Audio-Based Cough Detection in Clinic Waiting Rooms

Yumna Anwar, Sean M. Mullan, Octav Chipara, Alberto M. Segre, Philip Polgreen
Department of Computer Science
University of Iowa
Iowa City, United States of America
{yumna-anwar, sean-mullan, octav-chipara, alberto-segre@uiowa.edu, philip-polgreen}@uiowa.edu

***Abstract*—Automated cough detection has significant applications for the surveillance of diseases and supports medical decisions, as cough sounds can be a useful biomarker. However, the implementation and evaluation of robust cough detection models can be challenging due to the lack of real-world data. This paper introduces and makes available a collection of 2,883 coughs and 3,074 non-cough sounds recorded in clinic waiting rooms that we hope will become a baseline for this task. Using this dataset, we evaluate different convolutional network architectures for classifying short audio segments as cough or non-cough. An ensemble model of convolutional neuronal networks provides the most robust performance and has a ROC AUC of 98.1%. Equally important, we construct a cough counter that incorporates the ensemble model to compute the number of coughs per day. Then, a simple linear model estimates the number of visits in which the patients report cough symptoms from the cough counts. This simple regression model can predict the number of cough visits in the clinic with an absolute mean error of 4.26 cough visits per day. Using additional information about when patients are in the clinic helps a similar regression model reach a mean absolute error of 3.65 cough visits per day. These results demonstrate the feasibility of using cough detection as a biomarker for the spread of respiratory viruses within the community.**

***Index Terms*—Cough detection, Convolutional neural networks**

I. Introduction

Motivation. Coughing is one of the most common causes of primary healthcare visits. A wide range of diseases is associated with coughing, including asthma, gastroesophageal reflux disease, and some types of cancer. However, most new-onset coughs are caused by respiratory infections, most of which are easily transmitted to other individuals. Because of the importance of cough as a symptom of various diseases, a significant effort has focused on studying cough timing, intensity, and other aspects [1]. While some approaches involve patients keeping a cough diary, several approaches have instead performed automated cough counting using a variety of computational approaches to analyze audio recordings of coughing. Examples include detecting nighttime coughing as a marker for uncontrolled asthma and monitoring the success of tuberculosis treatment over time [2]. Other approaches have focused on the characteristics of a particular cough as a diagnostic aid, for example, using machine learning approaches to differentiate croup [3], whooping cough [4], or helping to diagnose COVID-19 [5], [6]. Indeed, the proliferation of mobile computing devices has made such analyses easier at an individual level.

Beyond coughing at an individual level, however, coughing also plays an essential role at a community or population level. Because coughing can distribute infectious droplets over relatively long distances, coughing is an effective mechanism for accelerating the transmission of respiratory infections. Thus, several contagious respiratory infections, including tuberculosis, pertussis, influenza, and respiratory syncytial virus (RSV), can spread through coughing. These diseases have associated morbidity and mortality and are therefore important from a public health perspective. Indeed, the current COVID-19 pandemic caused by the SARS-COV2 virus illustrates how rapidly infectious diseases can spread and highlights the value of automated cough detection, especially in indoor settings.

Outpatient clinic waiting rooms are particularly important environments in which to study coughing. The waiting rooms of primary or urgent care clinics are where infectious patients may first mix with those who have non-infectious complaints (e.g., well-child visits, musculoskeletal complaints, allergies, or dermatologic issues). Prior work has highlighted the importance of considering such transmission opportunities. For example, it has been established that the exposure associated with well-child visits increases infection risk among other family members [7]. Thus, detecting coughs within waiting rooms can inform interventions such as the deployment of universal droplet precautions, including requiring masking or helping to isolate coughing individuals in separate waiting rooms.

A second critical need for cough detection in waiting rooms is the opportunity to do real-time fine-grained syndromic surveillance. Current influenza and COVID-19 surveillance rely upon reports from healthcare providers, and in the case of influenza, reports lag current conditions by two weeks. A wide range of influenza (and now COVID-19) surveillance data are becoming available much faster than previously. However, most of these data are reported at the national, state, or county level. In addition, surveillance is usually testing-centric – if tests are not available or ordered, or if home testing is prevalent, then cases will not be reported. Thus, there is a need for new disease-agnostic surveillance approaches. Counting

This work is partly supported by NSF grants IIS-1838830 and CNS-1750155.

coughs in patient waiting rooms is one such approach that could work for COVID-19, influenza, RSV, or some new yet unknown respiratory pathogen. Furthermore, the techniques developed and validated for outpatient clinic rooms are likely to generalize to other settings such as schools or workplaces.

Technical Challenges. The development of disease surveillance tools based on cough detection from audio recordings faces several technical challenges. In realistic and open environments, numerous sounds have similar characteristics to coughs and may be misclassified as such. Examples of such confounders include throat clearing, laughing, doors closing, or background music (these sounds are similar to coughs in that they include high-energy peaks). Another critical challenge is that coughs infrequently occur even in outpatient clinics. Therefore, a cough detector must have very high specificity to avoid classifying other sounds as coughs. The detector should also have good sensitivity to recall a majority of coughs, although we may sacrifice some sensitivity in exchange for higher specificity. Another consequence of the low incidence of coughs is that realistic datasets are highly imbalanced, complicating the construction and training of accurate cough detectors. Finally, cough detectors must make robust predictions in a broad range of environments and be robust to hardware variation. Even microphones from the same manufacturer have noticeably different recording characteristics. We would also like our cough detectors to operate equally well in quick-care clinics, emergency waiting rooms, and schools despite differences in their acoustic characteristics. Finally, most coughs occur in the far-field, and, as a result, they are more prone to degradation due to interference from other concurrent sounds.

Limitations of Existing Datasets. One major impediment to progress in automated cough detection is the inadequacy of public datasets. Much prior work has relied on datasets that are not publicly available (e.g., [8], [9]). Unfortunately, publicly available datasets, such as the Augmented Multi-party Interaction (AMI) corpus [10], Audio Set [11], and ESC50 [12], are in general quite limited in terms of size and are sparsely annotated: specifically, they often lack fine-grained annotations identifying the temporal span of each cough. These kinds of annotations are essential to building effective cough detectors, and they are just not available for Audio Set, ESC50, and AMI, which only provide labels at the granularity of audio clips, which can range from a few seconds to minutes in length. Recently, researchers have re-annotated the AMI corpus with fine-grained annotations for each cough [13]. However, the number of coughs in the public dataset remains small: the ESC50, Audio Set, and AMI contain only 40, 871, and 1116 coughs, respectively.

