Real-Time Dynamic Physical Phenomenon Adapted Machine Learning Pipeline for Smart Acoustics

---

A Dissertation

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the requirements for the Degree

Doctor of Philosophy (Computer Science)

by

Mohsin Yusuf Ahmed

December 2019

Approval Sheet

This dissertation is submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

Mohsin Yusuf Ahmed

This dissertation has been read and approved by the Examining Committee:

Jack Davidson, Chairperson

John Stankovic, Advisor

Gabriel Robins, Member

Brad Campbell, Member

Laura Barnes, Member

Accepted for the School of Engineering and Applied Science:

Engineering Dean, School of Engineering and Applied Science

December 2019

Abstract

One vision for modern cyber-physical systems in smart environments is an underlying acoustic processing infrastructure providing various services to the residents of a smart space. Years of research on this domain have resulted in commercial in-situ and smartphone based systems which can do spoken conversation, take spoken commands, recognize speaker, and can detect agitated behavior, physiological sounds, disease sounds and body specific sounds. Based on the context, some of these systems are subject to some dynamic physical phenomena that may affect the performance of the underlying machine learning algorithms, addressing which has mostly remained an open problem. The overarching goal of this research is to utilize real-time sensing of such dynamic physical phenomena and adapt the underlying machine learning pipeline for better recognition of certain acoustic events.

To demonstrate our hypothesis, we have developed an acoustic event detection platform called LocoVocal, and three other solutions called mLung, DeepLung, and SocialSense as part of developing this thesis. LocoVocal is a novel in-situ human acoustic activity detection platform which localizes human subjects inside a room in real-time and chooses an appropriate location adapted acoustic model from a set of pre-computed models, and passes it to the application running on top of LocoVocal for improved performance. LocoVocal has been evaluated with three different applications: speech recognition, speaker identification, and distant emotion recognition to demonstrate its versatility. Two other novel solutions, mLung and DeepLung, have been developed in this thesis which act as a longitudinal monitoring service in the smartphone for pulmonary patients and

caregivers. mLung, and its deep learning counterpart DeepLung, are privacy-preserving mobile-cloud hybrid services for lung anomalous sound (like cough, wheeze) detection using classifiers in the cloud. Both systems filters speech in-phone for patient privacy, sense respiratory cycles in real-time using the inertial sensors as the phone is held by the patient's chest, and windows the audio dynamically based on the respiratory cycle for the in-cloud classifiers. Finally, we present SocialSense, which acts as a social interaction monitoring application in the smartphone detecting when a phone user is speaking, with who the user is speaking by discovering neighboring phones (assuming every phone is mapped to a unique user), and the mood of the user. By periodic Bluetooth neighbor sensing and collaboration among phones, SocialSense knows who is joining or leaving a social interaction, and maintains a non redundant classification model by keeping training instances only from people who are present in the social interaction.

To those who raised and cared
(parents, grandparents, and family)

Foreword

The seven years of apprenticeship for a Doctorate degree was quite like a roller coaster ride in a fantasy theme park. Looking back, it feels like going through a long assembly line of a factory to produce scientific minds, thus transforming from being immature and restless to somewhat grown up and focused. Along this ride, there were phases of lostness and despair while learning to steer a vessel in the ocean of unknown knowledge, no compass (let alone GPS) to navigate except stellar objects, and just hopes that a fish would voluntarily jump in the boat to provide a meal.

This journey towards the horizon of knowledge was mostly lonely and silent, passing by countless nights in Rice Hall 220 lab and later in Link Lab, living on microwave meals and digging through obscure codes and papers. Occasionally, there were lights and sparks of ecstasy, hope and some meaning to this process at the end of the tunnel when a new tool was taken mastery of, a bug got fixed, a new idea formed, some positive results found, a paper got written, submitted and accepted (even now, feels like yay!). This quest and test of time provided training and lessons strong enough to fuel and thrust the rest of this life. This degree trained to yearn for skills, values and virtues more than any materialistic glitters.

The grand designer (of this whole "simulation" that we like to think of as universe) provided resource in the form of a fatherly mentor and advisor, who was kind, patient, taught not to take shortcuts, trained to think deeply, wanted his mentees to truly learn the art of scientific process and creating knowledge, be independent, and provided a comforting hatching environment to let this skittish soul sprout and grow. Gratitude to him for his exemplary sincerity and everything.

A place to rest in this stressful time was provided by the simulation designer in this town in the form of a hillside little cricket club, where friendship was made by sharing pristine love for a silly game. Training there was hard for constant improvement and play was with heart melt passion, and later on coming back to

lab recharged for yet another week to work with sweet memories of the game (perhaps of an outstretched all-limb-in-the-air one-handed catch) and a piece of smile. The harder the agony, the harder were the slog sweeps, pulls and square cuts to wash the agony away. All these years of lessons of selflessness, composure, teamwork, discipline, and no shortcut focused hard-work for success learned across countless training and game sessions in cricket field were extrapolated not only in scientific research, but also in all aspects of life. Without this external refreshing resource that was direly needed, this PhD may not have finished.

Gratitude to the entity in the form of a loving family on the other side of the planet apart by one ocean and two continents to the east, who prayed relentlessly, loved unconditionally and were ready to provide anything they could for their son, grandson and sibling. Gratitude to all the friends and siblings from other mothers in Charlottesville, past and present peers in the research community, and all the brilliant lab mates for providing values, ideas, empathy, discussions, laughs, moments and memories. Gratitude to those who provided kind support in preparing part of this dissertation and presentation, in return for nothing.

Gratitude to the quals and PhD committee members for their kind service. Gratitude to the faculty members at UVA whose classes and lectures provided the inspiration and base for conducting core computer science research. Gratitude to all the co-authors in all papers for the scholarly contributions. Gratitude to all the teachers at all levels of education at all institutions. Gratitude to the University of Virginia for all the resources and the education, and the support staffs at CS department and engineering school for the care. Gratitude to my intern groups at Intel in Hillsboro, Oregon and Samsung Research at Mountain View, California for the time together to learn and grow. Gratitude to the general people and Governments of Bangladesh and the United States for funding and supporting my undergraduate and graduate studies for 11 years.

When a significant amount of time is passed at a place towards something which is earned over stretching one's abilities and test of time, there grows a special attachment to that time, place and objective. So is the bond for all these years

spent at UVA at this town for this degree. To the future mankind, if you happen to find this dissertation at one corner of UVA's then storage system hundreds and thousands of years from now, find me, my thoughts and my times in here.

Take care,

Mohsin Yusuf Ahmed

Link Lab, UVA

Charlottesville, Virginia

July 19, 2019

Contents

List of Tables

List of Figures

Chapter 1

Introduction

"How I wish I was like the wind,
Hearing each whisper, sound and thought,
A lonesome and wandering little wind,
Shattering all that has been sought."
- *Anonymous*

Alike the wishes of this anonymous poet to hear each whisper, sound, and thought, one vision for modern Cyber-Physical Systems (CPS) in smart environments is an underlying acoustic processing infrastructure providing various services to the residents of the smart space. Different acoustic applications in domains such as health monitoring, medical care, security, entertainment, information retrieval and e-commerce can be installed as a smart home or smartphone service for users. Sound is a ubiquitous noninvasive physical entity providing valuable inference of the activity, identity, location and physiology of a resident in a smart space. For example, every person has a unique voice, and our tones change based on our mood and emotion.

All acoustic activity detection tasks like speaker identification, emotion detection, speech recognition, agitated behavior (laughing, crying, screaming) detection, physiological sound (coughing, sneezing, yawning) detection, disease specific sound detection (wheezing for pulmonary patients) uses underlying machine

learning models to make the final classification. Years of research on this domain have resulted in commercial virtual assistants like Amazon Alexa [1], Google Home [2], and Siri [3] which can do spoken conversation, take spoken commands, and recognizes speaker. In addition, smartphone and contact microphone based active and passive acoustic sensing systems have been created for agitated behavior, physiological sound, disease sound and body specific sound detection. Based on the context, some of these systems are subject to some dynamic physical phenomena that may affect the performance of the underlying machine learning algorithms. For example, location of the human subject in a room interacting with a virtual assistant is a physical phenomenon that may affect the quality of the sound reaching the device due to room acoustic properties, and hence the application performance. Acoustic machine learning applications are all about finding patterns in a given data, hence it is important that the pattern of the training data and operational data are as similar as possible for good performance. The problem of addressing these dynamic physical phenomena for improved acoustic activity detection performance in both smartphones and in-situ home appliances (like Alexa) has mostly remained unsolved.

The overarching goal of this research is to utilize real-time sensing of such dynamic physical phenomena and adapt the underlying machine learning pipeline for better recognition of certain acoustic events. The technical goals include: real-time sensing of physical phenomena, user privacy, acoustic feature enhancement, user context, audio windowing and acoustic model adaptation.

In this chapter, we present the motivation behind this research (Section 1.1), identify the core research challenges (Section 1.2), and then state our thesis statement and provide an overview of our solution (Section 1.3), followed by a list of contributions (Section 1.4) and some caveats (Section 1.5). Finally, we describe the organization of this thesis (Section 1.6).

1.1 Motivation

If we think of all present or near future potential acoustic activity detection applications operating on an in-situ smart home (e.g. Alexa) or smartphone platform like speaker identification, speech emotion detection, agitated behavior (laughing, crying, screaming) detection, physiological sound (coughing, sneezing, yawning, snoring) detection, disease specific sound detection (wheezing for pulmonary patients), we notice that all these applications utilize a common underlying machine learning pipeline consisting of audio preprocessing and windowing, feature extraction and classification by a pre-trained model. For a streaming audio based machine learning application to perform well, it is important that the data pattern during training and operational scenario are as similar as possible.

We notice that certain acoustic activity detection tasks are affected by some dynamic physical phenomena as part of the operational context. These phenomena have either of the following two properties:

1. The phenomenon is a dynamic physical event that is the genesis of the sound, and time synchronized with the start and end of the sound to be detected.

2. The phenomenon is a dynamic physical event that directly impacts the underlying machine learning pipeline in the operational scenario.

To demonstrate the first property, snoring during sleeping is an acoustic event which is affected by the respiration cycle. Respiration is a constant dynamic physical phenomenon which is the cause and origin of snoring, and the start and end of a single snoring episode is time synchronized to the corresponding respiration cycle. For the second property, human originated sound or speech in an indoor location is an acoustic event which is affected by the human subject's location inside the room, as the sound heading to an in-situ audio capturing device (like Alexa) will be affected by room reverberation and ambient noise which depends on the human's location, and hence will adversely affect the machine learning operation to classify that speech or sound. The lack of an acoustic

activity detection platform which senses these dynamic physical phenomena in real-time and adapts the underlying machine learning pipeline is the primary reason certain applications' performances cannot be further improved. All existing systems adhere to an agnostic and static approach with regard to these dynamic physical phenomena.

To demonstrate these needs, we have developed an acoustic event detection platform called LocoVocal [4], and three other applications called mLung [5], DeepLung [6], and SocialSense [7] as part of developing this thesis. LocoVocal is an in-situ human acoustic activity detection platform which localizes the human subject inside a room in real-time. Based on the real-time human location, LocoVocal chooses an appropriate location dependent acoustic model from a set of pre-computed models, and passes it to the application running on top of LocoVocal. In an alternate run-time mode, LocoVocal creates a human location adaptive acoustic model on the fly for the end application. Two aspects of LocoVocal make it unique and ideal for indoor in-situ applications. First, the models created by LocoVocal are made from acoustic profiling of the specific room and specific human location, thus ensuring similar data pattern during model training and operational scenarios. Second, using the direction-of-arrival of sound in the microphone array in addition to real-time human localization, LocoVocal knows if a sound segment is coming from a human or non-human (machine) source. Such ability to separate human and non-human sound sources has the potential to improve safety and convenience in modern smart homes.

Two other related systems, mLung and DeepLung have been developed in this research which act as a longitudinal monitoring service in the smartphone for pulmonary patients and caregivers. mLung is a mobile-cloud hybrid service for lung anomalous sound (like cough, wheeze) detection for pulmonary patients using shallow support vector machine classifiers in the cloud. DeepLung does the same classification task, but uses deep convolutional neural networks instead. Both systems have the ability to sense respiratory cycles using the inertial sensors as the phone is held by the patient's chest, and windows the audio based on the respiratory cycle. The motivation behind this is to make sure that the distinct phases of lung events like coughing and wheezing reside in a single

event window, thus ensuring similar pattern between training and operational pulmonary events.

The final system developed in this thesis, called SocialSense acts as a social interaction monitoring application in the smartphone. The requirement is that every person involved in an in-person social interaction has to carry his own phone and provide short voice samples as training to his phone. Every phone has a binary speaker identification classifier which detects if its owner is speaking or not, and a mood detection multi-class classifier for detecting owner's mood. The physical phenomenon sensed in this context is the neighboring phones present in the social interaction by periodic scanning across the Bluetooth network. The motivation behind this is to make sure that every phone is using training samples from currently present phones only and thus avoiding any redundant training from absent people. If a phone does not have training samples from another phone, it requests it from that phone to send it through the Bluetooth network. Thus by periodic neighbor refreshing and collaboration among phones, SocialSense maintains an optimal classification model in each phone, and logs quantitative information about the social interaction and mood contagion of the conversation group.

1.2 Technical Challenges

This thesis addresses a number of key technical challenges in building the platform and applications which leverage the real-time sensing of appropriate dynamic physical phenomena and underlying machine learning model adaptation for superior acoustic event detection performance.

First, indoor acoustic environments are diverse in nature. Different rooms have different reverberation and ambient noise profiles. Building *a human acoustic activity detection platform that is capable of automatically profiling a wide variety of indoor acoustic environments and adapting the applications running on top* is challenging.

Second, human versus non-human source sensing is an important property to consider for an acoustic activity detection platform in smart homes. Given that modern virtual assistants like Amazon Alexa and Google Home can trigger home actuations like controlling doors, windows, HVAC, CCTV etc, it is a serious breach to home security if a maliciously programmed device can take human identity and controls home actuations. One challenge is therefore *a scheme to separate between human and non-human (machine) sound sources.*

Third, indoor human acoustic event detection using in-situ microphone is dependant upon the configuration of the classifier and acoustic features. A set of features which works well in close source to microphone distance may become too distorted to work well at a further distance. One challenge is therefore to *find a comprehensive set of acoustic features targeted to a particular application which are robust against the distance effect.*

Fourth, in order to facilitate a pulmonary anomaly detection mobile service, an important challenge is how to *design an intelligent architecture which would preserve patient privacy by not letting audio frames containing speech segments leave the local device, but still leverage the power of cloud and superior acoustic features for best lung sound classification performance.*

Fifth, in order to incorporate deep learning algorithms for smartphone based pulmonary anomaly detection, the challenges lie in *designing deep neural network architectures for both static and respiration cycle windowed audio, and to map acoustic features represented by low level descriptors and high level functionals into them.*

Sixth, as this entire thesis relies on sensing dynamic physical phenomena to demonstrate four different acoustic event detection platforms and applications, challenges are to *find these phenomena appropriate to the corresponding acoustic activity detection, and measuring them in real-time accurately.*

Acoustic Application	Operation	Microphone Type	Dynamic Physical Phenomenon	Sensing Modality	Actions on Machine Learning Pipeline
LocoVocal	Speech acoustic activity detection	In-situ	Human location	Skeletal tracking by Kinect	Use pre-computed models based on real-time human location
mLung/DeepLung	Lung anomaly detection	Smartphone	Respiration cycle	Inertial sensors	Window pulmonary audio based on respiration cycle
SocialSense	Social interaction detection	Smartphone	Neighboring people with smartphones	Bluetooth	Use non-redundant training instances based on periodic neighbor discovery
WhatEat	Type of food detection from eating sound	Contact	Bite/chew cycle	Inertial sensors	Window audio based on bite/chew cycle
SnoreDetect	Snoring detection while sleeping	Smart button	Respiration cycle	Inertial sensors	Window audio based on respiration cycle

Table 1.1: Example Acoustic Activity Detection Applications

1.3 Thesis Statement and Solution Overview

1.3.1 Thesis Statement

This dissertation investigates the following hypothesis:

By measuring appropriate dynamic physical phenomena in real-time using a set of secondary sensors, combined with underlying machine learning pipeline adaptation, it is possible to detect and classify certain acoustic events on a mobile or in-situ device which is highly accurate and adaptive to user context when compared to state-of-the-art static acoustic event detection systems.

1.3.2 Solution Overview

In order to prove our hypothesis, we have built a total of four systems: LocoVocal with three applications (speech recognition, speaker identification, distant emotion recognition), mLung, DeepLung, and SocialSense as parts of this thesis. In addition, we propose two more hypothetical applications which can be implemented using the idea of real-time dynamic sensing of physical phenomena presented in this thesis. These two applications are:

WhatEat: WhatEat detects the food being eaten from the biting and chewing sounds. It uses a small contact sensor consisting of a microphone and inertial sensors placed on the side of the user's cheek. WhatEat detects the biting/chewing cycle in real-time from the streaming inertial sensor data and windows the eating audio based on the biting/chewing cycles for the underlying machine

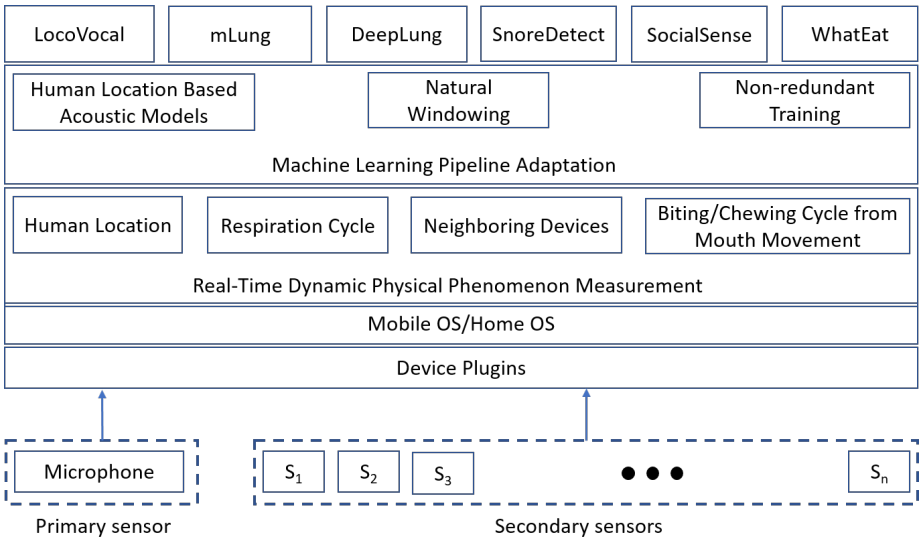


Figure 1.1: Conceptual integration of six applications: LocoVocal, mLung, DeepLung, SocialSense, WhatEat, and SnoreDetect.

learning pipeline. It is hypothesized that the biting/chewing based windowing of audio keeps the pattern of the operational audio data closer to the pattern of the training data compared to a static blind windowing of audio. A categorical classification is done to decide the type of food (e.g. chips, cookies, burger and pasta) being eaten after extracting appropriate featured from the biting/chewing windowed audio.

SnoreDetect: SnoreDetect detects snoring sounds while a person is sleeping using a smart button with microphone and inertial sensors. Like mLung and DeepLung, SnoreDetect detects respiration cycle in real-time from the inertial sensor data, and windows sleeping audio based on respiration cycle. A binary classifier makes a decision of snoring vs. not snoring after extracting features from the respiration cycle windowed sleeping audio.

Table 1.1 summarizes each of these implemented and proposed applications in terms of their operations performed, microphone type being used, the dynamic physical entity being sensed and its sensing modality, and the final action taken on the underlying machine learning pipeline. A conceptual integration of all these applications is shown in Figure 1.1.

At the bottom layer of the figure, we have the platform operating system (either

a mobile OS or home OS) connecting to the primary acoustic sensor (device microphone) and the set of secondary sensors to measure physical phenomena by the device plugins. Secondary sensors deployed in a home environment can be plugged into the platform and the device plugins layer contains the device drivers. In the next layer, the real-time dynamic physical phenomena measurement is done as appropriate to the target applications. For our six proposed applications, these physical phenomena to be measured in real-time are: human location (for LocoVocal), respiration cycle (for mLung, DeepLung, and SnoreDetect), neighboring smartphones carried by nearby people (for SocialSense), and biting and chewing cycles from mouth movement (for WhatEat).

In the next layer, the adaptation on the underlying machine learning pipeline is done as appropriate to the target application. Our conceptual platform in this layer chooses pre-computed human location based acoustic models for applications running on top of LocoVocal (speech recognition, speaker identification, distant emotion recognition) and does physiological and natural windowing of audio from patient/user respiration cycle for mLung, DeepLung, and SnoreDetect, do non-redundant precise training from people present in a social interaction for SocialSense, and windows eating audio based on biting/chewing cycles for WhatEat.

1.4 Contributions

The contributions of this dissertation are the following:

- LocoVocal, which is a novel location aware indoor human acoustic event detection platform for in-situ devices. The system was prototyped with a skeletal tracking based human localization using Kinect, and does indoor acoustic profiling in terms of synthetic reverberation simulation based on real-time human location and opportunistic ambient noise sampling to create human location specific acoustic models for end applications.

- We implement three human centered applications with LocoVocal: speech recognition, speaker identification and distant emotion recognition [8] to demonstrate the versatility of LocoVocal. Controlled experiments demonstrate that LocoVocal is 25%, 51.45%, and 10.46% more accurate for spoken word detection, spoken sentence detection and speaker identification, and up to 15.51% more accurate for distance emotion recognition, compared to state-of-the-art techniques. Home deployment of a LocoVocal prototype in a multi-person household with a speaker identification application demonstrates up to 17.30% improvement in performance in a real-world environment.

- We create the very first distance aware emotional corpuses to use in our distant emotion recognition experiments by 1) re-recording a popular emotional corpus with a microphone array with microphones placed at various distances, and 2) by injecting room impulse responses collected in a variety of rooms with various source-to-microphone distances into the same emotional corpus.

- A novel distorted acoustic feature elimination algorithm combined with best feature selection and classifier optimization techniques for distant emotion recognition, and in doing so, performing the most comprehensive feature analysis for distant emotion recognition using the largest known emotional feature set consisting of 6552 acoustic features.

- mLung, a novel privacy preserving, naturally windowed, mobile-cloud hybrid pulmonary care service for detecting unusual lung sounds like coughing and wheezing from streaming audio and inertial sensor data from a smartphone for pulmonary patients. mLung does in-phone speech detection and filtering by a lightweight classifier for patient privacy, and in-cloud lung and confounding sound classification by a heavyweight and expert supervised classifier.

- DeepLung, the deep learning counterpart of mLung, an end-to-end deep learning based audio sensing and classification framework for the same

lung anomaly detection task which incorporates 1-D and 2-D convolutional neural networks.

- Both mLung and DeepLung classifiers have been evaluated and verified by the first, extensive lung sound dataset recorded solely with a commodity smartphone without using any customized or contact microphone from 131 real pulmonary patients and healthy subjects, and annotated by professional crowd-sourcing company and expert researchers, which includes more than 45,000 instances of lung sounds, speech and other confounding sounds.

- We provide empirical evidence that mLung and DeepLung are 15%–25% and 15%-27% more accurate respectively in detecting lung sounds using only commodity smartphones when compared to a state-of-the-art smartphone based body sound detection system using specialized microphone hardware, with a best F1 score of 98%.

- SocialSense, which is an unobtrusive voice based speaker identification and mood detection system incorporating a practical, easy, and short training scheme using user's smartphone to detect a person's own speaking episode without using any on-body or contextual sensor.

- SocialSense uses a dynamic event-driven classification scheme where it performs speaker identification using a semi real-time classifier based on the periodic refresh of neighboring Bluetooth network of current users present in the scenario. This yields an average 30% increase in classification accuracy compared to static classification.

1.5 Caveats

The idea, platform and applications that are built during this research come with a few caveats:

- This whole thesis relies on real-time sensing of suitable dynamic physical phenomena to improve acoustic event detection applications by adapting

their underlying machine learning pipeline. It must be noted that not all acoustic event detection problems may be associated with a secondary dynamic physical phenomenon having properties described in Section 1.1, for which the hypothesis presented in this dissertation will not be applicable and hence will not improve the performance.

- The task of real-time sensing of dynamic physical phenomena incurs additional sensing overhead and operational cost for the acoustic event detection application. For example, practical human localization in a realistic room would require some modality having extended range and field-of-view than a Kinect, possibly room level WiFi infrastructure with each resident wearing an on-body device. Alternatively, if the acoustic activity detection system is smartphone based, the additional sensing will consume extra battery power.

- In mLung and DeepLung, we primarily focused on detecting pulmonary anomalies like coughing and wheezing. Aside from these, there are other medically defined pulmonary disease symptoms like rhonchus and crackles, which these systems cannot detect. It may happen that as more of these sounds become available to develop machine learning models, our present features and classifiers may not be good to detect them.

- We did not consider the effect of external noise while designing mLung and DeepLung, assuming they will be operational in a silent environment with minimal ambient noise. This is unlikely in a realistic noisy scenario where these systems are likely to be prone to false positives. We consider it future work.

- SocialSense assumes that every neighboring phone present in its Bluetooth vicinity (10 m) is involved in a social interaction with him, which may not be the case sometime. There may be multiple conversation groups present within a phone's Bluetooth range, and there might be phones within the range on the other side of a closed door or wall.

1.6 Dissertation Organization

The rest of the dissertation is organized as follows:

- Chapter 2 presents the state-of-the-art in technologies related to LocoVocal, mLung, DeepLung, and SocialSense systems that are developed in this thesis.
- Chapter 3 provides an overview of the LocoVocal system design and its prototype implementation.
- Chapter 4 describes evaluation of LocoVocal in terms of both controlled experiments and a real home deployment with two representative acoustic activity detection applications in terms of its ability to improve performance with real-time human localization.
- Chapter 5 describes how LocoVocal addresses distant emotion recognition problem in terms of feature and classifier enhancement, dereverberation and denoising, and training with synthetic reverberation. Also, the procedures to create first ever distance aware emotional corpus and experimental results in various acoustic environments are presented.
- Chapter 6 describes the design and architecture of mLung by providing the implementation details of the in-phone and in-cloud processing components, and the experimental evaluations.
- Chapter 7 describes the design of four different DeepLung architectures by providing the implementation details and the experimental results.
- Chapter 8 provides an overview of the SocialSense system design, its implementation, and the experimental evaluations.
- Chapter 9 concludes the dissertation by summarizing the contributions and discussing possible future work.

Chapter 2

Related Work

This chapter presents the state-of-the-art in technologies related to LocoVocal, mLung, DeepLung, and SocialSense systems that are developed in this thesis. The chapter covers existing systems that detect different types of acoustic events (Section 2.1), indoor human localization schemes which is used by LocoVocal system to localize human (Section 2.2), different pulmonary sound classification systems which are related to mLung and DeepLung (Section 2.3), various techniques for speaker and mood detection (Section 2.4) and sensing social interaction (Section 2.5) approaches that are related to SocialSense.

2.1 Application Specific Acoustic Event Detection

Acoustic sensing based on smartphone and in-situ microphone has been used in many works. Examples of speech sensing applications are speaker identification [9], emotion detection [10], stress detection [11] and conversation group detection [12]. Some applications do physiological sound sensing like cough detection [13, 14], heart beat detection using a special purpose microphone [15], or in-body sound detection [16, 17]. Systems like UbiEar [18] and SurroudSense [19] do ambient sound detection. Among in-situ systems, Resonate [20] does speaker

identification and mood detection by in-situ microphone using reverberant environment simulation, and VocalDiary [21] detects spoken voice commands for ground truth logging of home activities. Unlike the ideas presented in this thesis, none of these systems aims at sensing the dynamic physical phenomena which often originates from the sound to be classified.

2.2 Indoor Human Localization

Several systems exist that can accurately localize an individual [22, 23]. Many of these systems require the person to carry/wear a device to be localized and hence referred to as device-based systems [22]. These device-based systems use different sensing modalities such as WiFi [24–26], Bluetooth [27], ultrasound [28], GSM [29], FM signals [30, 31] etc. These systems may be practical in commercial environments, but sometimes have problems due to the so-called *forget to wear, forget to charge* problem [32]. Furthermore, many of these localization systems are also fingerprint based and hence require extensive training [23]. On the other hand, device-free systems localize an individual without requiring the person to carry/wear a device. Examples of such systems include camera-based systems [33,34], doorway tracking systems [35–37] etc. Camera based systems are subject to privacy concerns, especially given the possibility that the device could be hacked [38]. Doorway tracking systems are not vision-based and sense the person as they move from one room to another. However, they can only localize a person at a room-level granularity, which does not benefit AAD applications. Consequently, LocoVocal prototype uses a depth-sensor from a commercial device such as a Microsoft Kinect to accurately localize the person at a cm-level granularity [39] in a privacy-preserving manner without placing any onus on the person to carry/wear a device. However, LocoVocal is perfectly usable with other long range depth sensors and other device-based localization systems to tackle the limited field-of-view of depth sensors.

We point out that ours is not the first work to use a depth-sensor for localization [40, 41]. However, according to the best of our knowledge this is the

first work to use localization estimates to improve acoustic activity detection applications.

2.3 Pulmonary Activity Detection

Several shallow and deep learning based pulmonary activity detection works like wheeze detection [42–45, 45–50] and cough detection [13, 51, 52] exist in literature. For wheeze detection, various types of acoustic features have been used such as time-frequency spectrum [53, 54], entropy [55, 56], Mel-Frequency Cepstral Coefficients (MFCC) [57, 58], wavelet coefficients [59], standard deviation, peak frequency [60]. Along with these hand-crafted set of features different methods have been used for wheezing classification like Gaussian Mixture model, Neural Network [42, 43], Learning Vector Quantization (LVQ), Support Vector Machine (SVM) [44, 46], Random Forest [47] and Hidden Markov Model (HMM) [48–50].

