Methods 205 (2022) 53–62

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

HearCough: Enabling continuous cough event detection on edge computing hearables

Yuntao Wang [a,1], Xiyuxing Zhang [a,1], Jay M. Chakalasiya [b,1], Xuhai Xu [d], Yu Jiang [a], Yuang Li [c], Shwetak Patel [d], Yuanchun Shi [a,*]

[a] Key Laboratory of Pervasive Computing, Ministry of Education, Department of Computer Science and Technology, Tsinghua University, Beijing, China
[b] Global Innovation Exchange, Tsinghua Univetsity, Beijing, China
[c] Beijing University of Posts and Telecommunications, Beijing, China
[d] University of Washington, Seattle, WA, USA

ARTICLE INFO

Keywords:
Continuous cough monitoring
Active noise cancellation hearables
Health monitoring
Edge computing
Deep learning
On-chip Machine Learning

ABSTRACT

Cough event detection is the foundation of any measurement associated with cough, one of the primary symptoms of pulmonary illnesses. This paper proposes HearCough, which enables continuous cough event detection on edge computing hearables, by leveraging always-on active noise cancellation (ANC) microphones in commodity hearables. Specifically, we proposed a lightweight end-to-end neural network model — Tiny-COUNET and its transfer learning based traning method. When evaluated on our acted cough event dataset, Tiny-COUNET achieved equivalent detection performance but required significantly less computational resources and storage space than cutting-edge cough event detection methods. Then we implemented HearCough by quantifying and deploying the pre-trained Tiny-COUNET to a popular micro-controller in consumer hearables. Lastly, we evaluated that HearCough is effective and reliable for continuous cough event detection through a field study with 8 patients. HearCough achieved 2 Hz cough event detection with an accuracy of 90.0% and an F1-score of 89.5% by consuming an additional 5.2 mW power. We envision HearCough as a low-cost add-on for future hearables to enable continuous cough detection and pulmonary health monitoring.

1. Introduction

Pulmonary diseases are among the most prevalent classes of human health problems and a significant cause of mortality worldwide [1]. Cough is one of the most common and prominent symptoms associated with many pulmonary diseases, such as COPD, asthma, and tuberculosis. Automatically analyzing the coughing behavior can help diagnose the pulmonary health conditions and indicate the improvement or deterioration during treatment [3–5]. The key to achieve this is to capture and monitor the cough continuously [2,4,6–8,11,12].

The prevalence of ubiquitous computing enables new opportunities for cough monitoring. Specifically, earphones have become one of the most ubiquitous wireless accessories [13]. Modern earphones often go beyond being just audio listening devices, offering an expanding suite of sensors and micro-controllers with computational capability. Recent inventions and adoptions of active noise cancellation (ANC), a technique that utilize microphones to pick up environmental noises and the speaker to generate anti-phase acoustic signals for noise reduction, enables a more enjoyable listening experience[2]. These ANC microphones, which come with embedded computational units, provide a unique opportunity for edge intelligence. More importantly, these microphones are designed to be continuously working in the background, which brings the possibility of continuous health monitoring [34]. We envision opportunities for pulmonary health assessment using these edge computing ubiquitous earphones.

Using ubiquitous earphones for cough detection has advantages over other sensing solutions. Compared to smartphones [10,14], which are often put into enclosed places such as pockets or bags, earphones have much less obfuscated noise. Compared to smart speakers [9] that are located at certain places in the environment, earphones are wearable and can better capture users' cough sounds regardless of their location. Further, the earphone is close to the user's mouth; thus, it can capture

* Corresponding author.
E-mail addresses: yuntaowang@tsinghua.edu.cn (Y. Wang), shiyc@tsinghua.edu.cn (Y. Shi).
[1] These authors contributed equally to this work.
[2] https://blog.teufelaudio.com/hybrid-anc/

https://doi.org/10.1016/j.ymeth.2022.05.002
Received 31 January 2022; Received in revised form 9 April 2022; Accepted 7 May 2022
Available online 13 May 2022
1046-2023/© 2022 Elsevier Inc. All rights reserved.

clear coughing sounds.

Among all cough-related measurements, cough event detection is fundamental. Therefore, our goal is to enable effective, reliable, and privacy-safe continuous cough event detection method on edge computing ubiquitous earphones. In this paper, we present HearCough, a technique that repurposes always-on ANC microphones in the ubiquitous hearable hardware for continuous cough monitoring in daily scenarios. Our method is entirely on-advice, running on the computing unit in the earphone instead of processing the privacy-sensitive audio signal off the device (e.g., on the cloud). Despite the rich literature on cough detection [4–10,15–17], we are the first to validate the possibility of audio-based continuous cough detection using consumer hearables by finishing deployment of our detection model on hardware of consumer hearables and evaluating its effectiveness and reliability.. To achieve this purpose, we first proposed a lightweight end-to-end deep learning model - Tiny-COUNET that adopts a one-dimensional convolution neural network architecture and requires no pre-processing. As Fig. 1 shows, we adopted transfer learning to train Tiny-COUNET on public sound event dataset with a series of data augmentation methods. Then we evaluated Tiny-COUNET's performance on our evaluation dataset with acted cough and non-cough sounds, which were collected using the ANC microphones in two modern earphone form factors - earbud and headphone. Results indicate that Tiny-COUNET achieves an equivalent detection performance compared with the existing cutting-edge cough event detection methods — FluSense [9] and DeepCough [8]. However, Tiny-COUNET requires significantly less computational resources and spaces. Furthermore, Tiny-COUNET only requires microphones that are available on all ANC earphones. Then we implemented a HearCough wearable prototype using the existing ANC microphone in commodity earphones and a popular earphone micro-controller — BES2300YP. We quantized and deployed the pre-trained Tiny-COUNET to the micro-controller to build the continuous cough event detector.

Finally, we evaluated HearCough's effectiveness and reliability using cough sound data collected from a field study with 8 patients. As a result, it achieves a detection accuracy of 90.0% and an F1-score of 89.5% by consuming an additional 5.2 mW power. To our best knowledge, HearCough is the first work that validates the possibility of audio-based continuous cough detection using ubiquitous earphones. We envision HearCough to be a low-cost add-on for future hearables to enable continuous cough detection and pulmonary health monitoring.

This paper's main contributions have three folds.

1. We present a lightweight end-to-end deep learning model — Tiny-COUNET with its training method for continuous cough sound event detection. With a minimal requirement on storage space of 480 kB and computational requirement of 16.2M FLOPS, Tiny-COUNET can detect the cough event with an accuracy of 89.7% and an F-1 score of 89.9% on our collected acted cough dataset.

2. We present HearCough, a technique that enables continuous cough events detection using the built-in always-on ANC microphones in edge computing hearables. We prototyped HearCough wearable hardware with the existing ANC microphone in commodity earphones and a micro-controller — BES2300YP, on which the pre-trained Tiny-COUNET is deployed.

3. We evaluated HearCough's effectiveness and reliability on cough sound dataset from a field study with 8 patients. Results show that HearCough can realize cough event detection every 0.5 seconds at an accuracy of 90.0% and an F1-score of 89.5% by consuming an additional power of 5.2 mW.

2. Tiny-COUNET: An End-to-End deep learning model for cough detection

In this section, we present our end-to-end deep learning models for cough event detection, named **Tiny-COUNET** and its variants. These variants differ in input sizes, requiring varying computational resources. Finally, we present the training dataset that combines multiple open-access sound event datasets and the transfer learning procedure.

2.1. Tiny-COUNET

Tiny-COUNET is an end-to-end deep learning model designed to deploy on embedded processing units such as ARM M4F. The architecture was inspired by the work of Abdoli et al. [18] and VGG [19]. Tiny-COUNET takes a raw audio stream as input and outputs the probability of the existence of cough sound events. Tiny-COUNET learns a wave representation of inputs when processed through the network. The network architecture is shown in Fig. 2.