Contributions. This paper makes three specific contributions. First, we have created and now make publicly (https://msl.cs.uiowa.edu/project_cough.html) available an annotated dataset of involuntary coughs recorded in working real-world outpatient clinical settings. The dataset consists of 5,957 3-second clips. We have inspected each clip to ensure no personally identifiable information is included. A total of 2,883 clips contained 3,110 coughs, each annotated with their start and end times. The remaining 3,074 clips were selected to capture the challenges of building robust cough predictors. We hope that releasing the dataset will help advance research on this topic and become a standard benchmark for the community.

Second, we develop a robust and effective cough detector that uses deep learning techniques to identify coughs from audio recordings. The resulting detector is an ensemble of several deep learning models that use different architectures and modeling assumptions. We have validated the performance of our cough detector on a separate and even larger dataset containing 348 hours of audio collected under similar operational conditions (i.e., quick care clinics).

Third, we used our cough detector as the basis for a cough counter, and then used the output of the cough counter as the input to a linear regression model that estimates the number of patients who report cough as a symptom during their clinic visit. A naive model that would estimate the number of daily cough visits as the average number of days in the training set has a mean absolute error of 5.3 cough visits per day. A simple regression model that uses the cough counts reduces the mean absolute error to 4.26 cough visits per day. Using additional information about clinic hours and patient scheduling practices helps a similar regression model further reduce the mean absolute error to 3.7 cough visits per day. This result demonstrates the feasibility of building surveillance tools based on cough detectors and their potential use in other public settings such as schools or theaters where such validation data are unavailable.

II. Related work

This section compares prior work on cough detection on three dimensions: whether the operating environment is realistic, which machine learning techniques are used, and whether/how the resulting cough detectors are useful for disease surveillance. Table I summarizes prior work across all three dimensions and includes information about their performance. Note, however, that direct comparisons between different approaches based on reported performance may be misleading, as the performance was not measured on a standard dataset.

Operating Environment: A significant fraction of prior work relies on data that does not realistically capture the challenge of cough detection for disease surveillance. Early work on cough detection used voluntary coughs produced deliberately by participants on demand [15], [16]. It is unlikely that this approach can capture the natural diversity of real coughs and the significant differences across subjects. Similarly, many of the collected datasets were recorded in strictly controlled environments with little background noise. For example, Barata et al. [15] performed data collection in a laboratory setting where a microphone was placed 15 cm from the subject.

Other prior work relies on data collected from microphones worn directly by subjects [14], [16]. An advantage of this approach is that coughs tend to be louder and less susceptible

	Data	Audio environment	Subjects	Features	Size	Approach	Sens (%)	Spec (%)	PPV (%)	NPV (%)
Matos et al., 2006 [14]	involuntary	Microphone on chest during daily routine work	19	MFCC	821 min and 2473 cough signals	HMM	82			
Barata et al., 2019 [15]	voluntary	Cough, laughter, throat clearing, speech, forced expiration gathered in lab	43	mel-scaled spectrogram	6737 cough, 3985 laugh, 3695 throat, 731 speech, 443 expiration	CNN	91.7	90.1	92	89.5
Amoh et al., 2016 [16]	voluntary	Cough, breathing, heartbeats, cracklings and other from chest worn sensor	14	STFT	627 cough instances, of average 320ms	CNN, RNN	87.7	92.7		
Monge et al., 2018 [17]	involuntary	Cough, and other sounds from smartphone mic with participants performing task and with different background noise	13	MFCC and local Hu moments	Database of 1560 minutes with 5-18% cough samples	KNN*, SVM	88.5	99.77		
Imran et al., 2020 [6]	involuntary	ESC-50 dataset and self recorded sounds	-	MFCC	1838 cough sounds and 3597 non-cough	CNN	96.01	95.19		
Hossain et al., 2020 [8]	involuntary	Cough and other natural noises from public datasets. Evaluated on audio data from hospital waiting rooms	-	MFCC and Spectrogram	462220 seconds of audio samples. Evaluated on 2500 seconds samples from hospital waiting	CNN			85.4	84.5

TABLE I: Overview of the related literature on audio based automated cough detection models

to interference from other sounds due to the microphone's proximity to the subject. On the other hand, subjects must be willing to wear the devices consistently, which, in our experience, may engender compliance issues. Other researchers used mobile apps to collect sound recordings of coughs to identify whether a subject has COVID-19 (e.g., covid-19-sounds.org, [18]). However, these recordings also have different audio characteristics than those encountered in more passive disease surveillance contexts: the coughs are primarily voluntary and subject to less interference since subjects cough in proximity to their device. In contrast, FluSense [8] relies on recordings made in hospital waiting rooms to create a realistic dataset, but these data are not publicly available. The work reported here, like FluSense, relies on data collected by recording devices placed in real-world contexts: unlike FluSense, however, our extensive, annotated dataset is now available to all.

Machine Learning Techniques: All existing cough predictors tend to build models using similar features extracted from an audio signal, including Mel-Frequency Cepstral Coefficients (MFCC), Mel-scaled spectrogram, and Short-Time Fourier Transform (STFT). The cough detection problem is easily modeled as a binary classification problem, where the system is trained to label a window of audio data as either including a cough or not including a cough. Prior work has applied various traditional machine learning algorithms, including, for example, K-Nearest Neighbor and Support Vector Machines (SVM) [17] as well as Hidden Markov Models (HMMs) [14]. More recently, deep neuronal architectures such as convolutional neuronal networks (CNNs) [6], [8], [15] and recurrent neuronal networks (RNNs) [16] have also been applied to this problem. In this work, we apply various CNN architectures (e.g., CNNs, residual CNNs, depth-wise CNNs) as well as ensembles of these methods and also explore different training techniques (e.g., transfer learning) to build an effective cough detector.

