of 47% on a set of 91 patients, leading to an overall accuracy of 75%.

## 2.3 Summary of previous work

These collected works illustrate the wide variety of research questions surrounding coughs as a clinical tool and underscore the interest the medical community has in performing this analysis. We explicitly point out the varying applications and subsequent performance levels expected of published research in these fields. For some cases, such as automatic ambulatory cough detection, specificity (the ability for the algorithm to correctly reject non-cough sounds) appears to be very desirable, the key metric being the number of false alarms per hour.

However, in the case of a nighttime cough tracker, sensitivity is not quite as critical for the algorithm to be effective as the number and type of acoustic events requiring classification will be much more limited.

## 2.4 Deep Learning

Our cough detection and classification algorithms make use of the recent advances in machine learning technologies yielded by the advent of "deep learning" [34, 45]. Historically, signal processing algorithms and machine learning models would be expert systems, designed by humans with deep domain knowledge taking advantage of physical or mathematical properties of the signals being analyzed in order to extract relevant information. Deep learning seeks to reduce the required expert knowledge by allowing computations of the correct type and shape to automatically learn the parameters for the computations through back-propagation [45]. The counterpoint to this benefit is that a deep learning system, being a statistically-inferred model with many parameters, requires large amounts of training data to converge to a stable estimate for all parameters within the model. Data augmentation techniques such as perturbing inputs [51] or model augmentation techniques such as adding dropout [48] and introducing regularization constraints [32] can alleviate some of the need for vast quantities of data; however there remains an intrinsic relationship between the number of parameters within a model and the number of training examples required for a model to stably converge. In this work, we explicitly address the problem of insufficient data through a combination of models with a restricted set of parameters, data augmentation, and adversarial networks.

Convolutional neural networks [35] provide an excellent foundation for building detection and classification systems for signals that have information that is "localized" while retaining a relatively small number of parameters relative to a model that finds relationships across larger swaths of input data. A convolutional neural network finds shift-invariant joint probability distributions inherent within the input signal that can be used to classify the signal into the desired categories (e.g. *cough* vs. *non-cough*). This can be viewed as a kind of "pattern matching" upon the two-dimensional image generated by the spectro-temporal decomposition, where the patterns themselves are learned through back-propagation. Mathematically, the pattern match probabilities are calculated through the fundamental convolution equation:

$$\mathbf{O}(x, y, j) = \sum_{i=1}^{m} \sum_{u,v=0}^{s-1} \mathbf{K}(u, v, i, j)\mathbf{I}(x + u, y + v, i) \tag{1}$$

Where the input tensor $\mathbf{I} \in \mathbb{R}^{w \times h \times m}$ is transformed by the kernel tensor $\mathbf{K} \in \mathbb{R}^{s \times s \times m \times n}$ to create the output tensor $\mathbf{O} \in \mathbb{R}^{w \times h \times n}$, where each element of the output tensor $\mathbf{O}$ represents the likelihood of a particular pattern existing at that location within the input tensor. The parameters $w$ and $h$ represent the input tensor width and height respectively, while $m$ and $n$ represent the channels of the input and output tensors. We note that the learned parameters $\mathbf{K}$ are independent of input data size and are typically orders of magnitude smaller than the data they operate on. Convolutional networks serve to continually extract information and reduce the dimensionality of input data until a final detection/classification stage which is applied once the data rank is small enough to be amenable to more traditional neural network architectures, such as fully connected layers.

## 2.5 Adversarial Networks

Generative adversarial networks (GANs) are a deep learning method for transforming one signal domain to match the distribution of another [22]. GANs have been proven to be a state-of-the-art method for matching the distribution of datasets. For example, in a process known as style transfer, GANs are used to reconstruct an image in a manner stylistically similar to a target image [29]. A GAN operates by taking a standard deconvolutional network that operates upon noise as an input and produces an image and pairs it with a discriminator network that is trained to distinguish generated (fake) images from the target images. During training, the accuracy of the discriminator is used as a loss term for the generative network. The loss produced by the discriminator effectively forces the generative network to learn the distribution of the target dataset in order to consistently fool the discriminator.

In this work, we take the concept of a GAN and introduce a small modification we call the discriminative adversarial network (DAN). We apply a DAN to ensure that our classification algorithm learns features independent of data collection site, an essential detail when attempting to fuse two highly imbalanced data collections. This DAN operates in opposition to the classification network, forcing portions of the classification network to not make use of patterns or information within the input data that could be used to determine which data collection the data originated from. In essence, it forces the model to ignore certain pieces of information that would otherwise heavily bias the model predictions.

## 3 EXPERIMENTAL DESIGN

In this section, we detail the development and evaluation of our cough detection and classification machine learning models. All model training was performed using the Flux machine learning library [25] in the Julia programming language [12]. Model deployment and performance measurements were performed on a Raspberry Pi 3 B+ [3] using the MXNet machine learning library [16].

### 3.1 Data Corpus

Our machine learning models are trained upon a large cough sound database collected from two separate sites: an *Ambulatory* dataset and a *Clinical* dataset. Although each were collected for distinct purposes, we will show that the combination of both poses challenges and provides opportunities for developing more robust and useful machine learning models than either dataset used independently. The audio recordings in both cases were annotated by a team of expert annotators trained to detect cough sounds.

The *ambulatory* dataset was collected at *anonymous* and consists of 64.3 hours of audio taken from 17 subjects (7 female) already known to exhibit cough symptoms before enrollment; 8 participants were diagnosed with a cold, 5 with chronic cough due to various reasons including smoking, 3 with asthma and 1 with allergies. A total of 2420 coughs are represented in this dataset. . Represented within this dataset are a multitude of other audio sources such as human speech, motor vehicles, laughter, and sounds as the mobile device moved about.