The base architecture of Tiny-COUNET consists of one input layer, four convolution blocks, and three fully connected layers. Each convolution block has two convolution layers, followed by a pooling layer and dropout with one- dimensional convolution filters. Tiny-COUNET's last convolution block adopts a global average pooling layer that averages the 64 filters into a one-dimensional array. This global average pooling layer generates constant output size, regardless of the input size. Therefore, it allows experimenting with variable sampling rates and time window length while keeping the model architecture intact. Although the global average pooling layer comes at the cost of losing positional encoding of learned abstract features, it generates a smaller number of outputs compared to flattening the output of the last convolution layer. Thus it requires a smaller weight matrix when connected with the fully connected layer. Tiny-COUNET has 70 thousand parameters that require 480 kB storage with 16-bit integer data format at an 11.0 kHz sampling rate. The low computational needs and small buffer size enable the model to be deployed to an ARM M4F (230 MHz) micro-controller.

We designed Tiny-COUNET to be end-to-end for the following three reasons. 1) Audio spectrum [9,15,16] or short-time Fourier transform (STFT) [8] based deep learning methods require more parameters thus more computational resources. 2) Pre-processing for extracting spectrogram or other features (e.g., Mel-frequency cepstrum coefficients (MFCC) [20]) requires significant computational load. 3) End-to-end deep learning methods take raw audio signals as input; thus, do not miss any vital information.

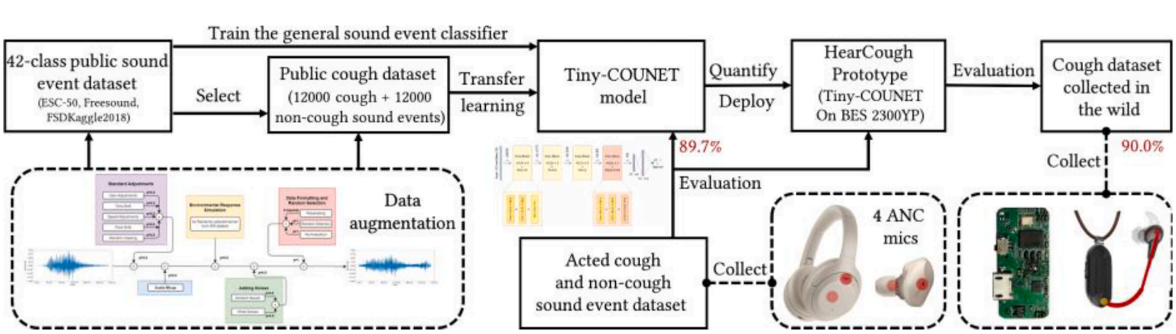

Fig. 1. Overview of HearCough.

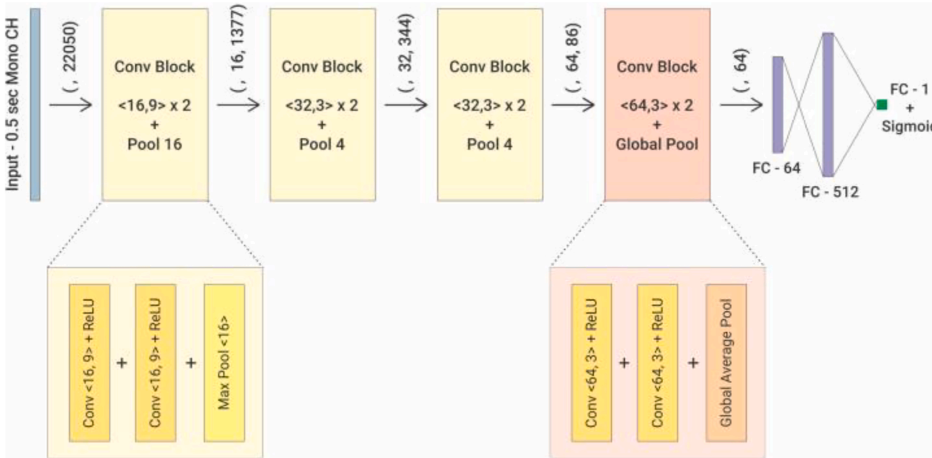

Fig. 2. Tiny-COUNET architecture.

2.2. Input size

Each frame's input size, consisting of an array of audio samples, equals the time duration by the sampling rate by the bit depth. Notably, a smaller input size can significantly reduce the size of the intermediate maps in the deep neural network, making it easier to be deployed on the microcontroller. We present the reasons for parameterizing the time duration per frame, sampling rate, and bit depth below.

2.2.1. Time duration per frame

The statistical analysis of cough audio samples shows that the individual cough instance was between 0.15 seconds to 0.4 seconds, with an average of 0.32 seconds. Although continuous cough sessions can last over 10 seconds, the cough event detector can repeatedly pass the whole session through a moving window. Our pre-study results show that a time duration that is longer than 0.5 seconds did not affect the cough event detection performance significantly, while a time duration shorter than 0.4 seconds deteriorated the performance. We observed consistent results across different sampling rates. Therefore, we adopted the time duration per frame as 0.5 seconds.

2.2.2. Sampling rate

A lower sampling rate decreases the input size but suffers from losing the high-frequency information. We evaluated the effects of 5 sampling rate parameters, including 44.1 kHz, 32 kHz, 22 kHz, 16 kHz, and 11 kHz.

2.2.3. Bit depth

CMSIS-NN [21] optimized 16-bit-integer based neurons and operations for deep learning on ARM M4F series of micro-controllers. Further, same bit-depth was used in the literature [8,9], thus we adopted 16-bit as the bit depth.

2.3. Tiny-COUNET training

We adopted transfer learning to train Tiny-COUNET and its variants to increase the robustness against irrelative sound events. As shown in Fig. 1, we first trained a 42-classes sound event classifier using the Tiny-COUNET with multiple open-access sound event datasets. Then we applied transfer learning to the trained Tiny-COUNET for binary cough event detection. We fine-tuned the model with open access cough sound dataset and a portion of our acted cough dataset during the transfer learning procedure. Below, we describe the training dataset, data augmentation, the training procedure.

2.4. Training dataset

We collected the following multi-class sound event and binary cough event datasets to train the Tiny-COUNET.

- **Multi-class Sound Event Dataset**. 42 sound event classes were selected from online open-access sound event datasets to train the general-purpose sound event classifier. These sound event dataset includes ***ESC-50*** [22] and a subset of the ***Freesound*** [23] for ***FSDKaggle2018*** [24] audio-tagging competition. Specifically, 36 classes were chosen from ESC-50, and 41 classes were chosen from Freesound. These sound events are selected from different categories including natural sound, musical instrument sound, animal sound, human sound, transportation sound, etc., including most of the non-cough sound events in users' life. The full list of sound events are: thunder, wind, water drops, tearing, sneezing, writing, finger snapping, fart, keyboard typing, cough, laughter, guitar, snare drum, bass/drum, double bass, flute, clarinet, harmonica, oboe, saxophone, trumpet, violin, piano, tambourine, burping/eructation, bus, drawer opening/closing, firework, gunshot, keys jangling, microwave, scissor, shatter, telephone, vacuum cleaner, dog barking, bird chirping, door knocking, cowbell, meow, squeak, and no sound. The dataset consists of a total of 18 hours (2 hours from the ESC-50 and 16 hours from the Freesound) of mono-channel audio, with a sampling rate of 44.1 kHz and bit-depth of 32-bit. The data was equally distributed across different sound event classes with a 10% variance in time duration. The dataset consists of a total of 18-hours (2 h from the ESC-50 and 16 h from the Freesound) of mono-channel audio, with a sampling rate of 44.1 kHz and bit-depth of 32-bit. The data was equally distributed across different sound event classes with a 10% variance in time duration.
- **Binary Cough Sound Event Dataset**. This binary cough sound event dataset is constructed from Audio-set [25], ESC-50, and Freesound. All the audio samples are mono-channel with a sampling rate of 44.1 kHz and a bit-depth of 32-bits. In total, we collected 12,000 cough audio samples that last for 6042 seconds (1.68 hours).