Applications: The closest related effort to our work is FluSense [8]. FluSense is a platform for disease surveillance that combines data from microphones and thermal imaging captured in waiting areas of a university hospital. To the best of our understanding, FluSense uses a CNN architecture to perform cough detection, and uses the output of its detector to predict the results of influenza laboratory tests. The cough detector relies on data from an array of microphones as well as data from thermal imaging sensors deployed in the same space. In contrast, we demonstrate that daily cough counts computed solely from audio data captured by a single waiting room microphone are correlated with the number of patients in the clinic that day who report cough as a medical complaint.

III. Deployment in Clinics

To evaluate the feasibility of using cough as a biomarker for disease surveillance, we have deployed our system in outpatient clinics in the Iowa City area from January 1st to March 31st, 2017. This period includes the preponderance of the 2017 flu season. Additionally, we obtained anonymized patient records detailing each visit to the outpatient clinic. The records include information regarding the diagnoses, symptoms reported by patients, and the start time and estimated duration of each visit.

Name	File duration	Total coughs events	Total files (duration in minutes)	Released
public	3 seconds	3,110	5,957 (297.85 minutes)	Yes
downtown	3 minutes	3,123	6,979 (20,937 minutes)	No
downtown*	3 minutes	unannotated	27,366 (82,098 minutes)	No

TABLE II: Summary of considered datasets. Note that a file may include several cough events.

Fig. 1: Recording device that integrates a Raspberry Pi, Blue Snowball USB microphone, and a real-time clock.

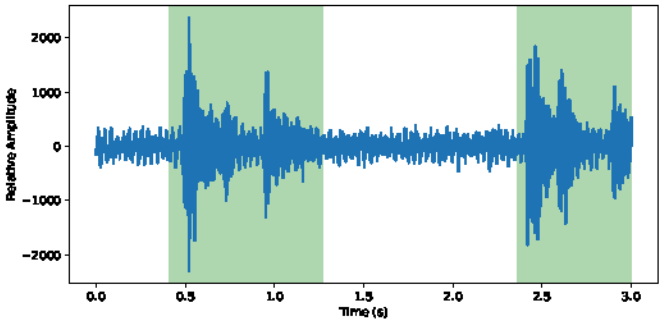


Fig. 2: Sample "hc/sample_hc_7484_74660_1.wav" from the public dataset. The highlighted green segments are the cough annotations. The first segment includes two plosives, while the second includes three plosives.

A. Methodology

Recording System. We have developed a system to record audio that incorporates a Raspberry Pi with an external Blue Snowball microphone (see Figure 1). In addition, we have integrated the Raspberry Pi with a real-time clock to obtain accurate timing information without requiring WiFi connectivity. The LED integrated into the power button was illuminated while recording audio. The recorded sounds are saved locally on a flashcard and downloaded periodically for archival and analysis. The audio was recorded at 44 KHz and saved uncompressed in WAV format.

Deployment. We deployed our system in four outpatient clinics in Iowa City, but in this paper, we focus on the data collected from one of these clinics that we will refer to as downtown*. We record the sounds from the waiting rooms from 8 AM to 6 PM. Patients who arrive at a clinic wait in a waiting room until medical personnel sees them. The acoustic environment of waiting areas is dynamic. It is common for music to be played in the background. Furthermore, patients speak with the nursing staff, the clerk, and each other. As the clerk manages the intake and discharge of patients, they commonly use staplers which can be a confounder. Other potential confounders include laughing, throat clearing, music, people talking, door slamming, and kids crying. Each clinic has unique acoustic characteristics. When a patient coughs, they are usually located several meters away from the microphone and at varying angles. Most of the coughs occur in the microphone's far-field. Consequently, they are soft and subject to interference from background music, speech, and other concurrent sounds.

Annotations. An involuntary cough starts with an inspiration, followed by a forced expiratory effort against the closed glottis, and ends with the glottis opening and rapid expiratory airflow [19]. The most notable aspect of coughing is its plosive phase, during which air is expelled. A common approach to counting coughs involves identifying each plosive cough sound. Unfortunately, labeling plosives is tedious when a subject coughs several times in quick succession. Sequences of coughs in quick succession are common in the data we captured in outpatient clinics. Instead of labeling plosives, we label coughs occurring in quick succession as a contiguous segment. An advantage of this approach is that it lowers the annotation burden as it is not necessary to make fine-grained judgments of when each plosive occurs. We instructed the annotators to label coughs as a contiguous segment when they are separated by less than 300 milliseconds. To illustrate these challenges, in Figure 2, we plot the waveform of a file from the public database. In this example, we label two segments as "coughs." The first segment involves two plosives in quick succession, while the second involves three. Figure 3 plots the distribution of cough annotation duration in seconds for all the annotated coughs in the dataset. The length of the coughs ranged from 0.1 seconds to 2.6 seconds. Segments exceeding 300 ms involve multiple coughs occurring in quick succession.