The *clinical* dataset was collected at *anonymous* and consists of 52.6 hours of audio taken from 56 participants (27 female), 45 of whom were previously diagnosed with pulmonary tuberculosis. A total of 1552 coughs are represented in this dataset, with 1378 of those coughs produced by TB-positive subjects. The non-tuberculosis coughs (referred to as "control" coughs) were produced by users with other pulmonary ailments. The *clinical* dataset was collected by participants sitting in a closed room with a TV playing in the background for 1 hour. . Represented within this dataset are human speech, TV noise, and various background sounds such as door closing and a whirring fan. We note that in many cases these sources of background noise are significantly louder than the cough sounds.

For cough classification tasks, it is imperative that mixing data from separate data collections does not cause the machine learning model to simply distinguish which site a recording came from due to the model learning some commonality within datasets such as the acoustics of the room. This would cause the model to learn about the environment the cough sounds were collected within as opposed to the cough sounds themselves. As detailed in the paragraphs above, our datasets are a prime example of unbalanced datasets where this would be a problem, as all tuberculosis coughs are contained within the *clinical* dataset, and almost all of the control coughs are contained within the *ambulatory* dataset. To address this imbalance, we will utilize a Discriminative Adversarial Network (DAN) to ensure that the machine learning model learns more than just differences in data collection sites when training classification models.

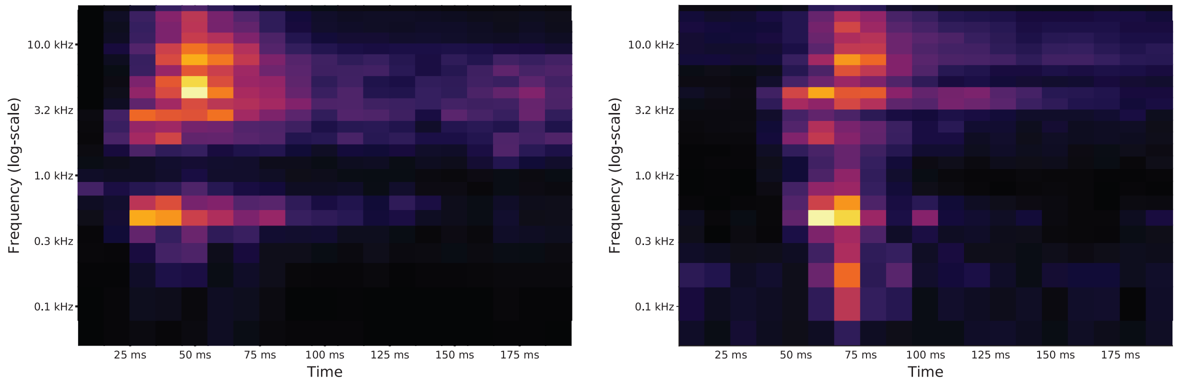


Fig. 1. Examples of cough signals visualized using the gammatone filterbank (GTFB) spectro-temporal decomposition

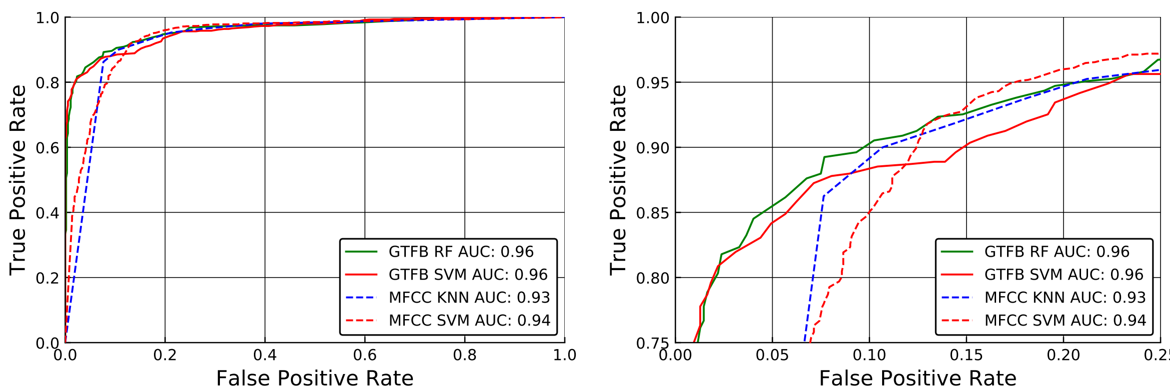


Fig. 2.

All data collections were approved by the Institutional Review Board at their respective collection sites.

## 3.2 Feature Preprocessing

Previous successful cough detection work hinges upon the usage of spectro-temporal decompositions coupled together with a learning model for making cough/non-cough classifications. One example of this is the work of Matos et al. using MFCC's as a spectro-temporal decomposition which are then input into an HMM to learn the relationships between different spectro-temporal patterns. Which signal decompositions and learning methods are most effective at signal detection remains an open research question; in recent years, however, great gains in signal processing and machine learning accuracy have been made through the usage of deep learning coupled with signal processing techniques to extract relevant information from signals. This data preprocessing and feature extraction is a critical step to ensuring that the machine learning models converge in a timely manner. Without this dimensionality reduction, the number of parameters that must be learned through back-propagation would far outstrip the amount of data available to train on.