2.5. Data augmentation

We applied data augmentation methods to the training dataset to expand its size and reduce the model's susceptibility to environmental factors. The data augmentation consisted of four stages, which are 1) standard audio data augmentation, 2) environmental noise augmentation, 3) audio mixup, and 4) data formatting and random selection. The data augmentation procedure is shown in Fig. 3.

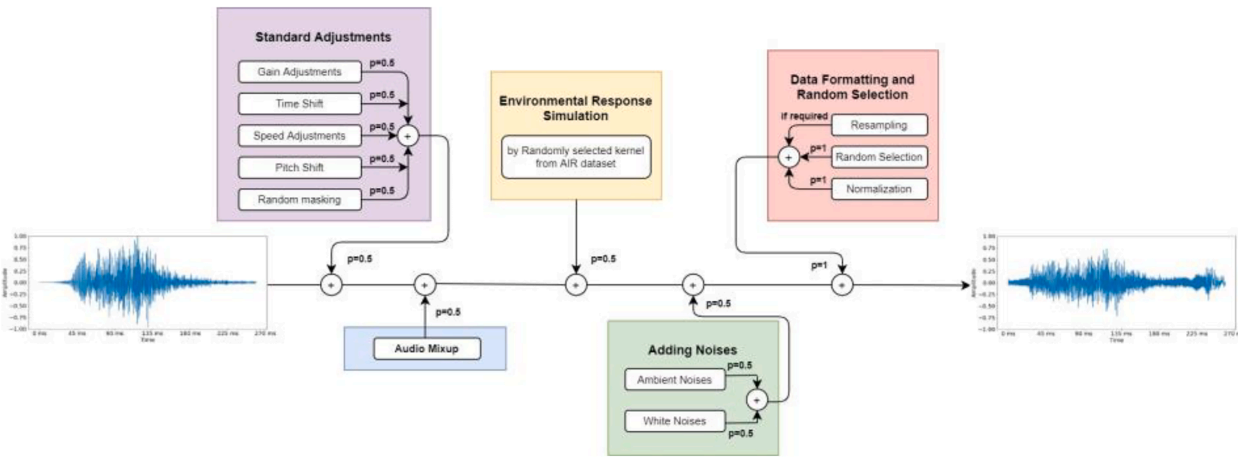

Fig. 3. Data Augmentation Flow.

2.5.1. *Standard audio data augmentations*

Standard audio data augmentation approaches included 1) gain adjustment (x0.5 to x2), 2) time shift (−0.15 to 0.15 seconds), 3) pitch shift (x0.5 to x2), 4) speed adjustment (x0.5 to x2), and 5) random masking by making 0–10% of random points zero. These adjustments are applied with a 50% probability each to preserve some originality.

2.5.2. *Environmental noise augmentation*

The environmental noise augmentation included:

1) mixing background noise from various environmental settings and 2) adding additive white Gaussian noise to simulate possible electromagnetic device interference. The augmentation was also applied with a random 50% probability.

Specifically, the background noises included sounds from the Audioset and environmental sound classification (ESC) [22] dataset. The selected environmental sounds included mundane event sounds, natural sounds, and daily scenario sounds. Machine simulated Gaussian noise[3] was added but maintained a minimum signal–noise ratio (SNR) of 25 dB.

2.5.3. *Audio mixup*

We applied audio mixup to improve the model's generalizability by diversifying the training distribution. The audio mixup simulates a scenario where multiple sounds are generated together (e.g., the sound of cough and whistle simultaneously). Given two audio samples, x_a from class A and x_b from class B, we generated a mixup sample x_m as $\lambda x_a + (1\text{-}\lambda)x_b$, where $\lambda \in [0.7, 1]$. We considered x_m belonging to class A since $\lambda > 0.5$. We randomly applied the audio mixup to 50% of the training data with a random λ value.

2.5.4. *Data formatting and random selection*

Since Tiny-COUNET and its variants require different input sizes, we re- sampled the source audio to match each model's parameter, including sampling rate and bit depth (e.g., then-chip deployed Tiny-COUNET model requires a sampling rate of 11.0 kHz and a bit-depth of 16 bits. As the dataset has a range of audio lengths, random offset and trimming were employed to select the required audio clip length. Finally, we normalized all audio samples.

2.6. *Training procedure*

The training procedure has two stages. We firstly trained a general-purpose classifier on the multi-class sound event dataset with data augmentation. We then applied transfer learning and fine-tuned the trained model for cough event detection. Below we describe each stage with details.

2.6.1. *Training model for general-purpose sound event classification*

We firstly modified Tiny-COUNET for multi-class sound event classification. This is achieved by replacing the last fully connected layer (FC $< 1 >$) with a 42-node fully connected layer (FC $< 42 >$) and changing the Sigmoid activation function to Softmax. Then we trained the model on a multi-class sound event dataset after data augmentation till convergence. The data augmentation workflow was running in real-time for each epoch. Tiny-COUNET were trained using a batch size of 64. As the Tiny-COUNET variants had a varied number of parameters, and we experimented with five different sampling rates of audio samples, the training time and amount of epoch to reach model convergence varied based on task complexity. For instance, Tiny-COUNET (0.5 s@11.0 kHz) was trained with a batch size of 64, and it took 2830 epochs to reach the convergence.

2.6.2. *Applying the transfer learning for cough event detection*

We firstly replaced back the converged model's last fully connected layer — FC $<42>$ to FC $<1>$, and the activation function — Softmax to Sigmoid. Then we trained the last two fully connected layers of the model mentioned above till convergence on the binary cough sound event dataset (see section 2.4). Finally, we fine-tuned the entire network at a lower learning rate (1e-4 to 1e-5). The data augmentation workflow was running in real-time for each epoch. Similar to the training model for general-purpose sound event classification, the complexity of the model and variety of sampling rates resulted into a varied number of epochs to reach convergence for each combination of model and sampling rate. For instance, Tiny-COUNET (0.5 s@11.0 kHz) was retrained with a batch size of 64. Each batch contains 32 cough and non-cough sound samples. Each epoch includes 375 batches. Tiny-COUNET took 1500 epochs to reach the convergence.

3. Tiny-COUNET evaluation and result

This section presents the data collection hardware, the collected evaluation dataset with cough and non-cough sounds, the evaluation procedure, and the major results.

3.1. *Evaluation dataset*

We present the hardware and the user study to collect real-scenario audio samples using built-in ANC microphones in commodity hearables. The acted cough audio samples and random non-cough sound events

[3] https://medium.com/analytics-vidhya/adding-noise-to-audio-clips-5d8cee24ccb8

were used as the dataset to evaluate the machine learning model described in section 2.

3.1.1. Data collection hardware

Figure 4 shows the hardware setup we deployed for our data collection. It has one SONY WF-1000XM3 true wireless stereo (TWS) ANC earbud[4] and one SONY WH-1000XM4 wireless ANC headphone[5]. We collected the raw audio signals from four microphones by wiring the raw audio signal directly from the always-on ANC microphones, as Fig. 4 shows. The four microphones include:

1) the left outer-ear-cup feed-forward microphone of the SONY WH-1000XM4 headphone (Fig. 4A); 1) the left inner-ear-cup feedback microphone of the SONY WH-1000XM4 headphone (Fig. 4B); 3) the right inner-ear feedback microphone of the SONY WF-1000XM3 earbud (Fig. 4C); 4) the right outer-ear feed-forward microphone of the SONY WF-1000XM3 earbud (Fig. 4D). The four aforementioned sound sources were simultaneously recorded with a Zoom multi-track handy recorder (model H6)[6]. All the microphones were powered by the handy recorder with VXLR to 3.5 mm converters. We set the Zoom H6 recorder's sampling rate at 96 kHz and resolution at 24 bits. The gain values of all channels were set to 5 out of 10. For each recording session, the Zoom H6 recorder generates one audio file in WAV format for each input channel with all files time-synchronized. Further, We utilized a noise meter (Smart Sense AR824) to measure the environmental noise level[7].