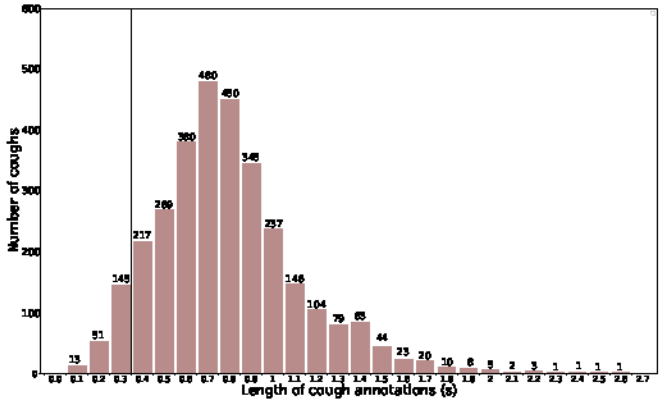


Fig. 3: Distribution of annotated cough duration in the `public` dataset

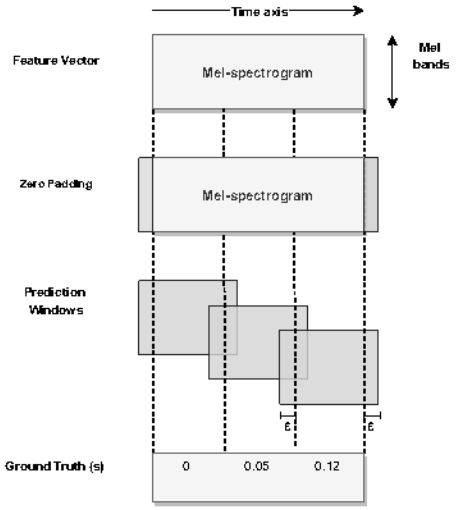


Fig. 4: Data processing pipeline

Datasets. We consider the following datasets: `downtown*`, `downtown`, and `public` (see Table II). The `downtown*` includes all the audio data captured in one of the clinics. A subset of the `downtown*` data was annotated, and we will refer to it as `downtown`. The audio files in `downtown*` and `downtown` are 3 minutes long. We used `downtown` data to train an initial CNN architecture similar to the one described in Section IV-B. The trained model was used to predict the likelihood that a sound segment included a cough. We computed the difference between the true and predicted labels for each segment. The `public` dataset was generated by sampling uniformly to extract 2,883 files that included coughs and 3,074 files that included non-coughs. The segments included in the `public` dataset are 3 seconds long. The sampling process ensured that the `public` dataset includes diverse sound segments, including some that are difficult to classify. We initially copied the labels from the `downtown` dataset to the `public` dataset. We then performed another round of annotation to ensure the correction of annotations and excluded any sound files that included personal information. The `public` dataset is available at https://msl.cs.uiowa.edu/project_cough.html.

IV. CNN Models

We model the problem of cough detection as a binary classification problem where an audio segment is classified as "cough" or "non-cough". We will consider different convolutional network architectures to identify which one performs the best for this problem. Our focus on CNNs is motivated by their state-of-the-art performance in many computer vision and audio processing tasks.

A. Feature extraction

The steps of our data processing pipeline are shown in Figure 4. Each audio file is divided into non-overlapping windows of one second in preparation for classification. Each window's label is computed by considering the maximum overlap between the window and all annotations. If the overlap exceeds a threshold value (which we set to 0.1), the window is labeled as "cough." Otherwise, windows are labeled as "non-cough." We can improve the classification accuracy by providing a classifier with a slightly larger wider window for classification. To this end, each prediction window is enlarged by ϵ, whose value is configured during hyperparameter search.

For each window, we perform the following feature extraction steps. First, we downsample the files from 44KHz to 16KHz. Next, the short-term Fourier transform is computed over windows of 512 samples with an increment of 256 samples and uses a Hann window. We then compute several Mel-scale spectrograms and Mel-Frequency Cepstral Coefficients. Finally, the computed features of each file are normalized to improve generalizability. The input to all the CNN networks has three dimensions: time, features extracted from the frequency domain, and two channels (i.e., MFCCs and Mel-scale spectrogram). We tuned the input size to maximize the performance for each network.

B. Convolutional Neural Networks (CNNs)

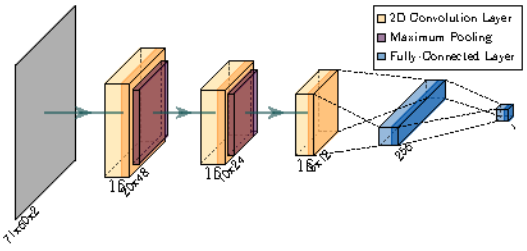


Fig. 5: CNN architecture based on by FluSense [8]

A traditional convolutional network consists of convolutional layers, pooling layers, and fully connected layers. FluSense [8] proposes a CNN architecture for classifying coughs (see Figure 5). The proposed architecture includes three convolution layers with kernel sizes of 20×48, 10×24, and 5×12. The number of channels at each layer is maintained

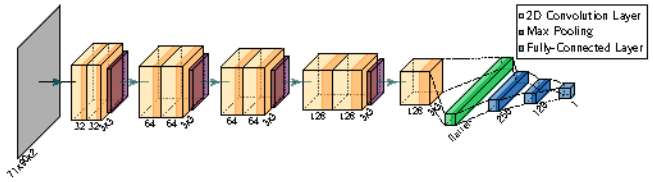


Fig. 6: CNN architecture whose depth and size of fully connected layers was optimized.

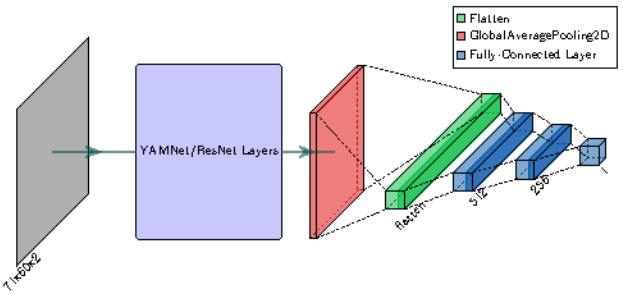


Fig. 7: Architecture based on YAMNet and ResNet

to be 16. A 2×2 max-pooling layer follows each convolutional layer. After the last convolutional layer, the output is flattened and fed into a fully connected layer. Group normalization is applied to improve performance. Finally, a dropout layer is applied after the final dense layer to prevent overfitting.

Inspired by more recent network designs, we experimented with a CNN architecture that uses smaller kernels and is deeper. We considered an architecture search where we started with an initial layer with a 3×3 convolutional kernel and 32 channels. Like ResNet and other CNNs, the subsequent layers include a pair of convolutional blocks that double the number of channels from the previous layer and reduce the spatial resolution in half. The output of the last convolutional layer is flattened and fed through two fully connected layers. We optimized the depth of the network and the size of the fully connected layer. The best-identified architecture is shown in Figure 6.

C. Transfer learning

We also used – YAMNet and ResNet [20] – two widely used architectures in sound classification. YAMNet trains the Mobilenet architecture [21] using the Audio Set corpus. The unique feature of Mobilenet is its use of depth-wise separable convolutions, which can be more computationally efficient than standard convolutions. ResNet is a residual neuronal network designed to address the challenge of training deeper networks. The critical insight idea is to reformulate the training problem to allow networks to skip layers via skip connections. We have adapted both network architectures to our design by adding two fully connected layers at the end (see Figure 7). A drop layer follows each fully connected layer to avoid overfitting.