For cough detection, [61] used a contact microphone on the throat with 91% sensitivity and 94% specificity, and [13, 51] used contact microphones on the chest with similar results. In [52], a free-field necklace microphone was used to collect cough data from 15 different pulmonary patients, which was used to train the models with 91% sensitivity and 99% specificity. [62, 63] used probabilistic neural networks with spectral and temporal features for cough sound detection. SymDetector [64] is a smartphone based system for detecting respiratory events like cough, sneeze, sniffle and throat clearing using a hybrid decision tree and SVM classifiers. Unlike mLung and DeepLung, none of these address the real-time respiration rate of the patient while detecting pulmonary activities, and none of these has been evaluated with a dataset as extensive as ours collected solely with commodity smartphone from 131 real patients and healthy subjects.

ResApp [65] provides smartphone based diagnosis and management for respiratory diseases. They use the smartphone microphone to passively capture and analyze the cough and breathing sounds to diagnose and measure the severity of pneumonia, asthma, bronchitis and chronic obstructive pulmonary disease.

They have focused on cough sounds of an extensive number of pediatric patients for diagnosis of acute respiratory illnesses in children [66], childhood pneumonia diagnosis [67–69], automatic cough segmentation in pediatric wards [70], wet and dry cough separation among children [71], and croup diagnosis among children from cough sounds [72]. ResApp solutions mainly focus on passive cough sound analysis on smartphones, while mLung and DeepLung detect wheezing too. Also ResApp solutions do not consider active respiratory cycle sensing and patient privacy for their solutions, which mLung and DeepLung do.

2.4 Speaker Identification and Mood Detection

Many of the existing speaker identification systems require the total number of speakers to be static, and they employ static classification schemes so that each speaker needs to train the system beforehand, which makes them less realistic [73]. SocialWeaver [12] uses a multi-level classification for speaker identification. The first level uses energy histogram classifiers while the second level uses a GMM based classifier. Neary [74] uses similarity of sound environment to detect conversational fields. These energy and loudness based approaches have greater error in noisy conditions and they fail if there is a person present in the scenario without his phone. SpeakerSense [9] is a speaker identification platform built on a heterogeneous multi-processor architecture. It attempts to reduce training overhead by training from real life events as phone calls and one-to-one conversations, but does not evaluate the system in noisy environments. Also, it requires the total number of speakers to be static and does not support realistic dynamic environments where speakers enter and leave on the fly.

There are a number of existing systems which detect user's mood from voice. EmotionSense [10] provides dual systems for speaker identification and emotion detection from user's voice using Gaussian mixture methods. [75] provides SVM based classifiers that recognize human emotional states from their utterances. However, these systems can only capture mood of a single person or entity and, therefore, are not suitable for social psychology experiments where a system would

need to know moods of everyone in a social interaction. Also, there is no evidence that these systems would operate well under real life noisy environments.

2.5 Sensing Social Interaction

Besides speaker identification and mood detection, there have been systems which detect other aspects of social interaction using different modalities. Some of the existing work on social interactions uses only on-body sensors such as accelerometers, gyroscopes, GPS, microphones, and cameras. Pierluigi et al. [76] built a badge having a triaxial accelerometer and a JPEG camera which is used to detect the presence of other people. Crowd++ [77] estimates the number of people talking in a certain place by unsupervised machine learning technique from smartphone audio inference. CenceMe [78] can automatically detect activities of individuals and share the sensing results through social networks.

Another type of work uses ambient sensors. [79] uses a sociometric badge equipped with infrared transmitter/receiver and microphone which senses and models human networks. In [80] four video cameras and audio collectors are placed in public areas such as the dining room, living room and hallway which can detect high-level social interactions among people such as greeting, standing conversation, and walking together.

2.6 Summary

This chapter presents the state-of-the-art in technologies related to LocoVocal, mLung, DeepLung, and SocialSense systems that are developed in this thesis. We have covered some systems and algorithms that are related to LocoVocal's human localization for acoustic activity detection, mLung and DeepLung's pulmonary anomaly detection technique, and SocialSense's collaborative speaker and mood detection technique.

Chapter 3

LocoVocal Design and Prototype Implementation

A realistic physical phenomenon when home residents interact with in-situ microphone based systems like commercial virtual assistants is their movements and locations inside the room. Based on human location, the quality of sound reaching the device varies because of room reverberation and ambient noise. In this chapter, we describe how LocoVocal addresses such issues to improve the performances of acoustic applications running on top of it. This chapter presents the detailed design and prototype implementation of LocoVocal, while the next chapter presents the experimental results.

The rest of this chapter is organized as follows. Section 3.1 discusses the motivation of using LocoVocal. Section 3.2 lists the challenges in designing a system like LocoVocal. Section 3.3 lists the technical contributions that this chapter makes. Sections 3.4 provides an overview of the LocoVocal prototype implementation. Section 3.5 provides detailed technical components of the real-time human localization. Section 3.6 provides the solution for indoor acoustic profiling and acoustic model adaptation of the end application. Section 3.7 discusses some aspects of the solution and finally Section 3.8 provides a summary of the chapter.

3.1 Motivation

Human acoustic activity detection (AAD) is rapidly gaining popularity in modern smart home solutions (Amazon Echo [81], Google Home [2]), with various applications utilizing speaker identification, speech recognition, speech emotion detection, physiological sound detection (coughing/sneezing) and agitated behavior detection (laughing, crying, screaming, curse words). These acoustic applications assist in many ways including the security, authentication, home healthcare and virtual assistance for the residents of a smart home.

In any human-centric indoor AAD application, the quality of the captured sound by an acoustic sensor depends on the human subject location in the room. Indoor environmental parameters like room reverberation, ambient noise and received sound intensity change with human subject location and affect the captured sound quality. Given that the sound capturing microphone is situated in a certain place of the room, the motion and location switching of the sound originating from a human subject create a time-sensitive dynamic acoustic profile in the room and affect the performance of the application. Most AAD applications use static machine learning models generally trained by clean sound/speech, and therefore the performance is adversely affected due to the discrepancy in static training data and dynamically captured run-time data due to human location and motion. In this chapter, we show that human-centric AAD application performance can be improved by real-time human localization and time-sensitive dynamic AAD model adaptation based on human location.

State-of-the-art AAD literature uses two kinds of sensing in general: invasive and non-invasive. Invasive systems require the human subject to carry or wear a microphone enabled device (smartphone [7, 82], smartwatch [83]) at all times and are not convenient for long-lasting AAD applications due to device energy constraints and extra overhead for human subjects. Non-invasive systems [20, 21], on the other hand, use a microphone deployed at a certain place in the room. While this may be practical for long-lasting AAD applications, they do not localize human subjects and adapt their models for improved AAD performance. In addition, these systems, along with commercial products like Amazon Alexa and

Google Home, cannot separate between human and non-human sound sources, and therefore have been reported to cause inconvenience being triggered by commands coming from TV advertisements [84]. Given that Amazon Alexa and Google Home can trigger home actuations like controlling doors, windows, HVAC, CCTV etc, it is also a serious breach to security if a maliciously programmed device can take human identity and controls home actuations. With real-time human localization, AAD systems with microphone arrays will be able to separate human and non-human sound sources for improved safety and convenience, in addition to improved performance.

3.2 Problem Demonstration with Alexa

To demonstrate the extent of the problem, we performed distant isolated word recognition experiments with a state-of-the-art, popular, and widely used commercial far-field voice controlled home automation virtual assistant, Amazon Alexa with the Amazon Echo Dot device [81]. The Echo Dot device listens to spoken sentences from across the room with a circular microphone array of seven microphones, beam-forming technology and noise cancellation and communicates with the cloud based virtual assistant Alexa for the natural language processing task. These commercial products are not yet capable of real-time human tracking and localization and separating between human and non-human sound sources.

During the experiment, Alexa was tasked with recognizing 10 spondaic words, i.e. two syllable words with equal stress on each syllable from 6 different distances ranging between 2 m to 12 m. We installed the Alexa app in a smartphone where we could see the real time speech-to-text transcription result by the device. If a word was not recognized correctly on the first attempt, it was repeated. If either the first or the second attempt was correct, the word was considered as correctly recognized. If both attempts were unsuccessful, the word was considered as not correctly recognized. Figure 3.1 demonstrates Alexa spondaic word recognition

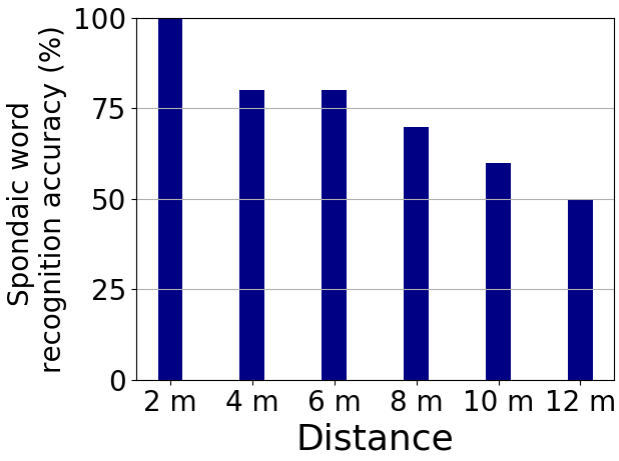


Figure 3.1: Alexa isolated word recognition accuracy drops to 50% at a distance of 12 meters.

accuracy at different distances, with accuracy dropping as low as 50% at 12 meter distance.

With such observations from a state-of-the-art commercial product, we hypothesize that had the Amazon Alexa service use dynamic human location aware physical models for its speech recognition, it would have been able to significantly improve its deficit in speech recognition for speech coming from moving human subjects in increased distances. In addition, it could have been able to detect if a command is coming from a human or non-human source for improved convenience and security. Although we were not able to use Alexa models for LocoVocal experiments since Alexa models are not publicly accessible and customizable, but we conducted LocoVocal experiments using other customizable AAD models for three different AAD applications and were able to validate our claim by improving their performance for moving human subjects.

3.3 Challenges

In this chapter we present LocoVocal, which is the first solution to propose the idea of real-time human localization for dynamically adapting AAD models with human location in a time-sensitive manner for improved performance of AAD applications in a real-world setup. Such real-time AAD model adaptation

based on human location is challenging due to several practical reasons. First, different people may move/walk at different speeds. Yet they have to be localized accurately in real-time for the end AAD application to update its training model in real-time accordingly. Second, there can be one or multiple human subjects present at any time and the AAD application should detect the sound-originating human subject. Third, there should be schemes for separating human and non-human (machine) sound/speech for safety and convenience issues. Finally, the target AAD application may be operational in different rooms with different acoustic properties (reverberation, ambient noise) and should be able to adapt its models with the acoustic profile in a real-time and time sensitive way.

3.4 Contributions

The contributions of LocoVocal are the following:

- Proposing and demonstrating the novel concept of real-time human localization with traditional AAD applications (e.g. speech recognition, speaker identification) for improved performance.
- Novel sound source detection in presence of multiple human and non-human sound sources by real-time human localization for improved security and convenience.
- Real-time human location based real-time acoustic profiling in terms of synthetic reverberation simulation and opportunistic ambient noise sampling for improved performance.
- Detailed sensitivity analysis of acoustic parameters to better understand operational behavior.
- A novel algorithm for auto-calibrating AAD model bank locations by Rapidly-Exploring Random Tree exploration under constraints.
- Two time-sensitive real-time variants of LocoVocal are proposed and experimented with: pre-computed model, and run-time model.

- Controlled lab experiments with two representative AAD applications are conducted that demonstrate up to 25% improvement in spoken word detection, 51.45% improvement of word error rate of spoken sentence recognition, and 10.46% improvement in speaker identification compared to the Resonate [20] system.

- A 2-person home deployment demonstrates its robustness in real-world adverse environment with 10.71%-17.30% improvement in speaker identification.

3.5 LocoVocal Conceptual Prototype Implementation

Commercial depth sensors, like the Microsoft Kinect [85], are capable of real-time human localization at a cm-level precision by skeletal joint tracking using a non-invasive infrared camera. Being inspired by its potential, we used a Kinect sensor for prototype implementation of LocoVocal, as shown in Figure 3.2. It should be noted that, Kinect is used for just this particular prototype, but LocoVocal can be implemented with any other human localization system using other modalities like WiFi, Bluetooth, FM, radar etc.

The *human and audio sensing* unit of LocoVocal prototype uses the real time skeletal tracking data from the Kinect sensor to localize the 3D coordinates of the head of the human subject. In addition, it also measures the direction-of-arrival of any incoming sound stream into the Kinect using beamforming in microphone array. The direction-of-arrival is useful to detect the source of sound in a multi-person scenario and when non-human sound sources (like TV) are present. The ability to separate human and non-human sound sources is novel and not present in state-of-the-art AAD systems both in literature [20, 21] and in commercial products like Amazon Echo or Google Home. Although these products use beamforming using their microphone arrays, they can detect only the source of the sound, but cannot detect if the source is human or machine.

The *acoustic profiling* unit does a real-time profiling of the indoor acoustic environment by a simulated representation of reverberation and opportunistic ambient noise sampling. This profiling can be done either offline or online, making LocoVocal operable in 2 different time-sensitive modes: *pre-computed mode* or *run-time mode*, respectively. The reverberation simulation takes different room-specific parameters, microphone location and real-time human location as its input configuration. The idea of real-time acoustic profiling based on real-time human location information is novel and hasn't been attempted before.

The *application model adaptation* unit uses this acoustic profiling to adapt the AAD training models running on top of LocoVocal. The real-time human location specific model is chosen for the application and incoming streaming audio intensity is adjusted based on human location for improved application performance.

As said earlier, LocoVocal can be configured in 2 different time-sensitive modes. In the ***pre-computed mode***, the model locations are predefined and model for each location is built before the system becomes operational, and models are used on-demand based on real-time human location. This is the fastest mode, but has the overhead of defining model locations and computing models in advance. In the ***run-time mode***, models are built from scratch on the fly based on real-time human location, and hence this mode is slower than *pre-computed mode*, but has no operational overhead.

3.6 Human Sound Source Localization

3.6.1 Skeletal Tracking and Localization by Kinect

LocoVocal prototype uses the streaming skeletal tracking data from the Kinect sensor to localize humans. Kinect can track 20 different body joints from skeletal data accurately in 3D coordinates with respect to a frame of reference centered at Kinect. LocoVocal localizes the human originated sound source by real-time

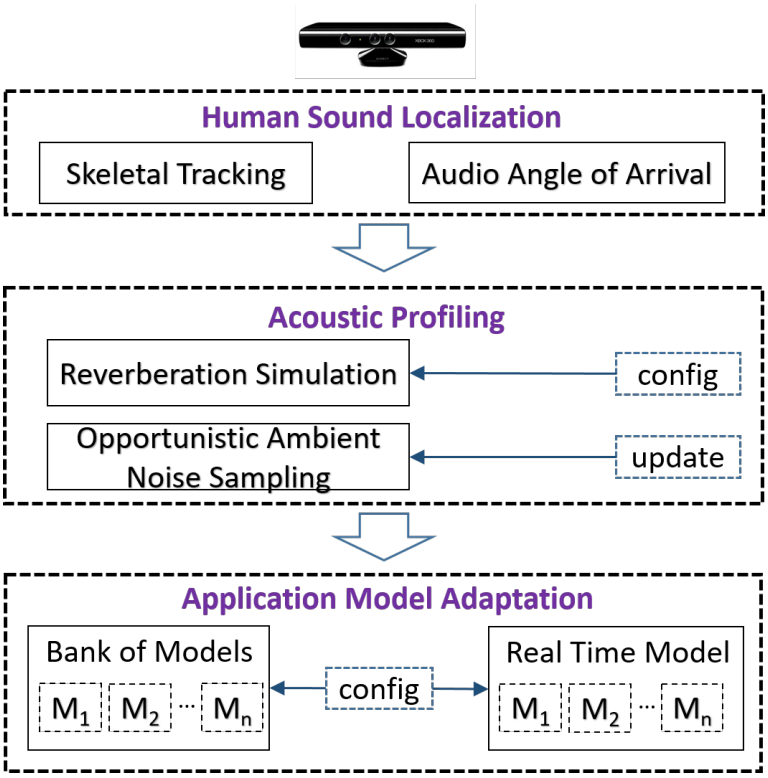


Figure 3.2: LocoVocal prototype architecture.

tracking of the skeleton head, as the head is the source for generating any speech or non-speech (laughter, cry, cough, sneeze) human originated sound.

The Kinect depth sensor is comprised of an infrared (IR) ray emitter and an IR sensor. The IR emitter emits IR beams to the distant human body (or any other obstacle) and the IR sensor reads the reflected beams. The reflected beams are converted to depth data which measures the depth or distance between the human body and sensor. Kinect can track up to 6 humans and among them, 2 humans can be tracked in detail for their 20 different body joints. However, all 6 humans can be localized in 3D coordinates.

3.6.2 Differentiating Multiple Human and Non-human Sources

The skeletal tracking is sufficient for source localization when only one human subject is detected as the only human is the obvious source for any human-originated sound. However, in scenarios where multiple human subjects and possibly non-

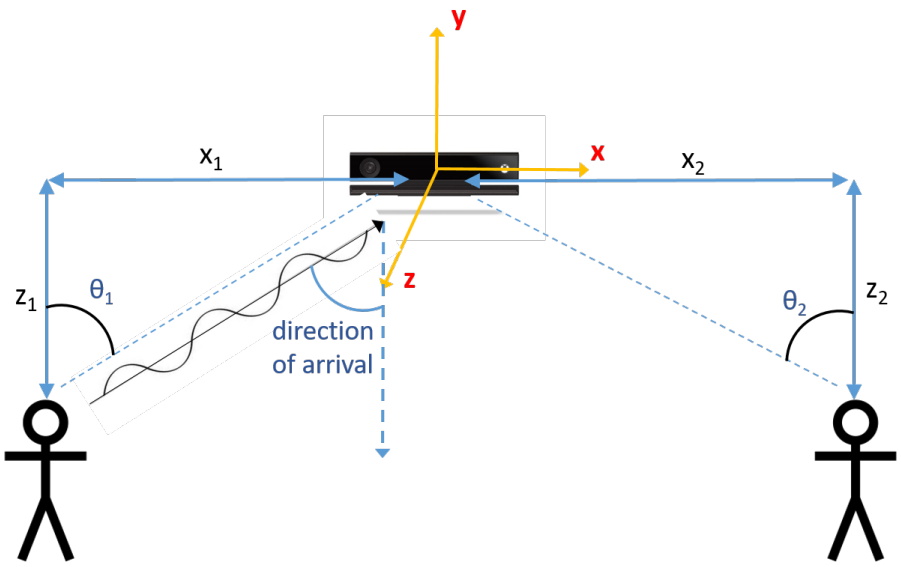


Figure 3.3: The direction of arrival of the incoming sound beam to the Kinect is mapped to the angular position of each 3D localized skeleton ($\theta = \arctan \frac{z}{x}$) to determine the speaking human. If sound is coming from a direction where no skeleton is found, then it is considered as non-human sound.

human sound sources are present, the system should detect which human is generating the sound, and ignore the source if it is non-human. LocoVocal solves this problem by measuring the direction-of-arrival of the incoming sound beam and mapping it to only one sound source, either human or non-human (like TV).

The Kinect audio capture system consists of a 4 microphone array, and can do beamforming by measuring the direction-of-arrival of an incoming sound beam by comparing the time difference of signal arrival in each pair of microphones. LocoVocal computes the angular position of each tracked skeleton with respect to the Kinect frame of reference from its 3D coordinates and by mapping angular position of each skeleton to the direction-of-arrival, it detects and localizes the sound originating human subject. If no skeleton is found along the sound direction-of-arrival, then it is considered as non-human sound and ignored. Commercial devices like Amazon Alexa products are not capable of separating between human and non-human sound sources, and therefore are prone to inconvenience and potential security threats.

3.7 Indoor Acoustic Profiling

LocoVocal attempts to minimize the discrepancy between training and run time acoustic data by providing a real time human location dependent acoustic profile of the room environment to the target AAD application. The end application uses this acoustic profile to adapt its training models and classifiers to improve performance in an operational scenario.

A source sound signal, $x(t)$, when sampled or recorded by a distant microphone is affected by room reverberation and noise. If the distant recorded signal is $y(t)$, then

$$y(t) = x(t) * R + n(t) \tag{3.1}$$

Where, $n(t)$ is the ambient noise, R is the room impulse response, and * is mathematical convolution. Room impulse response characterizes the acoustic properties of a room in terms of reflection and sound propagation between two given source and microphone locations, assuming that the room is a stable, causal linear time-invariant system. R is therefore a function of room configuration and source and microphone locations. When a human subject moves and speaks or generates any other sound, the room impulse response varies because of his change of location, causing dynamic reverberation effect in the captured microphone signal $y(t)$.

LocoVocal artificially simulates a set of room impulse responses from the real time localization of the human subject and room specific parameters. In addition, LocoVocal opportunistically samples and updates the room ambient noise $n(t)$ when there is no human subject present in its field-of-view. LocoVocal passes this acoustic profile tuple $R, n(t)$ to the target application, which then converts its clean speech training samples $x(t)$ to room acoustic and human location adaptive training samples $y(t)$ using Equation (3.1), and builds location adaptive models from $y(t)$. Finally, when the application is running with the updated

model and capturing sound, LocoVocal adjusts the received human-originated sound intensity based on the real-time location of the human subject.

3.7.1 Artificial Room Impulse Response Simulation

Room impulse response can be artificially simulated using the image source model (ISM) technique [86]. The ISM method calculates sound behavior using ray propagation assumption, where the sound beams travel directly from the source to receiver, and also travels indirectly by specular reflection from room walls. The rays can then be extrapolated as sound coming to the receiver from two sources, the second source being the image source behind the hypothetical mirror. The image source position can be calculated from the source and reflector position and angle. The straight line distance between the image source and the receiver is used to model the actual reflected sound path. There is an image source for every reflective surface, and a second order image source can be formed by the combined reflection off two surfaces. This can be extended to a desired number of sources for better room impulse response simulation.

LocoVocal uses a fast real time implementation of the ISM method [87] to simulate room impulse response R in Equation (3.1). This method requires the following room specific parameters: i) room dimensions, ii) microphone position, iii) human source position, and iv) Reverberation time T_{60} of room.

The reverberation time T_{60} of a room is the time the sound energy takes to decay by 60 dB after the source stops emitting. There are methods for measuring T_{60} if the room impulse response is known. In our case, we estimate the T_{60} using a statistical decay model proposed in [88].

These parameters have to be measured only once for each room where LocoVocal is operational, and can be done easily in a short time and hence practical. The room dimensions and the microphone/Kinect position in the room can be measured easily with a measuring tape or even faster with a laser range finder device. T_{60} can be measured by a blind estimation method from a short sound

clip recorded in the room, or more accurately with a commercial measurement device [89].

3.7.2 Opportunistic Ambient Noise Sampling

There can be various types of indoor ambient noise present where a human-centric acoustic activity detection application is operational, like from HVAC, refrigerator/freezer or a desk/ceiling fan. LocoVocal opportunistically samples the ambient noise $n(t)$ (in Equation (3.1)), when no skeleton is detected in Kinect, and passes the updated $n(t)$ to the application.

3.7.3 Adjusting Received Sound Intensity with Localized Source

The received sound intensity for an application decreases inversely proportional to the squared distance from the sound source. If intensity is I at distance r from source, then

$$I \propto \frac{1}{r^2} \tag{3.2}$$

LocoVocal adjusts the received signal intensity by amplifying it by the square of the distance from the human source, as measured by Kinect localization. This way, the perceived signal intensity is closer to the application training signal intensity, even though the training is updated and the model has been adapted.

3.7.4 Scaling of R and n(t)

Once the simulated R and sampled $n(t)$ is passed to the application by LocoVocal, the application updates its training instances using Equation (3.1), and scales the R and $n(t)$ components in the adapted training instances as follows:

$$y(t) = \alpha[x(t) * R] + (1 - \alpha)n(t); 0 < \alpha < 1 \tag{3.3}$$

α is the scaling parameter and it should be set based on indoor acoustic conditions. For rooms having high reverberation effect, α is set to a large value to put more weight in the R component during training model adaptation, and set to a small value in case of high ambient noise.

3.8 Discussion

3.8.1 Improve Commercial Virtual Assistants

We performed speech recognition experiments with the Amazon Alexa device with moving human subjects at different distances and observed that, for locations up to 5-meters distance, the device was very accurate in speech recognition with a 0 word error rate. Between 5-meters and 7-meters distances, the word error rate was 11%, and beyond 7-meters the word error rate increased to 22%. In addition, we performed another experiment from a 6-meter distance with ambient noise coming from water boiling in an electric kettle, where the word error rate increased to 33%. We also noted that, to avoid potential greater error rate for speech coming from increased distances, the device was more prone to ignore such speech instead of providing an output with more errors. In addition, the product cannot detect if a command is originating from a human or non-human source which is a potential for security threat and inconvenience. It is possible to improve these products at greater distances using location aware concept presented in LocoVocal, and make these products security-preserving and more convenient.

3.8.2 Range

Structured light based depth sensors like the Microsoft Kinect has a range of operation (4 m for Kinect) where they can operate. In addition, as Microsoft has

permanently ceased Kinect production, they will be out of the market in the near future. There are several other long range commercial depth sensors available like Intel RealSense [90], Zed stereo camera [91], Orbbec 3D camera [92], VicoVR [93] which have ranges up to 20 meters. Some of these products (VicoVR, Orbbec) come with a built-in human body tracking SDK while the others provide raw depth stream data only from which custom human body tracking is possible. All of these are usable in place of Kinect in the LocoVocal prototype. In addition, LocoVocal can be used in an extended range using wireless (WiFi, GSM, FM) signals, and computer vision technologies.

3.8.3 Field-of-view

Kinect and all other commercial depth sensors have a limited field-of-view (55 degrees for Kinect), which means if placed at the center of a room, it cannot track and localize human subjects beyond this field-of-view in a 360 degree space. If bound to depth sensor based solution, one solution is to use a circular array of several depth sensors to cover the entire 360 degree, which may need as many as 4 Kinects. An alternate is to use Lidar technology for 360-degree human tracking [94]. However, cost is a concern for the very expensive commercial Lidar sensors generally used in autonomous vehicles. If using wireless localization, field-of-view is not an issue.

3.8.4 Reverberation Time

We used a blind estimation technique to estimate T_{60} which is not always fully accurate, and this technique can estimate T_{60} up to 1.2 seconds. More accurate measurement of T_{60} is especially important in highly reverberant environments. There are commercial instruments [89] available to more accurately measure T_{20} or T_{30} from which $_{T60}$ can be interpolated.

3.8.5 Obstacles

There can be furniture and other obstacles present in a room within LocoVocal's range and hence those locations cannot be a candidates for model banks. If a floor map of the obstacles is provided to the RRT simulation and model bank auto-calibration, the algorithm can make sure that no RRT edge intersects an obstacle region, and no selected vertex for a candidate bank location lies on an obstacle. Thus, the system can automatically decide the model bank locations meeting all constraints and avoiding the obstacles.

3.9 Summary

This chapter provides an overview of the design and prototype implementation of LocoVocal. We propose the idea of sensing a physical phenomenon like real-time human location to dynamically adapt AAD acoustic models for improved performance. We describe the conceptual prototype implementation of LocoVocal using a Kinect based human skeletal tracking, human versus non-human source separation using the sound direction of arrival, and human location based acoustic model adaptation using room impulse response simulation and opportunistic ambient noise sampling.

Chapter 4

LocoVocal Evaluation

In this chapter, we evaluate LocoVocal by both controlled experiments and a real home deployment with two representative AAD applications in terms of its ability to improve AAD performance with real-time human localization. LocoVocal uses human location specific customized acoustic models for improved sound/speech activity detection performance. To do that, LocoVocal does indoor acoustic profiling in terms of artificial room impulse response simulation, opportunistic ambient noise sampling and sound intensity adjustment as described in chapter 3. We demonstrate the ability of LocoVocal in the improvement of speaker identification and speech recognition by various controlled experiments in a conference room and home deployment in a 2-person household. We also compare LocoVocal's performance with that of the state-of-the-art solution Resonate [20] which uses static physical environment adaptation for AAD applications. We also analyze real-time properties of LocoVocal in the run-time mode. Finally, we discuss a method for auto-calibration of LocoVocal model bank locations instead of pre-computed bank locations.

4.1 Controlled Experimental Setup

For the controlled experiments, we evaluate LocoVocal prototype with a speech recognition application and a speaker identification system set up in a conference room, thus demonstrating its versatility for these 2 AAD tasks.

4.1.1 Speech Recognition

CMUSphinx Tool

We set up the CMUSphinx [95] speech recognition tool as our acoustic application for our experiments. We chose CMUSphinx as this provides real-time model adaptation capability compared to other alternatives (Microsoft Speech). CMUSphinx provides a tool named SphinxTrain to train acoustic models for any language or indoor environment, which LocoVocal uses to adapt CMUSphinx default acoustic models. An acoustic model in a speech recognition system characterizes how sound in speech changes over time. Each speech sound, named phoneme, is modeled by a sequence of states and signal observation probabilities, which describes the likely distribution of sounds in that state.

CMUSphinx also needs a *dictionary model*, which is a decoder with a large list of words with a sequence of phonemes, which describe the pronunciations of the words. The acoustic model detects the phonemes in speech from speech features and try to match their order to a specific word in the dictionary. Every language has a specific set of phonemes with which all the words are made of. For example, the pronunciation of the word "Eleven" in terms of phonemes in American English is: `AX L EH V AX N`

Finally, CMUSphinx needs a *language model*, which describes the probability of next word sequence once a sequence is detected. LocoVocal dynamically adapts the acoustic model of CMUSphinx based on room profile and human subject location, and uses the default English dictionary and language models.

Word ID	Word	Sentence ID	Sentence
w1	forward	s1	Shall we start?
w2	reverse		
w3	right	s2	I like sugar in my coffee.
w4	left		
w5	up	s3	He knows my number.
w6	down		
w7	stop	s4	It is raining outside.
w8	move		
w9	jump	s5	Drinking milk is healthy.
w10	rotate		

Table 4.1: Words and sentences in test corpus

Test Corpus Creation

We made a corpus consisting of 5 sentences and 10 individual words for our speech recognition experiments. These words and sentences of our test corpus are listed in Table 4.1.