3.1.2. Acted cough sound event collection

We first conducted a user study to collect acted cough sound event dataset under various background noises. The study was approved by the Institutional Review Board (IRB). All participants agreed to open-source their cough sound data for public research purposes.

3.1.2.1. Participants. We recruited 20 participants with an average age of 21.3 (s.d. = 2.6) from a local university. None of them had respiratory disease.

3.1.2.2. User study design and procedure. Here we define the data collection on each participant as one trial. Each trial has three conditions in different environments, including a quiet room (43 dB), a sidewalk of the street (64 dB), and a noisy canteen (75 dB) during rush hours. We recorded two sessions of acted cough events under each condition. During each session, participants coughed 10 times.

Firstly, participants were informed of the user study's purpose and procedure. All participants signed a data open-source agreement since they all agreed to do so during recruitment. Then they wore the hardware and practiced multiple times in the quiet room. Through the whole user study, each participant wore the SONY WH-1000XM4 headphone on the right side and the SONY WF- 1000XM3 earbud on the left side, as Fig. 5 shows. After the practice, the experimenter guided each participant to record two sessions of cough data under each environmental condition. We followed the order of the quiet room, the sidewalk, and then the noisy canteen.

3.1.3. Non-cough sound event collection

To further evaluate our method's false-positive performance, the experimenter wore the hardware and went to 13 different locations for data collection of random non-cough sound events. These locations include a subway train (96 dB), a pedestrian street (78 dB), two offices (58 dB and 46 dB), two bus stops (90 dB and 84 dB), a moving car (76 dB), two households (49 dB and 52 dB), two shopping malls (69 dB and 77 dB), and two canteens (69 dB and 72 dB). The experiment recorded various actions, including standing still (5032.71 s), walking (4312.26 s) and speaking (5253.72 s). Specifically, we recorded more speaking events due to its similarity with cough events [10]. We recorded 2 min of audio data in each location, resulting $2 \times 13 \times 3 = 78$ minutes audio recordings from each microphone.

3.1.4. Acted cough sound event dataset

In total, we collected 20 participants $\times$ 4 microphones $\times$ 6 sessions $\times$ 10 cough samples = 4800 cough audio samples under various background noises. We manually labeled and segmented cough events. The dataset collected two types of cough: single cough and continuous cough. On average, each single cough lasts for 0.37 seconds (s.d. = 0.12) and each continuous cough lasts for 0.79 seconds (s.d. = 0.30). We obtained 3050 single cough events and 1750 continuous cough events, respectively.

3.2. Evaluation and result

We present and compare Tiny-COUNET's performance against existing baseline cough event detection methods. We also analyze the effect of sampling rate (input size) to the detection performance. All machine learning models were implemented and trained using TensorFlow[8].

3.2.1. Baselines

We selected and re-implemented two existing cutting-edge cough event detection methods as baselines. Both Tiny-COUNET and these two baselines were trained and evaluated on the same training and validation dataset for a fair comparison. We also selected two cough event detection methods based on earbuds platforms. In total, we compared Tiny-COUNET with four baselines.

Fig. 4. The hardware setup and the location illustration of the four ANC microphones. This figure was recreated using product pictures from official website and 52Audio.

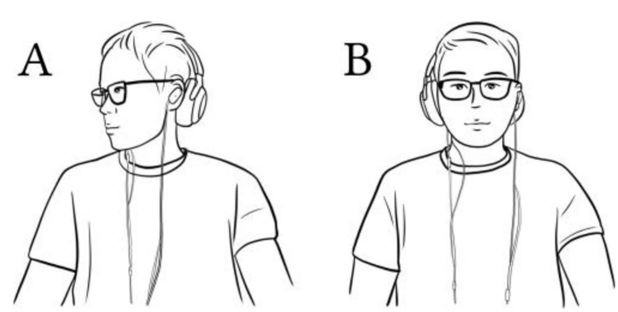

Fig. 5. Each user wore the earbud on the left ear and the earphone on the right ear during the user study.

4 https://www.sony.com/electronics/truly-wireless/wf-1000xm3
5 https://www.sony.com/electronics/headband-headphones/wh-1000xm4
6 https://zoomcorp.com/en/us/handheld-recorders/handheld-recorders/h6/
7 http://www.52audio.com/
8 https://www.tensorflow.org/

(1) **FluSense** [9]**.** FluSense is a convolution neural network (CNN) that enables accurate cough event and cough type detection. It adopted two-dimensional CNN on the spectrogram extracted from the raw audio stream. We adopted a 1-second window duration according to the original paper. We also adopted a sampling rate of 8 kHz according to the original paper.
(2) **DeepCough** [8]**.** Similar with FlueSense, DeepCough adopted CNN architecture on the STFT + MFCC features extracted from the raw audio for cough event detection. Specifically, we used a superior version of the DeepCough by running it on a 320 ms audio clip with a 64 ms moving window.
(3) **CoughBuddy** [16]**.** CoughBuddy adopted a sensor-fusion method usingsignals from both the microphone and the inertial measurement unit (IMU). It utilized feature-based classification algorithms and dynamic time warping (DTW) for cough detection. We did not re-implemented this work because of the vague of the specific audio algorithms. Nevertheless, the work was worth comparison since they were designed for earphones.
(4) **CoughTrigger** [15]**.** CoughTrigger utilizes a lower-power-conception IMU in earbuds as a cough detection activator using a multi-center classifier (MCC) to trigger the microphone for audio signal processing and classification. We selected this method as one of our baselines for the same reason as our selection of CoughBuddy.

3.2.2. Evaluation metrics

Aiming to produce a micro-controller deployable solution, we consider the following threefold evaluation metrics: 1) detection performance, 2) size of parameters and intermediate variables, and 3) inference latency. The detection performance is directly related to the effectiveness and reliability, while second and third evaluation metrics correlate with the usability and adaptability of the micro-controller. Below we describe each metric in detail.

- **Detection performance.** We used overall accuracy, specificity, sensitivity, and F-1 score as standard performance metrics.
- **Size of parameters and intermediate variables.** The edge computer unit has limited onboard storage that stores the model and the temporary inputs and outputs at each layer. Thus we measured the size of model parameters and intermediate variables as one metric.
- **Inference latency.** The inference latency has to be less than the inference window's duration to facilitate the continuous cough event detection. The complexity of the model and the clock rate of the computing unit directly impacts the latency. Thus, we measured and compared the computing latency among different machine learning models.

3.2.3. Results and findings

In this evaluation, all machine learning models were implemented and trained on a workstation with Intel Xeon E5-2670 Octa-Core CPU, 64 GB RAM, and 12 GB NVIDIA Titan X GPU. We present the results and major findings below.

Tiny-COUNET is effective for cough event detection. As shown in Table 1, Tiny-COUNET with a sampling rate below 16.0 kHz achieves comparable detection performance when compared to the baseline methods. However, Tiny-COUNET has significantly less requirement in computing resources and storage space. Meanwhile, Tiny- COUNET yields the lowest computing latency. Compared with CoughBuddy and CoughTrigger, Tiny-COUNET has less requirement for input data since multi-modal data is not available on the majority of earbuds platforms.