We considered two approaches to training the networks. First, we used transfer learning by downloading the pre-trained models and training only the parameters of the newly added fully connected layers. Alternatively, we retrained the models from scratch. We have optimized the network architecture to determine the size of the fully connected layers. Additionally, we have also optimized the depth of the YAMNet network.

D. Model Ensembles

A common technique to improve the performance of a classifier is to create an ensemble of several models. We have created an ensemble that incorporates the best trained CNN and the retrained YAMNet and ResNet models. Each model's independent predictions were fed into a single fully connected layer whose size was configured during training. The output of the fully connected layer is a weighted combination of the outputs of each model.

V. Evaluating Cough Prediction Accuracy

This section evaluates the performance of the considered CNN architectures. Using these data, we would like to answer the following questions:

1) Can we develop a cough classifier that predicts whether short real-world audio segments contain coughs with high specificity and good sensitivity?
2) If so, what CNN architectures are most effective?
3) Can the number of visits during which patients report cough as a symptom be estimated from cough predictions?

To this end, we divide the `public` dataset into five folds to perform cross-validation. We observed a significant variation in the sound characteristics of different days, even in the same clinic. Accordingly, we stratified the samples to ensure that the same-day recordings were either in training or the testing set (but not in both). Additionally, construct the five folds to have a similar number of coughs. All models were trained using the Adam optimizer and a cosine-decay learning schedule. We use the binary cross-entropy as a loss function. ReLU is used as the activation function in all but the final layer. The final layer uses a softmax activation function to output the probability of the binary class (cough and not cough).

We use Optuna [22] to optimize hyperparameters and network architectures. Table IV summarizes the configured parameters. Optuna uses a Tree-structured Parzen Estimator to search this large parameter space efficiently. We also use early stopping to terminate the training process when the error rates do not improve for several epochs.

We evaluate the performance of the models according to the following metrics. AUC is the area under the Receiver Operating Characteristic (ROC) curve. An AUC of 50% can be achieved by random guessing, while an AUC of 100% is achieved by a perfect classifier. Since the `public` dataset is balanced, we focus on maximizing the area under the ROC curve on this dataset. Our primary method for handling

Model	ROC-AUC mean $\pm\sigma$	MCC mean $\pm\sigma$	Specificity 97(%) SEN mean $\pm\sigma$	95(%) SEN mean $\pm\sigma$	90(%) SEN mean $\pm\sigma$	87(%) SEN mean $\pm\sigma$
CNN	96.99 ± 0.37	0.84 ± 0.02	72.57 ± 3.06	83.57 ± 3.401	93.81 ± 2.26	95.89 ± 1.81
FluSense	95.51 ± 0.70	0.78 ± 0.01	62.05 ± 6.66	76.43 ± 3.13	88.02 ± 2.33	91.87 ± 2.05
YAMNet (frozen)	95.52 ± 0.63	0.78 ± 0.02	62.75 ± 3.36	74.82 ± 3.89	87.88 ± 2.47	91.46 ± 2.69
YAMNet (retrained)	$\mathbf{97.42 \pm 0.39}$	$\mathbf{0.84 \pm 0.01}$	$\mathbf{79.22 \pm 3.54}$	$\mathbf{87.29 \pm 1.35}$	$\mathbf{94.17 \pm 1.07}$	$\mathbf{96.42 \pm 0.49}$
ResNet (frozen)	73.81 ± 5.90	0.37 ± 0.04	17.06 ± 8.24	23.75 ± 9.49	33.55 ± 10.27	37.87 ± 10.89
ResNet (retrained)	96.91 ± 0.55	0.84 ± 0.01	70.48 ± 6.85	84.00 ± 2.72	93.56 ± 1.91	95.57 ± 1.07
Aggregate	98.02 ± 0.36	0.87 ± 0.01	83.29 ± 3.80	90.94 ± 2.57	96.48 ± 0.54	97.82 ± 0.37
Ensemble	98.11 ± 0.38	0.87 ± 0.01	85.11 ± 3.77	91.67 ± 2.64	96.79 ± 0.82	97.79 ± 0.59

TABLE III: Prediction results on the `public` dataset. The results of the best individual model is highlighted in bold. The *aggregate* model is the mean predicted probability of CNN, YAMNet (retrained) and ResNet (retrained) for each audio segment. The Ensemble model is a single dense layer trained on the predicted probabilities of CNN, YAMNet (retrained) and ResNet (retrained).

label imbalance was to downsample the non-cough class while creating the `public` dataset from `downtown`. We also report the Matthews correlation coefficient (MCC) and the specificity at different sensitivity levels. This reflects the relative importance placed on minimizing false positives, i.e., incorrectly predicting a cough. We report the average and standard deviation across the five folds for all metrics.

Tuned parameters	Model
Prediction window overlap (ϵ)	All models except ensemble
Initial learning rate	All models
Batch size	All models
Weight of positive class	All models
Drop probabilities	All models except ensemble
Number of dense units	All models

TABLE IV: Tuned parameters

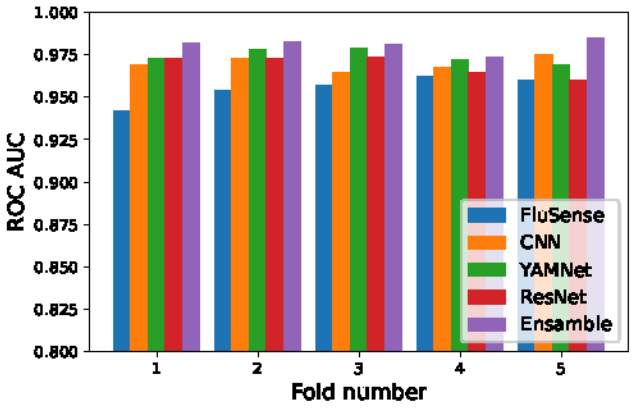


Fig. 8: AUC over the five folds.