CMUSphinx acoustic model adaptation requires some speech samples provided to it in the operational environment as initial training. We recorded the test corpus words and sentences near the microphone with good sound quality in the conference room where experiments were performed, and provided as training samples as .wav files. However, any speech samples (not necessarily the test corpus samples) with enough phonetic variations would suffice as training.

The evaluation metric used for spoken word transcription is average accuracy, where,

$$Accuracy = \frac{Number\ of\ words\ correctly\ transcribed}{Total\ number\ of\ words} \tag{4.1}$$

For sentence transcription, word error rate (WER) was used as the metric. If the original sentence has N words, and the transcribed sentence has I inserted words, D deleted words, and S substituted words, then,

$$WER = \frac{I + D + S}{N} \tag{4.2}$$

4.1.2 Speaker Identification

We implemented an off-the-shelf real time speaker identification system for our second AAD application. We trained the system with speech samples from 3 persons. Mel-frequency cepstral coefficient features with cepstral mean normalization were extracted from the samples and were used to train a Gaussian mixture model (GMM) for each speaker. During real time experiments, captured live speech from a test corpus made of the 3 persons is filtered through an energy threshold based silence filter and a voice activity detection filter [96] to get rid of silence and non-human sounds. After all this filtering, the resulting human speech is segmented into 5 second windows, and log-likelihood ratio of each segment is calculated for each speaker's GMM model. The corresponding speaker of the GMM model having highest log-likelihood score is assigned to the segment.

4.1.3 Room and Acoustic Model Bank Setup

Experiments were performed in a conference room with dimensions 7.19 m x 5.4 m x 2.7 m. The room had moderate ambient noise and reverberation effect. The Kinect was set in the center of the room on a table and the height was adjusted with a tripod to see standing and walking human subjects in its field-of-view. A sample of room ambient noise was recorded, and the position of the Kinect in the room in terms of 3D coordinates were noted.

We set up 9 different spots for acoustic model banks ranging between 1-3 meters in varying angles within the Kinect field-of-view. We chose these 9 locations as they evenly covered the entire field-of-view neither being too close nor too far. The banks are named from a to i, as shown in Figure 4.1. The bank locations were marked down for ease of visualization during the experiment. The training samples recorded in the room were passed to the T_{60} estimator, and T_{60} was estimated to be 0.44s. The floor of the room was carpeted which is mostly sound absorbing, and the walls and ceiling had moderate reverberation property.

The Kinect position, location of each of the 9 banks, estimated T_{60}, absorption coefficients of the room surfaces, and the room dimensions were passed to the ISM simulator and simulated room impulse response for each of the bank location was obtained. These impulse responses were convolved with the clean training speech samples of CMUSphinx and the speaker identification tool, and were added with the sampled room ambient noise. An α scaling factor was used as in Equation (3.3), and it was set to 0.5, as both the room ambient noise and reverberation were moderate.

The resulting and transformed .wav files for each bank are the basis for CMUSphinx and speaker identification model adaptation. We used several SphinxTrain tools to extract features from the newly transformed CMUSphinx .wav files and adapt its acoustic model for each of the bank locations for this particular conference room. For the speaker identification tool, we retrained and adapted the GMM models for each location bank from the transformed training files.

4.2 Controlled Experimental Results: Pre-computed Mode

Our controlled experiments were done in a conference room with the two AAD applications running on top of LocoVocal: speech recognition and speaker identification. These experiments were done with both standing and walking human subjects while LocoVocal was switching models real-time, and with 1-person, 2-person and non-human sound source (loudspeaker) present. Also, experiments were done with ambient noise from a running table fan to demonstrate how LocoVocal improves AAD performance by periodic noise sampling. A number of system specific parameter (T_{60}, room dimension, source-microphone distance, α) sensitivity experiments were done to better understand system behavior with regard to these parameters. Finally, LocoVocal performance for the two AAD applications were compared with the Resonate system to demonstrate how real-time human location based physical model adaptation improves AAD performance from static physical model adaptation. Most of these experiments were done

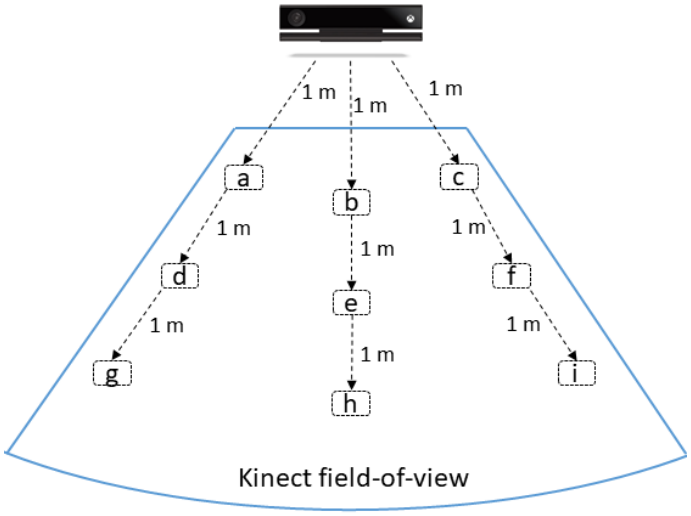


Figure 4.1: 9 different acoustic model bank locations in the Kinect field-of-view.

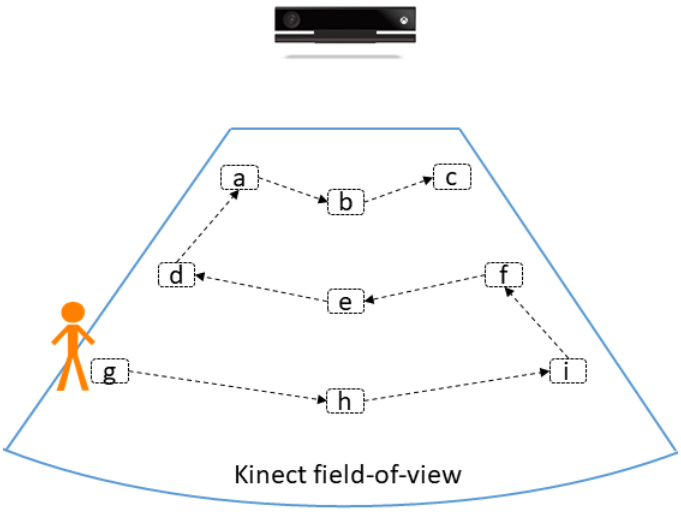


Figure 4.2: User route, starting from bank *g*, across all 9 bank locations in walking-and-talking stress test.

in pre-computed mode, while the latency of run-time mode was measured and deadline for model switching was defined for different walking speeds.

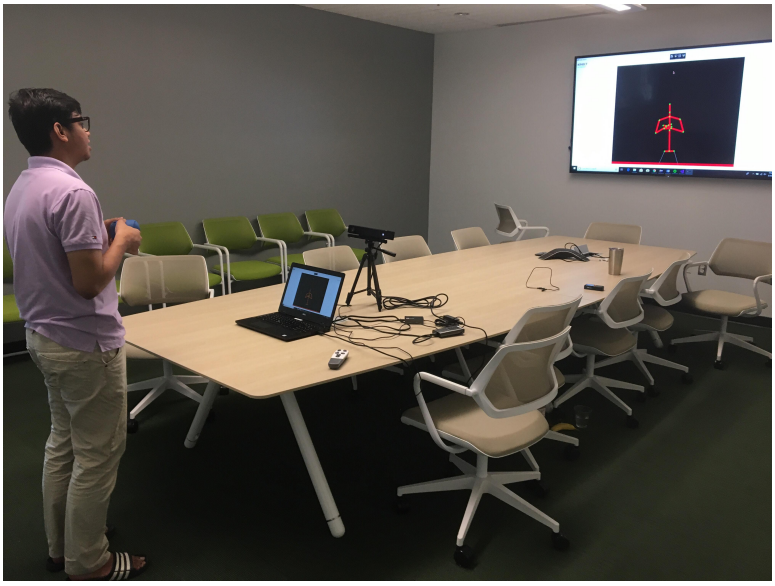


Figure 4.3: A subject playing the test corpus with a Bluetooth loudspeaker during Walk-Stop-and-Speak test.

4.2.1 1-Person Walk-Stop-and-Speak Test

In this experiment, we tested LocoVocal's skeletal tracking and the word, sentence transcription, and speaker identification accuracy in different bank locations for a single human subject, as the person walks to different banks, stops, and speaks the test corpus. The LocoVocal system was configured with the 9 acoustic model banks as in Figure 4.1. The subject carried a Bluetooth loudspeaker with him, and walked to each of the 9 bank locations one-by-one, as in Figure 4.3. Once he reached the bank location, he stopped and played the whole test corpus with the loudspeaker, then moved to the next bank. The bank markings on the floor helped the subject to move to correct bank location. The reason behind playing recorded corpus by loudspeaker instead of subject directly uttering them was for spoken input consistency across all locations. Each experiment was repeated 3 times for both speech recognition and speaker identification.

For baseline performance comparison, we used a basic model with no acoustic adaptation. We repeated the walk-stop-and-speak test with the baseline model and no model switching. Tables 4.2 and 4.3 show the WER and accuracy of sentence transcription, word transcription and speaker identification for 1-person walk-stop-and-speak test for location adaptation, intensity adjustment and the

baseline.

It is evident from Table 4.2 that sentences s2 and s5 were harder for the CMUSphinx tool to transcribe compared to the other 3 sentences, resulting in higher WER for these 2 sentences even after location adaptation and intensity adjustment. However, regardless of the varying difficulty of transcribing the 5 different sentences, LocoVocal was able to improve the WER for all 5 sentences compared to the baseline system. Similarly, LocoVocal improved the word transcription and speaker identification accuracy for each of the 9 bank locations for the 1-person walk-stop-and-speak test.

Sentence ID	**Model type**	**Word error rate for each bank**								
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
s1	Baseline (no adaptation)	0.33	0.11	0.11	0.66	0.66	0.56	0.78	0.89	1
	Location adapted	0	0	0	0.61	0	0	0.33	0.22	0.44
	Intensity adjusted	-	-	-	0	0	0	0.22	0.08	0
s2	Baseline (no adaptation)	0.66	0.78	0.55	0.83	0.83	0.78	0.78	0.88	0.83
	Location adapted	0.44	0.55	0.38	0.61	0.67	0.66	0.61	0.72	0.83
	Intensity adjusted	-	-	-	0.22	0.44	0.28	0.5	0.72	0.77
s3	Baseline (no adaptation)	0.17	0.08	0.17	0.33	0.41	0.33	0.92	0.67	0.83
	Location adapted	0	0	0	0.08	0.08	0.25	0.25	0.33	0.17
	Intensity adjusted	-	-	-	0.08	0	0	0.08	0.17	0
s4	Baseline (no adaptation)	0	0	0	0.25	0.33	0.33	0.5	0.5	0.33
	Location adapted	0	0	0	0.25	0.08	0.25	0.25	0.17	0.17
	Intensity adjusted	-	-	-	0.25	0.08	0	0.25	0	0.17
s5	Baseline (no adaptation)	0.58	0.58	0.67	0.75	1	0.83	1	1	1
	Location adapted	0.25	0.17	0.17	0.58	0.92	0.75	0.92	0.83	1
	Intensity adjusted	-	-	-	0.5	0.42	0.75	0.66	0.75	0.92

Table 4.2: Sentence recognition WER for baseline, location adapted and intensity adjusted models.

Application	**Model type**	**Accuracy of each bank**								
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**
WT	Baseline (%)	53.30	50	46.70	53.30	50	40	33.30	36.70	33.30
	Location adapted (%)	63.03	66.60	56.60	56.60	53.30	46.60	43.30	46.60	40
	Intensity adjusted (%)	-	-	-	60	66.70	50	46.70	50	53.30
SID	Baseline (%)	80	82.76	83.33	61.29	60	57.14	50	48.28	51.72
	Location adapted (%)	93.33	96.67	90	68.96	67.85	66.66	55.17	53.33	54.84

Table 4.3: Word transcription (WT) and speaker identification (SID) accuracy for baseline, location adapted, and intensity adjusted LocoVocal models.

4.2.2 1-Person Walking-and-Speaking Test

In the walk-stop-and-speak test, LocoVocal was getting enough time for acoustic model switching as the human subject remained stationary in each bank while playing the corpus. In this experiment, we did a stress test on its model switching frequency. This experiment was done with the word corpus only. The user started walking from bank g, and followed the path shown in Figure 4.2 covering all 9

bank locations until he reached bank c. He started speaking, i.e. playing the word corpus of 10 words in his loudspeaker at bank g when he started his tour, and finished playing 1 full recording of the word corpus when he finished his tour at bank c. For a baseline, we loaded the original unaltered model into CMUSphinx and repeated the test with no location based model switching. Both the experiments were repeated 5 times.

LocoVocal has 18.18% average word transcription accuracy improvement for walking-and-speaking test compared to baseline. The frequent bank switching of human subject across the 9 banks causes continuous model switching by LocoVocal. The model switching has a latency and it may happen that speech captured in one bank may be processed by the model of a previous bank along the human subject's tour path.

4.2.3 Multi-Person Source Detection Test

In this experiment, we tested LocoVocal's ability in detecting human originated sound source by beam direction-of-arrival when multiple skeletons are detected by Kinect. Also, we tested this in presence of a non-human sound source (loudspeaker) to see if LocoVocal can detect it. Traditional location agnostic systems (Resonate, Amazon Echo) cannot detect if a sound is originating from a human or non-human (TV) subject, and therefore can inappropriately and insecurely trigger AAD applications targeted for humans but originated from a non-human sound source.

We conducted a 2-person test with a non-human source (loudspeaker) in various bank locations. LocoVocal could detect the sound originating skeleton every time, and sound from loudspeaker was detected as non-human sound in all cases and hence ignored for further processing, as shown in Table 4.4.

Person 1 Position	Person 2 Position	Loudspeaker Position	Original Source	Detected Source
d	f	e	Person 1	Person 1
d	f	e	Person 2	Person 2
d	f	e	Loudspeaker	Loudspeaker
d	e	f	Person 1	Person 1
d	e	f	Person 2	Person 2
d	e	f	Loudspeaker	Loudspeaker

Table 4.4: Sound Source detection with multiple human and non-human sources. LocoVocal detects the correct sound source every time from 2 humans and a loudspeaker.

4.2.4 Opportunistic Ambient Noise Sampling Test

In this experiment, we tested how opportunistic ambient noise sensing during system idle time when no skeleton is detected improves speech detection performance. We placed a running table fan beside the LocoVocal system and played the test corpus from bank e position, which is 2 m directly in front of the Kinect, from a Bluetooth loudspeaker placed on a table. LocoVocal opportunistically sampled the fan noise as there was no human in front of it, and updated its acoustic model within 2 seconds. For performance comparison, we did the same experiment with a non-noise-adapted model having only regular conference room ambient noise (not fan noise). As seen from Figure 4.4, WER for sentence transcription reduced between 13.79-71.79% for the 5 different sentences. Also, the average accuracy for word transcription and speaker identification improved by 39.5% and 45.35%, respectively.

4.2.5 Parameter Sensitivity Test

We wanted to learn the speech transcription and speaker identification performance sensitivity due to change of various acoustic and scaling parameters of LocoVocal system. We did the sensitivity testing by varying T60, room dimension parameter, source-microphone distance and α (scaling parameter) and measuring respective system performance. All these experiments were done with the source located at bank position e.

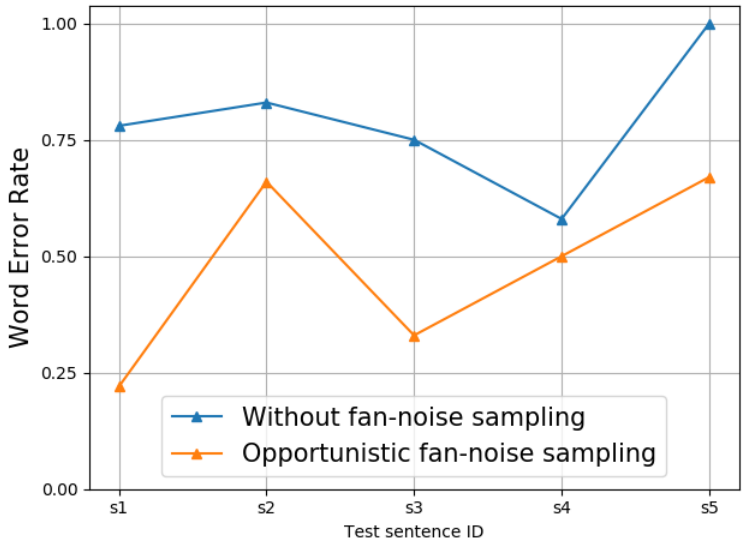


Figure 4.4: Sentence transcription performance with and without opportunistic noise sampling.

Varying T_{60}

The reverberation time, T_{60} for the conference room was found to be 0.44 s using the estimation method. We varied it to $0.25T_{60}$ (0.11 s), $0.5T_{60}$ (0.22 s), $2T_{60}$ (0.88 s), and $4T_{60}$ (1.76 s), and measured transcription and speaker identification performance with source located at bank position e.

As seen from Figure 4.5 and Table 4.5, reducing the T_{60} actually improves both sentence and word transcription performance in both cases. This is because, we used a blind estimation method to estimate reverberation time from only some speech recorded in the room. There is scope of some error in this estimation, and the blind estimator probably over-estimated the T_{60}. Therefore, system performance improved when T_{60} was reduced, and degraded when T_{60} was increased. As mentioned earlier, sentences s2 and s5 were more difficult to transcribe compared to others for the CMUSphinx tool, hence incurring more overall WER.

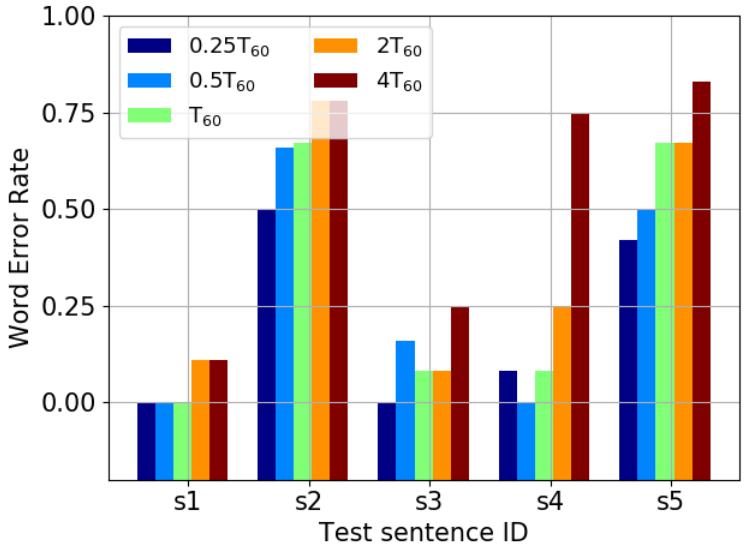


Figure 4.5: Sentence transcription word error rate with varying T_{60}.

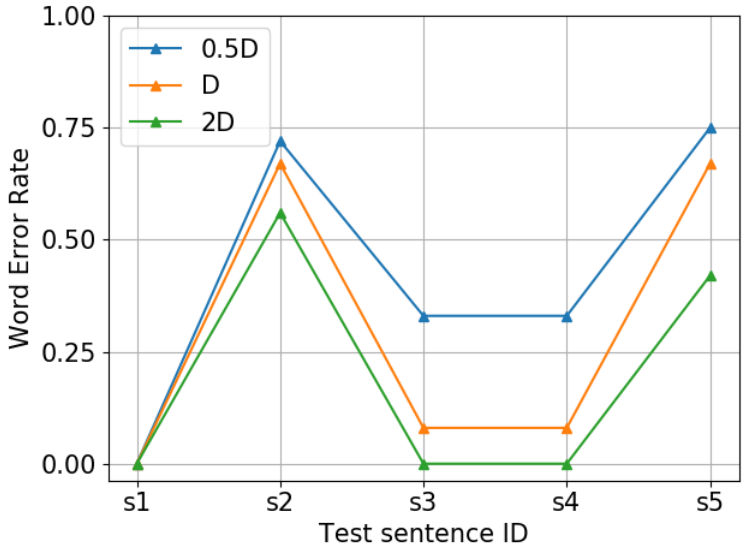


Figure 4.6: Sentence transcription word error rate with varying room dimensions.

T_{60} **level**	**0.25**T_{60}	**0.5**T_{60}	T_{60}	**2**T_{60}	**4**T_{60}
WT accuracy (%)	63.30	56.70	53.30	50	46
SID accuracy (%)	72.41	68.96	67.85	46.67	39

Table 4.5: Word transcription (WT) and speaker identification (SID) accuracy for varying T_{60} levels.

Varying Room Dimensions

All the 3 room dimension parameters (length, width, height) were varied by a factor of 0.5 and 2 during acoustic profiling and model adaptation, and system

Room dimension	**0.5D**	**D**	**2D**
WT accuracy (%)	48	53.30	70
SID accuracy (%)	65.51	67.85	75.86

Table 4.6: Word transcription (WT) and speaker identification (SID) accuracy for varying room dimension parameters.

Source position	**b**	**e**	**h**
WT accuracy (%)	76.70	53.30	40
SID accuracy (%)	93.10	67.85	51.72

Table 4.7: Word transcription (WT) and speaker identification (SID) accuracy for varying source-microphone distance.

α **level**	**0.1**α	**0.3**α	**0.5**α	**0.7**α	**0.9**α
WT accuracy (%)	36.60	51	53.30	53.30	46.70
SID accuracy (%)	53.57	63.33	67.85	65.71	51.72

Table 4.8: Word transcription (WT) and speaker identification (SID) accuracy for varying α.

performance was measured. In synchrony with the results obtained from varying T_{60}, system performance for sentence recognition improved by 48.8% on average across 5 sentences, and word recognition and speaker identification improved by 31.33% and 11.81% respectively, when room dimensions were increased, and vice versa, as seen from Figure 4.6 and Table 4.6. As our T_{60} was overestimated by the blind estimator, the effect of overestimation got balanced when the room dimensions were increased, i.e. reducing reverberation effect.

Varying Source-microphone Distance

We used the acoustic model adapted for bank e position to do experiments from bank b and h, both of which are located 1 m away from bank e, each on either side. Figure 4.7 and Table 4.7 show that playing test corpus from the nearer bank position b, when the running model was actually adapted for bank e, improves both word and sentence transcription performance, and vice versa for the far bank position h, where performance degrades. The signal quality and intensity is much better at nearer positions with less reverberation effect. Therefore, even the acoustic model adapted for a different far location cannot degrade the performance of a source speaking from a nearer position. This result

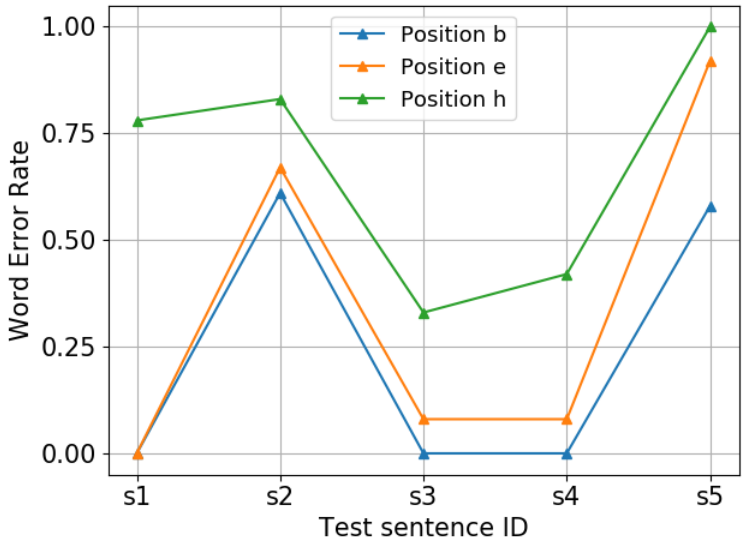


Figure 4.7: Sentence transcription word error rate with varying source-microphone distance.

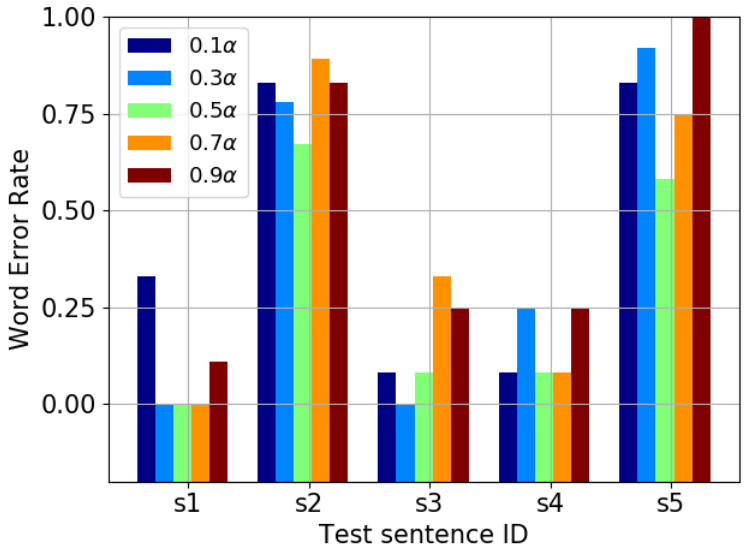


Figure 4.8: Sentence transcription word error rate with varying α.

demonstrates that, LocoVocal concept is implementable with other less accurate human localization systems (like WiFi, acoustic).

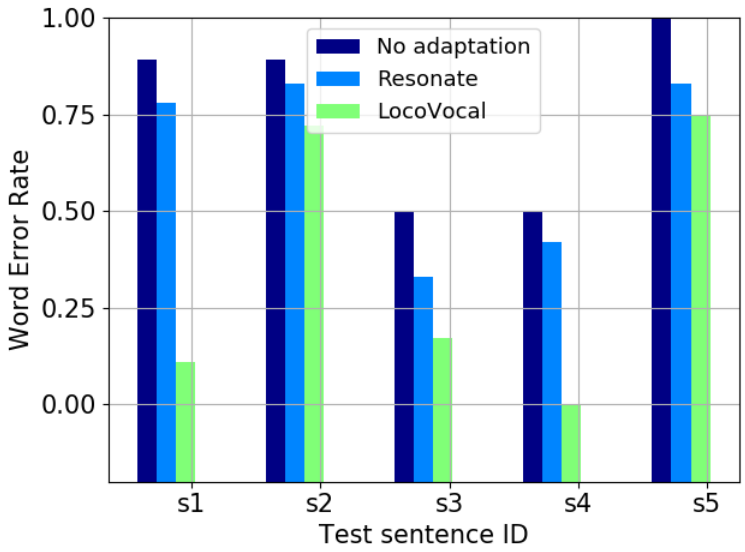


Figure 4.9: Comparison of LocoVocal speech recognition performance from bank h with the Resonate system.

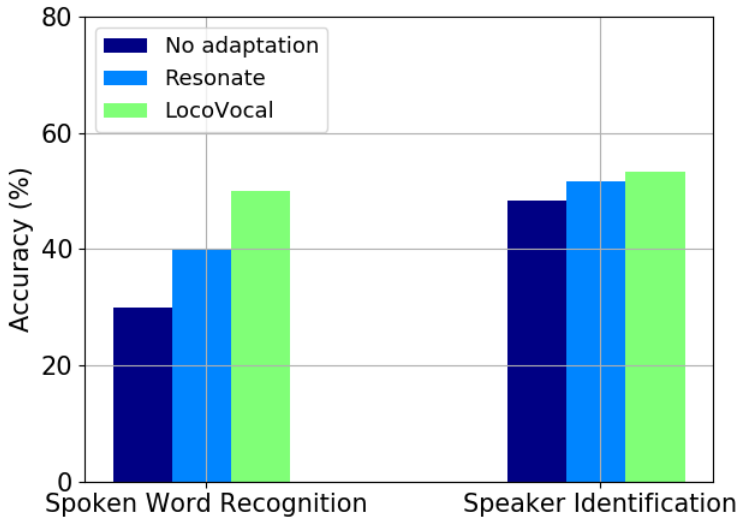


Figure 4.10: Comparison of LocoVocal spoken word recognition and speaker identification performance from bank h with the Resonate system.

Varying α

The reverberation and ambient noise component scaling parameter, α, which was originally set at 0.5, was varied to factors of 0.1, 0.3, 0.7, and 0.9, and system performance was measured. As seen from Figure 4.8 and Table 4.8, both word and sentence transcription and speaker identification performance was best for

0.5α in most cases, and degraded for the other 3 values of α. The value of α sets the weight on reverberation and ambient noise component on the adapted training data, with $\alpha = 0.5$ setting the equal weight between 2 components. The conference room where we did our experiments had reverberation and ambient noise which were both moderate, therefore $\alpha = 0.5$ yielded best performance. However, in rooms with high reverberation and comparatively lower ambient noise, α should be set to a higher value and vice versa.

4.2.6 Comparison with RESONATE

We compare LocoVocal performance for speech recognition and speaker identification with the Resonate [20] system, which is the state-of-the-art in literature for static physical acoustic environment adaptation for AAD applications. We performed these experiments from bank location h, which is 3 m away from the Kinect. We developed a location independent static model as in Resonate and compared with LocoVocal and baseline (no adaptation) for our two AAD applications.

As seen from Figure 4.9, LocoVocal reduces the WER by 51.45% on average compared to static Resonate system for spoken sentence recognition across our 5 test sentences. Also, LocoVocal improves spoken word recognition accuracy by 25% and speaker identification accuracy by 10.46% compared to Resonate when experiments were done from bank location h, as seen from Figure 4.10. It should be noted that, since these experiments where done in a moderate volume from our furthest bank h situated 3 meters away from the Kinect, the overall performance was relatively poor.