Worth mentioning, the two baselines — Flusense and DeepCough — require additional computing resources to calculate the spectrogram and short-time Fourier transform (STFT), respectively. Since these pre-processing steps vary depending upon the implementation, Table 1 excludes the related computational complexity. Since Tiny-COUNET is end-to-end, it does not require any pre- processing. Therefore, Tiny-COUNET is more efficient and suitable for on-chip deployment compared to the baselines.

A sampling rate at 22 kHz is sufficient for cough event detection. A higher sampling rate benefits the recognition performance since it provides more high-frequency features. However, we observed a slight decrease (0.3%) in detection performance when down-sampling the audio from 44.1 kHz to 22.05 kHz. Then a distinct decrease in the detection performance happens when the sampling rate is below 16 kHz. Thus, a sampling rate at 22 kHz is sufficient for Tiny-COUNET to achieve accurate cough event detection.

Tiny-COUNET (0.5 s@11.0 kHz) is the best fit for the on-chip deployment. Considering the detection performance, the computational need, and the latency, Tiny-COUNET (0.5 s@11.0 kHz) is more compatible with the micro-controllers with only hundreds of MHz of clock speed and ultra-low storage capacity (e.g., hundreds of kB). Therefore, in section 4, we adopted Tiny-COUNET (0.5 s@11.0 kHz) for quantization and on-chip deployment.

4. HearCough: Continuous cough event detection on hearables

This section presents the key components of HearCough, including the hardware prototype and the on-chip deployment of the Tiny-COUNET.

4.1. HearCough wearable prototype

e deployed our model on a popular micro-controller — BES2300YP by the Bestechnic[9]. It was adopted by many popular true wireless stereo (TWS) earphones, such as JBL FREE II, Samsung Galaxy Buds Live, and Huawei FreeBuds 2 Pro. It has dual ARM-Cortex M4F processors with up to 300 MHz CPU, 992 kB SRAM, and 4 MB flash storage. We only use one single processor in this work since the other processor runs the Bluetooth stack and digital signal processing (DSP) related algorithms. Further, the SRAM is shared by the two processors. The Bluetooth and the operating system take more than 400 kB of the SRAM. To prevent the memory overflow, we limited the SRAM space usage for the machine learning model to be around 500 kB. Fig. 6C shows the development board that run the pre-trained Tiny-COUNET (0.5 s@11 kHz) for cough sound event detection and evaluation. The board is powered at 5 V. The three analog microphone ports include two for active noise cancellation and one for speech audio collection. During the evaluation, the audio samples were streamed onto the on-chip SRAM using an TF card (8 GB) through SPI protocol. The data collection wearable is a necklace pendant with its hardware as shown in Fig. 6B. Inside the wearable prototype, we designed a printed circuit board (PCB) with the BES2300YP micro-controller that records the audio signal from the analog microphone, as shown in Fig. 6C. The audio signal is streamed to another PCB with the TF card. The wearable prototype is powered with a 130 mAh Li-Po battery that can continuously record for 6 h. It starts the recording when the switch is ON and stops when the switch is OFF. The audio is in WAV format, with the sampling rate at 48 kHz and encoding depth at 16 bits. We adopted an always-on feed-forward microphone in a Bose Quietcomfort 20 earbud[10] as Fig. 6D shows. This is to evaluate the scalability of HearCough to different microphone hardware in commodity hearables.

4.2. On-chip model quantization, deployment and optimization

In our implementation, we adopted the Tiny-COUNET with 0.5 s @ 11025 Hz input size for on-chip deployment. This is because only the Tiny- COUNET with the parameters mentioned above (480 kB) can run on the micro-controller due to the space limit of around 500 kB. We adopted a moving window strategy to store the input audio stream on

[9] http://www.bestechnic.com/Home/Index/index/lan_type/2

Table 1
Evaluation results of different models with various input size. Acc., Rec., Spec., and F1 represent Accuracy, Recall (Sensitivity), Specificity, and F1-Score.

Model	Input Size	Flops (M)	Space (kB)*	Acc.	Sen.	Spec.	F1	Latency (ms)
Tiny-COUNET	0.5 s @ 44.1 kHz	64.9	1514	92.9%	93.2%	92.6%	92.9%	16.8 ± 4.2
Tiny-COUNET	0.5 s @ 32.0 kHz	47.0	1136	92.5%	91.4%	93.7%	92.6%	11.1 ± 1.8
Tiny-COUNET	0.5 s @ 22.05 kHz	32.5	824	92.7%	93.6%	91.9%	92.6%	8.0 ± 1.7
Tiny-COUNET	0.5 s @ 16.0 kHz	23.5	636	91.3%	90.4%	92.2%	91.4%	6.5 ± 1.2
Tiny-COUNET	0.5 s @ 11.0 kHz	16.2	480	89.7%	88.3%	91.1%	89.9%	5.2 ± 0.1
FluSense+	1.0 s @ 8.0 kHz	56.4	882	87.2%	87.7%	86.7%	87.1%	7.0 ± 2.5
DeepCough+	0.32 s @ 16.0 kHz	10.0	1200	91.7%	89.2%	94.5%	91.9%	10.0 ± 3.3
CoughBuddy#	—	—	—	88.8%	83.0%	91.3%	83.2%	—
CoughTrigger#	—	—	—	73.0%	82.0%	55.0%	—%	—

* The space includes model parameters and intermediate variables in 16-bit integer data format.
\+ We exclude the requirement in computational resources and storage space for pre-processing.
\# We exclude the metrics uncertain from the original paperworks for models we did not re-implement.

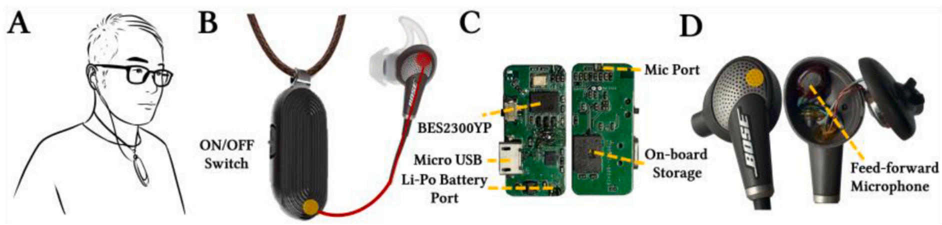

Fig. 6. The data collection wearable and hardware in the field study. A: The patient wore the necklace hardware during the field study. B: The necklace prototype for collecting cough sound events. C: The printed circuit boards and wiring for the HearCough prototype. D: The feed-forward microphone's location in the earbud.

the SRAM. Therefore, HearCough requires 2 window duration of data stored on the SRAM, resulting in an extra space overhead of 11 (kHz) × 16 (bit depth) / 8 (bits per Byte) × 1 (second) = 22 kB. Therefore, the continuous cough event detection algorithm on the BES2300YP micro-controller consumes 480 kB + 22 kB = 502 kB SRAM space in total.

The pre-trained Tiny-COUNET's weights and activations were stored in a 32-bit float data format. Since the ARM-M4F chip-set does not support float-point arithmetic, the model must be quantized before deployment. Therefore, we transformed the Tiny-COUNET using CMSIS-NN [21] compiler. We firstly converted the floating-point values to fixed-point, considering the compatibility and computational efficiency. Specifically, we adopted a 16-bit integer data format to avoid significant accuracy reduction. Therefore, all weights and activations were quantized symmetrically around zero with power-of-two scaling using the bitwise shifting function in the CMSIS-NN kernel. All the layers and functions of Tiny-COUNET are available in the CMSIS-NN library. We further optimized the deployed Tiny-COUNET by identifying the most suitable implementations in different versions of CMSIS-NN kernels depending on the values of the layer dimensions.