A. Results on the `public` dataset

Table III includes the results of the best model found during hyperparameter search for each model architecture. The following discussion will focus on ROC AUC (or AUC, henceforth), but similar trends can be observed for the other metrics. Figure 8 plots the AUC for each model and validation fold.

The FluSense model performs relatively poorly on our dataset. The worse performance of FluSense can be attributed to the fact that the number of channels in its convolutional layers remains fixed as the spatial dimension is reduced. In contrast, our CNN architecture which increases the number of channels with the depth of the network, consistently has better performance than FluSense. Using the pre-trained YAMNet and ResNet50 also typically led to lower AUC values, particularly in the case of ResNet50. A possible explanation is that the final features produced by ResNet50 (optimized for image classification) are not effective in distinguishing between coughs and other sounds.

Retraining the YAMNet and ResNet50 produces better performance. Two factors contribute to these improvements. First, during the retraining, we optimize the depth of each of the networks, tailoring each architecture better to the amount of available data. Second, the parameters are trained using our data rather than another proxy dataset or tasks. As a result, the retrained YAMNet has the best average performance across all the considered metrics.

The best performance is obtained by combining the performance of different models. We consider two approaches to combine the predictions of models. The Aggregate model averages the predictions of the CNN, retrained ResNet, and retrained YAMNet. It achieves an average AUC of 98.02%, which improves over YAMNet's 97.42% average AUC. The Ensemble model includes a fully connected layer that outputs a weighted combination of the predictions of the individual models as the final prediction. Due to this additional flexibility, the Ensemble model further boosts the performance to an 98.11% AUC. As shown in Figure 8, the Ensemble model provides the best performance for all folds with slight variations across the folds. It is important to note that even though the AUC improvements may seem minor, they have a significant cumulative effect for a cough predictor. Since each model makes one prediction for each second during the working day, there are a total 32,400 of predictions per day.

Given the large number of predictions required, even small changes in AUC can lead to significant changes in the number of false positives and negatives.

B. Results on the downtown dataset

In the following, we evaluate how well the models trained on the public generalize on the larger downtown dataset. The downtown and public datasets share most of the same cough data. However, the downtown data includes a significant fraction of non-cough sounds from the downtown clinics[1]. The balance of labels in the downtown dataset is representative of a clinic's waiting room.

Due to the overlap in data between the public and downtown, we need to avoid making predictions on parts of files that were used during training, which would overestimate the performance of the models. We constructed five evaluation sets using the downtown data to avoid this situation. Each evaluation set e_i ($i = \{1, 2, 3, 4, 5\}$) is initialized to include all the files in downtown. Then, we consider iteratively the i-th folds of the public dataset that were used to train the models and remove from the evaluation set e_i all files that included data used for training in public. The statistics reported below are computed over the five evaluation sets.

Table V shows the performance of the best-performing models on the downtown dataset. The AUC results on the downtown dataset are slightly worse than those on the public dataset. The YAMNet remains the best performing single model, followed by our optimized, and ResNet, each providing average AUCs of 97.40%, 97.01%, and 96.95%, respectively. The Ensemble model achieved the best performance with an AUC of 97.81%. The reduction in the AUC can be attributed to a higher false-positive rate, which is partly expected given the significantly larger number of non-cough files considered. The increase in the false positive rate is responsible for reductions in Matthews Correlation Coefficients (MCCs). These results give us confidence that despite the significant variations in the audio environment of clinic rooms, it is possible to build an accurate cough detector. The Ensemble provides a specificity of nearly 80% with a sensitivity of 97%. Alternatively, a specificity of 88% may be achieved for a slightly lower sensitivity of 95%. In the remaining section, we will configure the cough classifier to maximize the specificity at 95%-sensitivity.

C. Predicting Cough Visits Per Day from Audio Recordings

Having demonstrated that a cough detector can make accurate predictions, we evaluate whether these cough predictions can be related to the clinic visits during which patients report having a cough. We have obtained a deidentified record of the patient visits from the clinic where we deployed the recording system. A visit record includes several valuable pieces of information: the chief complaint reported by the patient, provider comments, the start time of the visit, and an estimated duration for a visit. We define a visit as a "cough visit" if the word cough appears either in the chief complaint or the notes fields. Our goal is to predict the number of cough visits per day in the clinic solely from the audio recordings.

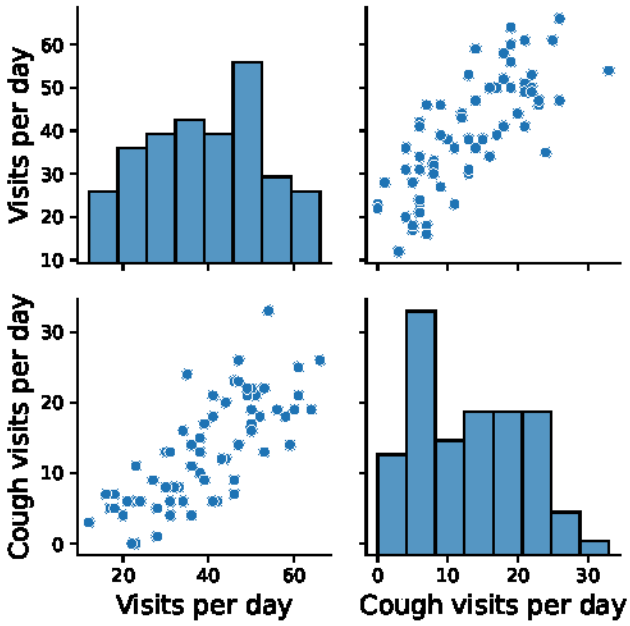


Fig. 9: The distribution of daily visits, cough visits, and their relationship.

We use the Ensemble model trained using the first fold to estimate the number of coughs daily. We have performed the following analysis with the other folds and obtained similar results. From all the available recordings, we consider the days that (1) have complete audio for the entire 10 hours workday and (2) were not used for training in the first fold. A total of 37 days between January 1st and March 31st meet this constraint. There were 2560 visits at the clinic, including 856 cough visits. There are an average of 39 visits per day (range 12 – 66) and an average of 13.06 cough visits per day (range 0 – 33). Figure 9 shows the distribution of total visits, cough visits, and their relationship. The algorithm to estimate the number of coughs per day depends on a prediction threshold θ and minimum time between predictions Δ and works as follows. First, we use the Ensemble model to the likelihood of a cough for each second (of the ten hours) during which the clinic is open. If the predicted likelihood exceeds θ, then the considered window is labeled as having a cough; otherwise, it is labeled as non-cough. When considering a current window labeled as cough, the counter is incremented if the time between the previous window and the current window exceeds Δ seconds.