4.3 Controlled Experimental Results: Run-time Mode

Pre-computed mode, although being fast, has the overhead of building models for each bank location in advance. Run-time mode, on the contrary, builds and

Task	Average Execution Time (ms)
ISM simulation	491
Convolution and ambient noise addition	47
CMUSphinx model adaptation	1822
CMUSphinx model load/switch	1615
Total	*3975*

Table 4.9: Execution time for different acoustic adaptation tasks for CMUSphinx. LocoVocal run-time and pre-computed modes have a latency of 3.975 and 1.615 seconds for speech recognition, respectively.

Walking Type	Average Walking Speed
Slow walk	0.77 m/s
Normal walk	1.11 m/s
Fast walk	1.74 m/s

Table 4.10: Average walking speed of different kind of walking.

executes models from scratch at run time. Using run-time mode, LocoVocal can be capable of autonomously deciding how many model banks are necessary and auto-calibrate those model banks instead of providing pre-computed arbitrary model banks, as described in detail in section 4.5.

In computing the operational latency of LocoVocal speech recognition run-time mode, the skeletal localization time is very small and negligible. The main latency is contributed by the ISM simulation, convolution and ambient noise adding, and the CMUSphinx model adaptation. It should be noted that, convolution and ambient noise addition is done to the entire training data, and hence, the latency is linearly proportional to the training size. Therefore, run-time mode may not be feasible for applications having very large amounts of training data.

We measured the average execution time of different stages of acoustic profiling and adaptation of LocoVocal over several samples in an Intel Core i7 2.9 GHz machine with 8 GB memory, as shown in Table 4.9. Adding them all yields a latency of 3.975 seconds for LocoVocal run-time mode. Note that, only the CMUSphinx model load/switch latency of 1.615 seconds is the latency for the pre-computed mode.

To connect this latency with human movement time, we collected average walking time from 2 people for 3 different walking speeds: slow walk, normal walk, and fast walk across 5 samples each, as summarized in Table 4.10.

We define distance precision for a given mode as the minimum distance a human subject needs to cross in a given walking speed so that the system has enough time for adapting a new model. As walking speed ranges between [0.77 m/s, 1.74 m/s] from Table 4.10, the best distance precision in run-time mode would be in the range of [3.06, 6.92] m, considering a 3.975 second latency. For pre-computed mode, this distance precision would be between [1.24, 2.81] m range, as model bank switching latency is only 1.615 second. Assuming that majority of people will walk at a normal speed (1.11 m/s from Table 4.10), the distance precision of a run-time and pre-computed mode of LocoVocal would be 4.41 m and 1.79 m, respectively.

4.4 Home Deployment Experiment Results

To evaluate LocoVocal's performance in a real world environment, we deploy it with a speaker identification application in the kitchen and dining room of a 2-person household for 1 day. The purpose of this deployment was to observe LocoVocal in a natural environment with natural movement of individuals.

Before the deployment began, each of the 2 persons provided training samples to the speaker identification program by reading a short script. The LocoVocal system was set at a side of the dining room so that most of the kitchen and dining room falls within the range of the Kinect. 6 model bank locations were set by the sink, stove, fridge, kitchen counter and the dining table area. Two real time speaker identification systems were set to run for the entire day in the LocoVocal workstation: one on top of LocoVocal with location adapted models, and another baseline system with no localization and model adaptation.

The experiments were done in 3 rounds each at different times of the day. During each round, the two residents of the household were asked to engage in conversations with each other. However, they were not constrained on what other actions (household chores) to perform, the order of the actions and their position in the kitchen and dining area. For ground truth, the entire deployment audio was recorded. Each round of the conversation in the morning, afternoon

and evening session lasted between 12-20 minutes. These were some surrounding added noise due to the household chores (dish washing, cooking) the residents were doing during their conversation.

After each round of conversation, we evaluated the speaker identification accuracy for both the LocoVocal and baseline system from the ground truth recording. As seen from Table 4.11, LocoVocal improved the speaker identification accuracy by 10.71%-17.30% from the baseline for every round of the home deployment. However, there were few instances when a participant went to the other end of the kitchen to use the pantry, which was out of range of the Kinect. In those cases, the real-time skeletal tracking and localization froze, but LocoVocal was still running and using the previously best found model. When the participant appeared within the range, LocoVocal started the localization process immediately.

Round	**Noise source**	**Baseline accuracy**	**LocoVocal accuracy**	**Improvement**
Morning	No noise	77.78%	86.11%	10.71%
Afternoon	Dish-washing	72.22%	84.72%	17.30%
Evening	Cooking	73.66%	82.67%	12.23%

Table 4.11: LocoVocal improves accuracy for speaker identification in different rounds of home deployment by 10.71-17.30%.

4.5 Auto-calibration of Model Banks

We used nine pre-computed model banks in nine predefined locations in the Kinect field-of-view in the pre-computed mode. This mode has the overhead of computing each model beforehand by providing location specific input. In this section, we explore the possibility of the system auto-calibrating the required number of model banks itself in the run-time mode.

If 2 adjacent model banks are located too close, the following issues may arise:

1. The models may become too similar to each other and hence redundant.

2. As model switching has some latency, a moving human subject may cross the neighboring bank area before the system was able to switch to that model, making the neighboring bank unusable.

The system has to 'explore' the space (room of operation) starting from an initial position and keep finding new points in the unexplored area meeting the above constraints. A random data structure like Rapidly-Exploring Random Tree (RRT) [97] can be used for such path planning in the space. Unlike other randomized path planning algorithms, RRT vertices are heavily biased to explore toward the unexplored region of the space.

Let, λ be the minimum difference in performance between two neighboring models, and τ be the minimum distance between neighboring models based on average human movement speed. The RRT exploration will start from an initial point and keep exploring the space by finding new unexplored points. For each point, the algorithm will check if the new point is at least τ units away, and have performance at that point (with a model based on that point) at least λ units different than the previous model. Once such a point is found, it is selected for a bank location. The algorithm will explore the entire space by RRT exploration and choose appropriate vertices from the RRT as candidates of model locations, based on the λ and τ constraints. After this process converges, the resulting Voronoi region of each bank location will determine that particular bank's area of operation.

4.6 Summary

This chapter describes the performance of LocoVocal in terms of its ability to improve 2 representative AAD applications (speaker identification and speech recognition) by sensing a secondary physical phenomenon like real-time human localization. Using human localization information and microphone array in the AAD hardware, LocoVocal can separate human and non-human sound sources for improved safety and convenience. Through controlled experiments by a prototype implementation using a Kinect based human localization system, we demonstrated LocoVocal performing spoken word recognition, sentence recognition and speaker identification with 25%, 51.45% and 10.46% improvement respectively compared to a state-of-the-art system in literature using location

agnostic static models. The real-time human tracking and localization ability of LocoVocal for AAD applications makes it a possible solution for further improving the state-of-the-art commercial products like Amazon Alexa and Google Home.

Chapter 5

Distant Speech Emotion Recognition

In this chapter, we address and provide solution to another problem, distant speech emotion recognition, using LocoVocal techniques. However, we performed non real-time experiments for this problem as opposed to previous problems (speech recognition and speaker identification) in Chapter 4 where we did real-time localization with moving human and live acoustic event detection. Non real-time experimental setup allowed us to test distant emotion recognition performance in a variety of rooms with different acoustic configurations and speaker-to-microphone distances (using a room impulse response dataset), where the experiments in Chapter 4 were done in just one room. Therefore, using non-real time experiments in this chapter, we demonstrate the versatility of LocoVocal for different tooms, acoustic configurations and source-to-microphone distances.

The rest of this chapter is organized as follows. Section 5.1 discusses the motivation of doing distant emotion recognition. Section 5.2 lists the technical contributions that this chapter makes. Section 5.3 lists the challenges in providing a solution for distanct emotion recognition. Sections 5.4 describes the method behind finding distance robust features, dataset preparation, feature extraction

and selection, and associated evaluations. Sections 5.5 provides algorithms and experimental results of signal cleaning from reverberation and noise. Section 5.6 provides the solution for training with artificial reverberation. Section 5.7 presents the CPU time benchmarking of various system components. Section 5.8 provides experimental results of a YouTube dataset evaluation. Section 5.9 discusses some aspects of the solution and finally Section 5.10 provides a summary of the chapter.

5.1 Motivation

Extracting emotional components from human speech (speech emotion recognition) in real time has been a challenging problem for several decades. Speech is the most common and natural communication medium in humans. Therefore, accurate real time speech emotion recognizers have far more potential for real world deployment centric applications because 1) speech has pervasive reachability to nearby sensors (microphone) as opposed to video/facial expression based emotion recognizers, and, 2) speech is less intrusive as opposed to galvanic skin resistance based emotion recognizers. A real time speech emotion recognizer will have profound impact on a wide range of applications if it can be accurate in a wide range of environments with different acoustic configurations and different source-to-microphone distances. If these challenges can be met, then the solution can be used in many applications requiring real time emotion recognition. For example, it can be used in an advanced driver assistance system to detect real time mood of a vehicle driver, as aggressive driving behavior may lead to accidents. Similarly, real time cockpit behavior for airline pilots can be monitored for possible depressive syndromes leading to suicidal tendencies. In general, people with suicidal tendencies can be monitored for mood to support just in time interventions. Certain medical conditions like heart diseases are likely to worsen due to anger and excitement, therefore such patients can be monitored in real time for emotional outbursts and subsequent interventions can possibly avoid heart attacks.

In a real time indoor speech emotion recognition system, the microphones are deployed in certain places of the room. These microphones capture speech signals originating from sources (human) situated at various distances. Increasing source-to-microphone distance reduces signal-to-noise ratio and induces noise and reverberation effects in the captured speech signal, thus degrading the quality of captured speech, and hence the performance of the emotion recognizer. A related area of research is distant-speech-recognition (DSR) i.e. converting speech to text by distant microphones, which is an extension of the automatic-speech-recognition (ASR) problem, where a lot of progress has been made [98–100] in recent years. However, real time distant emotion recognition (RTDER) is an area not much explored before to the best of our knowledge. It is important to note that, the solutions of the DSR problem are not generalizable to solve DER problem because of the difference in the nature of the core problems. DSR targets translating captured speech in distant microphones to text, while RTDER targets classifying captured speech into certain emotional classes in real time. Speech emotion recognition requires a large number of local and global acoustic features, a static or dynamic emotion training set and uses classifiers like support vector machines (SVM), Gaussian mixture models (GMM) and random forest whereas automatic speech recognition needs a limited number of features (MFCC) with hidden Markov models (HMM) and uses a different technique involving phonemes and language models.

A real time speech emotion recognition system is generally evaluated using one or more emotional speech database/corpuses. An emotional speech corpus is generally made from real-world incident recordings or from acted/elicited artificial emotional utterances in sound laboratories by professional/semi-professional/non-professional actors. A majority of the existing emotional speech databases are made artificially because of the legal and moral issues of using real life recordings for research purposes. An extensive list of state-of-the-art emotional corpuses can be found in [101], where the corpuses have been made by professional/non-professional actors and from extracted movie clips. A common characteristic of all the existing emotional corpuses is that all of them are made of clean speech recorded by closely situated microphones, often in a noise-proof anechoic sound

studio. All the existing real time speech emotion recognition results are based on these clean speech recordings and, hence, these results are not applicable to a real world environment where acoustic sensors are likely to be situated far from the speakers.

In this chapter, we address different stages of the processing pipeline of a real time speech emotion recognition system to solve the previously unexplored RTDER problem and increase real time emotion recognition accuracy in distant microphones in different room types.

5.2 Contributions

The main contributions of this chapter are:

- We identify several challenges in different stages of a real time distant speech emotion recognition pipeline and provide solutions to them with empirical results obtained over extensive evaluations.

- We create the very first distance aware emotional corpuses to use in our experiments by 1) re-recording a popular emotional corpus with a microphone array with microphones placed at various distances, and 2) by injecting room impulse responses collected in a variety of rooms with various source-to-microphone distances into the same emotional corpus. We plan to make these distance aware emotional corpuses freely available for research purpose.

- Our novel combination of distorted acoustic feature elimination, best feature selection and classifier optimization techniques improves the real time distant emotion detection accuracy between 1.31% (for the worst case scenario of a large church hall) to 6.12% from the baseline in a variety of rooms with various source-to-microphone distances. We perform the most comprehensive feature analaysis for RTDER using the largest known emotional feature set consisting of 6552 acoustic features. Note that, we considered loud background noisy environments and extremely large

rooms with very high reverberation effects for possible worst-case situation analysis and achieved improvement from the baseline even in the worst scenarios.

- At the signal acquisition stage, we use 2 state-of-the-art dereverberation and denoising techniques to clean the distant emotional speech signal. In addition, we combine these approaches with our novel combination of distorted feature elimination, best feature selection and classifier optimization techniques to achieve up to 10.84% improvement from the baseline in a variety of rooms with various source-to-microphone distances, with the final classification accuracy ranging between 79.44%-94.95%.

- At the classifier training stage, we train our classifiers with synthetic reverberation obtained from a room impulse response generator to reduce the discrepancy between training and testing conditions in a RTDER environment. Our training approach only requires emotion samples with clean speech at a close microphone. In addition, we combine this with our novel feature and classifier enhancement techniques to obtain up to 15.51% improvement from baselines across all the rooms in a variety of distances, with the final accuracy ranging between 87.85%-95.89%.

- We evaluate the above mentioned techniques on a YouTube video dataset consisting of 37 clips spanning over 3 hours from lectures, public speech, talk-shows, and personal statements from both actors and real people and obtain a maximum of 7.30% of improvement in recognizing real world emotions at various source-to-microphone distances in different rooms, with a maximum accuracy of 93.68%.

- We experimentally evaluate the CPU runtime of each component of our system and demonstrate the real time capability of our system.

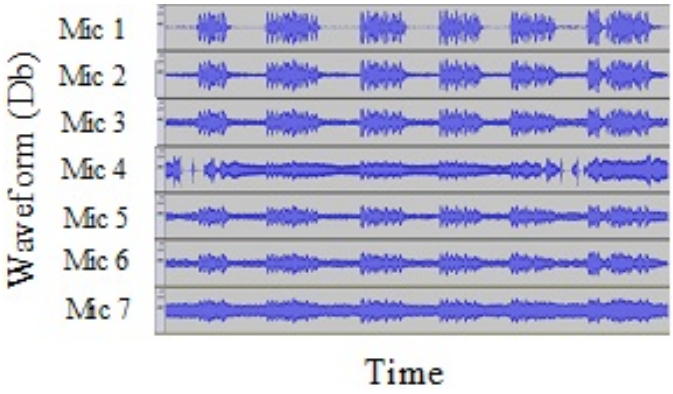


Figure 5.1: Raw waveforms of an angry utterance containing 6 separate sentences in 7 microphones situated at different distances.

5.3 Problem Formulation and Challenges

Speech based real time emotion detection is a complex problem due to the diversity in the way different people speak and express emotions, linguistics of different languages and accents, and the expression of a wide range of emotions by human. However, when speech is captured by distant microphones (as opposed to right next to the speaker), it adds further complexity to the real time emotion detection problem due to room reverberation, noise, and reduced signal-to-noise ratio.

In the past 2 decades, various emotional corpuses have been made by the affective computing community from clean emotional speech recorded in anechoic (non-reverberant) and noise-free sound studios simulated by professional or non-professional actors, and hence, all the existing emotion detection results are based on clean emotional recordings. However, for a realistic real time emotion detection system deployed in open environments, one or more microphones will be situated at certain places of a room, and hence capture sound waves coming from distant sources (human subjects). We formally call this a *Real Time Distant Emotion Recognition* (RTDER) problem. As an example, we record 6 sample angry utterances from an emotional corpus in front of a microphone array consisting of 7 microphones situated at various distances, and Figure 5.1 shows the recorded waveforms. The waveform of microphone 1 demonstrates clear recording with highest signal-to-noise-ratio (SNR), where microphone 1

is situated nearest to the speaker. However, as the speaker to microphone distance increases, background noise and room reverberation are injected into the recordings, and hence the SNR decreases, as observed in recordings from microphones 2-7. This induced noise and reverberation drastically affect the emotion detection performance, as we demonstrate in later sections.

Figure 5.2 demonstrates a standard acoustic emotion detection pipeline. A clean speech emotional corpus is created by induced or acted emotional utterances from professional or non-professional actors. A training and test set is generated from the emotional corpus, and emotional models are generated from the training set after extracting emotion correlated acoustic features. Finally, the same features are extracted from the test set and the emotional models are applied on the test set to classify emotion. In the context of RTDER, the training set is obtained from a clean speech signal while the test set is obtained from distant speech; hence the challenge arises because of discrepancy in training and test data. In this paper, we address different stages of the acoustic emotion detection pipeline in context of RTDER with an objective of making training and testing conditions similar, and thus increase emotion classification accuracy. The challenges we address and solve are:

- **Challenge 1:** Can we find a set of emotion correlated acoustic features which are robust against microphone-to-speaker distance?
- **Challenge 2:** Can we clean the test speech recorded over distance from noise and room reverberation?
- **Challenge 3:** Can we add artificial room acoustic configuration into the clean speech training to reduce the discrepancy between training and test scenario?
- **Challenge 4:** Can we execute various system components of our solution in real time on standard available hardware of our target safety centric applications?

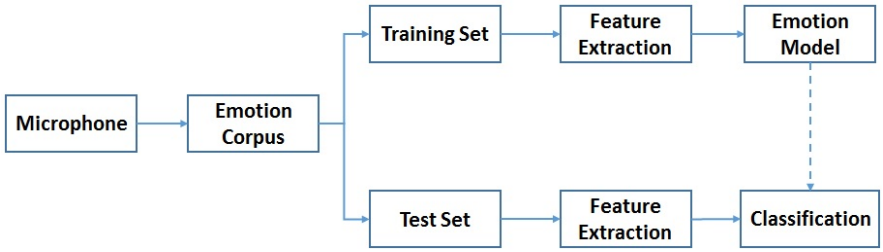


Figure 5.2: Stages of a standard acoustic emotion detection pipeline.

5.4 Finding Distance Robust Features

When emotional speech signal is recorded by a distant microphone, the recorded signal becomes distorted compared to the original signal because of room ambient noise and reverberation. The amount of distortion depends on the acoustic properties of the room and amount of noise. For our solution, we empirically find a set of acoustic features which are robust across distance as well as correlated to target emotions. We then use these distance robust features for both training with clean emotional speech and testing on distant speech. Since these features are robust across distance, their distortion with distance is minimal, hence the discrepancy between training and testing in RTDER is also minimal, and accuracy is improved.

In our solution, we calculate the distortion of a particular feature f at distance d using the following formula:

$$distortion_d = \left| \frac{f_0 - f_d}{f_0} \right| \times 100\% \tag{5.1}$$

Where, f_0 = feature value for clean signal (distance 0),
f_d = feature value for signal at distance d.

For the rest of this section, we discuss the data preparation strategy for our experiments, feature extraction, distorted feature filtering, best feature selection and classifier optimization methods.

5.4.1 Data Preparation

We used the Berlin Emotional Speech Database, also known as Emo-DB [102], in our experiments to find distance robust emotional speech features. Emo-DB is a well-known and widely used freely available emotional corpus in the affective computing domain. It contains short sentences in German each spanning between 2-5 seconds in 7 different emotion categories: anger, anxiety, boredom, disgust, happiness, neutrality and sadness. There are 535 utterances in total in Emo-DB spanning these 7 emotions spoken by 10 professional actors (5 males and 5 females).

Just like most other emotional corpuses, Emo-DB contains only clear speech recordings. We apply the following 2 techniques to impose distance effect in Emo-DB recordings.

Re-record Emo-DB with a microphone array: We played the Emo-DB recordings with a loudspeaker and recorded them with a VocoPro UHF-8800 microphone array consisting of 4 microphones placed at distances of 1m, 3m, 5m and 7m from the loudspeaker. The recording was done in a 10m x 5m lab at 16 KHz sampling rate and 24-bit precision. There was loud background HVAC noise present while recording. Most indoor applications would fall within 7m of speaker-to-microphone distance hence we did not record for any further distance. To the best of our knowledge, this is the first emotional corpus recorded with different speaker-to-microphone distances. We refer this dataset as "Emo-DB-Array" for the remaining of the paper.

Inject distance effect into Emo-DB from AIR impulse response database: The Aachen Impulse Response (AIR) [103] database is a collection of room impulse responses (IR) measured in a variety of rooms with various acoustic configurations and with different source-to-microphone distances. A room impulse response can be used to describe the acoustic properties of a room in terms of sound propagation and reflections for a specific source-microphone configuration. The distant reverberated signal $s(n)$ is represented as a convolution of the source (clean) signal $x(n)$ with the room IR $r(n)$

Room	Dimensions	Speaker-to-Microphone Distances
Office room	5m x 6.4m x 2.9m	1m, 2m, 3m
Meeting room	8m x 5m x 3.1m	1.45m, 1.7m, 1.9m, 2.25m, 2.8m
Lecture room	10.8m x 1.09m x 3.15m	2.25m, 4m, 5.56m, 7.1m, 8.68m, 10.2m
Aula Carolina church	19m x 30m	1m, 2m, 3m, 5m, 15m, 20m

Table 5.1: Room configurations of IRs from AIR database

$$s(n) = x(n) * r(n) \tag{5.2}$$

We convolved the Emo-DB recordings with room impulse responses obtained from the AIR database. Convolving the Emo-DB recordings with these IRs injects the acoustic and various distance effects of AIR database rooms into the Emo-DB database recordings. We refer this dataset as "Emo-DB-AIR" for the remaining of the paper. Table 5.1 summarizes the dimensions and speaker-to-microphone distances of various rooms from the AIR database whose IRs were convolved with Emo-DB recordings to construct Emo-DB-AIR.

5.4.2 Feature Extraction

We used the widely used OpenSMILE feature extraction toolkit [104] to extract a large number of 6552 features as 39 functionals of 56 acoustic low-level descriptors (LLD) related to energy, pitch, spectral, cepstral, mel-frequency and voice quality and corresponding first and second order delta regression coefficients. The 39 statistical functionals are applied to the LLDs computed from each of the emotional utterances to map a time series of variable length onto a static fixed size (6552) feature vector. Features were extracted on the utterance level, i.e., 1 feature vector per sentence. These 6552 features constitute the Emo-Large feature set of OpenSMILE toolkit, which is the largest emotion specific feature set known to date in terms of number of features. We chose the largest feature set because it would allow us to know more emotion correlated features which get distorted over distance, and hence would help to build a feature set robust to distance by keeping the distance agnostic emotion correlated features only. Such an approach to find distance robust emotional features has not been attempted before, to the best of our knowledge.

5.4.3 Iterative Distorted Feature Cut (IDFC) Procedure

For each of the 6552 features, distortion of each feature with respect to its corresponding clean signal feature value is calculated using equation 5.1, for both the Emo-DB-Array and Emo-DB-AIR datasets. The features are then sorted by their distortion value from highest to lowest, and iteratively discarded (cut) from the train and test sets one by one. In each step of the iteration, a new emotion model is built with the updated reduced-by-1 feature set, and tested upon the test set, and corresponding classification accuracy is logged. A support vector machine (SVM) classifier is used for training and testing, as SVM is reported to have best performance in emotion detection in prior works. The features are normalized to the range [-1, 1] before training and testing. This procedure iterates across all 6552 features and returns the best accuracy achieved across all iterations and the corresponding best feature cut.

5.4.4 Feature Selection and SVM Parameter Optimization

We chose a large feature set consisting of 6552 features so that we can identify the highest number of distance sensitive distorted features. Although the Emo-Large feature set is an emotion detection feature set, not all the 6552 features are equally important for the emotion detection task. Allowing a lot of less-correlated features overfits the classification model resulting in greater errors, in addition to increased latency for real time operation. Therefore, we used an algorithm presented in [105] which ranks the features by their importance to the classification problem by calculating their F-scores. The larger the F-score is, the more likely this feature is more discriminative. We also calculated optimized hyperparameters for the SVM using grid search, with the cost $c = 4$ and gamma $\gamma = 0.00195$.

For a 2-class (binary) classification problem, given training vectors $x_{\mathrm{k}}, k = 1, ..., m$, if the number of positive and negative instances are n^{+} and n^{-}, respectively, then the F-score of the i^{th} feature is defined as:

$$F(i) = \frac{(\bar{x_i}^{(+)} - \bar{x_i})^2 + (\bar{x_i}^{(-)} - \bar{x_i})^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n^+}(x_{k,i}^{(+)} - \bar{x_i}^{(+)})^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n^-}(x_{k,i}^{(-)} - \bar{x_i}^{(-)})^2} \tag{5.3}$$

where $\bar{x_i}$, $\bar{x_i}^{(+)}$, $\bar{x_i}^{(-)}$ are the mean of the i^{th} feature of the whole, positive and negative data sets respectively, and $\bar{x_{k,i}}^{(+)}$, $\bar{x_{k,i}}^{(-)}$ are the i^{th} feature of the k^{th} positive and negative instance, respectively. The Emo-DB dataset has 7 different emotion labels; hence this is a multi-class classification problem, as opposed to binary classification. For multi-class classification, the algorithm constructs $C(k, 2) = \frac{k(k-1)}{2}$ binary classifiers between each possible pair of the original k classes. F-score based feature ranking is calculated for each of these binary classifiers, and finally it selects the same feature subset for every binary classifier to maximize the average accuracy over all classes.

Next, we sort all the features by their F-score importance, and evaluate if choosing a smaller subset of more important features improves the classification performance. We iteratively chose larger subsets of important features, and do a 10-fold cross validation on the training set. Figure 5.3 shows that choosing a subset of 3271 most important features yields highest cross-validation accuracy (88.78%) with the optimized SVM model ($c = 4, \gamma = 0.00195$) with a Radial-Basis kernel.

Finally, we again apply the iterative distorted feature cut procedure on the best 3271 features to eliminate the most distorted features among these best 3271 features to further increase the emotion detection accuracy, as seen from results in the next section.

5.4.5 Evaluations

We experimented with the IDFC, feature selection with F-score and SVM parameter optimization procedures on Emo-DB-AIR and Emo-DB-Array datasets. We set the baseline as when no feature and classifier enhancements are done, and training is done on clean speech from Emo-DB and testing is done on noisy and reverberated speech from Emo-DB-AIR and Emo-DB-Array. On average,

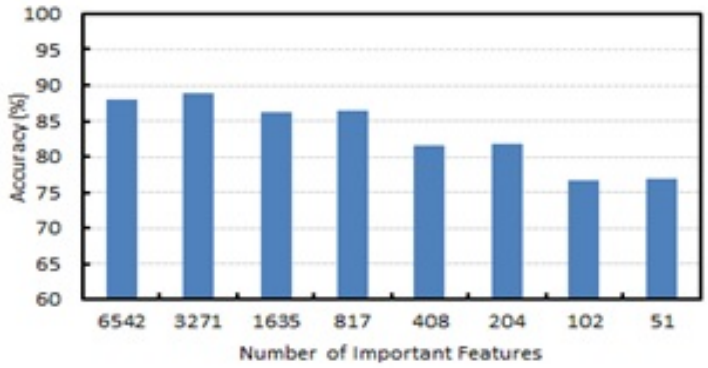


Figure 5.3: We picked the subset of most important features. Choosing the most important 3271 features yielded highest cross-validation accuracy of 88.78%.

we get 2.15%, 2.93%, 2.09% and 1.31% classification improvement for Aula Carolina church, lecture, meeting and office rooms with a final average classification accuracy of 85.14%, 81.53%, 93.19%, and 90.97%, respectively, for the Emo-DB-AIR dataset, and 6.12% average improvement for the Emo-DB-Array dataset with a final average accuracy of 87%. Small distances (like 1m) in small rooms (meeting, office, lab) yields least improvement, as the signal gets little distorted with such small distance, hence the IDFC procedure is less effective. But for larger distances, the IDFC procedure accompanied with best feature subset and optimized SVM is effective in most cases.

While the average accuracy increase may seem low, we show in sections 5.5 and 5.6 that these distorted feature elimination, best feature selection and classifier optimization techniques, when accompanied by signal cleaning and training transformation techniques, yield accuracy improvement as much as 15.51% from the baseline.

5.5 Cleaning Signal from Reverberation and Noise

As stated earlier, speech signals captured in distant microphones are infected with reverberation and noise. In this section, we address the signal acquisition stage of the pipeline presented in Figure 5.2. We use 2 state-of-the-art dereverberation and denoising techniques to clean the distant signal, as described below.

5.5.1 Dereverberation and Denoising Algorithms

Weighted-Prediction Error (WPE)

WPE performs inverse filtering of room acoustics based on linear prediction. For each sample time t, the WPE method [106] linearly predicts the reverberation component contained in an observed speech sample, $x(t)$ from its preceding samples $x(u); u < t$. Let $y(t)$ be the distant speech signal at time t containing reverberation and background noise. Let $y_n[k]$ denote a short-time-Fourier-transform (STFT) coefficient calculated from $y(t)$, where n and k are the time frame and frequency bin indices, respectively. $y_n[k]$ is dereverberated at each frequency bin k using a linear filter as follows:

$$x_n[k] = y_n[k] - \sum_{\tau=T_\perp}^{T_T} g_\tau^*[k] y_{n-\tau}[k] \tag{5.4}$$

where * is the complex conjugate operator, and $T_\perp$ and T_T is the effective time period of the filter. We set $T_\perp = 3$ and $T_T = 50$ to deal with long-term reverberation. $G = (g_{T_\perp}, ..., g_{T_T})$ is a set of filter coefficients optimized to minimize the following objective function:

$$F_{WPE} = \sum_{n=1}^{N} \left(\frac{\left| y_n[k] - \sum_{\tau=T_\perp}^{T_T} g_\tau^*[k] y_{n-\tau}[k] \right|^2}{\theta_n} + \log \theta_n \right) \tag{5.5}$$

where N is the total number of time frames and $\theta = (\theta_1, ..., \theta_N)$ is a set of auxiliary variables optimized jointly with G. The optimized filter F_{WPE} is applied to $y_n[k]$ to generate dereverberated and denoised STFT coefficient $x_n[k]$.

Coherent-to-Diffuse Power Ratio Estimation (CDR)

This method has been proposed by Schwarz and Kellermann [107] to clean the speech signal from reverberation and noise. This technique estimates the ratio between direct and diffuse (reverberation and noise) signal components, also called as coherent-to-diffuse power ratio (CDR), from the measured coherence between

speech captured in two omnidirectional microphones. The CDR estimators are applied in a spectral subtraction postfilter for reverberation suppression.