| **Layer** | **Output Shape** | **Parameters** | **Runtime** |
|---|---|---|---|
| Feature preprocessing (GTFB) | $24 \times 19 \times 1$ | 0 (0.0%) | 36.9ms (13.4%) |
| 3x BN-CNN-LReLU | $24 \times 19 \times 8$ | 1316 (26.2%) | 153.2ms (55.6%) |
| Max Pooling | $12 \times 9 \times 8$ | 0 (0.0%) | 4.7ms (1.7%) |
| 3x BN-CNN-LReLU | $12 \times 9 \times 8$ | 1848 (36.7%) | 64.3ms (23.3%) |
| Max Pooling | $6 \times 4 \times 8$ | 0 (0.0%) | 1.0ms (0.4%) |
| 3x BN-CNN-LReLU | $6 \times 4 \times 8$ | 1848 (36.7%) | 15.2ms (5.5%) |
| Global Average Pooling | $1 \times 1 \times 8$ | 0 (0.0%) | 0.2ms (0.1%) |
| Fully Connected | 2 | 18 (0.0%) | 0.1ms (0.0%) |
| Total | | 5030 | 275.6ms |

Table 3. The cough detection network architecture, with learned parameter distribution and runtimes measured per-layer. All runtimes measured on a Raspberry Pi 3 B+, using a batch size of 100, equal to processing a full second of audio at once.

## 3.3 Baseline Comparison

## 3.4 Model Design

After feature preprocessing, the input to the machine learning model can be viewed as an image with a single channel, the result of applying the gammatone filterbank to the input audio data, broken up into overlapping time frames. An example of this decomposition is shown in Figure 1, where two exemplary cough signals are shown side-by-side with color mapping the energy at each point in time and frequency. Note that the GTFB yields an exponentially widening filterbank, and as such the frequency axis is log-scale.

*3.4.1 Detection Model.* Convolutional networks can be stacked on top of each other to find larger and more complex patterns within data. We employ stacks of batch normalization [26] (BN), Convolutional neural network (CNN) and LeakyReLU (LReLU) [38] layers to create "blocks" of convolutional kernels. Each block contains three consecutive sequences of a BN-CNN-LReLU layer grouping. We separate these blocks with max pooling layers to reduce image resolution, culminating in a global average pooling layer [36] and a final fully connected layer with softmax activation to output probabilities for the two classes (*cough* vs. *non-cough*). . Throughout the network, all convolutional layers output eight channels, and all kernel sizes are $3 \times 3$. We report the overall runtime and memory requirements for the model as measured on a Raspberry Pi 3 B+ in Table 3.

*3.4.2 Classification Model.* The classification model for classifying coughs into *tuberculosis* vs. *control* coughs is extremely similar to the detection network. The differences are that the feature preprocessing stage generates 32 filterbank outputs instead of 24, the network contains 4 stacked blocks of convolutions rather than 3, and each convolutional layer generates 6 channels instead of 8. These changes, while relatively minor, allow the classification network greater latitude in building complex patterns out of the stacks of convolutional kernels. This in turn significantly boosts the recognition accuracy while affecting a relatively minor impact on performance without increasing the number of parameters significantly. We note that the runtime of this model is similar to that of the detection model; however, as runtime is rarely a concern for classification models we will forgo any analysis of our cough classification model performance.

As mentioned in Section 3.1, the datasets for cough classification are highly imbalanced; due to differences across the datasets, it is necessary to take steps to prevent the classification model from simply learning the difference between the *clinical* and *ambulatory* datasets rather than learning the difference between a *tuberculosis*

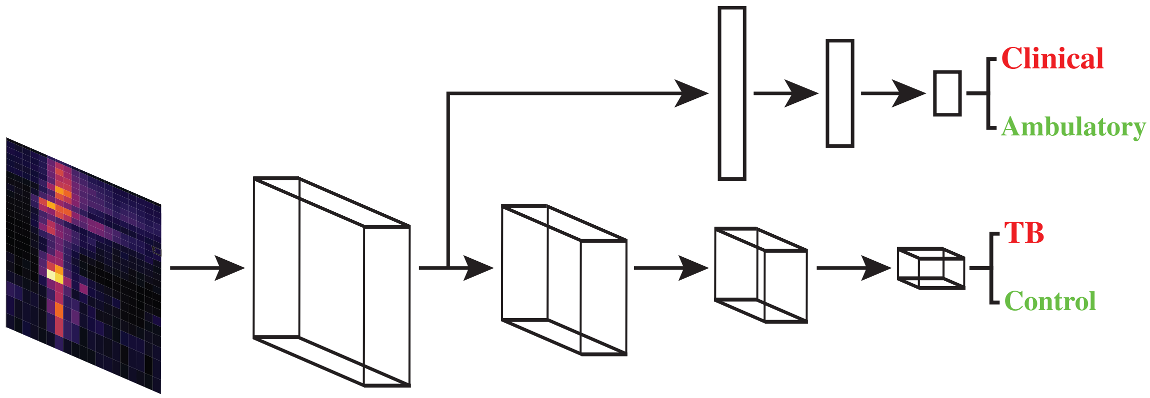


Fig. 3. Cough classification architecture with DAN and CNN architectures visualized. Along the bottom lies the convolutional classification network that classifies cough type, with the outputs from the first layer feeding into a multilayer perceptron that classifies dataset source.

cough and a *control* cough. In essence, the gradients created by dataset differences overwhelm the gradients imposed by the actual differences between classes. To remedy this, we employ a Discriminative Adversarial Network (DAN) to take the outputs of the first convolutional "block", and attempt to classify the input data sample as either stemming from the *clinical* dataset or the *ambulatory* dataset. This is shown graphically in Figure 3, with the typical convolutional network classifying what type of cough was input, and a second DAN classifying the dataset from which the cough originated. The DAN takes the form of a 3-layer multilayer perceptron with linearly decreasing layer sizes and LeakyReLU activations between layers. The last layer feeds into a softmax activation that predicts the dataset for the given input cough samples. The ability of the DAN to correctly classify an input data sample is then pitted against the ability of the main network to classify an input cough. The network loss $L$ is therefore a function of both the classifier network loss $L_C$, and the discriminator loss $L_D$ as shown in the training procedure given in Algorithm 1, although this changed loss is relevant only to the shared first block, as that is the only section of the network that is updated by both the discriminator and the classifier.