Finally, we built the HearCough prototype by deploying the Tiny-COUNET on the BES2300YP micro-controller using the Keil MDK[10] software development environment. Thus HearCough can continuously detect the cough sound event in the background. To further lower the computing load and power consumption, we added an additional screening method with an adaptive threshold. The threshold is set to be twice the average amplitude of the last 5 s of audios. Therefore, HearCough activates the Tiny-COUNET for cough event detection only when the average amplitude of the current 0.5-second audio clip is above the threshold.

5. HearCough evaluation and result on cough data from patients

This section describes the field study for further evaluating HearCough's performance on continuous cough event detection with actual patients, including the study design, study procedure and the evaluation results.

5.1. Participants

We recruited 8 patients who suffered from pharyngitis (P1, P5, P6, P7), bronchitis (P2, P3), allergy (P8), or fever (P4). They have an average age of 31.0 (s.d. = 11.9). We recruited all the participants from a local hospital. They coughed in daily routine with different counts and frequencies. All participants were tested negative for the Covid-19. The field study was approved by the Institutional Review Board (IRB).

5.2. Study design and procedure

We first introduced the purpose and the device of the study to the participants. Participants randomly picked one device for either left or right ear. We asked participants to wear the device while performing daily routine tasks for 5 hours. Therefore, for each participant, we obtained 5 hours of recording data. Finally, we conducted a quick follow-up interview.

5.3. Data labeling

We collected 5 × 8 = 40 hours audio recordings. We first applied a threshold to filter out irrelevant sound events such as quiet room, As a result, 17.03 hours of audio recordings were left. Then we sampled all recording files into 20-second audio clips. Then we recruited 12 raters into 4 groups to identify and segment cough events in all the audio clips. Raters in the same group process the same audio clip set (4.25 hours). We consider an audio clip valid as at least 2 out of 3 raters labeled as positive. In total, we collected 2713 cough audio event samples. There are 2041 single cough events and 336 continuous cough events. The single cough events have an average duration of 0.38 (s.d. = 0.32) seconds. The continuous cough events average time duration of 1.03 (s. d. = 1.41) seconds. Continuous cough events were segmented into multiple single cough events for the evaluation.

[10] https://www2.keil.com/mdk5

5.4. Results and findings

We further fine-tuned the pre-trained Tiny-COUNET with our collected acted cough and non-cough dataset (section 3.1.2) with data augmentation methods in section 2.5. We splited the acted cough and non-cough dataset into training set and validation set, using early stopping strategy to avoid overfitting. We re-trained Tiny-COUNET for 400 epochs, which is around the early stopping point. This is to further increase Tiny-COUNET's performance on cough event detection. Then we deployed the re-trained Tiny-COUNET with the steps presented in section 4. Finally, we evaluated its performance using the cough sound data and non-cough sound data from real patients under various conditions. We balanced this data by randomly choosing an equal amount of non-cough sound audios. We describe our major results and findings below.

5.4.1. Detection Performance and Error Analysis

Results show that HearCough achieved 90.0% accuracy, 85.2% sensitivity, 94.8% specificity, and 89.5% F1-score on the cough sound dataset from the real patients. We believe that the transfer learning presented in section 2.6 has a significant contribution to HearCough's high performance. Therefore, Tiny-COUNET learned the features from the cough sound and features from a comprehensive set of non-cough sound events.

In order to further analyse the performance, we exported the false-recognized samples and recruited 3 experienced raters different from 12 raters in 5.3. In aggregate, 246 false-positive samples and 422 false-negative samples are exported and analysed by the raters.

An analysis of the false-positive samples in the dataset reveals that coughs are most often confused for ambient noise with burst (61%, such as plate crushing) and for speech actions (31%). The remaining 8% of false positives are indeed cough samples but not cognizant by the raters. The reason for recognizing noise with burst as cough may derive from the similarity of the pattern, both of which start with silence followed with a burst sound. Speech actions are erroneously recognized due to its similarity with cough [10].

Among the false-negative samples, 50% of the sounds are indeed cough sounds but with little volume, most of which are considered similar to the ambient noise with burst by our raters. 23% of the samples are coughs with extremely high volume, which may be falsely taken as wind due to the similarity of the sonic boom and incessant high volume featured by both sounds. 17% of the sounds are falsely labeled as cough before in 5.3, among which 7% are not coughs and 10% are inhalation phase sounds before cough. Throat clear actions takes up for 10%.

Our model has a lower performance on cough-like sounds (such as speech sounds and ambient noise with burst) may derive from the shallowness of our CNN model, which limits the model's ability to extract more essential features and distinguish more subtle differences between similar sounds.

There is space to improve HearCough's performance further. Without the fine-tuning procedure using our acted cough dataset, the quantized Tiny-COUNET achieved 78.5% accuracy, 72.2% sensitivity, 84.8% specificity, and 77.0% F1-score. Therefore, the fine-tuning procedure can significantly increase the detection performance. Tiny-COUNET has a lower detection performance when compared with results on the acted cough dataset. We believe in the following two reasons.

1) Although recorded in various scenarios, the acted cough dataset did not cover a wide range of background noises as the collected dataset from this field study.
2) Tiny-COUNET was trained using public datasets using different recording devices but not the ANC microphones in the earphones. Different microphones and their locations can significantly affect the detection performance. We believe that we can further improve Tiny-COUNET's performance using an incremental learning strategy after deployment.

5.4.2. Hardware Power Consumption and Scalability

The average power consumption of the BES2300YP development board was 65.1 mW without running the Tiny-COUNET. The average power consumption increases to 70.3 mW while Tiny-COUNET runs for continuous cough event detection. Thus the Tiny-COUNET consumes an additional — 5.2 mW power, only 8% of the original power consumption. HearCough can work across hearable devices. In this paper, we re-trained HearCough dataset collected by Sony WF-1000XM3 and Sony WH-1000XM4. Then we evaluated HearCough's performance on dataset collected by Bose QC 20. Results show that HearCough can work across devices despite the ANC microphone's specs.

5.4.3. Preliminary Analysis on the Cough Behavior

Results show that the 8 patients coughed an average of 339.12 (s.d. = 350.28) times during the 5 hours recording. According to the user survey, participants perceived an average cough amount of 103.75 (s.d. = 82.62) times. There is a significant difference ($Z = 2.1$, $p = 0.03$) between the perceived and actual amount of coughs when using Wilcoxon signed-rank test for the significance analysis. This indicates that human's recall of their cough amount is inaccurate. Fig. 7 shows the number of coughs per 15 min by the time of all participants. We noticed that different patients with different diseases have different coughing behavior. For example, patients who suffer from pharyngitis (P5 - P7) coughed the most after their meals and gradually coughed less. However, the patient (P4) with fever suffered from a severe cough continuously within the 5 hours.

6. Discussion

In this section, we discuss our major findings, design recommendations, limitations, and future work.

6.1. Edge intelligence on consumer hearables

Earphones have become one of the most ubiquitous devices after smartphones. Embedded with more sensing techniques and computational resources, we foresee many opportunities to make these ubiquitous hearable devices intelligent for health purposes for four reasons. 1) These ubiquitous hearables have many sensors (e.g., microphone, motion sensor, and proximate sensor) that are ready to capture rich physiological signals. 2) The hearable devices are located on our heads thus can provide a large amount of biometric information from the respiratory [26], cardiac [27–31], and nerve [32] systems. 3) The computational power of the micro-controller in modern earphones is becoming stronger, supporting many machine learning-based intelligent applications. 4) Current wireless earphones with the BLE protocol can support firmware updates over the air, making application deployment to earphones easy.

6.2. Developing deep learning techniques on Micro-controllers

Despite potential applications for developing deep learning techniques on micro-controllers, additional deployment constraints other than accuracy including storage space, computational resource, inference latency, and power consumption need to be considered. Therefore, we comprehensively investigated the model's architecture, size, and input for deployability during the design phase. Further, since the micro-controller is limited in computational resources, we design Tiny-COUNET and its variants as end-to-end; thus, HearCough requires no computationally intensive pre-processing methods.