We experimented with 3 different CDR estimators to suppress reverberation:

- Known direction of arrival (DOA) and noise coherence
- Unknown DOA
- Unknown noise coherence

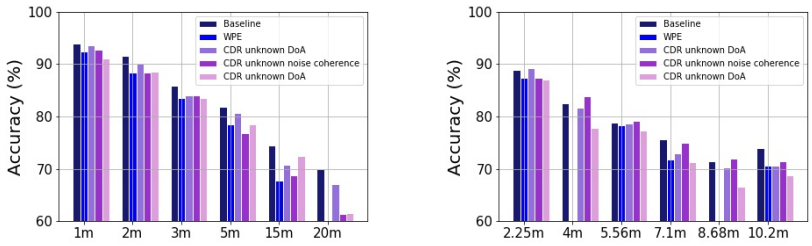
DOA is the angle between the received sound wave axis and microphone axis. Sounds which propagate directly to microphone have a DOA of 0, but reverberated sound being reflected from room walls and objects have a non-zero DOA. The details of the CDR estimators are beyond the scope of this paper, and interested readers should refer to [107] for the details.

5.5.2 Results

We applied the CDR dereverberation technique on Emo-DB-AIR dataset and WPE dereverberation and denoising technique on both the Emo-DB-AIR and Emo-DB-Array datasets. The CDR algorithm requires having 2 omnidirectional microphones for coherence and CDR estimation. The AIR database is a binaural impulse response database, which means IRs were collected with 2 microphones, which justifies our use of Emo-DB-AIR dataset for CDR based dereverberation.

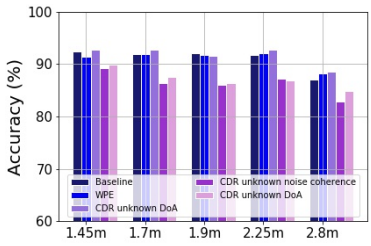
Emo-DB-AIR Evaluation

Figure 5.4 shows the comparative results of different dereverberation and denoising techniques on the Emo-DB-AIR dataset. For these evaluations, we set the baseline as when no signal cleaning is done, i.e. training on clean speech from Emo-DB and testing on noisy and reverberated speech from Emo-DB-AIR. The metric for these analyses is the percentage accuracy of correctly classified

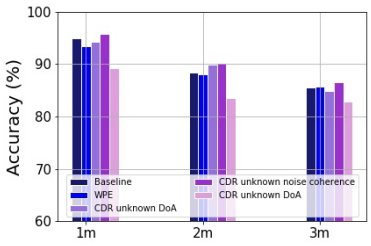


(a) Aula Carolina church

(b) Lecture room

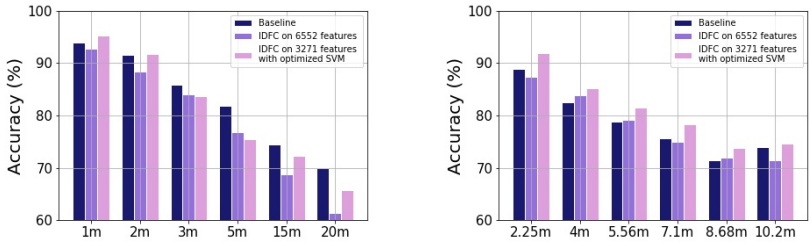


(c) Meeting room



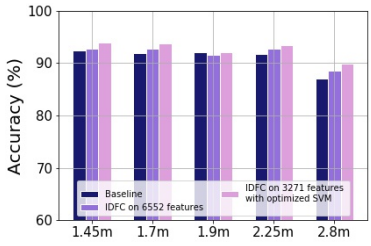
(d) Office room

Figure 5.4: Performance of WPE and CDR dereverberation and denoising techniques on Emo-DB-AIR dataset. CDR with unknown noise coherence consistently outperformed the baseline in most cases (except Aula Carolina church).

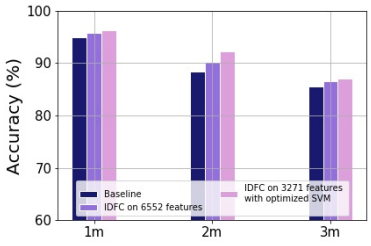


(a) Aula Carolina church

(b) Lecture room



(c) Meeting room



(d) Office room

Figure 5.5: IDFC on original 6552 features improves performance in most cases (except Aula Carolina church due to its very large dimensions) for the Emo-DB-AIR dataset under CDR with unknown noise coherence. Another IDFC on best 3271 features (from Figure 3) with optimized SVM parameters ($c = 4, \gamma = 0.00195$) further boosts performance in all source-to-microphone distances.

emotional utterances (total 535). For baseline, we use the SVM classifier with original 6552 features for classification.

The room dimensions of different rooms in the AIR dataset are provided in Table 5.1. From Figure 5.4, we can see that, for the Aula Carolina church which has the largest dimension and very high reverberation effect, none of the dereverberation and denoising technique can improve the baseline. For the other 3 rooms, the CDR with unknown noise coherence technique consistently improves from the baseline in most cases, the improvement ranging between 1 to 10 utterances. Best improvement was obtained in the office room, which has the smallest dimensions. As expected, the number of correctly classified utterances decreases in all rooms with increasing speaker-to-microphone distance.

The performance of the WPE algorithm in most cases was below the baseline and in some instances it severely degraded the performance. To investigate the issue, we listened to some of the emotional clips from EMO-DB-AIR dereverberated by the WPE algorithm. Our perception was that this algorithm distorts the original signal significantly as a side effect of dereverberation, which causes performance degradation. The performance of other 2 CDR estimation based techniques also ended up being sub baseline.

The CDR with unknown noise coherence technique was found to be the best performing one from the comparative performance study of all the dereverberation and denoising techniques. We further improve its performance by applying the iterative distorted feature cut procedure and feature selection and classifier parameter optimization techniques introduced in section 5.4. The results are shown in Figure 5.5. For the Aula Carolina church, the IDFC procedure with best 3271 features and optimized SVM improves performance for 1m and 2m source-to-microphone distances, although without any feature enhancement the performances were sub-baseline for all distances. For lecture, office and meeting rooms, there is a 2.34%, 1.57% and 2.25% accuracy improvement, with an average final accuracy of 80.90%, 92.70%, and 91.96%, respectively.

Emo-DB-Array Evaluation

The result of WPE dereverberation technique on Emo-DB-Array dataset is shown in Figure 5.6. Unlike Emo-DB-AIR dataset, WPE technique improves from the baseline in all distances for Emo-DB-Array except 1m as the signal distortion due to reverberation and noise at 1m distance is too small to be dereverberated. The lab environment where we recorded Emo-DB-Array using a microphone array had more surrounding noise component from HVAC than reverberation, while the Emo-DB-AIR dataset has stronger reverberation effect than noise. Hence, the lesson learned is that WPE technique performs better on noisy signals rather than reverberated signals.

We further applied the iterative distorted feature cut, best feature selection and classifier optimization procedures from section 5.4 to further improve the WPE dereverberation and denoising performance on the Emo-DB-Array dataset. An improvement of 1.12%, 8.79%, 10.84% and 8.41% was obtained with a final accuracy of 94.95%, 92.15%, 86.17%, and 79.44% for 1m, 3m, 5m and 7m distances, respectively, when we applied the IDFC procedure on the best 3271 features with an optimized SVM classifier.

From these results, we can conclude that the best feature selection, IDFC and classifier optimization techniques combined with the CDR and WPE techniques significantly improve emotion detection performance in various indoor environments across a wide range of source-to-microphone distances even in worst reverberant (Aula Carolina) and noisy (lab) conditions.

5.6 Training with Artificial Reverberation

The objective of our approach in section 5.5 was to make the testing condition similar to the training condition by reducing noise and reverberation from the distant speech and making it as clean as possible like the clear speech training. In this section, we take the opposite approach: making the training condition as similar as possible to testing condition. We do this by injecting artificial

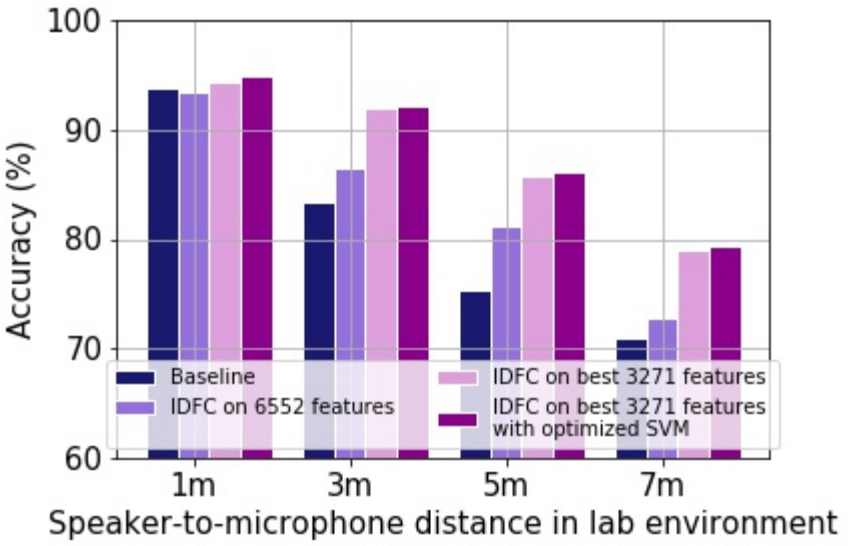


Figure 5.6: Performance of WPE dereverberation and denoising technique on Emo-DB-Array dataset. Combining WPE with feature selection, IDFC and optimized SVM results in a performance boost.

reverberation into clear emotional speech and use this artificially reverberated speech for training the classifier.

5.6.1 The Artificial Room Impulse Response Generator

In equation 5.2, we showed that the distant reverberated sound signal is the convolution of the clean speech signal and the room impulse response. Impulse response represents the acoustic physical property of a room in terms of sound propagation and reflection which is essentially a FIR filter. Room impulse responses can be synthetically generated using the image source model [86] (ISM), which acts as a transfer function between a sound source and an acoustic sensor (microphone) in a given environment if some acoustic parameters of the room are known. Once such a room impulse response is available, a sample of distant audio data at any distant microphone can be obtained by convolving the impulse response with the clean speech signal, as in equation 5.2.

We used the Lehmann's modified and improved ISM simulation technique [87] to generate artificial room impulse responses. The model requires the following room specific parameters:

- Room dimensions
- Source and microphone positions
- Reverberation time T_{60}
- Absorption coefficients of 6 wall surfaces of the room (optional)
- Sound velocity in the room (optional)

Absorption coefficients represent the acoustic absorption capability of a surface. The value is in the range of 0 to 1. The higher the value, the more sound absorbing the surface is (hence, less reverberant). Anechoic chambers are made of fully absorbing wall, floor and ceiling surfaces. This parameter to the ISM simulation model is optional. If omitted, the model is built with equal relative absorption coefficients for all 6 wall surfaces.

The reverberation time, T_{60} is the time required for the sound energy to decay by 60 dB after the sound source has been turned off. A standard method for measuring T_{60} from the room impulse response has been presented by Schroeder [108]. But in our case, we have to estimate T_{60} blindly, as we do not know the impulse response. T_{60} only depends on the room's physical properties, and hence it can be estimated from a signal reverberated in that particular room. We used a blind T_{60} estimation method proposed in [88] which estimates reverberation time from a reverberated sound signal using a statistical model for sound decay. The advantage of this method is that it can be used to estimate T_{60} just from the (reverberated) sound recordings and without any additional measurement. The limitation of this method is that, this algorithm allows estimating the T_{60} within a range of 0.2 s to 1.2 s and assumes that source and receiver are not within the critical distance. Hence, for very large rooms or halls (like Aula Carolina in the AIR dataset) having high T_{60}, this method will not work. But for almost every practical in-house scenario, the method works.

To test the accuracy of the T_{60} estimation, we took a number of clean recordings from the Emo-DB dataset and convolved them with the real impulse responses from the lecture, meeting and office rooms from AIR dataset with different

Room	**Speaker-to-Microphone Distances**	**Average True T_{60} (Schroeder's method)**	**Average Estimated T_{60}**
Office room	1m, 2m, 3m	0.56s	0.63s
Meeting room	1.45m, 1.7m, 1.9m, 2.25m, 2.8m	0.30s	0.25s
Lecture room	2.25m, 4m, 5.56m, 7.1m, 8.68m, 10.2m	0.84s	0.80s

Table 5.2: Average true T_{60} vs. average estimated T_{60} in different rooms

source-to-microphone distances. Then we blind estimated the corresponding T_{60} of the reverberated signal for that particular room and source-to-microphone distance. We also measured the true T_{60} from the impulse responses in the AIR dataset using Schroeder's method [108], and compared the estimated T_{60} with true T_{60}, as shown in Table 5.2. We found the discrepancy between average true and estimated T_{60} being 7 ms, 5 ms and 4 ms for office room, meeting room and lecture room, respectively. It must be noted that, the T_{60} estimation for Aula Carolina church was not possible because of its very large dimensions (18m x 30m x 15m), and the true T_{60} for Aula Carolina using Schroeder's method was found to be 4.5+ seconds, where the estimation can estimate T_{60} up to 1.2 second.

5.6.2 Results

Evaluations were done on meeting, lecture and office rooms from the Emo-DB-AIR dataset and on the Emo-DB-Array dataset.

Emo-DB-AIR evaluation

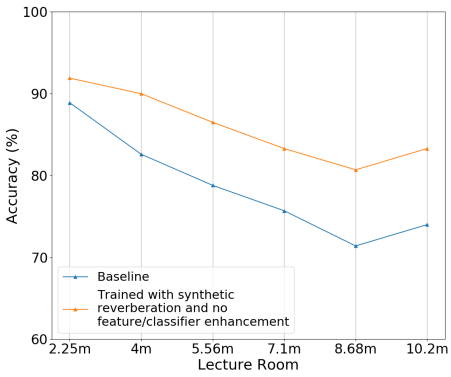
We used the room dimensions, different source and microphone positions and T_{60} reported in [103] for the rooms in Emo-DB-AIR dataset as input to the ISM simulation model. Figure 5.7 shows results of when training is done with speech from Emo-DB and testing on noisy and reverberated speech from Emo-DB-AIR (baseline). When we incorporate training with synthetic reverberation, there is a performance boost, as seen from Figure 5.7. We get 3.37%-12.56%, 2.28%-4.93%

and 0.63%-3.62% improvement for lecture, meeting and office rooms, respectively at different distances when we train with synthetic reverberation compared to the baseline. Figure 5.8 shows the results when the best 3271 features (instead of original 6552 features) with the IDFC procedure and classifier optimization has been used with synthetic reverberation which results in 3.78%-18.81%, 0.53%-3.55%, and 1.14%-4.97% improvement for lecture, meeting and office rooms, respectively at different distances. The final emotion classification accuracy achieved has an average accuracy of 88.75%, 94.84% and 93.89% for lecture, meeting and office rooms, respectively.

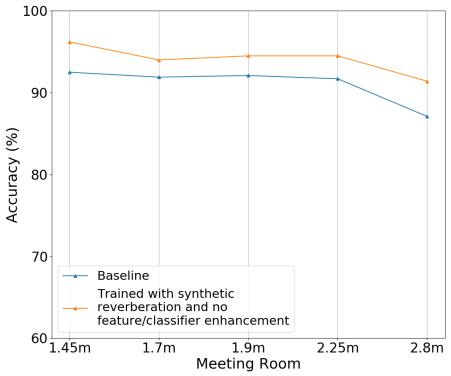
Emo-DB-Array evaluation

We measured the lab dimensions where we recorded the Emo-DB-Array dataset, with positions of the loudspeaker and microphones for various source-to-microphone distances, and estimated T_{60} from Schroeder's method as input to the ISM simulation model. However, in contrast to Emo-DB-AIR dataset, we used the default setup of absorption coefficients in the ISM simulation model with equal relative absorption coefficients (instead of real absorption coefficients) for the lab walls to observe the performance under this constraint.

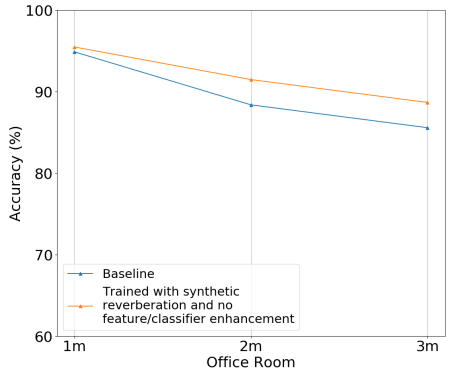
Figure 5.9 shows the performance for baseline (no training transformation done) compared with when training with synthetic reverberation. Training with synthetic reverberation slightly degrades performance in shorter distances but results in 6.37%-13.66% improvement across longer source-to-microphone (5-7 m) distances. Figure 5.10 shows the results when the best 3271 features (instead of original 6552 features) with the IDFC procedure and classifier optimization has been used with synthetic reverberation which slightly degrades performance in shorter distances but results in upto 5.41% improvement across longer source-to-microphone distances for Emo-DB-Array dataset. At the end, a final classification accuracy of 95.89%, 93.08%, 87.85% and 86.54% are achieved for 1m, 3m, 5m and 7m source-to-microphone distances, respectively, with an improvement of 2.06%, 9.72%, 12.52% and 15.51% from the original baseline, respectively. The improvement increases with increasing source-to-microphone distance, as the



(a) Lecture room

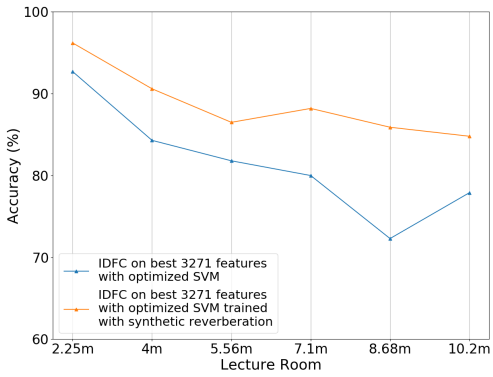


(b) Meeting room

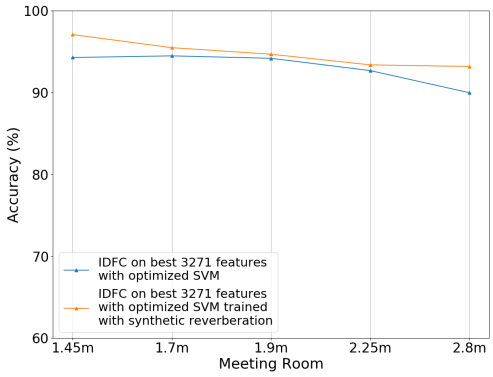


(c) Office room

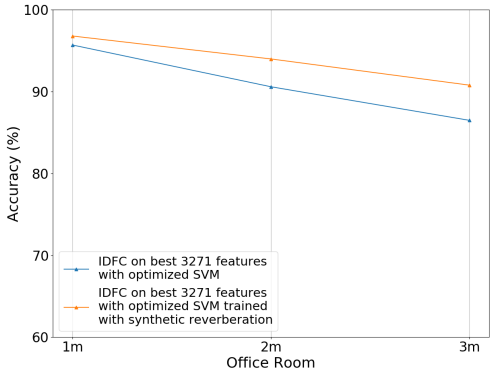
Figure 5.7: Training with synthetic reverberation results in 3.37%-12.56%, 2.28%-4.93% and 0.63%-3.62% improvement for lecture, meeting and office rooms, respectively at different distances for Emo-DB-AIR dataset.



(a) Lecture room



(b) Meeting room



(c) Office room

Figure 5.8: Training with synthetic reverberation accompanied with IDFC on best feature selection and classifier optimization results in 3.78%-18.81%, 0.53%-3.55%, and 1.14%-4.97% improvement for lecture, meeting and office rooms, respectively at different distances for Emo-DB-AIR dataset.

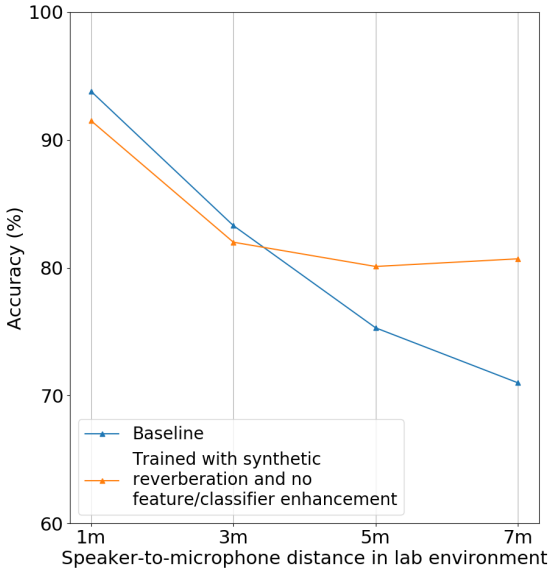


Figure 5.9: Training with synthetic reverberation slightly degrades performance in shorter distances but results in 6.37%-13.66% improvement across longer source-to-microphone distances for Emo-DB-Array dataset.

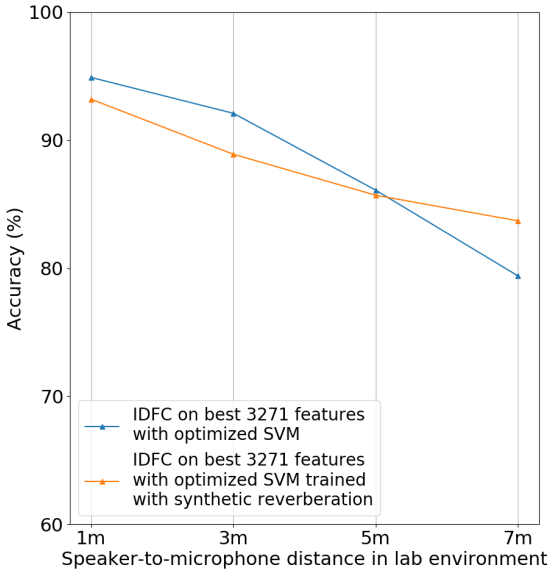


Figure 5.10: Training with synthetic reverberation accompanied with IDFC on best feature selection and classifier optimization slightly degrades performance in shorter distances but results in upto 5.41% improvement across longer source-to-microphone distances for Emo-DB-Array dataset.

Task		Computation time (s)
SVM training & classification		0.15
6552 feature extraction		0.03
CDR	No noise coherence	0.21
	No DoA	0.21
	Known DoA and noise coherence	0.26
WPE (per utterance)		1.91
T_{60} estimation	Schroeder's method	0.02
	Blind estimation	11.17
ISM simulation & convolution		1.28

Table 5.3: Computation time for various system tasks

signal distortions at near distances are too small for the training transformation to be as effective at further distances. Note that, the lab had loud background HVAC noise present, under which these improvements were obtained.

From these results, we conclude that the feature and classifier enhancement techniques combined with training with synthetic reverberation improves distant emotion recognition in a variety of situations, even with loud background noise.

5.7 Benchmarking

5.7.1 CPU Time

We did all our experiments on a workstation having a Core i7-2600 CPU with 3.40 GHz clock frequency and 8 GB memory. We benchmark the computation time for SVM model building and classification, feature extraction, CDR and WPE dereverberation and denoising, T_{60} estimation using Schroeder's method and blind T_{60} estimation and impulse response simulation using ISM method with fast convolution, as shown in Table 5.3. Computation time is the time spent running the particular task plus running OS code on behalf of the task.

As seen from Table 5.3, some tasks (like WPE dereverberation, blind T_{60} estimation, ISM simulation) have high CPU execution times even for a powerful workstation we used in our experiments, and pose a challenge for RTDER. Other tasks have low CPU execution times and can run in real time. Blind T_{60} estima-

tion, which has extremely high latency, is needed only once for a particular room for input into the ISM model, and needs not to be run in real time. The ISM simulation and convolution computations are also needed once for a particular room and a speaker-microphone configuration. If the speaker position is static (like sitting by a dining table or in a living room), ISM model needs to be computed once for training and hence needs not run in real time. However, if the speaker is moving, ISM needs to be updated as the speaker moves, which needs real time ISM computation based on speaker position. Schemes like advanced computation of all possible ISM models can be incorporated to minimize real time ISM computation (discussed in section 5.9). And, low latency CDR can be used instead of comparatively higher latency WPE for dereverberation and denoising for smooth real time operation.

Note that, the CPU times reported in Table 5.3 will increase in orders of magnitude if run on more resource constrained hardware like the Arduino/Raspberry Pi or a smartphone. In such cases, the corresponding system components may not be able to execute real time without a cloud service. However, we argue that the most likely applications of a RTDER system are safety-critical in nature (vehicle/aircraft safety, patient safety, occupant safety in smart homes) and therefore it is expected that the system components would run on powerful machines to ensure real time execution.

5.7.2 Localization Error Sensitivity

LocoVocal needs to do real-time human localization for choosing a location relevant model for distant emotion recognition. Although we prototyped LocoVocal with a Kinect which does cm level accurate human localization, it is not practical to use due to its limited range. A practical alternative is to use the in-home WiFi infrastructure for human localization, which has coverage in the entire room. However, WiFi based localization is not as accurate as a Kinect and will result in some errors, typically of several meters. Therefore, we evaluate how human localization error affects overall emotion detection accuracy. The evaluation was done using the Emo-DB-Array dataset recorded in the lab, and the results are

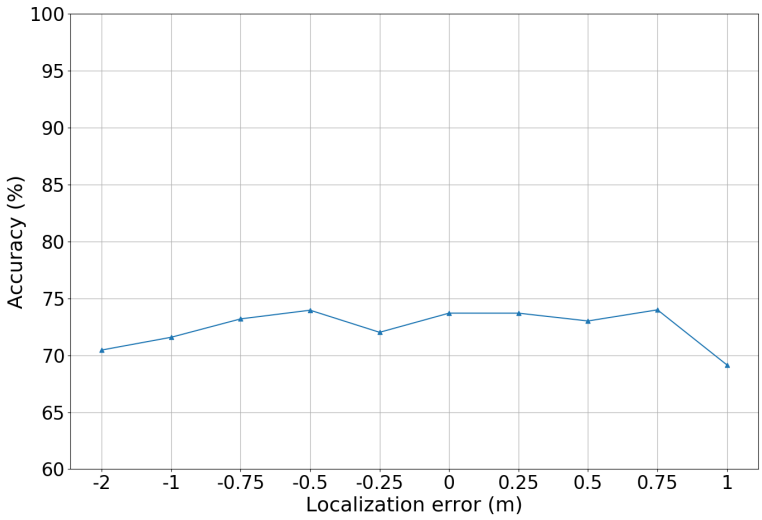


Figure 5.11: Localization error effect on emotion detection performance.

shown in Figure 5.11. The results show that if there is small localization error (up to 0.75 meters), then accuracy drop is minimal compared to no localization error. However, accuracy drops by up to 4.56% if the localization error is 1 meter or more. In rooms with higher reverberation effect, this degradation due to localization error is likely to be higher.

5.8 YouTube Dataset Evaluation

All our evaluations in sections 5.4, 5.5, and 5.6 were based on the Berlin Emotional Speech Database Emo-DB and its two distance aware variants created by us: Emo-DB-AIR and Emo-DB-Array. In this section, we experimented with a different dataset made from both acted and real emotional incidents taken from a number of YouTube video clips, as opposed to only acted artificial utterances of Emo-DB.

We collected 37 emotional clips from YouTube spanning more than 3 hours of emotional speech. Two persons labeled them into 4 emotion categories: angry, happy, neutral and sad. The angry recordings included clips from the talk show "The Daily Show" with the host reacting on the South Carolina church shooting

in 2015, a talk show from CNN with the participants reacting about gun control, a heated speech from a presidential candidate of the US national election 2016, and a number of clips taken from TV shows and movies from YouTube. Among the happy recordings are some personal statements released by a number of people and some funny clips taken from NBC talk shows. The neutral recordings consisted of a documentary recording about the US constitution, with a few others. The sad recordings included a number of personal statements about abuse, depression, monologues about deceased people and a number of clips from TV shows and movies.

We split these recordings into a total of 1124 10 second utterances and convolved them with the meeting and office room impulse responses from the AIR database with various source-to-microphone distances. We transformed the training by injecting synthetic reverberation from the impulse response simulator, as described in section 5.6. We also selected the best 3255 features from the F-scores of the original 6552 features and applied the IDFC procedure on the features of both training and test set with the optimized SVM. We achieve an average accuracy increase of 3.11% and 7.30% from the baseline (no training transformation or feature enhancement done) in the meeting room and office room, respectively, with a final emotion detection accuracy of 93.68% and 93.15%, respectively, across all the source-to-microphone distances in corresponding rooms.

From these results, we verify that our approach improves distant emotion recognition accuracy for realistic emotional speech data as well as acted corpus as shown in prior sections.

5.9 Discussion

5.9.1 Public Dataset vs. Real Deployment

We limited our experiments to the 2 distance aware variants of the Emo-DB public dataset: Emo-DB-Array and Emo-DB-AIR, which we customized according to

our needs. These 2 are the very first distance aware emotional datasets, to the best of our knowledge. Since RTDER is a new area of research, we present our preliminary results based on variants of the public Emo-DB dataset in this paper. In our future subsequent work, we plan to include RTDER results based on real deployments in real families.

5.9.2 Real Time Computation Scenario for Static vs. Dynamic Speakers

For static fixed position speakers, feature extraction, signal cleaning (WPE or CDR), and SVM classification needs to be done real time with the audio stream. Reverberation adaptation (T_{60} estimation, ISM simulation, convolution) and SVM training need not to be done real time, as they need to be computed just once for static speakers and a certain indoor environment.