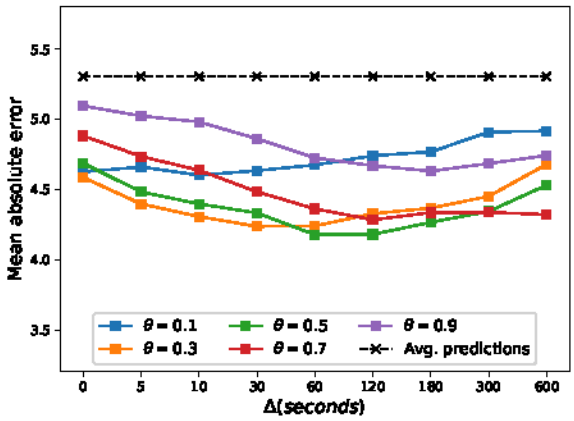
The parameter θ controls the classifier's trade-off between sensitivity and specificity. As expected, increasing θ results in increases in sensitivity but reductions in specificity. It is common for a cough to result in several one-second windows to be marked as a cough. This is due to a patient coughing several times in a row. Such events are a common occurrence,

[1]We do not make the downtown dataset publicly available to the significant effort required to ensure that no personal information is exposed.

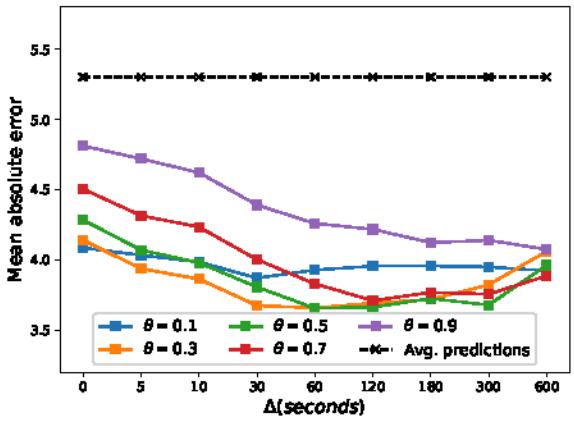
Model	AUC	MCC	Specificity			
			97(%) SEN	95(%) SEN	90(%) SEN	87(%) SEN
	mean $\pm\sigma$	mean $\pm\sigma$	mean $\pm\sigma$	mean $\pm\sigma$	mean $\pm\sigma$	mean $\pm\sigma$
CNN	97.01 ± 0.22	0.38 ± 0.12	60.37 ± 31.21	85.89 ± 4.21	96.07 ± 0.29	97.97 ± 0.12
YAMNet (retrained)	**97.40 ± 0.52**	**0.51 ± 0.11**	**55.03 ± 30.46**	**86.78 ± 6.26**	**97.16 ± 1.26**	**98.58 ± 0.64**
ResNet (retrained)	96.95 ± 0.30	0.34 ± 0.92	63.63 ± 6.07	84.36 ± 3.56	96.26 ± 0.90	98.11 ± 0.58
Ensemble	97.81 ± 0.21	0.52 ± 0.07	79.97 ± 7.64	88.82 ± 2.40	98.35 ± 0.60	99.19 ± 0.34

TABLE V: Performance results on the downtown dataset

as previously shown in Figure 3. Ideally, we would like a cough counter to increase by one after each coughing episode. Increasing Δ can prevent counting a coughing episode multiple times. However, larger values of Δ would merge multiple cough episodes.



(a) Mean absolute error for different values of θ and Δ.



(b) Mean absolute error for different values of θ and Δ when predictions are limited to when patients are in the clinic.

Fig. 10: Linear regression is used to predict the number of cough visits from the cough counts. We evaluate the impact of θ and Δ on predicting the number of cough visits per day.

We use linear regression and leave-one-out cross-validation to predict the number of cough visits per day from the predicted given a pair of θ and Δ values. Figure 10a plots the mean absolute difference between the actual and predicted number of coughs per day. We selected the best values for θ and Δ by minimizing the mean absolute difference on the training data. The best configuration is when $\theta = 0.5$ and $\Delta = 120$ seconds and the same as the best configuration shown in the figure. As a general trend, for a fixed value of θ, increasing the Δ reduces the mean absolute error until a minimum is reached. Beyond this point, further increasing Δ will increase the mean absolute error.

Figure 10a also plots a simple baseline that predicts the number of coughs per day as the average number of coughs computed over the training data. Against this baseline, the cough detector reduces the means absolute error from about 5.3 to 4.17, a reduction of 21.3%. This result demonstrates that we can predict the number of cough visits to the considered clinic with an absolute error of 4.17 visits. By analyzing the record of patient visits to the clinic, we observed that there are times when no patients are in the clinic, particularly during weekends when significantly fewer patients are seen in the clinic. During these periods, the cough detector may make incorrect predictions that coughs occur (e.g., due to background music or discussions among staff). Using the information about the start time of patient visits and the visit's estimated duration, we performed the same analysis as above but limiting predictions to only when one or more patients are in the clinic. The mean absolute error is shown in Figure 10b. The accuracy of the baseline is most unchanged by this additional information. In contrast, the regressors based on the cough counts significantly reduce the error. We can predict the number of cough visits per day with an accuracy of 3.65 visits per day when $\theta = 0.5$ and $\Delta = 60$ seconds.

VI. Discussion

The cough counter presented in Section V-C incorporates simple regression models that predict the number of cough visits from a single variable – the cough count. Such models are attractive due to their simplicity and are useful baselines for future work. In the following, we describe some of the potential sources of error for this model and ways to build better cough prediction models. Inaccuracies in the obtained records may impact our results. For example, a visit may not be labeled as cough (even though it should be) when the provider did not use the word "cough" in their notes or diagnostic code. Furthermore, our results also depend on correct start and end times. Nevertheless, we do not expect these possible sources of error to significantly impact our

models due to the large number of records involved (over 2,560 records).