## 3.5 Model Training Methodology

For all model training, the input dataset was partitioned by participant and randomly split into 5 folds, one of which was set aside to be used as the test set to measure model accuracy. Detection models were trained on minibatches containing a prescribed ratio of cough samples and non-cough samples randomly selected from the training set and then tested by running whole recording files from the test set through the model and classifying each time window; those classification results were compared to the ground truth annotations. Classification models were trained on minibatches containing a balanced mix of tuberculosis and control coughs randomly selected from the training set and then validated by running each cough in the testing set through the model. A label was determined for each sample and compared to the ground truth diagnosis for the participant from whom the cough recording came. Model training was performed using the RMSProp optimizer [52] until test loss stopped decreasing for ten training epochs. The classification model weights were trained according to the DAN training procedure given in Algorithm 1, with a $\lambda$ parameter of 0.1 and a learning rate $\eta_t$ of 0.001. Training a single instance of the detection model required roughly 48 hours on a single workstation.

**ALGORITHM 1:** Classification and DAN training procedure

**Input:** Sample $X$ with Label $Y$ from dataset $S$, learning rate $\eta_t$, DAN strength parameter $\lambda$.
**Output:** Updated weights $W_D$ and $W_C$ of discriminator and classifier networks.
$\hat{Y}, \hat{S}$ = Forward($X$);
`// Update Discriminator`
$L_D$ = SoftmaxCrossEntropy($S, \hat{S}$);
$\frac{\partial L_D}{\partial W_D}$ = Backward($L_D, W_D$);
${W_D}^{t+1}$ = UpdateParameters($W_D$, $\frac{\partial L_D}{\partial W_D}, \eta_t$);
`// Update Classifier`
$L_C$ = SoftmaxCrossEntropy($Y, \hat{Y}$);
$L = L_C - \lambda L_D$;
$\frac{\partial L}{\partial W_C}$ = Backward($L, W_C$);
${W_C}^{t+1}$ = UpdateParameters($W_C$, $\frac{\partial L}{\partial W_C}, \eta_t$);

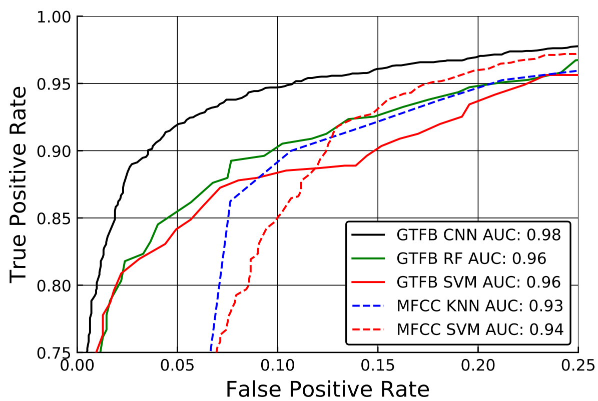


Fig. 4.

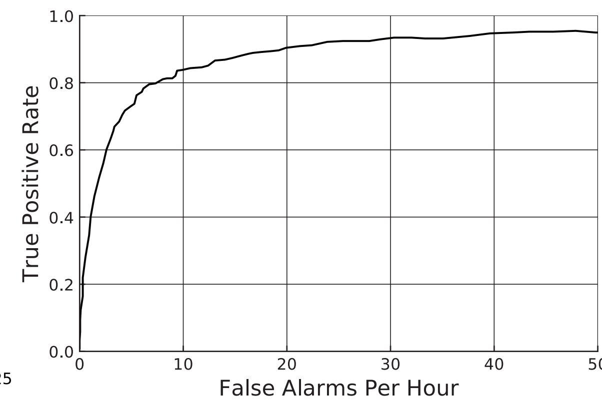


Fig. 5. False alarm curve showing detection performance of CNN model with median filtering.

## 4 RESULTS AND DISCUSSION

Using the models designed in Section 3.4 and applying them to the datasets described in Section 3.1, we now detail the performance of the models on both detection and classification tasks. In the case of detection, we first examine the ability of the model to classify a single window of time as either a cough or a non-cough. We then examine the coherence of the model predictions across time, finally comparing the predictions to ground truth annotations to obtain a count of correctly identified coughs and an estimate of the rate of false alarms per hour. In the case of classification, we examine the ability of the model to classify a pre-segmented cough as either a tuberculosis cough or a control cough, paying extra attention to the fact that the datasets used are highly unbalanced. We examine the efficacy of using a DAN to ensure the trained model can successfully integrate information from both datasets to create a single unbiased model. Finally, we measure the runtime performance of the detection model to evaluate its deployability on embedded platforms.

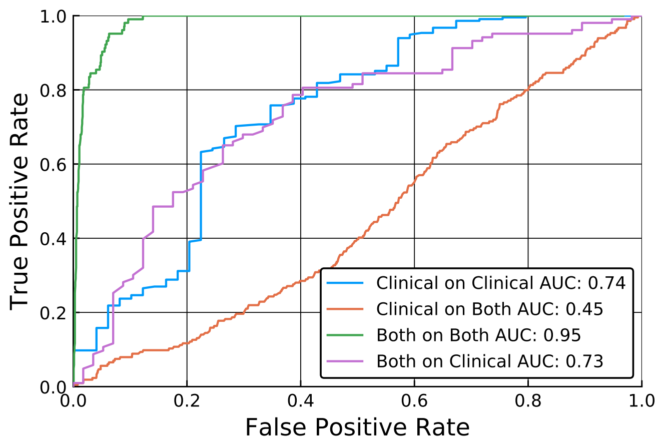


Fig. 6. ROC curve showing cough classification model performance for four different training/testing dataset configurations.