There is a trade-off between detection performance and deployability in enabling machine learning based applications on edge devices as demonstrated in our paper. Although the Tiny-COUNET (0.5 s@11

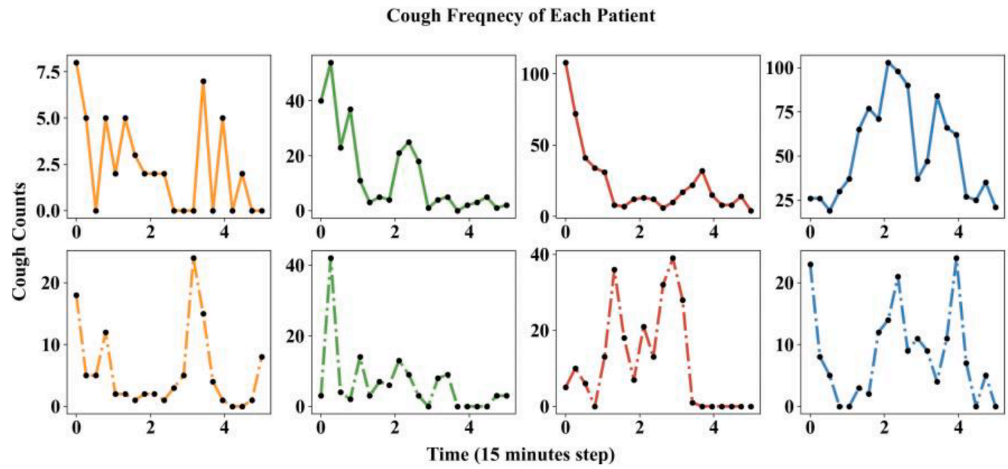

Fig. 7. Each patient's cough counts every 15 min along with the time (P1 - top left figure, P8 - bottom right figure).

kHz) did not have the highest detection performance, it achieved a fair accuracy on the cough dataset collected in the field study with real patients. However, it has the minimum requirements on the storage space, computational resources, and power. We believe that HearCough can be further improved by applying the incremental learning as we did in section 5 using the data collected after deployment.

The input of the deep learning model has a significant effect on deployability and real-time capability. As Table 1 shows, a larger stream of input would provide more information for inference but can significantly increase the model size. Thus it requires more storage space and computational resources, making the detection method less deployable. In our implementation, HearCough adopted Tiny-COUNET with 0.5-second input window duration at 11.0 kHz sampling rate. Therefore, HearCough can detect the existence of a cough event every 0.5 s. Our method can handle different input sizes thanks to the convolutional layers in the Tiny-COUNET. Thus, to further lower the computational resources needed or increase the real-time capability, one can leverage a smaller input window duration or lower sampling rate, with the cost of lower detection performance. For example, the 0.33-second window size will decrease the performance by around 2% on average.

6.3. Unique opportunities of the Always-on active noise cancellation Microphones

HearCough utilizes the ANC microphones in the earphones for cough event detection. These microphones were designed to be always-on. Therefore, HearCough starts to work once users put on the earphones. Further, it will not consume extra power from the sensors except for running the detector on the micro-controller. With an increasingly powerful computing unit deployed on hearables, we believe that the built-in always-on ANC microphones can open more new opportunities for edge computing or ubiquitous computing. Worth mentioning, HearCough uses the outer-ear feed-forward microphone rather than the in-ear feedback microphone because the feedback microphone will pick up audios (e.g., music) as large noises from the speaker.

6.4. Continuous cough monitoring for health management

Continuous cough event detection can enable cough frequency monitoring, valuable for respiratory and otolaryngological health. Healthy people usually cough twice per hour while patients cough 43 times on average per hour [33]. In our field study, patients coughed more than 60 times per hour. Quantitative analysis shows that people recalled a significantly lower number of their coughs. Indicating the low credibility of people's recall memory on their cough behavior, not to mention cough frequency analysis. Therefore, HearCough can be valuable for people to be aware of their coughing behavior and health status. Using ubiquitous earphones for cough event detection has advantages over other sensing solutions, including smartphones [10,14] or smart speakers [9]. The earphone is close to the user's mouth; thus, it can capture clear coughing sounds. Further, earphones can provide more long-term monitoring capability since they are wearable.

6.5. Limitations and future work

This paper explored the feasibility and evaluated the performance of deploying a deep learning cough event detector on a popular computing unit available in many commodity hearables. However, we have not designed a prototype with all components working together. We leave this for future work. Further, continuously listening for intermittent cough events is computationally powerconsuming. If we can filter out potential cough events and only run inference on these filtered events, the power efficiency could be improved. One potential way to achieve this is through filtering the input stream for coughing-induced sudden bursts of power. For instance, a DSP-based algorithm can filter out non-cough events and wake the model only for potential cough events.

Currently, HearCough can only detect the cough sound event, which is the foundation of any cough measurements. Although cough event detection alone can be used for self-health management, cough type detection will also help medical diagnostics. We expect future work to explore on-chip machine learning based on cough type detection. Further, we expect future work to explore how the segmented cough audio samples and the statistical analysis can help medical experts with diagnostics or therapeutic evaluation.

7. Conclusion

We present HearCough, a technique that enables the always-on active noise cancellation microphones in commodity hearables for continuous cough event detection. We first proposed a lightweight end-to-end deep learning model — Tiny-COUNET and its transfer learning based training approach with data augmentation methods. When evaluated on our collected acted cough event dataset using a 0.5-second window and a sampling rate of 11 kHz, Tiny-COUNET achieved an accuracy of 89.7% and an F1-score of 89.9%. It also required significantly less computational resources (16.2 MFlops) and spaces (480 kB) than cutting-edge cough event detection methods. We further implemented HearCough by quantizing and deploying the pre-trained Tiny-COUNET to a popular micro- controller — BES2300YP in modern earphones. Finally, we proved HearCough's effectiveness and reliability using a field study with eight patients. Results show that HearCough achieved continuous cough event detection with an accuracy of 90.0% and an F1-

score of 89.5% by consuming an additional 5.2 mW power. To the best of our knowledge, this work is the first to validate the possibility of audio-based continuous cough event detection using the hardware in consumer hearables. We envision HearCough as a low-cost add-on for future hearables to enable continuous cough detection and pulmonary health monitoring.

Funding

This work was supported by the National Science Foundation of China (NSFC) under Grant No.6213000120 and No.62002198, Independent Research Plan of Tsinghua University, Beijing Key Lab of Networked Multimedia, andInstitute for Guo Qiang of Tsinghua University.

Author contributions

Yuntao Wang: Conceptualization, Formal analysis, Methodology, Validation, Visualization, Funding acquisition, Writing – original draft, Writing – review & editing. **Xiyuxing Zhang:** Validation, Visualization, Formal analysis, Software, Writing – original draft, Writing – review & editing. **Jay M Chakalasiya:** Formal analysis, Validation, Methodology, Software, Visualization, Writing – original draft. **Xuhai Xu:** Conceptualization, Methodology, Writing – review & editing. **Yu Jiang:** Visualization, Writing – review & editing. **Yuang Li:** Methodology, Software, Writing – original draft. **Shwetak Patel:** Methodology, Resources, Supervision, Writing – review & editing. **Yuanchun Shi:** Conceptualization, Methodology, Funding acquisition, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Science Foundation of China (NSFC) under Grant No.6213000120 and No.62002198, Independent Research Plan of Tsinghua University, Beijing Key Lab of Networked Multimedia, and Institute of Guo Qiang of Tsinghua University. We want to acknowledge all theparticipants for volunteering in our user studies.