A limitation of the simple regression models is that they do not discriminate between coughs produced by different individuals. For example, consider the case when three coughs are produced within five minutes. The three coughs could be have been from one, two, or three individuals. The models considered in this paper do not discriminate between these cases; they estimate the average number of cough visits when observing a given number of coughs. Based on their sound characteristics, we could improve these models by estimating the likelihood that coughs come from the same or different individuals. We will investigate such models as part of future work.

VII. Conclusion

The ability to count coughs accurately is an essential element of several key medical applications. This paper considers the challenge of building an effective cough counting application that relies only on real-world audio data. We make three contributions: (1) We will open-source our `public` dataset, which includes carefully annotated cough and non-cough sounds captured in clinic waiting rooms. We hope that our dataset will become a useful standard benchmark dataset for the cough detection community and lead to further advancements in the state-of-the-art. (2) We have evaluated several different convolutional network architectures trained and tested on our large cough dataset. A model based on the YAMNet architecture has the best results with an average AUC of 97.42% as measured by five-fold cross-validation. Combining different YAMNet with ResNet and an optimized CNN to create an Ensemble provides the best overall performance with an AUC of 98.11%. (3) We created a cough counter that incorporates the Ensemble model. Our findings show that a simple linear model can predict the number of cough visits from daily cough counts extracted from audio. Over a dataset that includes 37 days of clinic data and audio, the model predicts the number of cough visits with a mean absolute error of 4.17 cough visits per day. Furthermore, using additional information about clinic and patient scheduling practices further reduce the mean absolute error to 3.65 cough visits per day.

References

[1] J. Smith, A. Woodcock, Cough and its importance in copd, International journal of chronic obstructive pulmonary disease 1 (3) (2006) 305.

[2] G. Botha, G. Theron, R. Warren, M. Klopper, K. Dheda, P. Van Helden, T. Niesler, Detection of tuberculosis by automatic cough sound analysis, Physiological measurement 39 (4) (2018) 045005.

[3] R. V. Sharan, U. R. Abeyratne, V. R. Swarnkar, P. Porter, Automatic croup diagnosis using cough sound recognition, IEEE Transactions on Biomedical Engineering 66 (2) (2018) 485–495.

[4] D. Parker, J. Picone, A. Harati, S. Lu, M. H. Jenkyns, P. M. Polgreen, Detecting paroxysmal coughing from pertussis cases using voice recognition technology, PloS one 8 (12) (2013) e82971.

[5] J. Laguarta, F. Hueto, B. Subirana, Covid-19 artificial intelligence diagnosis using only cough recordings, IEEE Open Journal of Engineering in Medicine and Biology 1 (2020) 275–281.

[6] A. Imran, I. Posokhova, H. N. Qureshi, U. Masood, S. Riaz, K. Ali, C. N. John, M. Nabeel, Ai4covid-19: Ai enabled preliminary diagnosis for covid-19 from cough samples via an app, arXiv preprint arXiv:2004.01275.

[7] J. E. Simmering, L. A. Polgreen, J. E. Cavanaugh, P. M. Polgreen, Are well-child visits a risk factor for subsequent influenza-like illness visits?, Infection Control & Hospital Epidemiology 35 (3) (2014) 251–256.

[8] F. Al Hossain, A. A. Lover, G. A. Corey, N. G. Reich, T. Rahman, Flusense: a contactless syndromic surveillance platform for influenza-like illness in hospital waiting areas, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4 (1) (2020) 1–28.

[9] B. Rocha, D. Filos, L. Mendes, I. Vogiatzis, E. Perantoni, E. Kaimakamis, P. Natsiavas, A. Oliveira, C. Jácome, A. Marques, et al., A respiratory sound database for the development of automated classification, in: International Conference on Biomedical and Health Informatics, Springer, 2017, pp. 33–37.

[10] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, et al., The ami meeting corpus, in: Proceedings of the 5th international conference on methods and techniques in behavioral research, Vol. 88, Citeseer, 2005, p. 100.

[11] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, M. Ritter, Audio set: An ontology and human-labeled dataset for audio events, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017, pp. 776–780.

[12] K. J. Piczak, Esc: Dataset for environmental sound classification, in: Proceedings of the 23rd ACM international conference on Multimedia, 2015, pp. 1015–1018.

[13] P. Leamy, D. Berry, T. Burke, D. Dorran, Re-annotation of cough events in the ami corpus, in: 2019 30th Irish Signals and Systems Conference (ISSC), IEEE, 2019, pp. 1–5.

[14] S. Matos, S. S. Birring, I. D. Pavord, H. Evans, Detection of cough signals in continuous audio recordings using hidden markov models, IEEE Transactions on Biomedical Engineering 53 (6) (2006) 1078–1083.

[15] F. Barata, K. Kipfer, M. Weber, P. Tinschert, E. Fleisch, T. Kowatsch, Towards device-agnostic mobile cough detection with convolutional neural networks, in: 7th IEEE International Conference on Healthcare Informatics (ICHI 2019), 2019.

[16] J. Amoh, K. Odame, Deep neural networks for identifying cough sounds, IEEE transactions on biomedical circuits and systems 10 (5) (2016) 1003–1011.

[17] J. Monge-Álvarez, C. Hoyos-Barceló, P. Lesso, P. Casaseca-de-la Higuera, Robust detection of audio-cough events using local hu moments, IEEE journal of biomedical and health informatics 23 (1) (2018) 184–196.

[18] T. Xia, D. Spathis, J. Ch, A. Grammenos, J. Han, A. Hasthanasombat, E. Bondareva, T. Dang, A. Floto, P. Cicuta, et al., Covid-19 sounds: A large-scale audio dataset for digital respiratory screening, in: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021.

[19] A. Morice, G. Fontana, M. Belvisi, S. Birring, K. Chung, P. V. Dicpinigaitis, J. Kastelik, L. McGarvey, J. Smith, M. Tatar, et al., Ers guidelines on the assessment of cough, European respiratory journal 29 (6) (2007) 1256–1276.

[20] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861.

[22] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.