| | Predicted Positive | Predicted Negative |
|---|---|---|
| TB | 81.9% (176) | 18.1% (39) |
| Control | 44.9% (22) | 55.1% (27) |

Table 4. Clinical on Clinical confusion matrix

| | Predicted Positive | Predicted Negative |
|---|---|---|
| TB | 93.9% (201) | 6.1% (13) |
| Control | 92.7% (2294) | 7.3% (180) |

Table 5. Clinical on Both confusion matrix

| | Predicted Positive | Predicted Negative |
|---|---|---|
| TB | 94.2% (97) | 5.8% (6) |
| Control | 6.1% (86) | 93.9% (1315) |

Table 6. Both on Both confusion matrix

| | Predicted Positive | Predicted Negative |
|---|---|---|
| TB | 80.6% (83) | 19.4% (20) |
| Control | 40.4% (23) | 59.6% (34) |

Table 7. Both on Clinical confusion matrix

### 4.1 Detection Results

Cough detection performance is measured on an instantaneous frame-by-frame basis, and as grouped “cough events” compared against ground truth annotations. Figure 4 shows a receiver operating characteristic (ROC) curve for frame-by-frame processing across the test set for our cough detection model as compared against the best baseline methods on the same dataset. The ROC curve visualizes the trade-off between true and false positive rates and is generated by sweeping the detection probability threshold. This allows applications with a need for a high true positive rate (e.g. a cheap algorithm to identify coughs in long recordings for further analysis by a more complex algorithm or human) and applications that require a low false positive rate (e.g. cough stations placed in a noisy environment such as a doctor’s office) to both make use of the same base model. Figure 4 shows that in order to obtain, for example, an **80%** true positive rate, the model will generate a **1.1%** false positive rate.

Moving beyond instantaneous detection results, multiple temporal detections are combined into a set of cough events. A median filter of length 9 frames is employed to reject spurious detections and smooth the detection signal. Contiguous cough classifications are coalesced together to form single cough events, and these cough events are then compared to the ground truth annotations. A true positive (a properly detected cough event) vs. false alarms per hour metric can thus be calculated, as shown in Figure 5. We plot the percentage of true cough events detected versus the false positive rate per hour (labeled in literature as “false alarms per hour”) in Figure 5. This figure shows that to reach a true positive rate of **80%**, the model will generate **7.3** false alarms per hour.

### 4.2 Classification Performance

Cough classification performance is primarily measured by the true positive and false positive rates produced by a model as it classifies coughs into the two relevant classes (*tuberculosis* vs. *control*). To test the classification model in isolation from detection accuracy, all tests of the classification model are performed upon pre-segmented cough samples. Figure 6 shows receiver operating characteristic (ROC) curves for four situations along with area under curve (AUC) numbers, showing the difference in overall model performance depending on whether the model was trained on only data from the *clinical* dataset or both datasets, and whether the model was tested on only data from the *clinical* dataset or both datasets. Models trained on both datasets employed a DAN to avoid overfitting onto dataset variations. Tables 4, 5, 6 and 7 list confusion matrices for the four different configurations, picking representative points along the ROC curves given in Figure 6.

The "Clinical on Clinical" results in Figure 6 and table 4 show that the classification model is able to learn to detect *tuberculosis* coughs, however when tested upon the both the *clinical* and *ambulatory* datasets in the "Clinical on Both" condition, classification accuracy is extremely poor. This is due to significant dataset differences, and the fact that the *clinical* dataset contains a mere 174 control coughs. To remedy this, we train a new model upon both datasets, to make use of the large number of control coughs within the *ambulatory* dataset. Training the model upon the two highly unbalanced datasets is in itself insufficient, as the model simply keys off of the differences between datasets and associates all *clinical* data as *tuberculosis* cough samples and all *ambulatory* data as *control* cough samples. We therefore use a DAN to force the classification network to learn features that cannot be used to simply discover which dataset a cough sample comes from. Testing this model that was trained on both datasets across the entire test set is shown as the "Both on Both" condition within Figure 6 and Table 6. We also show that this same model, when tested upon *clinical* data only (shown as "Both on Clinical" in Figure 6 and Table 7), performs similarly to the original "Clinical on Clinical" condition. We note that the randomized test set partitioning described in 3.5 causes the total numbers for each class within the test set to change slightly between the model trained on *clinical* data only, versus the model trained on both datasets.

To compare against a non-acoustic tuberculosis detection system, sputum testing for tuberculosis using a GeneXpert G4 system yields a true positive rate of 52.8% and a false positive rate of 5.0% [42]. When testing "Both on Both" (chosen as the most similar experiment to the GeneXpert evaluation) we are able to obtain an example true positive rate of **94.2%** and an example false positive rate of **6.1%**.

### 4.3 Runtime Performance

Our models have been designed from the ground-up for usage in embedded processing environments, with low computational resources required and an emphasis on models that can be deployed onto resource constrained devices. Table 3 gives detection model timings as measured on a Raspberry Pi 3 B+. The timings were calculated using the built-in profiling mechanisms of MXNet. Feature preprocessing and the first convolutional block account for the majority of the runtime requirements, adding up to nearly 70% of the total CPU time used per batch. The numbers reported in Table 3 are representative of calculations with a batch size of 100, which corresponds to processing a full second of audio at once. This reduces library overhead to less than 1% of total runtime and is a realistic latency target for all applications considered within this paper. We highlight the low memory requirements of the trained detection model: The model consists of **12504** static parameters, (including, for example, GTFB values) and **5030** learned model parameters, yielding a total model memory footprint of approximately **70 kB**. Peak memory usage by the model during classification with a batch size of 100 remains less than **1.5 MB**, easily fitting within even the smallest embedded DSP platforms. This is in strong contrast to many previous works that depend on machine learning models such as random forests or support vector machines, which routinely require much more memory than is available in embedded devices.