However, for dynamically moving speakers, the room IR needs to be updated real time as the speaker moves, therefore ISM simulation, convolution, and SVM training also needs to be computed in real time. As discussed in section 5.7, ISM simulation and convolution are computation heavy tasks, and challenging to be computed in real time, especially in resource constrained platforms like Raspberry Pi. One possible solution is advanced computing of all possible room IRs by all possible speaker-to-microphone distances into an IR cache, and use the appropriate IR from the cache depending on the latest speaker position for convolving with the speech signal. Another solution is to use opportunistic room IR computation based on latest speaker position. For example, if the current speaker-to-microphone distance is 3 meters, then compute IRs with 2.9m and 3.1m in advance and use the one whichever happens to be the next distance (assuming 0.1m distance precision) with the opportunistic scheme. T_{60} estimation, even for dynamically moving speakers, needs to be computed just once for a particular room, hence need not to be computed in real time.

True	Classified As						
	Anger	**Anxiety**	**Boredom**	**Disgust**	**Happy**	**Neutral**	**Sad**
Anger	117	0	0	1	8	1	0
Anxiety	4	56	0	0	6	3	0
Boredom	0	0	70	0	0	5	5
Disgust	0	2	1	38	1	2	2
Happy	13	3	0	1	53	1	0
Neutral	0	1	3	0	0	75	0
Sad	0	0	2	0	0	2	59

Table 5.4: Confusion matrix for detected emotions

5.9.3 Confusion Matrix

With our proposed techniques, we demonstrated improvement in real time emotion recognition performance over distance. However, scope exists for betterment of the core solution, i.e., emotion detection with clear speech, upon which our solutions stand. We analyzed the confusion matrix of 7 different emotions from Emo-DB after a 10-fold cross validation on the clear speech data, as shown in Table **??**. We noticed that a significant number (around 33%) of misclassifications occur between the angry and happy emotions. Techniques involving hierarchical classifiers with specialized angry vs. happy separators, textual features added with acoustic features after a speech-to-text conversion can improve the misclassification rate between angry and happy classes.

5.10 Summary

We present novel solutions to various challenges in the processing pipeline of an acoustic RTDER system. The solutions we present are useful in real time recognition of emotions from distant speech in a variety of rooms with various acoustic configurations and source-to-microphone distances. We provide empirical evidence that our novel combination of feature selection, classifier optimization and distorted feature elimination technique combined with WPE and CDR dereverberation and denoising algorithm is capable of increasing emotion detection accuracy as much as 10.84%, with the final accuracy ranging between 79.44%-94.95%. In addition, our feature and classifier enhancement and distorted feature elimination technique combined with training with synthetic

reverberation from a room impulse response generator increase emotion detection accuracy as much as 15.51% across various rooms, acoustic configurations and source-to-microphone distances, with the final accuracy ranging between 87.85%-95.89%. We considered challenging situations with extremely reverberant church halls and noisy backgrounds with loud HVAC noise and obtained improvement even in such conditions. A case study on a dataset made from 37 realistic YouTube videos spanning 4 different emotions demonstrates a maximum of 7.30% increase in accuracy for detecting real world distant emotions in different rooms, with a maximum accuracy of 93.68%. We evaluated the CPU runtime of various system components and demonstrate the real time execution capability of our system. We concluded with discussing some limitations of our current solutions, and proposed methods to solve those in future works.

Chapter 6

mLung and DeepLung Design and Implementation

Smartphones have a great potential to be used as a home healthcare and patient monitoring tool. In this chapter, we present two novel smartphone based solutions, mLung and DeepLung, for pulmonary anomalous sound detection like cough and wheeze for longitudinal remote monitoring of pulmonary patients. We describe how they sense a physiological phenomenon like respiratory rate in real-time and adapts the machine learning pipeline for improved performance of detecting lung anomalies.

The rest of this chapter is organized as follows. Section 6.1 discusses the motivation of using mLung and DeepLung. Section 6.2 lists the technical contributions that this chapter makes. Sections 6.3 provides an overview of the dataset preparation for our experiments. Sections 6.4, 6.5, and 6.6 describes mLung system architecture, in-phone processing details, and in-cloud processing details. Section 6.7 provides the details of various DeepLung system components. Finally, Section 6.8 provides a summary of the chapter.

6.1 Motivation

Chronic pulmonary diseases are one of the main causes of death in the United States causing nearly 7 percent of all deaths in 2015 [109]. In the last 35 years, more than 4.6 million Americans died from a range of these diseases, like chronic obstructive pulmonary disease (COPD), pneumoconiosis, and asthma [109]. An estimated 40 million people in the US presently are living with these diseases [110, 111], causing more than $130 billion in health-care costs [112]..

Chronic pulmonary diseases cause obstruction of the respiratory airway. Therefore, COPD exacerbation or asthma attacks are often accompanied by abnormal lung activities in the patient like coughing and wheezing due to airflow into the narrowed airway. With the proliferation of smartphones and its associated sensors in the patient population, it is possible to use the smartphone as a pulmonary care device to monitor these abnormal lung activities like coughing and wheezing. Such physiological audio sensing by the smartphone microphone is challenging due to limited computation and battery power of the device, and concerns about the patient's privacy. In addition, there is a paucity of well annotated pulmonary audio activity data collected by commodity smartphones to make appropriate machine learning models.

To address these needs, in this chapter we present the architectural concept of mLung and its deep learning counterpart DeepLung, which are privacy preserving, naturally windowed abnormal lung sound detection services in smartphone for pulmonary patients. The mLung system, when held by the chest, listens to the internal body sounds and detects unusual lung sounds like wheezing and coughing. mLung detects the respiration cycle in real-time from chest movements using inertial sensors, and segments the audio data by the natural respiration window. This ensures that all the distinct phases of a cough or wheeze resides in a single audio segment as opposed to traditional static length windowing which is agnostic about the physiological endpoints of a lung event. Our proposed mLung architecture employs a mobile-cloud hybrid classification scheme where an in-phone lightweight classifier filters out speech frames, while an in-cloud

expert and heavyweight classifier performs the complex lung sound classification task. Thus mLung not only ensures the patient's privacy by not allowing raw speech frames to leave the local device, but also off-shores compute intensive lung sound classification tasks to the powerful cloud for better performance. Similar state-of-the-art body sound sensing systems [16, 17] are bound to use limited acoustic features, static length windows, and external microphone hardware to solely run on the energy constrained phone to protect user privacy, while mLung runs just with commodity smartphone hardware with a better architectural design by dividing tasks between the phone and the cloud, still protects patient privacy, and results a better classification performance.

DeepLung also listens to the internal body sounds using the built-in phone microphone and fine trained convolutional neural network (CNN) either in the phone or in the cloud and detects these unusual lung sounds like coughing and wheezing. DeepLung can adopt either a dynamic respiration cycle based physiological and natural windowing of audio, or a static length windowing. The respiration cycle is detected from the expansion and contraction of the patient's chest from the phone's inertial sensors' readings. DeepLung uses the Interspeech 2010 Paralinguistic Challenge features [113] as input to it's 1-D and 2-D CNNs for classifying lung and other confounding body sounds. DeepLung classifiers have been designed considering 2 possible system architectures: i) a mobile-cloud hybrid architecture where the phone captures audio, does windowing, preprocessing and speech filtering, and then the lung sounds are classified in the cloud, and ii) a mobile in-situ architecture where all processing is done in the phone. DeepLung ensures patient privacy in the mobile-cloud architecture by filtering out speech frames in the local phone with high accuracy using a shallow classifier.

To tackle the challenge of limited available pulmonary activity data to make appropriate deep learning models, we collected audio and inertial sensor data using a smartphone from 131 pulmonary patients and healthy subjects over controlled and well designed pulmonary activities. We collaborated with a crowdsourcing company to ensure high quality annotation of different pulmonary anomalous events in this novel dataset. A series of experiments were performed

using this novel dataset and performance was compared with the BodyBeat [16] system which uses specialized microphone hardware to classify body sounds on a smartphone.

Several other shallow and deep learning based pulmonary activity detection works like wheeze detection [42–50] and cough detection [13, 51, 52] exist in the literature. However, they often use limited training data which is not collected with a commodity smartphone. Our models are built using the first ever pulmonary activity data collected from 131 real patients and healthy subjects using smartphones, and trained using extensive acoustic features which are tailored to detect non speech human sounds.

6.2 Contributions

The contributions of mLung and DeepLung are:

- Experimental results verified by the first, extensive lung sound dataset recorded with a commodity smartphone from 131 pulmonary patients and healthy subjects, and annotated by professional crowd-sourcing company and expert researchers, which includes more than 45,000 instances of lung sounds, speech and other confounding sounds.
- A novel dynamic respiration based natural windowing of audio for pulmonary sound classification.
- Designing and comparing 1-D and 2-D CNNs for pulmonary anomaly detection using Interspeech 2010 Paralinguistic Challenge acoustic features.
- Comparison of lung sound detection performance between respiration cycle based windowing and static sized windowing on this novel dataset.
- Intelligent architecture comprising mobile-cloud hybrid classification scheme ensuring patient privacy and better classification performance than the baseline without using any additional specialized microphone due to natural windowing and superior features and classifier in the cloud.

6.3 Data Preparation

6.3.1 Pulmonary Data Collection

We create an empirical pulmonary activity dataset by collecting audio and inertial sensor data using a Samsung Galaxy Note 8 smartphone on the chest from 131 human subjects. Among them are 91 chronic patients (41 male, 50 female). 69 of them were diagnosed with asthma, 9 with COPD, and 13 exhibited co-morbidity of both asthma and COPD. For breathing cycle ground truth, a Zephyr Bio-harness was worn by each subject. On average, we collected around 40 minutes of continuous audio and inertial sensor data from each subject.

We collected pulmonary activity data across seven different tasks from the 131 subjects with the phone placed on their chest to make the dataset rigorous in terms of data quality and variability. These tasks were chosen because they were likely to invoke pulmonary anomalous sounds like coughing and wheezing in the subjects. The study protocol consisted of the following tasks:

1. *Pulmonary Function Test*: A total of three efforts for pulmonary function test were recorded.

2. *Cough*: Cough voluntarily for up to two minutes based on the comfort of the subject, in case the subject is not having natural coughs. Patients were asked to cough as naturally as possible.

3. *A-vowel*: Making 'Aaaa....' sound continuously for as long as the subject can. This task lasted up to one minute.

4. *Speech*: Speaking freely on any topic the subject is comfortable with. This task lasted 3-5 minutes.

5. *Reading*: Reading a neutral paragraph loudly for 3-5 minutes.

6. *Sit-silent*: Sitting silent for one minute and counting the breaths while keeping the phone on the chest.

7. *Supine-silent*: Lying down for one minute and counting the breaths while keeping the phone on the chest.

6.3.2 Data Annotation

Due to the challenge of annotating the large amount of collected data, we collaborated with a crowd-sourcing company to collect non-expert annotation of various pulmonary activities. The annotators are given an interface to mark the start and end of each cough, wheeze, speech and confounding sounds. The same interface also provide training to the non-expert annotators with examples of each of the events to annotate, and how to mark each of them in the audio. This annotation is also given to another set of annotators to review and correct possible incorrect annotations. A part of the non-expert annotations are also revised by expert researchers in this domain to assess and confirm the accuracy of the annotations by the non-expert contributors. In total, we have a dataset with more than 3,500 instances of cough, wheeze, throat-clearing, abdominal sounds, and other confounding noises, and more than 42,000 instances of speech. To the best of our knowledge, this is the first pulmonary activity dataset collected solely with commodity smartphone sensors without using any additional customized microphone from such a large number of real pulmonary patients.

6.3.3 Data Normalization

The volume of each raw audio frame in the dataset was normalized to 1 by dividing the whole frame by the frame peak value. This was done to eliminate any bias due to frame volume for the classifiers.

6.4 mLung System Architecture

mLung is comprised of three major architectural components: sensing and breathing detection, in-phone admission control, and in-cloud sound classification, as shown in Fig. 6.2.

Figure 6.1: Example usage of mLung by a pulmonary patient.

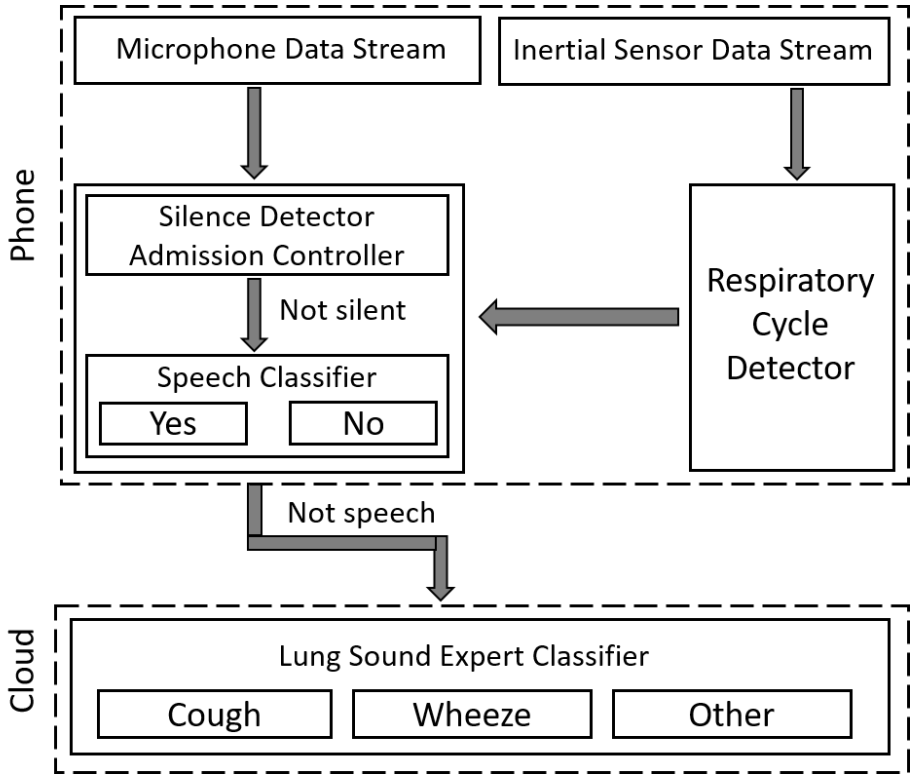


Figure 6.2: mLung architecture comprising of respiration cycle detector, phone admission controller and speech detector, and cloud expert classifier.

mLung detects the expansion and contraction of the chest due to breathing and hence the respiration cycle time-stamps using the tri-axial accelerometer and gyroscope of the phone when held against the chest by a pulmonary patient as

shown in Fig. 6.1. The respiration cycles define the windows for the corresponding audio stream for lung sound detection.

mLung listens to the lung and confounding body sounds by the smartphone microphone and segments the streaming audio according to the respiration cycles. The in-phone admission controller uses an energy-threshold based filter to filter out silent windows, and a lightweight binary classifier to filter out frames containing speech and non-body external sounds like external noise. Then it uploads the remaining sound frames containing only in-body sounds to the cloud for further processing. Filtering out speech frames from being sent to the outside world ensures user privacy.

The cloud component of mLung extracts an extensive number of energy, spectral, cepstral and mel-frequency based acoustic features with a number of high-level statistical functionals from the in-body sound frames and uses a fine-tuned supervised expert classifier to detect common lung originated sounds like cough and wheeze, and other confounding body sounds like abdominal sounds (e.g., bloating) and throat clearing. This expert classifier is offline trained, aggressively tested and highly tuned to recognize lung-specific sound activities.

6.5 mLung In-Phone Processing

6.5.1 Respiration Cycle Detection

mLung uses a modified version of InstantRR [114] to identify the respiration cycles from the raw inertial sensor data. mLung fuses 3-axis accelerometer and 3-axis gyroscope data when the phone is placed against the chest. Data is smoothed along each axis using a butterworth bandpass filter between 0.13Hz to 0.66Hz corresponding to a breathing rate between 8 BPM to 40 BPM, followed by application of a Savitzky-Golay filter [115] to further smooth the data and extract the breathing waveform. Fast Fourier transform is applied on the smoothed signal to compute the dominant frequency on each timeseries signal as a candidate axis for breathing cycle identification. Finally, we select the axis that shows the

maximum periodicity [116] and apply a peak detection algorithm to detect peaks and valleys of each respiratory cycle [117]. A dynamic window corresponding to a complete breath cycle covers the duration between two successive valleys of the breathing waveform.

6.5.2 Admission Control by Silence Filtering

mLung windows the streaming audio based on the respiration cycle and analyzes it to detect sound activity. The silence detector is a threshold-based classifier which acts as an admission controller for the subsequent classification phase. It computes the frame energy, compares it against a configurable threshold, and if it exceeds this threshold, the window is passed on to the in-phone speech classifier.

6.5.3 Speech Filtering

mLung uses a lightweight binary classifier with a limited number of acoustic features suitable to run in the phone to separate and filter out speech frames from non-speech lung sounds to protect the user's privacy. Body originated non-speech lung sounds are generated by complex physiological lung processes and these sounds are situated within the lower region of the frequency spectrum, ranging between 20Hz to 1300Hz, while speech ranges in a much higher region between 300Hz to 3500Hz [16]. Because of the high separability between speech and lung sounds with limited features as shown in Fig. 6.3, it is possible to accurately separate them with a lightweight classifier to run on a resource constrained smartphone.

We explored a number of frame level acoustic features like root mean square energy, 12 mel-frequency cepstral coefficients (MFCC), zero crossing rate (ZCR), voicing probability, fundamental frequency, and a number of statistical functionals on the breathing window level segments like arithmetic mean, standard deviation, skewness, kurtosis and 2 linear regression coefficients. We selected the best features based on the mutual information, which chooses mutually independent

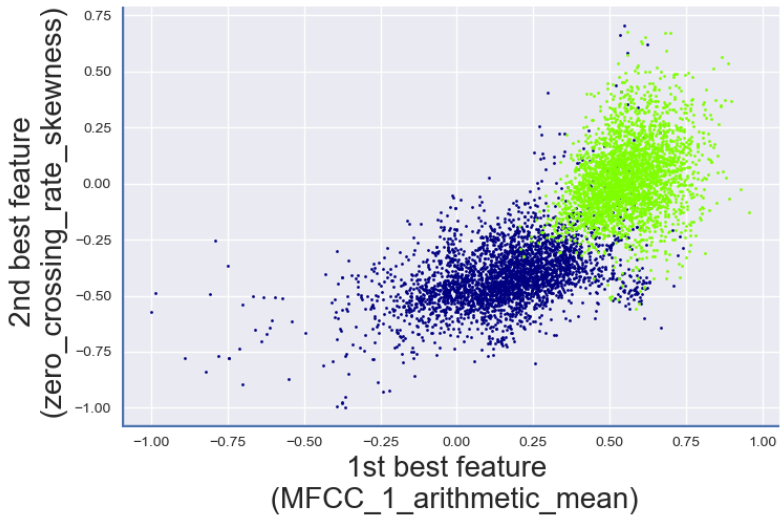


Figure 6.3: Scatter-plot of 6000 speech (light green) and lung sounds (dark blue) in a 2-dimensional best 2 feature space shows high degree of separability.

features highly correlated to the target class. As shown in Section 7.1, the speech detection f1-score varies between 96%-98% when the 5-15 best features are used. mLung filters out the speech frames for user privacy, and uploads the lung and body sound frames to the cloud for further processing.

6.6 mLung In-Cloud Processing

At the cloud, mLung normalizes the raw lung sound frame volumes to 1 by dividing the whole frame by the frame peak value to eliminate bias due to variation of loudness across different sound frames. Then it extracts a total of 1582 features as 21 functionals of 38 acoustic low-level descriptors (LLD) and corresponding first order delta regression coefficients related to energy, pitch, spectral, cepstral, mel-frequency, voicing probability, shimmer and jitter. The 21 statistical functionals are applied to the LLDs computed from each of the lung sound frame within a respiration cycle to map a time series of variable length (based on breathing cycle) onto a static fixed size (1582) feature vector. We chose these features (listed in Table 6.1) as they have been demonstrated to perform well in human paralinguistic sound detection [113].

Low Level Descriptors	Functionals
PCM loudness	position maximum/minimum
MFCC (0-14)	arithmetic mean, standard deviation
Log Mel Frequency Band [0-7]	skewness, kurtosis
Line Spectral Pair Frequency [0-7]	linear regression coefficient 1st, 2nd
Fundamental Frequency	linear reg. error (quadratic, absolute)
Fundamental Frequency Envelope	quartile 1st, 2nd, 3rd
Voicing Probability	quartile range 2—1/3—2/3—1
Jitter local	percentile 1st, 99th
Jitter consecutive frame pairs	percentile range 99—1
Shimmer local	up-level time 75/90

Table 6.1: mLung and DeepLung low and high level features used in cloud.

mLung scales each feature vector and uses a supervised support vector machine (SVM) classifier to recognize common lung sounds i.e., cough and wheeze. SVM classifiers are lightweight and have been demonstrated to be suitable to use in a smartphone [118]. To handle other sounds, it has an additional class for non-lung originated confounding body sounds like abdominal sounds and throat clearing. The scaled feature vectors for each class are used to train the SVM, and the SVM hyperparameters are optimized (polynomial kernel, $c = 10$ and $\gamma = 0.1$) using a grid search.

6.7 DeepLung System Components

6.7.1 Architecture

We design and develop DeepLung algorithms considering two different implementation architectures for comparison purposes as follows:

Mobile-cloud Hybrid Processing

In this architecture, the phone normalizes and windows the audio stream coming from the patient's body, and filters out speech frames using a shallow in-phone classifier for patient privacy. Then it uploads the body sound frames to the cloud where a CNN does lung sound (cough and wheeze) classification. For handling confounding body sounds (throat clearing, abdominal sounds) there

is an additional class, hence the CNN in the cloud does a 3-class classification: cough, wheeze, and other.

Mobile In-situ Processing

In this architecture, the mobile does in-situ end-to-end audio processing, and the sound frames never leave the device (as opposed to uploading to the cloud). An offline trained CNN in the phone does a 4-class classification: speech, cough, wheeze, and other.

6.7.2 Audio Windowing

For comparison, we experimented with two different audio windowing schemes for DeepLung as following:

Respiration Cycle Based Natural Windowing

Since DeepLung operates when a patient holds the phone by the chest, the respiratory cycle can be detected in real-time from the expansion and contraction of the chest using the phone's inertial sensors [114], and the audio stream can be windowed by the respiratory cycle for classification. Such respiratory cycle based natural windowing ensures that all the distinct phases of a cough and a wheeze resides in a single window.

Static Windowing

We also experimented with static fixed size windows of 60 ms length and 10 ms shift to extract low level acoustic features to train the CNN. Such static windowing results in a much larger amount of training instances compared to respiration based natural windowing.

6.7.3 In-phone Speech Filtering

In the mobile-cloud hybrid processing architecture, DeepLung uses a shallow support vector machine (SVM) classifier to separate and discard speech frames from being uploaded to the cloud for patient privacy. Because of the complex physiological generation process, lung and other in-body sounds are situated at a much lower region (20-300 Hz) of frequency spectrum than speech (300-3500 Hz) [16], and hence have high separability. A similar approach has been shown to separate speech from internal body sounds with high accuracy using a limited number of acoustic features [5].

6.7.4 Feature Extraction

We used the Interspeech 2010 Paralinguistic Challenge feature set [113] to extract a total of 1582 acoustic features from the normalized audio frames using the OpenSmile [119] tool, as these features have been tailored to detect the non-speech human sounds. The tool first extracts 38 acoustic low-level descriptors (LLD) related to energy, pitch, spectral, cepstral, mel-frequency, voicing probability, shimmer and jitter, and their first order delta regression coefficients. Then 21 different statistical functionals are applied to the LLDs to map a audio time series signal of variable length (based on respiration window duration) into a static sized (1582) feature vector. For respiration cycle based windowing of audio, we used all the 1582 features, while for static sized 60 ms window, we used only the 38 LLDs as features. Table 6.1 summarizes the features.

6.7.5 Deep Neural Networks Structure

For our analysis to classify the lung sounds, we used both 1-dimensional and 2-dimensional CNN models, so that we can compare their performances with each other.

Network	Detail
1-D CNN for respiration windowed audio	conv1: 8 filters, kernel size 3, 1 stride pooling: pool_size 2, 1 stride classification layer: softmax
1-D CNN for static windowed audio	conv1: 8 filters, kernel size 3, 1 stride conv2: 8 filters, kernel size 3, 1 stride pooling: pool_size 2, 1 stride fully connected layer: 100 neurons classification layer: softmax
2-D CNN for respiration windowed audio	conv1: 8 filters, kernel size 3x3, 1x1 stride pooling: pool_size 2x2, 1x1 stride classification layer: softmax
2-D CNN for static windowed audio	conv1: 8 filters, kernel size 3x3, 1x1 stride conv2: 8 filters, kernel size 3x3, 1x1 stride pooling: pool_size 2x2, 1x1 stride fully connected layer: 100 neurons classification layer: softmax

Table 6.2: Structure of different CNN models

1-D CNN Models

For respiration cycle windowed audio represented as 1582 features, we used a CNN with 1 convolutional layer and 1 max pooling layer. Instead of adopting a fully-connected layer for classification, the classifier outputs the category confidence via the max pooling layer, and then the resulting vector is fed into the output (softmax) layer [120]. Since, respiration cycle based samples are comparatively fewer in number, we did not use a fully connected layer to reduce the number of parameters and reduce chances of overfitting [121]. For static windowed audio represented as 38 features, we used a CNN with 2 convolutional layers, 1 max pooling layer, and a fully connected layer, as samples are much larger in quantity (more than 800,000).

2-D CNN Models

We transformed the 1-D feature vectors into 2-D by transposing it and taking a dot product with the original feature vector. Since the feature space for respiration cycle windowed audio is high (1582), we reduced it's dimension by choosing the best 50 features using mutual information before transforming it to a 50x50 dimension 2-D feature vector to reduce memory consumption. The static windowed 1-D features were transformed to 38x38 size 2-D vectors similarly. Then we use them to train 2-D CNNs similar in structure to the 1-D

CNNs. Table 6.2 describes the structures of the different CNNs we used to make DeepLung models.

6.8 Discussion and Summary

This chapter presents the overview of mLung and DeepLung which are privacy preserving, naturally windowed, mobile-cloud hybrid pulmonary care service for detecting lung activities. During use, a patient holds the phone by his chest for a short time (few minutes) and mLung detects respiratory cycles from the breathing motions of the chest and naturally windows the audio stream. A lightweight in-phone classifier filters speech for user privacy, and a heavyweight, fine tuned, expert in-cloud classifier does lung sound classification. DeepLung, the deep learning counterpart of mLung, uses 1-D and 2-D CNNs for the lung sound classification task.

Smartphones will be the basis for many future medical, health and wellness applications serving the patient and elderly population, and caregivers. mLung and DeepLung will be an enabling technology by enabling caregivers to routinely check the patients' pulmonary disease status and effect of a particular medication remotely, as long as patients use the system regularly as per doctor's suggestion. The chance for privacy invasion is significantly reduced because the in-phone classifier filters speech with high accuracy, which will be comforting for people to install the application in their phones. Finally, mLung and DeepLung are low cost and promising to be user friendly as no external microphone is needed.

We primarily focused on detecting pulmonary anomalies like coughing and wheezing. Aside from these, there are other medically defined pulmonary disease symptoms like rhonchus and crackles. As more of these sounds become available, models for detecting these can be developed with our existing CNNs. There is scope to further engineer DeepLung 2-D CNNs for further improvement of performance. As deep neural networks have been demonstrated to be more noise resilient and location independent [18] compared to shallow learners, DeepLung is more realistic to use in a real world noisy environment.

Chapter 7

mLung and DeepLung Evaluation

In this chapter, we evaluate mLung and DeepLung in terms of their ability to detect anomalous lung sounds. mLung does speech detection and filtering in the phone for user privacy, naturally windows the audio based on user's respiration cycle, and performs the complex lung sound classification in the cloud with powerful acoustic features and a shallow SVM classifier. DeepLung, on the other hand, uses 1-D and 2-D CNNs for lung sound classification using the Interspeech 2010 Paralinguistic Challenge features. Using the novel lung activity dataset collected by smartphones, we demonstrate the ability of mLung and DeepLung in speech detection, respiration cycle detection, and their ability in classifying various lung sounds by several experiments. We compare the performance of mLung and DeepLung with the BodyBeat [16] system which uses customized microphone hardware to detect in-body sounds.

	# of best features			
	3	**5**	**10**	**15**
f-1 score	0.93 ± .027	0.96±.004	0.98±0	0.988±.004

Table 7.1: Speech Detection F-1 Score for Various Number of Features

7.1 mLung Experimental Results

7.1.1 Respiration Cycle Detection

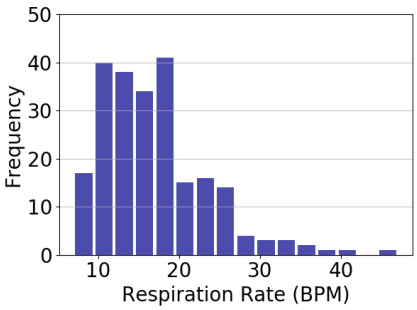
The modified InstantRR algorithm was run on the sit-silent and supine-silent inertial sensor data collected from the participants. Participants' self counts of breath cycles (histogram shown in Fig. 7.1) were used as the gold standard ground truth to compare the performance of the algorithm. Figure 7.1 shows the histogram of the differences between estimated breath counts and ground truth breath counts. The mean absolute error is 3.4 breaths per minute with most differences centered around 0.

7.1.2 Speech Detection Accuracy

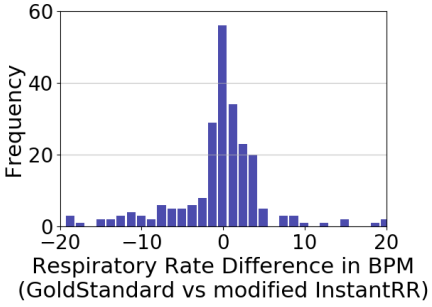
We evaluate the F1 score (harmonic mean of precision and recall) of our lightweight in-phone speech detection classifier for different feature sizes ranging between 3-15 features. The features were chosen based on mutual information. For each experiment, we randomly select a subset of 3000 speech samples (out of 42000+ samples), and a subset of 3000 non-speech lung and body sounds (out of 3500+ samples). We train our SVM classifier using two-thirds of the data, and run tests on the rest. Each experiment is repeated 5-10 times for statistical confidence. As seen from Table 7.1, F1 score is 96%+ when the 5+ best features are used.