References

[1] Global initiative for chronic obstructive lung disease - global initiative for chronic obstructive lung disease (May 2020). URL https://goldcopd.org/.
[2] J. Amoh, K. Odame, Technologies for developing ambulatory cough moni- toring devices, Critical Reviews™ in Biomedical Engineering 41 (6).
[3] H.A. Bickerman, S.E. Itkin, The effect of a new bronchodilator aerosol on the air flow dynamics of the maximal voluntary cough of patients with bronchial asthma and pulmonary emphysema, J. Chronic Dis. 8 (5) (1958) 629–636.
[4] C. Zhu, L. Tian, X. Li, H. Mo, Z. Zheng, Recognition of cough using features improved by sub-band energy transformation, in: in: 2013 6th International Conference on Biomedical Engineering and Informatics, 2013, pp. 251–255.
[5] S. Birring, T. Fleming, S. Matos, A. Raj, D. Evans, I. Pavord, The leicester cough monitor: preliminary validation of an automated cough detection system in chronic cough, Eur. Respir. J. 31 (5) (2008) 1013–1018.
[6] Y.A. Amrulloh, U.R. Abeyratne, V. Swarnkar, R. Triasih, A. Setyati, Au- tomatic cough segmentation from non-contact sound recordings in pediatric wards, Biomed. Signal Process. Control 21 (2015) 126–136.
[7] J. Amoh, K. Odame, Deep neural networks for identifying cough sounds, IEEE Trans. Biomed. Circuits Syst. 10 (5) (2016) 1003–1011.
[8] J. Amoh, K. Odame, Deepcough: A deep convolutional neural network in a wearable cough detection system, in: 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS), IEEE, 2015, pp. 1–4.
[9] 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, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4 (1) (2020) 1–28.
[10] X. Xu, E. Nemati, K. Vatanparvar, V. Nathan, T. Ahmed, M. Rahman, D. Mccaffrey, J. Kuang, J. U. N. A. Gao, Listen2Cough : Leveraging End- to-End Deep Learning Cough Detection Model to Enhance Lung Health Assessment Using Passively Sensed Audio, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 5 (1) (2021) 1–22.
[11] M. You, W. Wang, Y. Li, J. Liu, X. Xu, Z. Qiu, Automatic cough detection from realistic audio recordings using c-bilstm with boundary regression, Biomed. Signal Process. Control 72 (2022), 103304.
[12] N. Simou, N. Stefanakis, P. Zervas, A universal system for cough detection in domestic acoustic environments, in: 2020 28th European Signal Processing Conference (EUSIPCO), IEEE, 2021, pp. 111–115.
[13] Canalys, Global smart device shipment forecasts 2020 to 2023, last accessed January 2020 (2020). URL https://www.canalys.com/newsroom/ canalys-worldwide-smart-device-shipments-2023.
[14] J. Monge-Alvarez, C. Hoyos-Barcelo, P. Lesso, P. Casaseca-de-la-Higuera, Robust detection of audio-cough events using local hu moments, IEEE J. Biomed. Health Informatics 23 (1) (2019) 184–196, https://doi.org/10.1109/JBHI.2018.2800741.
[15] S. Zhang, E. Nemati, M. Dinh, N. Folkman, T. Ahmed, M. Rahman, Kuang, N. Alshurafa, A. Gao, Coughtrigger: Earbuds imu based cough detection activator using an energy-efficient sensitivity-prioritized time series classifier, arXiv preprint arXiv:2111.04185.
[16] E. Nemati, S. Zhang, T. Ahmed, M. M. Rahman, J. Kuang, A. Gao, Coughbuddy: Multi-modal cough event detection using earbuds platform, in: 2021 IEEE 17th International Conference on Wearable and Implantable Body Sensor Networks (BSN), IEEE, 2021, pp. 1–4.
[17] T. Röddiger, M. Beigl, M. Hefenbrock, D. Wolffram, E. Pescara, Detecting episodes of increased cough using kinetic earables, in: Augmented Humans Conference 2021, 2021, pp. 111–115.
[18] P.C. Sajjad Abdoli, A.L. Koerich, End-to-End Environmental Sound Classification using a 1D Convolutional Neural Network, Expert systems with applications.
[19] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
[20] Wikipedia contributors, Mel-frequency cepstrum — Wikipedia, the free encyclopedia, [Online; accessed 5-August-2021] (2021). URL https://en.wikipedia.org/w/index.php?title=Mel-frequency_cepstrum&oldid=1021923846.
[21] L. Lai, N. Suda, V. Chandra, Cmsis-nn: Efficient neural network kernels for arm cortex-m cpus, arXiv preprint arXiv:1801.06601.
[22] K.J. Piczak, ESC: Dataset for Environmental Sound Classification (2015). 10.7910/DVN/YDEPUT. URL 10.7910/DVN/YDEPUT.
[23] E. Fonseca, X. Favory, J. Pons, F. Font, X. Serra, FSD50K: an open dataset of human-labeled sound events, CoRR abs/2010.00475. arXiv:2010.00475. URL https://arxiv.org/abs/2010.00475.
[24] E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, X. Favory, J. Pons, X. Serra, General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline, CoRR abs/1807.09902. arXiv:1807. 09902. URL http://arxiv.org/abs/1807.09902.
[25] J.F. Gemmeke, D.P.W. 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: Proc. IEEE ICASSP 2017, New Orleans, LA, 2017.
[26] T. Röddiger, D. Wolffram, D. Laubenstein, M. Budde, M. Beigl, Towards Respiration Rate Monitoring Using an In-Ear Headphone Inertial Measurement Unit, in: International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp'19), 1st International Workshop on Earable Computing Systems, 2019, pp. 48–53.
[27] S. Vogel, M. Hülsbusch, D. Starke, S. Leonhardt, In-ear heart rate monitoring using a micro-optic reflective sensor, Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings (2007) 1375–1378 doi:10.1109/IEMBS.2007.4352554.
[28] T. Wartzek, S. Vogel, T. Hennigy, O. Broderseny, M. Hülsbuschz, M. Herzogz, S. Leonhardt, Analysis of heart rate variability with an in-ear micro- optic sensor in view of motion artifacts, Proceedings - 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009 (2009) 168–172doi:10.1109/BSN.2009.19.
[29] M.Z. Poh, K. Kim, A.D. Goessling, N.C. Swenson, R.W. Picard, Heart- phones: sensor earphones and mobile application for non-obtrusive health monitoring, Proc. – Int. Sympos. Wearable Comput. ISWC (2009) 153–154, https://doi.org/10.1109/ISWC.2009.35.
[30] S.F. Leboeuf, M.E. Aumer, W.E. Kraus, J.L. Johnson, B. Duscha, Earbud-based sensor for the assessment of energy expenditure, HR, and VO2max, Med. Sci. Sports Exerc. 46 (5) (2014) 1046–1052, https://doi.org/10.1249/MSS.0000000000000183.
[31] V. Goverdovsky, D. Looney, P. Kidmose, D.P. Mandic, In-Ear EEG from viscoelastic generic earpieces: robust and unobtrusive 24/7 monitoring, IEEE Sens. J. 16 (1) (2016) 271–277, https://doi.org/10.1109/JSEN.2015. 2471183.
[32] B.D. Looney, P. Kidmose, C. Park, M. Ungstrup, M.L. Rank, Rosenkranz, Recording Concept, Ieee (december) (2012) 32–42.
[33] S.S. Birring, S. Matos, R.B. Patel, B. Prudon, D.H. Evans, I.D. Pavord, Cough frequency, cough sensitivity and health status in patients with chronic cough, Respir. Med. 100 (6) (2006) 1105–1109, https://doi.org/10.1016/j.rmed.2005.09.023.
[34] Kenneth Christofferson, Xuyang Chen, Zeyu Wang, Alex Mariakakis, Yuntao Wang, Sleep Sound Classification Using ANC-Enabled Earbuds, 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) (2022), https://doi.org/10.1109/PerComWorkshops53856.2022.9767394.