### 4.4 Future Work

Future effort in cough detection and classification technologies has multiple promising paths forward:

*Runtime performance*: Using convolutional networks to naïvely classify streams of data results in large amounts of wasted work; due to the shift invariance of convolutional networks, the first layer of the CNN can re-use the previous timestep's output for all but one time frame, significantly reducing the computational load of the most expensive portion of the network. Network optimizations such as the usage of separable convolutions [17] and the usage of extremely low precision numerics [30] have grown in popularity within the machine learning community and could serve to further reduce computational load; however, further investigation is warranted to quantify the effect on network accuracy such optimizations would enact.

*Further data collection*: As stated repeatedly throughout this paper, the quantity and quality of the data collected for a machine learning model directly impacts the efficacy of the model. More data for cough classification in particular would enable deeper and more complex models to be learned, and the more the datasets can be balanced the better for model training and stability. Although the DAN is able to correct for some large-scale bias within the learned models, it is not a panacea.

*Forced cough experiments*: Beyond simply collecting more data for deeper models, there are interesting avenues for new kinds of data collection, such as studying whether a *forced* cough contains the same pulmonary information as a natural cough. The data collection process for natural coughs is time-consuming and uncertain; some patients cough over a hundred times per hour, whereas others cough merely once or twice an hour. Further investigation is warranted to force users to cough on-demand, creating an acoustic event that may contain diagnostically relevant information, and to see if machine learning models can be trained upon such a dataset. This would obviate the need for patient screening that entails an hour-long process waiting for a natural cough to occur.

*Comparison with other pulmonary ailments*: Another area of investigation is to collect datasets from a wider range of pulmonary ailments such as pneumonia and cystic fibrosis to determine the ability of the machine learning model to disambiguate between multiple sputum-generating diseases. Different pulmonary ailments affect different areas of the pulmonary system in different ways, generating sputum in different areas of the pulmonary system. We hypothesize that the acoustic signature of coughs stemming from lower respiratory infections versus diseases that effect higher areas of the lungs differ sufficiently that automated cough classification systems could be built to detect them.

## 5 CONCLUSION

In conclusion, we have detailed the design of novel cough detection and classification algorithms that enable new sensing modalities and applications for automated cough counting and classification. We have shown that our algorithms are both accurate and efficient enough to be deployed onto the mobile computing platforms that are becoming ever more prevalent and pervasive in daily life. These new models enable new sensing modalities and preserve user privacy, performing all computation locally and not requiring the storage or transmission of raw audio from the user's device. We demonstrate the application of cutting edge machine learning techniques to train machine learning models to perform difficult tasks while compensating for gross dataset imbalance.

## REFERENCES

[1] 2014. Darknet. (2014). http://pjreddie.com/darknet
[2] 2018. Amazon Echo. (May 2018). https://developer.amazon.com/echo
[3] 2018. Raspberry Pi. (May 2018). https://www.raspberrypi.org/products/
[4] Udantha R. Abeyratne, Vinayak Swarnkar, Amalia Setyati, and Rina Triasih. 2013. Cough Sound Analysis Can Rapidly Diagnose Childhood Pneumonia. *Annals of Biomedical Engineering* 41, 11 (Nov. 2013), 2448–2462. DOI: http://dx.doi.org/10.1007/s10439-013-0836-0
[5] Mahmood Al-khassaweneh and Ra'ed Bani Abdelrahman. 2013. A signal processing approach for the diagnosis of asthma from cough sounds. *Journal of Medical Engineering & Technology* 37, 3 (April 2013), 165–171. DOI: http://dx.doi.org/10.3109/03091902.2012.758322

[6] Justice Amoh and Kofi Odame. 2013. Technologies for developing ambulatory cough monitoring devices. *Critical Reviews™ in Biomedical Engineering* 41, 6 (2013). http://www.dl.begellhouse.com/journals/4b27cbfc562e21b8,179a953a4da02786,60a9081a3e2a5c35.html

[7] J. Amoh and K. Odame. 2016. Deep Neural Networks for Identifying Cough Sounds. *IEEE Transactions on Biomedical Circuits and Systems* 10, 5 (Oct. 2016), 1003–1011. DOI: http://dx.doi.org/10.1109/TBCAS.2016.2598794

[8] Yusuf A. Amrulloh, Udantha R. Abeyratne, Vinayak Swarnkar, Rina Triasih, and Amalia Setyati. 2015. Automatic cough segmentation from non-contact sound recordings in pediatric wards. *Biomedical Signal Processing and Control* 21 (Aug. 2015), 126–136. DOI: http://dx.doi.org/10.1016/j.bspc.2015.05.001

[9] Samantha J. Barry, Adrie D. Dane, Alyn H. Morice, and Anthony D. Walmsley. 2006. The automatic recognition and counting of cough. *Cough* 2 (2006), 8. DOI: http://dx.doi.org/10.1186/1745-9974-2-8

[10] Antony Barton, Patrick Gaydecki, Kimberley Holt, and Jaclyn A Smith. 2012. Data reduction for cough studies using distribution of audio frequency content. *Cough (London, England)* 8 (Dec. 2012), 12. DOI: http://dx.doi.org/10.1186/1745-9974-8-12

[11] Antony James Barton. 2013. Signal Processing Techniques for Data Reduction and Event Recognition in Cough Counting. (Nov. 2013). https://www.escholar.manchester.ac.uk/uk-ac-man-scw:213303

[12] J. Bezanson, A. Edelman, S. Karpinski, and V. Shah. 2017. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.* 59, 1 (Jan. 2017), 65–98. DOI: http://dx.doi.org/10.1137/141000671