7.1.3 Lung Sound Detection Accuracy

We use BodyBeat [16] as our comparison baseline for these experiments, as this is a similar smartphone system for detecting in-body sounds which includes lung



(a) Histogram of number of breath cycles per minute (BPM) from self counted gold standard ground truth from 131 subjects.



(b) Respiration cycle detection errors are mostly centered around 0.

Figure 7.1: Respiration cycle detection.

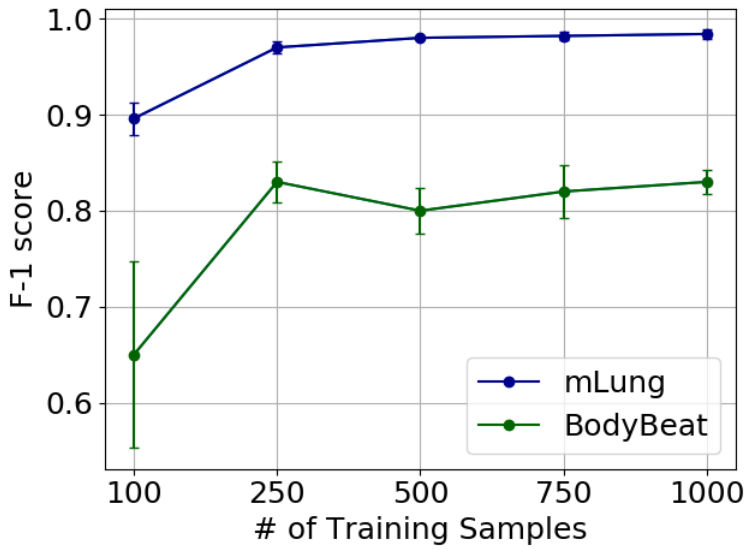


Figure 7.2: mLung is 15-25% more accurate than BodyBeat for different training sizes for detecting lung and confounding sounds like cough, wheeze and other (abdominal sounds and throat clearing).

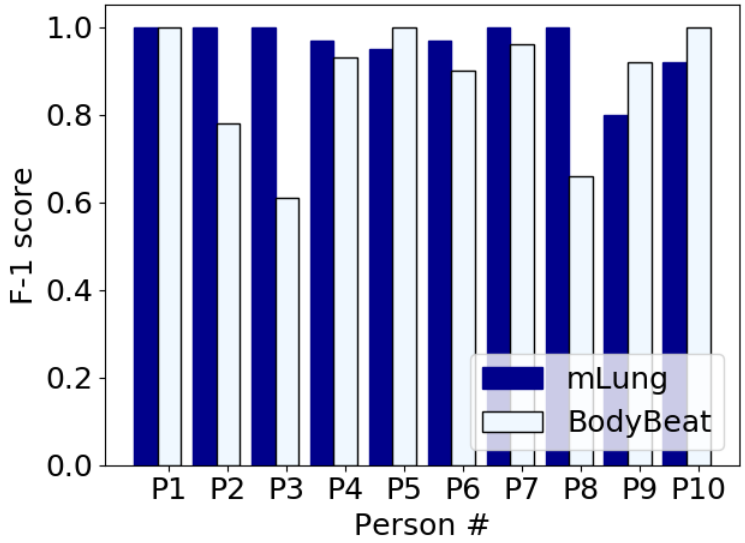


Figure 7.3: mLung outperforms BodyBeat in 7 out of 10 subjects in the personalized test by an average of 15.71%.

sounds like coughing too. We implemented BodyBeat's algorithm with a linear discriminant classifier with 30 best features based on correlation feature selection. We perform generalized, leave-1-person-out, and leave-1-sample-out personalized experiments for the 3 target classes: cough, wheeze, and others, and compare our results with the BodyBeat classifier.

Training Amount

We evaluate the F1 score of our expert in-cloud lung sound classifier in classifying cough, wheeze and other (abdominal sounds and throat clearing) confounding sounds for different amount of training ranging between 100-1000 training samples. For each experiment, we randomly divide the entire dataset into 2 halves, use 1 half for testing, and randomly choose the target amount of training samples from the other half to make classification models. Each experiment is repeated 5-10 times.

Fig. 7.2 shows the F1 score of the 2 systems for training set sizes of 100-1000 samples. BodyBeat, being run solely in phone and hence restrained to use limited features, performs poorly with limited training, and converges to 83% F1 score

when provided the largest training set. mLung, on the other hand, is always leading and its F1 score reaches to 98%, which is 15%-25% higher than BodyBeat, and proves its efficiency when limited training data is available.

Leave-1-Person-Out Test

This experiment resembles a more realistic scenario where a patient attempts to classify his lung sounds when mLung is trained with samples from other people. For this experiment, we divided the samples of our dataset to the corresponding subjects, trained the models by samples from all but 1 subjects, and tested on samples from the remaining subject. This was done for each of the 131 subjects, and each experiment was repeated 5-10 times for statistical confidence. mLung's f-1 score reaches as high as 98% for the leave-1-person-out test, which is 13% higher than BodyBeat for the leave-1-person-out test, and hence more effective in a realistic scenario.

Leave-1-Sample-Out Personalized Test

This experiment resembles a usage scenario where a patient has collected sufficient amount of his own lung sound samples over time which are then used to train a personalized mLung classifier for the individual. We picked 10 subjects who have maximum instances of cough, wheeze and other (abdominal sounds, throat clearing) samples in our dataset for this experiment, so that each personalized classifier has enough training samples from all classes. For samples for each test subject, we trained on all but 1 samples, and tested on the remaining sample. Fig. 7.3 shows that mLung outperforms BodyBeat system for 7 subjects by an average of 15.71%. BodyBeat however, is slightly better than mLung for 3 subjects (P5, P9, P10). Overall, mLung is 8.5% more accurate than BodyBeat across all the 10 subjects in the personalized test with 96.1% average f-1 score per person.

7.2 DeepLung Experimental Results

7.2.1 Training amount

We evaluate the F1 scores of our different CNNs for different amounts of training. For the dataset windowed by respiration cycles, we vary the training amount between 50-1000, and for the static windowed dataset, we vary the training amount between 10,000-200,000. We use 80% of the dataset for training, and remaining 20% for validation. From the training set, we randomly select the target amount of samples to make the classification model, and test on the validation set. Each experiment is repeated 5-10 times.

The results are presented in Table 7.2. We discuss our observations in the following sections:

1-D CNN vs. 2-D CNN

It is noted from Table 7.2 that 2-D CNN performs really poorly compared to 1-D CNN for respiration cycle windowed audio. The reason is that, we reduced the dimensionality of the feature vector from 1582 to 50 by mutual information based feature selection for memory efficiency before transforming it to 2-D, which cost loss of information leading to poor performance. For static windowed audio, the 1-D and 2-D CNN performances are close as no dimension reduction was done on the original feature space. For both cases, F1 scores increase with increase in training samples.

Natural windowing vs. static windowing

Respiration cycle based natural windowing of lung sounds performs between 4-20% better than static windowed lung sounds, as seen from Table 7.2. Respiration based physiological and natural windowing ensures that the distinct phases of lung anomalous sounds resides in a single frame, and hence have better separability.

However, we hypothesize that with a deeper and better neural network design, it is possible to further improve the performance of static windowed classifiers.

3-class vs. 4-class classification

Performances of both 3-class and 4-class CNNs are similar when a larger amount of training is available. 4-class models have an additional class of speech, which is easily separable from lung and other body sounds due to its position in a higher frequency spectrum. 4-class CNN performance is a little poorer than 3-class for respiration windowed audio when limited training is provided, but it becomes close to the 3-class models as additional training samples are provided.

Respiration windowed					**Static windowed**				
	3-class classification (in cloud)		**4-class classification (in phone)**			**3-class classification (in cloud)**		**4-class classification (in phone)**	
# of training samples	**1-D CNN**	**2-D CNN**	**1-D CNN**	**2-D CNN**	**# of training samples**	**1-D CNN**	**2-D CNN**	**1-D CNN**	**2-D CNN**
50	.80±.08	.63±.251	.72±.114	.52±.267	**10,000**	.65±.034	.71±.038	.68±.037	.69±.03
100	.92±.035	.70±.284	.84±.06	.48±.3	**50,000**	.72±.028	.76±.02	.77±.017	.78±.022
500	.96±.012	.34±.137	.96±.009	.68±.384	**100,000**	.79±.015	.80±.009	.81±.009	.80±.01
1000	.98±.002	.60±.296	.98±.004	.97±.011	**200,000**	.82±.012	.82±.01	.84±.005	.82±.01

Table 7.2: F1 scores of different CNN models of DeepLung.

7.2.2 Leave-1-person-out test

This experiment was conducted considering the realistic scenario when a patient tries to use a DeepLung system which has been trained by speech and lung sound samples from other people. This experiment was done on the 3-class data using 1-D CNN. For this experiment, we randomly picked 80% samples from each of the 2 datasets (respiration and static windowed), divided these samples to their corresponding 131 subjects, trained the CNN on all-but-1 subject, and tested on the samples from the remaining subject. This was done for each of the 131 subjects, with the experiment being repeated 5-10 times. The average F1 score (.995±.025) for respiration windowed audio was better than static windowed audio (.796±.15) for this test.

	# of training samples		
	100	**500**	**1000**
DeepLung	.924±.035	.967±.012	.983±.002
BodyBeat	.65±.097	.80±.024	.83±.013

Table 7.3: DeepLung 1-D CNN F1 score comparison with BodyBeat.

7.2.3 Comparison with BodyBeat

We compare DeepLung 1-D CNN performance with the BodyBeat [16] classifier, which is a mobile based body sound detection system using specialized microphone hardware. We implemented BodyBeat's algorithm with a linear discriminant classifier with 30 best features chosen using mutual information based feature selection from the 1582 original features. An 80-20% random split was done on the respiration windowed dataset to make the training and validation sets. Both classifiers were trained using same amount of training from the training set and tested on the validation set. Table 7.3 shows that DeepLung is 15-27% better than BodyBeat for various amount of training, although using only commodity smartphone hardware (as opposed to external microphone).

7.3 Summary

This chapter describes the performance of mLung and DeepLung in classifying anomalous lung and confounding sounds like cough, wheeze, abdominal sounds and throat clearing for pulmonary patients by sensing a secondary phenomenon like the respiration cycle from chest motion and windowing audio accordingly. Experiments were performed using novel lung activity data collected by smartphone from 131 real patients and healthy subjects. The shallow in-phone classifier demonstrated to detect and filter speech from lung sounds with 98% F1 score using as low as 10 features. Both mLung and DeepLung had significant better performance (upto 27%) in classifying lung sounds than the baseline BodyBeat system when limited training data were available. Also, mLung and DeepLung demostrated better performance than BodyBeat in multiple realistic leave-1-person-out test and leave-1-sample-out tests. The promising performance of

mLung and DeepLung with real time respiration cycle detection on lung activity data collected by chest-held smartphones from an extensive number of real patients and healthy subjects makes it a possible solution for remote longitudinal monitoring of pulmonary patients by the caregivers.

Chapter 8

SocialSense Design and Implementation

Humans are getting more and more intimately with their smartphones, and therefore these devices got great potential for behavioral and psychological monitoring. In this chapter, we present a how a social interaction monitoring solution in the smartphone, SocialSense, senses a dynamic physical entity like human presence in social interaction groups, and adapts its underlying machine learning pipeline for better recognition of social interaction .

The rest of this chapter is organized as follows. Section 8.1 discusses the motivation of using SocialSense. Section 8.2 lists the technical contributions that this chapter makes. Section 8.3 provides a comparison of SocialSense with the state-of-the-art. Section 8.4 provides detailed technical components and operation of SocialSense. Section 8.5 provides experimental results. Section 8.6 discusses some aspects of the solution and finally Section 8.7 provides a summary of the chapter.

8.1 Motivation

Speaker identification systems based on in-home/on-body/smartphone microphones are used for various applications such as voice based authentication, home health care, security, and daily activity monitoring. With the pervasive usage of smartphones in everyday life, it is an exceptionally suitable unobtrusive platform for speaker identification reducing the overhead of on-body or contextual sensors. Besides speaker identification, speaker mood detection is another important problem in human interaction studies and social psychology research. The challenges of smartphone based speaker identification and mood detection include preserving user privacy, maintaining identification accuracy, accurate operation of the system irrespective of smartphone location, resilience against ambient noise and operating under energy constraints.

Both speaker and mood identification are part of a bigger and important health sensing problem, detection of human social interaction, which is an important indicator of mental and physical health in people of all ages. Regular good social interaction brings many health benefits including reduced risk for cardiovascular and Alzheimer's disease, some cancers, osteoporosis and rheumatoid arthritis, steady blood pressure and reduced risk of depression and other mental disorders. On the other hand, social isolation culminates to loneliness and depression, physical inactivity and overall having a greater risk of death for older people. Therefore, a system able to detect people's social interactions and mood would be greatly beneficial for caregivers to more accurately diagnose and treat patients suffering from psychological disorders.

In this chapter, we present SocialSense, a collaborative smartphone based speaker and mood identification and reporting system which logs user speaking and mood episodes from his/her voice. A person can be uniquely identified by his smartphone Bluetooth ID. After SocialSense detects its user's speaking episode and mood, it broadcasts a message containing the user ID, the speaking episode timestamp, and corresponding mood to all neighboring phones via Bluetooth broadcasting. Thus every phone logs a global scenario of the social interaction environment. In a nutshell, SocialSense can answer the following questions:

- When is the phone user speaking?
- What is the mood of the user during speaking?
- With whom is the user speaking to? Who else are present around?
- When are the other persons in the environment speaking?
- What are the moods of other persons while they speak?

Besides detecting the smartphone user's mood, understanding the moods of other persons present in a social interaction is an important indicator of the global mood, hence the quality of the social interaction. Having this feature, SocialSense can potentially be used to demonstrate and verify the effect of mood contagion, i.e. how multiple individuals in a social interaction reach a mood convergence [122]. Using the idea of mood contagion, SocialSense can be used as a recommender system where it can recommend happy persons as potential conversation partners of sad persons to cheer them up. The actual use of SocialSense for mood contagion is outside the scope of this work.

Our prime target for the usage of SocialSense is in assisted living facilities for the elderly where the prevalence and magnitude of depression is of major concern. More than 1 million Americans reside in assisted livings presently. Studies found that, 20% to 24% of assisted living residents have symptoms of major or minor depression which is likely to cause physical, cognitive, and social impairment and delayed recovery from medical illness and surgery to these elderly. The scary fact is that, many depressive older adults end up committing suicide. Among men of age 75 and over, rate of suicide is 37.4 per 100,000 population. Several diagnostic barriers exist for the screening and treatment of depression in assisted livings which includes lack of regulatory requirements, privacy concerns, cost, and misinterpretation of depression. It is suggested that assisted living staff (nurses, therapists, medical directors) should proactively assess for depressive syndromes instead of self-reporting of mood changes by the residents. SocialSense can be used as an automated diagnostic tool to monitor the mood and social interactions of the assisted living residents where each of the residents is provided with a smartphone with our system. [123]

Since SocialSense can capture the global scenario of a social interaction setting, it can be used as a data collection system for various social psychology and human interaction research. In addition, SocialSense incorporates a dynamic event-driven classification scheme for speaker identification. New people can enter into a social interaction while some people may leave at any time. SocialSense periodically refreshes its Bluetooth neighbor set and whenever it detects a change in the set, i.e., some people entered or left, it recreates the classification model based on the new neighbor set. For this purpose, it imports the user training feature files from the newly arrived phones to re-compute the classification model.

8.2 Contributions

The main contributions of this chapter are:

- Presenting SocialSense, which is an unobtrusive voice based speaker identification and mood detection system using user's smartphone without using any on-body or contextual sensors thus contributing to mobility and user-friendliness.

- A practical, easy, and short training scheme to train a phone to detect a person's own speaking episodes. One key novelty of SocialSense is that it avoids the need of exhaustive training by all users in a social interaction setting and still accurately detects all speakers by collaboration among the phones.

- SocialSense maintains user privacy as no audio samples are recorded or stored in the phone and features are extracted in real time and after classification they are removed from the system. There is no way to reconstruct the original audio signal at a later time from SocialSense.

- SocialSense's voice based mood detection module in every phone is conventional, however by collaboration among the phones, it can detect the mood of the members of a conversation group and the change of one's mood

when he/she switches between conversation groups, i.e., demonstrate mood contagion. This novel idea hasn't been explored before and such a system would be invaluable for further ex-periments on social mood dynamics. Also, using a random forest classifier for mood detection compared to GMM and SVM classifiers used in baseline systems [10, 75], SocialSense has a 4% to 20% increase in accuracy compared to the baselines.

- SocialSense supports real life environments where new people enter and existing people leave the social interaction environment. SocialSense periodically refreshes its Bluetooth neighbor set to detect such changes in the environment.

- Another novel feature of SocialSense is its dynamic event-driven classification scheme where it performs speaker identification using an up-to-moment classifier based on the current users present in the scenario. This yields an average 30% increase in classification accuracy compared to static classification.

- Evaluation with respect to noisy environments has been performed by injecting various artificial noise to simulate real life noisy environments and results demonstrates that SocialSense improves speaker identification accuracy in noise by 10%-43% based on different types of noise and mood detection accuracy in noise by 33% compared to the state-of-the-art systems. SocialSense has been evaluated by training with noise to yield these performance boosts, which hasn't been done in baseline approaches.

8.3 SocialSense Comparison with State-of-the-art

We compare SocialSense with some state-of-the-art smartphone based sensing systems in table 8.1. As seen from this table, SocialSense has better performance than other systems for speaker identification and mood detection, and can operate under ambient noise.

System	Operations	Classifiers used	Results
EmotionSense [10]	Speaker identification, mood detection	Gaussian Mixture Model (GMM)	90% speaker ID accuracy, 70% mood detection accuracy
SpeakerSense [9]	Speaker identification	GMM	95% speaker identification accuracy
Neary [74]	Detect conversational fields i.e detecting multiple persons who are in a conversation	No classifier	96.6% precision and 67.9% recall achieved in a controlled experiment
Qiang et al [124]	User activity, speaker ID, proximity, location	Naive Bayes, Discriminant, Boosted tree, Bagged tree	92% accurate speech detection
SocialSense	Speaker identification, mood detection, mood contagion sensing	Logistic regression, Random forest	94% speaker ID accuracy, 90% speaker ID accuracy in noise, 80% mood detection accuracy, 76% mood detection accuracy in noise

Table 8.1: Comparison of State-of-the-art

8.4 SocialSense System Design and Operation

The assumption behind SocialSense is that every user in a social interaction setting carries his/her own phone with the SocialSense app running in it. However, if one or more persons is present without his phone, only his speaking and mood episodes will remain undetected and unreported, while all other users' speaking and mood episodes will be detected and broadcasted.

Figure 8.1 shows the system diagram. The SocialSense app runs continuously in each phone listening to audio streams. Silent frames are detected by comparing each frame's energy to a threshold, and filtered from further processing to save energy. Each phone periodically updates its phone-set within its Bluetooth proximity range, which is around 10 meters. It is required that the system meets the energy constrains of mobile devices in order to make it usable in realistic scenarios. SocialSense is capable of running for 10 to 14 hours continuously in smartphones and tablets which is good enough for its usage as a healthcare, research and data collection tool in assisted living. SocialSense is made up of a number of modules described in the following sections.

8.4.1 Phone-set Formation

A phone's phone-set is defined as the set of phones running the SocialSense app situated within the Bluetooth proximity range from that phone. This module running in every phone refreshes its phone-set periodically (generally every 30s)

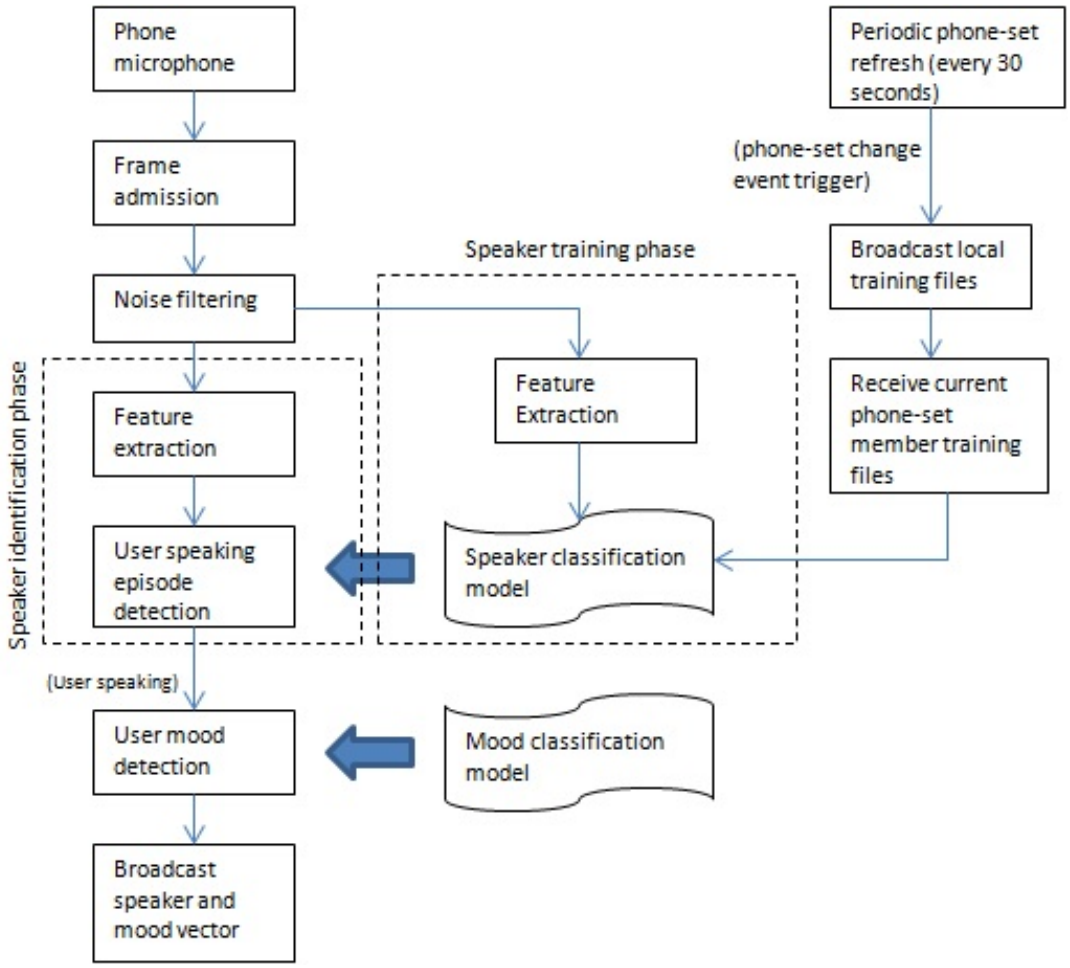


Figure 8.1: SocialSense block diagram

to keep the most recent neighboring phones in its phone-set. The periodic interval is set so that it is neither too short to trigger redundant phone-set discovery process nor too long to miss significant changes in the phone-set, considering realities of human social inter-action. All members of a phone-set are assumed to be close enough to participate in a conversation. Conversely, phones not belonging to the phone set are assumed to be not participating in a conversation.

8.4.2 Speaking Episode Detection Module

This module determines whether a voice segment belongs to the phone user or not. The speaker identification is a binary classification problem where every

non-silent audio segment must be classified into one of two classes: "phone user's voice" or "anything else" (e.g. others' voice or ambient noise). It uses a dynamic logistic regression based classifier, which can be easily trained by the user (or support personnel in assisted living). The user trains the speaker classification system by speaking for 60 seconds in front of the phone in normal tone and loudness. This simple, easy-to-use and short self- training scheme allows the classifier being updated with the latest voice samples of the user. In assisted living facilities, this training will be done by the staff.

Some existing smartphone based speaker identification systems classify speakers based on the loudness of the perceived audio signal [12, 124]. The hypothesis behind those works is that, a user's voice is loudest in his own phone in a particular time instant compared to any other neighboring phones at the same time (as the user is supposed to be the closest person to his phone). However, this scheme doesn't work well in noisy environments, and also in the situation where a person without any phone is talking with people having their phones. In the latter case, when the person not having his phone is talking, his voice will be loudest to the person's phone who is closest to him, so that phone will incorrectly assume that the person without his phone is its user and classify positively, which is incorrect. Other systems like SpeakerSense [9] require training a speaker model for each individual who needs to be recognized, thus incurring large training overhead and resulting in complex, power-hungry classifiers.

SocialSense, on the other hand, uses a simple logistic regression based binary classifier with very little training overhead using 39 MFCC (mel-frequency cepstral coef-ficient) features. The phone-user (or staff) can train the system easily by speaking for 60 seconds in front of the phone in normal tone and volume to create a speaker classification model. As human voice may occasionally change depending on his physiological state, using this easy-to-use training scheme, the system can detect when its user is speaking irrespective of his voice quality, in the presence of noise and even when a person without his phone is present in the scenario as well. Unlike volume based systems, SocialSense does not fail when a user is present without his phone. Only his speaking and mood episodes remain undetected, but the systems in other users' phones work fine. The presence of a

user without his phone does not incur any error or failure in the overall system operation.

8.4.3 Mood Detection Module

Detecting speaker mood in a mobile platform is a major challenge in this work. If a voice segment has been classified as a user's voice by the speaking episode detection module, this module further determines the user's mood (happy, sad, angry, neutral) from his voice. Then it generates a speaking and mood vector consisting of the starting and stopping timestamp of the user's speaking episode and mood during that speaking episode. This module extracts 39 MFCC coefficients from each user utterance window and calculates 9 different statistics on each MFCC coefficient culminating to 351 audio features. These statistics are: geometric mean, harmonic mean, arithmetic mean, range, skewness, standard deviation, z-scored average, moment and kurtosis. The MFCC coefficients combined with these statistics carry a large amount of prosodic and energy based information correlated to emotion. It then uses these features to train a random forest classifier from the EMA emotional utterance dataset [125] for detecting mood.

8.4.4 Message-Exchange Module

SocialSense forms a Bluetooth network among all members of a phone-set. When a phone has a speaking and mood vector to send, it broadcasts the vector using flooding over the network. It has an incoming thread and an outgoing thread to handle incoming and outgoing messages, respectively. It maintains a message queue, new vectors to be broadcasted are enqueued in the queue and the outgoing thread sends vectors one by one from the queue.

8.4.5 Dynamic Event-Driven Classification Module

For speaker identification, the logistic regression classifier uses a positive training file to keep training samples from the phone's user, and uses another negative training file to keep training samples from all other users. During startup of a conversation, SocialSense broadcasts its local positive training file to all neighbors which they use for their negative training. If there are 4 phones in the scenario, each phone uses its local file for positive training and 3 other files received from others for negative training. The phone-set discovery process triggers every 30 seconds to refresh the phone-set. If there is a change in the phone-set during a periodic phone-set refresh (an old user leaves or a new user enters), an event is triggered. When the event triggers, each phones broadcasts its positive training file over the network and updates its negative training using only the files received from phones present in the current scenario, and then rebuilds an updated classifier for speaker identification. This improves classification accuracy by 33% on average compared to static training and makes the training process for each user simple, which is shown in the evaluation section.

8.5 Evaluation

The evaluation consists of multiple parts. First, we evaluate how accurate SocialSense is in identifying speakers. Then we evaluate the effectiveness of mood identification. We have done these evaluations in quiet and noisy (artificially injected) environments, showing that training with noise in noisy environments yields good increase in performance. We have demonstrated the impact of window size, amount of training data, and dynamic classification for speaker identification. We also compare our results with some state-of-the-art solutions.

8.5.1 Speaker Identification Evaluation

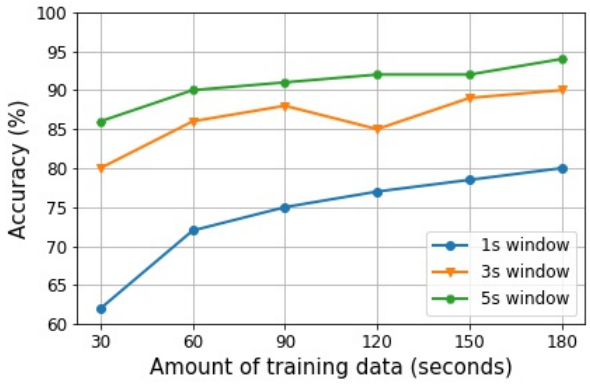
We have evaluated the performance of SocialSense's speaker classification module in terms of the classification accuracy, which is the overall correctness of the

model. The data for these evaluations are taken from voice segments collected from 7 persons. There were 4 females and 3 males among them. A 1.5 hour long conversation on various random topics between two of these females was recorded by us. Another 6 conversations, each around 5 minutes in length, between a male and a female, were collected from the internet. We collected 3 solo speech recordings from the remaining 3 persons for 10 minutes each. We extracted individual voice recordings from each of these 7 speakers separately from these recorded conversations and simulated 2, 3, 4 and 5 person conversations from these. We performed all the speaker identification experiments from these simulated conversations. For example, for simulating 3 person conversations, we trained the logistic regression classifier with one person as positively trained, and the other two persons as negatively trained, with all 3 combinations of three persons, and all 35 possible selections of 3 persons from a set of 7 persons.

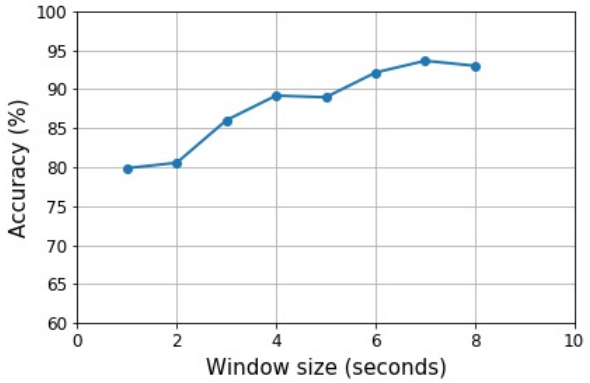
Training Size

Intuitively speaker identification accuracy increases with the increase of training data, as the classifier can encode more information with a longer training. This phenomenon is shown in figure 8.2. The training and testing data were taken from voice samples collected from 7 persons, with 2, 3, 4 and 5 person simulated conversations. The accuracy for 3 separate window sizes is shown for training up to 180 seconds.