[13] S. S. Birring, T. Fleming, S. Matos, A. A. Raj, D. H. Evans, and I. D. Pavord. 2008. The Leicester Cough Monitor: preliminary validation of an automated cough detection system in chronic cough. *The European Respiratory Journal* 31, 5 (May 2008), 1013–1018. DOI: http://dx.doi.org/10.1183/09031936.00057407

[14] G. H. R. Botha, G. Theron, R. M. Warren, M. Klopper, K. Dheda, P. D. van Helden, and T. R. Niesler. 2018. Detection of tuberculosis by automatic cough sound analysis. *Physiological Measurement* 39, 4 (April 2018), 045005. DOI: http://dx.doi.org/10.1088/1361-6579/aab6d0

[15] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (Oct. 2001), 5–32. DOI: http://dx.doi.org/10.1023/A:1010933404324

[16] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. *arXiv:1512.01274 [cs]* (Dec. 2015). http://arxiv.org/abs/1512.01274 arXiv: 1512.01274.

[17] François Chollet. 2016. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv:1610.02357 [cs]* (Oct. 2016). http://arxiv.org/abs/1610.02357 arXiv: 1610.02357.

[18] T. Cover and P. Hart. 2006. Nearest Neighbor Pattern Classification. *IEEE Trans. Inf. Theor.* 13, 1 (Sept. 2006), 21–27. DOI: http://dx.doi.org/10.1109/TIT.1967.1053964

[19] S. Davis and P. Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28, 4 (Aug. 1980), 357–366. DOI: http://dx.doi.org/10.1109/TASSP.1980.1163420

[20] T. Drugman, J. Urbain, N. Bauwens, R. Chessini, C. Valderrama, P. Lebecque, and T. Dutoit. 2013. Objective Study of Sensor Relevance for Automatic Cough Detection. *IEEE Journal of Biomedical and Health Informatics* 17, 3 (May 2013), 699–707. DOI: http://dx.doi.org/10.1109/JBHI.2013.2239303

[21] Donald Enarson. 2013. *Respiratory diseases in the world: realities of today - opportunities for tomorrow.* European Respiratory Society, Sheffield. OCLC: 868996973.

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[23] Marti A. Hearst. 1998. Support Vector Machines. *IEEE Intelligent Systems* 13, 4 (July 1998), 18–28. DOI: http://dx.doi.org/10.1109/5254.708428

[24] J. Y. Hsu, R. A. Stone, R. B. Logan-Sinclair, M. Worsdell, C. M. Busst, and K. F. Chung. 1994. Coughing frequency in patients with persistent cough: assessment using a 24 hour ambulatory recorder. *European Respiratory Journal* 7, 7 (July 1994), 1246–1253. http://erj.ersjournals.com/content/7/7/1246

[25] Mike Innes. 2018. Flux: Elegant Machine Learning with Julia. *Journal of Open Source Software* (2018). DOI: http://dx.doi.org/10.21105/joss.00602

[26] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]* (Feb. 2015). http://arxiv.org/abs/1502.03167 arXiv: 1502.03167.

[27] Richard S. Irwin, Michael H. Baumann, Donald C. Bolser, Louis-Philippe Boulet, Sidney S. Braman, Christopher E. Brightling, Kevin K. Brown, Brendan J. Canning, Anne B. Chang, Peter V. Dicpinigaitis, Ron Eccles, W. Brendle Glomb, Larry B. Goldstein, LeRoy M. Graham, Frederick E. Hargreave, Paul A. Kvale, Sandra Zelman Lewis, F. Dennis McCool, Douglas C. McCrory, Udaya B. S. Prakash, Melvin R. Pratter, Mark J. Rosen, Edward Schulman, John Jay Shannon, Carol Smith Hammond, Susan M. Tarlo, and American College of Chest Physicians (ACCP). 2006. Diagnosis and management of cough executive summary: ACCP evidence-based clinical practice guidelines. *Chest* 129, 1 Suppl (Jan. 2006), 1S–23S. DOI: http://dx.doi.org/10.1378/chest.129.1_suppl.1S

[28] F. Itakura. 1975. Line spectrum representation of linear predictor coefficients of speech signals. *The Journal of the Acoustical Society of America* 57, S1 (April 1975), S35–S35. DOI: http://dx.doi.org/10.1121/1.1995189

[29] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. 2017. Neural style transfer: A review. *arXiv preprint arXiv:1705.04058* (2017).

[30] Jong Hwan Ko, Josh Fromm, Matthai Philipose, Ivan Tashev, and Shuayb Zarar. 2017. Precision Scaling of Neural Networks for Efficient Audio Processing. *arXiv:1712.01340 [cs, eess]* (Dec. 2017). http://arxiv.org/abs/1712.01340 arXiv: 1712.01340.

[31] J. Korpáš, J. Sadloňová, and M. Vrabec. 1996. Analysis of the cough sound: an overview. *Pulmonary pharmacology* 9, 5 (1996), 261–268. http://www.sciencedirect.com/science/article/pii/S0952060096900344

[32] Anders Krogh and John A. Hertz. 1992. A Simple Weight Decay Can Improve Generalization. In *Advances in Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson, and R. P. Lippmann (Eds.). Morgan-Kaufmann, 950–957. http://papers.nips.cc/paper/563-a-simple-weight-decay-can-improve-generalization.pdf

[33] Eric C. Larson, TienJui Lee, Sean Liu, Margaret Rosenfeld, and Shwetak N. Patel. 2011. Accurate and privacy preserving cough sensing using a low-cost microphone. In *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 375–384. http://dl.acm.org/citation.cfm?id=2030163

[34] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521 (27 05 2015), 436 EP –. http://dx.doi.org/10.1038/nature14539

[35] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. 1990. Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in Neural Information Processing Systems 2*, D. S. Touretzky (Ed.). Morgan-Kaufmann, 396–404. http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf

[36] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network In Network. *arXiv:1312.4400 [cs]* (Dec. 2013). http://arxiv.org/abs/1312.4400 arXiv: 1312.4400.

[37] Robert G. Loudon and Linda C. Brown. 1967. Cough Frequency in Patients with Respiratory Disease. *American Review of Respiratory Disease* 96, 6 (Dec. 1967), 1137–1143. DOI: http://dx.doi.org/10.1164/arrd.1967.96.6.1137

[38] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier Nonlinearities Improve Neural Network Acoustic Models. *International Conference on Machine Learning* (2013), 6.

[39] S. Matos, S. S. Birring, I. D. Pavord, and H. Evans. 2006. Detection of cough signals in continuous audio recordings using hidden Markov models. *IEEE Transactions on Biomedical Engineering* 53, 6 (June 2006), 1078–1083. DOI: http://dx.doi.org/10.1109/TBME.2006.873548

[40] K. McGuinness, K. Holt, R. Dockry, and J. Smith. 2012. P159 Validation of the VitaloJAK™ 24 Hour Ambulatory Cough Monitor. *Thorax* 67, Suppl 2 (Dec. 2012), A131–A131. DOI: http://dx.doi.org/10.1136/thoraxjnl-2012-202678.220

[41] J. Monge-Alvarez, C. Hoyos-Barcelo, P. Lesso, and P. Casaseca-de-la Higuera. 2018. Robust Detection of Audio-Cough Events using local Hu moments. *IEEE Journal of Biomedical and Health Informatics* (2018), 1–1. DOI: http://dx.doi.org/10.1109/JBHI.2018.2800741

[42] Vladyslav Nikolayevskyy, Paolo Miotto, Edita Pimkina, Yanina Balabanova, Irina Kontsevaya, Olga Ignatyeva, Alessandro Ambrosi, Girts Skenders, Arvydas Ambrozaitis, Alexander Kovalyov, Anna Sadykhova, Tatiana Simak, Andrey Kritsky, Svetlana Mironova, Olesya Tikhonova, Yulia Dubrovskaya, Yulia Rodionova, Daniela Cirillo, and Francis Drobniewski. 2015. Utility of propidium monoazide viability assay as a biomarker for a tuberculosis disease. *Tuberculosis* 95, 2 (March 2015), 179–185. DOI: http://dx.doi.org/10.1016/j.tube.2014.11.005

[43] D. O'Shaughnessy. 1988. Linear predictive coding. *IEEE Potentials* 7, 1 (Feb. 1988), 29–32. DOI: http://dx.doi.org/10.1109/45.1890

[44] R. D. Patterson, Ian Nimmo-Smith, John Holdsworth, and Peter Rice. 1987. An efficient auditory filterbank based on the gammatone function. In *a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, Vol. 2. http://www.academia.edu/download/42900653/An_efficient_auditory_filterbank_based_o20160221-23516-9rsfvo.pdf

[45] Juergen Schmidhuber. 2015. Deep Learning in Neural Networks: An Overview. *Neural Networks* 61 (Jan. 2015), 85–117. DOI: http://dx.doi.org/10.1016/j.neunet.2014.09.003 arXiv: 1404.7828.

[46] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]* (Sept. 2014). http://arxiv.org/abs/1409.1556 arXiv: 1409.1556.

[47] Arietta Spinou and Surinder S. Birring. 2014. An update on measurement and monitoring of cough: what are the important study endpoints? *Journal of Thoracic Disease* 6, 7 (Oct. 2014), S728–S734. http://jtd.amegroups.com/article/view/3346

[48] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. (????), 30. https://dl.acm.org/citation.cfm?id=2670313

[49] S. Subburaj, L. Parvez, and T. G. Rajagopalan. 1996. Methods of Recording and Analysing Cough Sounds. *Pulmonary Pharmacology* 9, 5 (Oct. 1996), 269–279. DOI: http://dx.doi.org/10.1006/pulp.1996.0035

[50] Vinayak Swarnkar, Udantha R. Abeyratne, Anne B. Chang, Yusuf A. Amrulloh, Amalia Setyati, and Rina Triasih. 2013. Automatic Identification of Wet and Dry Cough in Pediatric Patients with Respiratory Diseases. *Annals of Biomedical Engineering* 41, 5 (May 2013), 1016–1028. DOI: http://dx.doi.org/10.1007/s10439-013-0741-6

[51] Luke Taylor and Geoff Nitschke. 2017. Improving Deep Learning using Generic Data Augmentation. *arXiv:1708.06020 [cs, stat]* (Aug. 2017). http://arxiv.org/abs/1708.06020 arXiv: 1708.06020.

[52] T. Tieleman and G. Hinton. 2012. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning. (2012).

[53] B. H. Tracey, G. Comina, S. Larson, M. Bravard, J. W. López, and R. H. Gilman. 2011. Cough detection algorithm for monitoring patient recovery from pulmonary tuberculosis. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 6017–6020. DOI: http://dx.doi.org/10.1109/IEMBS.2011.6091487

[54] World Health Organization. 2017. The top 10 causes of death. (Jan. 2017). http://www.who.int/mediacentre/factsheets/fs310/en/

[55] Mingyu You, Zeqin Liu, Chong Chen, Jiaming Liu, Xiang-Huai Xu, and Zhong-Min Qiu. 2017. Cough detection by ensembling multiple frequency subband features. *Biomedical Signal Processing and Control* 33 (March 2017), 132–140. DOI: http://dx.doi.org/10.1016/j.bspc.2016.11.005

Received May 2018; accepted June 2018