As we can see from figure 8.2, there is a sharp increase in accuracy between 30s and 60s of training, and beyond that the accuracy increases slowly. Also the accuracy is highest for a 5s window size. These values are the lower bounds on the training data needed to accurately identify the speaker on the phones, i.e. a minimum of 60 seconds of training is required with a minimum of a 5 second window size.



(a)



(b)

Figure 8.2: Speaker identification accuracy vs. (a) Amount of training data; (b) Window size.

Window Size

This test was conducted on 2 person conversations. One person was trained as positive while the other was trained as negative for 60 seconds. The window size was varied from 1 to 8 seconds and each window was classified using the logistic regression classifier.

Results from figure 8.2 suggest that, a window size between 5 to 7 seconds is optimal for speaker identification. 3-4 second long window sizes yield accuracy of 86-89% which is acceptable. 5-7 second long window sizes can be used for warm conversations where each speaker talks for a long time before switching turns, while a 3-4 second window can be used for cold conversations with frequent turn-takings with short speaking duration in each turn.

Effect of Noise

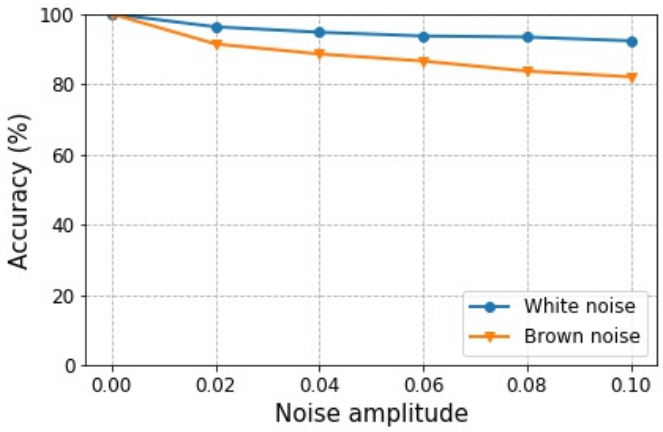
Noise is a very important and realistic issue to consider to evaluate a smartphone based speaker identification system. It is very likely that users will move with their phones to different places (both indoor and outdoor) engaging in social interactions. Therefore, the system must be able to correctly identify its speaker under various types of noise.

Evaluation has been done to test the effect of artificial noise on speaker recognition accuracy. These tests were also done using two person conversations collected from seven speakers. We used Audacity [126], an open-source sound editing software to inject artificial white and brown noise into voice samples, and observed classification accuracy under different levels of noise. White noise is quite similar to television static or the humming of an air conditioner and brown noise is similar to gusty wind. Therefore, these artificial noises can simulate similar indoor and outdoor noisy environments.

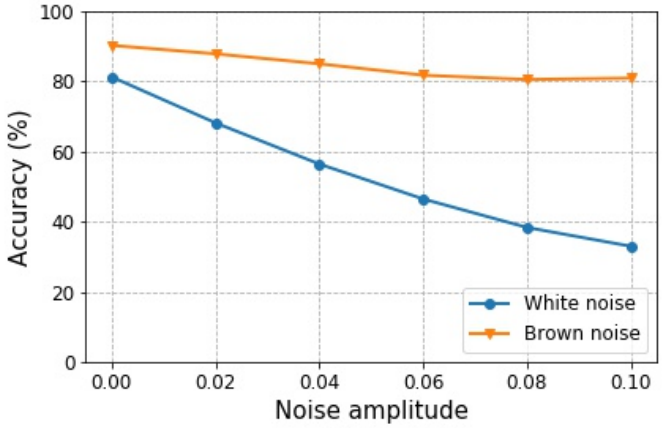
Figure 8.3a shows the effect of white and brown noise on SocialSense speaker recognition accuracy with similar train test set. During no noise, the accuracy is best at 100%, while during maximum noise, the accuracy degrades to 82%, which is an 18% drop. However, because the train and test sets are similar in this case, this is not a realistic scenario. Figure 8.3b shows the effect of noise under different train-test sets. Here, the best accuracy during no noise is 90.2% and the worst accuracy is only 33%, which is a shocking 57% drop, and demonstrates how the system will fail in the presence of noise, if no measure is taken.

It is a design characteristic of SocialSense that a phone user can have his own phone trained for detecting his speaking episodes. This adds a lot of flexibility to the system. In noisy environments, the user can have his phone trained in noise to enhance speaker identification accuracy. Because of the short training session and each user needing to train only himself (as opposed to other systems where all user need to train every phone), the training overhead is low.

Figure 8.4 shows the effect of noise when the training is done in noise as well. It shows that even in worst noisy conditions the accuracy drops to 76% for white



(a) Similar train-test



(b) Different train-test

Figure 8.3: Effect of noise on speaker recognition accuracy, (a) With similar train-test set; (b) With different train-test.

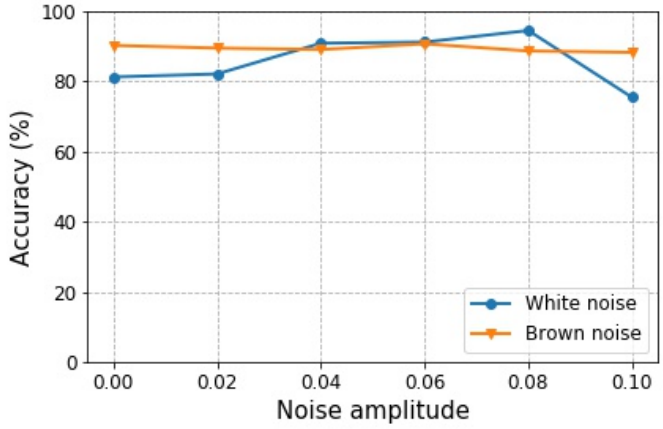


Figure 8.4: Effect of noise on speaker recognition accuracy, with training in noise.

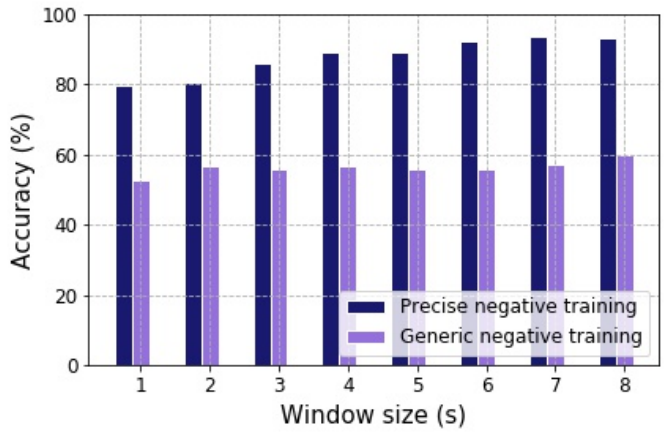


Figure 8.5: Effect of dynamic training vs. generic training.

noise and 89% for brown noise, which is a 43% performance boost for white noise and 10% boost for brown noise. We have limited the noise amplitude to 0.1 in these experiments as this level is commensurate to real life extremely noisy environments.

Effect of Dynamic Classification

Because of the dynamic event-driven classification scheme in SocialSense, every phone is trained with a precise negative training set comprised of the voices of all other persons present in the social interaction setting. The phones update their training files by message exchange whenever a new person enters or leaves the Bluetooth range. Without the dynamic classification, every phone had to use a generic negative training comprising of generic voices from arbitrary persons, since no apriori knowledge of the users is available.

We used voice samples from 5 persons for precise training, with 1 trained as positive and other 4 trained as negative. The testing samples had voices from all 5 persons. For generic training, we used a separate voice collection from 3 people (The EMA dataset [125]) for negative training, and used the same test set as precise training.

The performance comparison of precise and generic training is shown in 8.5, which shows a significant classification improvement (30% on average) due to

dynamic training. Consequently, this novel aspect of our solution results in a major performance improvement.

Worst Case Analysis of Dynamic Speaker Classification

The phone-set refresh process triggers once in every 30 seconds. If there is a change in the phone-set immediately after a refresh process, all the phones will stay with out-dated classifiers for 30 seconds in the worst case. There are 2 cases to consider: i) some new phones arriving, ii) some existing phones leaving. In the first case, if some new phones arrive right after the refresh process, they will remain unknown to the existing phones for 30 seconds until the next refresh process. In this time, the newly arrived phones cannot send or receive any vectors, so their social interactions will not be logged. Also, during this time, the newly arrived phones will have a blank negative set, so all persons' speaking episodes will be considered as positive in these phones. To avoid classification errors due to the initialization in the newly arrived phones, voice segments during this initialization window are ignored. The second case, where some existing phones leave right after the refresh process, is less complicated than the first case. In this case, all the remaining phones will stay with redundant negative sets, containing trainings from people who doesn't exist anymore. But, there will be no classification error in these phones unlike the first case. For both cases, situation comes back to normal in at most 30 seconds, after the immediate next phone-set refresh.

8.5.2 Mood Detection Evaluation

Because of the difficulty associated to get real life data for mood evaluation, we performed both training and testing from the Electromagnetic Articulography dataset [125], which contains 680 acted utterances of a number of sentences in 4 different emotions (anger, happiness, sadness and neutrality) by 3 speakers. We used 3 different classifiers to model each mood using MFCC features with 9

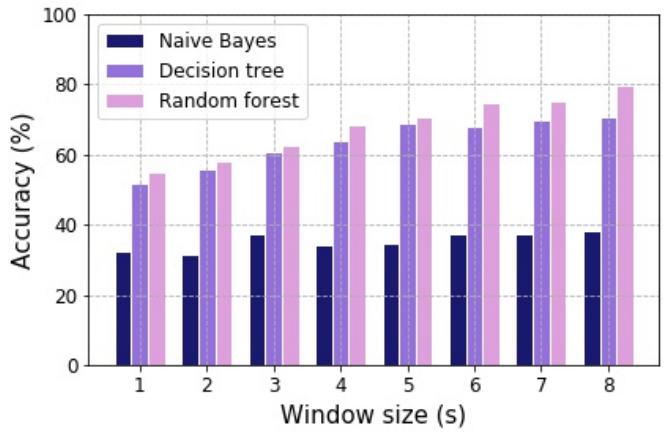


Figure 8.6: Effect of audio sample length on emotion recognition accuracy with various classifiers.

Real mood	Classified as			
	Angry	Happy	Neutral	Sad
Angry	144	19	1	2
Happy	32	110	4	18
Neutral	4	11	125	16
Sad	1	4	23	154

Table 8.2: Confusion matrix for emotion recognition with random forest classifier

statistics (total 351 features), naive Bayes, random forest, and decision tree. We also varied the acoustic window size from 1 to 10 seconds.

The random forest classifier yielded best cross classification results for 10 folds, as shown in figure 8.6. This classifier resulted in a 4% to 10% increase in accuracy compared to the baseline EmotionSense [10] with varying window size, and a 20% increase for speaker independent model compared to [75]. The results for the baselines are taken from corresponding existing works. The figure demonstrates that mood classification accuracy increases with increasing window sizes, however beyond 6s window size it becomes stable and doesn't change much. The confusion matrix for the random forest classifier is shown in Table 8.2. It is interesting to note that, most misclassifications occur between angry-happy and neutral-sad class pairs. The reason is that both angry and happy are high arousal, while both neutral and sad are low arousal emotions.

Similar to the speaker identification module, we evaluated the performance of the mood detection module under noise. The effect of white noise has been noticed

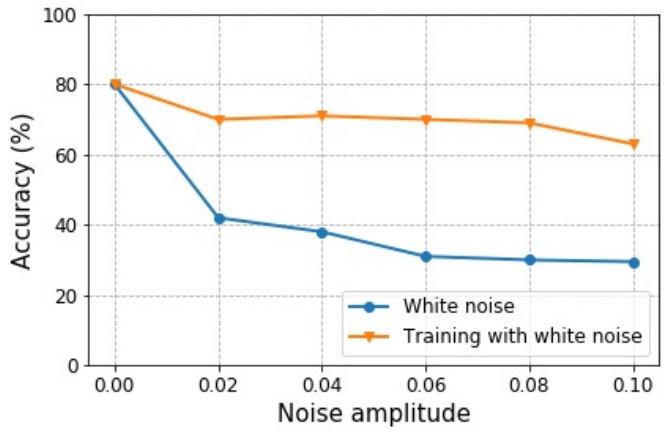


Figure 8.7: Effect of noise and improvement with training with noise for mood classification.

as to be more detrimental than brown noise, so we have done this experiment for white noise only. We injected white noise with amplitudes varying from 0.02 up to 0.1 into the mood dataset. We trained with mood utterances without noise and tested with utterances in noise.

As expected, the performance dropped drastically, as shown in figure 8.7. However, similar to the speaker identification module, we trained the mood dataset in noise and performed a 10 fold cross validation, which yielded a 33% performance boost in the worst 0.1 noise amplitude, as shown in figure 8.7. The existing mood detection systems haven't evaluated the possibility of training with noise, which in our case, yielded a significant increase in performance.

8.5.3 Energy

SocialSense consumes energy in two ways: (i) idle listening, and (ii) once a speech episode is identified, it runs various modules and classifiers. Experiments were run to determine the lifetime of tablets and smartphones running SocialSense. The least energy cost for SocialSense is if it is idle listening and there are no speech episodes to process. Our experiments showed that Nexus 7 tablet ran for 14 hours and the HTC one smartphone ran for 12 hours for this best situation. When SocialSense is actively processing speech episodes there are 5 modules in the system which consume the majority of energy: i) acoustic

processing and feature extraction, ii) logistic regression speaker identification classifier, iii) random forest mood detection classifier, iv) speech and mood vector transmit/receive, and v) periodic phone-set refresh, training file exchange and classification model file recreation. In a second set of experiments we modified the system to run all these modules continuously as if there was continuous speech. This is the worst case in regards to energy costs. In these experiments the Nexus 7 tablet ran for 12 hours (down from 14 hours) and HTC one smartphone ran for 10 hours (down from 12 hours). Consequently, SocialSense can operate between 12-14 hours on a tablet, and between 10-12 hours on a smartphone. This demonstrates that SocialSense can indeed be used as a healthcare device in assisted living since such devices can be charged over night.

8.6 Discussion

8.6.1 Social Interaction History

SocialSense detects speaking episodes and the mood of a user, and by collaboration it imports the speaking episodes and moods of the neighboring users as well. A user interface can be built upon this fine grained information showing the social interaction history of a user within a particular time-frame. Such a user interface will be able to display a user's common conversation partners, his amount of participation and engagement during a conversation with a particular partner, his mood during a conversation and hence mood during that time of that particular day, change of his mood with time or change of conversation partner and so on. Many of these quantities are of interest to psychologists when they treat a potentially depressive patient, and hence ask him relevant questions. The patients' answers are often vague, confusing and erroneous because most of the time they do not remember their social interaction history and mood for a very long time. SocialSense can eliminate the need for these oral questionnaires and hence avoid all the errors as it logs the social interaction data of a user with his moods. Therefore, this system can be used in places like an assisted living

facility where depression and related psychological disorders are common among the occupants.

8.6.2 Robustness

As we argue that SocialSense is usable among the elderly in assisted living facilities, we are aware of the fact that the elderly are prone to forgetfulness, and it is very likely that they may sometimes forget to carry their phones during a social interaction. Though SocialSense is most accurate if every person carries his smartphone in order to detect everybody's speaking episodes and moods, the system does not break down if such assumption is violated. If a person does not carry his phone during a social interaction, his own speaking and mood episodes will remain undetected and unreported and others will not have his information for complete mood contagion. All the other persons' speaking and mood episodes will be detected and reported correctly. This is a major system design enhancement compared to volume based systems [12, 124] which fail when one or more persons forget to carry their phones. It is also important to note that overall diagnosis involves many conversations over multiple days and some missing information when smartphones are forgotten or turned off does not necessarily cause problems.

8.6.3 Training in Assisted Livings

SocialSense's easy to use individual training scheme and adaptability to noisy environments is very suitable for its usage in assisted living. We have shown in the evaluation section that it only takes 60 seconds of training for the system to work in any particular environment. Assisted living residents generally pass specific time of their days in specific locations (e.g., mornings in the hall room, noon at lunch room, afternoon in the garden). The assisted living support person can train the smartphone for each of these common environments. If a resident moves to a new location where the system needs to be retrained because of

different noise levels, the support person can do the training very easily with 60 seconds of data.

8.6.4 Mood Contagion

Using SocialSense it is possible to detect not only the mood of an individual user, but also the moods of others present in the social interaction setting. According to the best of our knowledge, no such system has been built yet which can detect such a global mood. Thus, SocialSense can be used as a platform to verify and conduct experiments on mood contagion which is a psychological process by which a group of people engaged in a social interaction reaches emotional convergence, i.e. they all have similar feelings after a certain time though their initial feelings may be different. It is hypothesized that interventions based on knowledge of mood contagion can be used to help treat depression in the elderly.

8.6.5 "In-Phone" vs. "In-Cloud" Scheme

We adopted an "in-phone" processing scheme as opposed to "in-cloud" processing as in [127]. The term "in-phone" means that all data acquisition, feature extraction, and classification are performed in the phone itself. A reasonable alternative to or solution is an "in-cloud" solution, where unprocessed raw data (conversation recordings) or semi-processed data (features) are sent to a central server where a web service performs further processing and classification. However, the "in-cloud" approach requires connectivity to the internet by wi-fi or 3G which is not always available or is sometimes unreliable. To handle the unreliability of connections various buffering and upload schemes have to be developed. A high-speed 3G/4G connection also imposes additional operating cost for each phone. The "in-phone" approach is cheaper and better supports mobility and could be used even when residents are away from the assisted living facilities.

8.6.6 Concurrent Speaking Episodes

In our experiments described above, we assumed that users did not speak concurrently. In reality, speakers do speak concurrently on some occasions. So we also evaluated our system to test how it performs when users speak concurrently. Ideally, when two or more users are speaking concurrently, each of their systems should detect their own speaking episodes and log them as "speaking" in their individual phones. We performed experiments with 4 speakers (2 male, 2 female), with two concurrent speakers at a time for all 6 possible pairs of conversations. As expected, the system performance degraded. On average, SocialSense was 55% accurate in detecting a particular user's speaking episode when 2 concurrent users were speaking. While this sounds low, this result only applies to the portion of the speaking episode when there is actual concurrency, e.g., when two people first both start speaking (but then one usually backs off) or when someone interrupts a speaker.

8.7 Summary

This chapter presents the design, implementation, and evaluation of SocialSense which is a collaborative mobile platform for speaker identification and mood and mood contagion detection from users' voice. Aside from its ability to recognize speaker and mood with significant accuracy, we have demonstrated its performance relative to the amount of training data and length of window size, culminating in an optimal benchmarking of these parameters. We provide empirical evidence that SocialSense performs well under various noisy environments when trained with noise, with an easy-to-use training scheme. Also, with a dynamic classification scheme, SocialSense is 30% more accurate in speaker identification compared to generic training with static classification. SocialSense is 4%-20% more accurate in speaker independent mood sensing compared to the baseline state-of-the-art mood sensing systems. It was also shown that SocialSense lifetime on various devices is between 10 to 14 hours.

Chapter 9

Conclusion

9.1 Summary and Key Contributions

This thesis presents the idea of real-time sensing of dynamic physical phenomena and thereby adapting the underlying machine learning pipeline to improve performance of certain acoustic activity detection applications. As part of this thesis, we have developed one platform and three systems, each of which has been demonstrated to verify the claim of this thesis.

9.1.1 A Location Aware Platform for Indoor Human Acoustic Event Detection

At the beginning of this thesis, we present the LocoVocal platform. LocoVocal is a novel solution for providing human location aware acoustic models to applications running on top it for improved performance. LocoVocal prototype was implemented with a Kinect depth sensor which localizes human subjects accurately in real-time by skeletal tracking. Two real-time acoustic applications, speech recognition using the CMUSphinx tool and speaker identification were experimented with LocoVocal to demonstrate its versatility and to show that it is 25%, 51.45%, and 10.46% more accurate for spoken word detection, spoken

sentence detection and speaker identification compared to a state-of-the-art static acoustic adaptation system. Home deployment of a LocoVocal prototype in a multi-person household with a speaker identification application demonstrates up to 17.30% improvement in performance in a real-world environment.

9.1.2 Distant Emotion Recognition from Speech

Due to acoustic signal distortion to a distant in-situ microphone, applications like speech emotion detection perform poorly. We propose a solution to this problem which utilizes a novel feature selection algorithm based on iterative distorted feature elimination combined with the acoustic adaptation method of LocoVocal platform. For our experiments, we take a popular emotional corpus and apply distance effect in it by re-recording it with a microphone array in our lab, and convolving with a room impulse response dataset collected in four different rooms at various distances. The overall performance results show as much as 15.51% improvement in distant emotion recognition over baselines, with a final emotion recognition accuracy ranging between 79.44%-95.89% for different rooms, acoustic configurations and source-to-microphone distances.

9.1.3 Lung Anomaly Detection for Pulmonary Patients

Then, we present two novel smartphone based pulmonary anomaly monitoring systems, named mLung and DeepLung, which can be used for longitudinal remote monitoring of pulmonary patients. These systems, when held by the patient's chest, detects respiration cycle using the phone inertial sensors and windows the audio stream according to the respiration cycles to detect pulmonary anomalous sounds, like cough and wheeze. Thus they sense a physical phenomenon like respiration rate in real-time and windows the audio for the machine learning pipeline. For patient privacy, these systems detect and filter speech in-phone with a lightweight classifier, and does the lung sound classification in the cloud. mLung used a shallow support vector machine classifier while DeepLung uses convolutional neural networks for classifying lung sounds. mLung and DeepLung

classifiers have been empirically evaluated with a novel and extensive pulmonary sound dataset collected from 131 real pulmonary patients and healthy subjects using commodity smartphones. We provide empirical evidence that mLung and DeepLung are 15%–25% and 15%-27% more accurate respectively in detecting lung sounds due to natural windowing and better features and classifier when compared to a state-of-the-art smartphone based body sound detection system using specialized microphone hardware, with a best F1 score of 98%.

9.1.4 Social Interaction Monitoring

We finally present SocialSense, which is a smartphone based social interaction monitoring tool. SocialSense can detect when a phone user is speaking using a speaker identification module, with who the user is speaking by discovering neighboring phones (assuming every phone is mapped to a unique user), and what is the mood of the user with a mood detection module. SocialSense incorporates an easy and short training scheme where the user has to train the speaker identification module by providing a short speech. By periodic Bluetooth neighbor sensing, SocialSense knows who is joining or leaving a social interaction, and by collaboration and training exchange with neighboring phones, maintains a non redundant speaker identification classification model by keeping instances only from people who are present in the social interaction in its models. We demonstrate that such dynamic event driven classification yields an average 30% increase in classification accuracy compared to static classification.

9.2 Limitations and Future Improvements

First, in pre-computed mode, the number and positions of LocoVocal bank locations have to be manually determined during system setup at the room of operation, and models for each of the bank locations for the target application have to be computed beforehand. This is quite an overhead from the user's perspective. In a potential solution to automate this process, the system has to 'explore' the space (room of operation) starting from an initial position and

keep finding new points in the unexplored area meeting the constraints that adjacent bank locations should not be too close to be redundant, and that they should not be too close that a human is likely to reach and go beyond the neighboring point before the system can actually switch to the model for that point, as model switching may have some latency. A random data structure like Rapidly-Exploring Random Tree (RRT) [97] can be used for such path planning in the space. Unlike other randomized path planning algorithms, RRT vertices are heavily biased to explore toward the unexplored region of the space.

Second, in mLung and DeepLung, the respiration cycle detection experiments were done only when the subjects were in sit-silent and supine-silent position. It is likely that if the subjects were more mobile (like walking) the algorithm performance will drop. The algorithm performance can be improved under different motions by separating the respiration periodicity from the particular motion periodicity.

Third, mLung and DeepLung classifiers were not experimented under the effect of various indoor and outdoor noises, assuming they will be operational in a silent environment with minimal ambient noise. This is unlikely in a realistic noisy scenario where these systems are likely to be prone to false positives. An external noise detector can be added before these systems and the detected noise can either be suppressed or the classifiers can be adapted by training with the detected noise.

Fourth, apart from cough and wheeze, there are other medically defined pulmonary disease symptoms like rhonchus and crackles, which mLung and DeepLung presently cannot detect. It may happen that as more of these sounds become available to develop machine learning models, our present features and classifiers may not be good to detect them. We may need better deep neural networks to detect these new sounds.

Fifth, SocialSense assumes every phone present in its Bluetooth network is involved in a social interaction. This may not always be true as there may be multiple conversation groups within a room. Conversation groups can be inferred by pairwise detection of synchronicity of conversation between two persons,

considering the principle that if two persons are in a conversation, generally one is speaking and the other is silent as he is listening and vice versa.

Bibliography

[1] Amazon alexa. https://developer.amazon.com/alexa, 2019.

[2] Google home. https://store.google.com/us/product/google_home?hl=en-US, 2018.

[3] Apple siri. https://www.apple.com/siri/, 2019.

[4] Mohsin Y Ahmed, Avinash Kalyanaraman, and Jack Stankovic. Locovocal: Improving indoor human acousticactivity detection by real-time human localization. unpublished, 2019.

[5] Mohsin Y Ahmed, Md Mahbubur Rahman, Viswam Nathan, Ebrahim Nemati, Korosh Vatanparvar, and Jilong Kuang. mlung: Privacy-preserving naturally windowed lung activity detection for pulmonary patients. In *Proceedings of the IEEE-EMBS international conference on Wearable and Implantable Body Sensor Networks*, 2019.

[6] Mohsin Y Ahmed, Md Mahbubur Rahman, and Jilong Kuang. Deeplung: Smartphone convolutional neural network-based inference of lung anomalies for pulmonary patients. In *Interspeech*, 2019.

[7] Mohsin Y Ahmed, Sean Kenkeremath, and John Stankovic. Socialsense: A collaborative mobile platform for speaker and mood identification. In *European Conference on Wireless Sensor Networks*, pages 68–83. Springer, 2015.

[8] Mohsin Y. Ahmed, Zeya Chen, Emma Fass, and John A. Stankovic. Real time distant speech emotion recognition in indoor environments. In *MobiQuitous*, 2017.

[9] Hong Lu, AJ Bernheim Brush, Bodhi Priyantha, Amy K Karlson, and Jie Liu. Speakersense: energy efficient unobtrusive speaker identification on mobile phones. In *International conference on pervasive computing*, pages 188–205. Springer, 2011.

[10] Kiran K Rachuri, Mirco Musolesi, Cecilia Mascolo, Peter J Rentfrow, Chris Longworth, and Andrius Aucinas. Emotionsense: a mobile phones based adaptive platform for experimental social psychology research. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 281–290. ACM, 2010.

[11] Hong Lu, Denise Frauendorfer, Mashfiqui Rabbi, Marianne Schmid Mast, Gokul T Chittaranjan, Andrew T Campbell, Daniel Gatica-Perez,

and Tanzeem Choudhury. Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 351–360. ACM, 2012.

[12] Chengwen Luo and Mun Choon Chan. Socialweaver: Collaborative inference of human conversation networks using smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 20. ACM, 2013.

[13] Justice Amoh and Kofi Odame. Deepcough: A deep convolutional neural network in a wearable cough detection system. In *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*, pages 1–4. IEEE, 2015.

[14] Ebrahim Nemati, Md Mahbubur Rahman, Viswam Nathan, and Jilong Kuang. Private audio-based cough sensing for in-home pulmonary assessment using mobile devices. In *EAI International Conference on Body Area Networks*, 2018.

[15] Shahriar Nirjon, Robert F Dickerson, Qiang Li, Philip Asare, John A Stankovic, Dezhi Hong, Ben Zhang, Xiaofan Jiang, Guobin Shen, and Feng Zhao. Musicalheart: A hearty way of listening to music. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 43–56. ACM, 2012.

[16] Tauhidur Rahman et al. Bodybeat: a mobile system for sensing non-speech body sounds. In *MobiSys*, volume 14, pages 2–13, 2014.

[17] Koji Yatani and Khai N Truong. Bodyscope: a wearable acoustic sensor for activity recognition. In *UbiComp*, pages 341–350. ACM, 2012.

[18] Liu Sicong, Zhou Zimu, Du Junzhao, Shangguan Longfei, Jun Han, and Xin Wang. Ubiear: Bringing location-independent sound awareness to the hard-of-hearing people with smartphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):17, 2017.

[19] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 261–272. ACM, 2009.

[20] Robert F Dickerson, Enamul Hoque, Philip Asare, Shahriar Nirjon, and John A Stankovic. Resonate: reverberation environment simulation for improved classification of speech models. In *Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on*, pages 107–117. IEEE, 2014.

[21] Enamul Hoque, Robert F Dickerson, and John A Stankovic. Vocal-diary: A voice command based ground truth collection system for activity recognition. In *Proceedings of the Wireless Health 2014 on National Institutes of Health*, pages 1–6. ACM, 2014.

[22] Dimitrios Lymberopoulos, Jie Liu, Xue Yang, Romit Roy Choudhury, Vlado Handziski, and Souvik Sen. A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In *Pro-*

ceedings of the 14th international conference on information processing in sensor networks, pages 178–189. ACM, 2015.

[23] Jiang Xiao, Zimu Zhou, Youwen Yi, and Lionel M Ni. A survey on wireless indoor localization from the device perspective. *ACM Computing Surveys (CSUR)*, 49(2):25, 2016.

[24] Jie Xiong and Kyle Jamieson. Arraytrack: A fine-grained indoor location system. In *NSDI*, pages 71–84, 2013.

[25] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single wifi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 165–178. USENIX Association, 2016.

[26] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 269–282. ACM, 2015.

[27] Giorgio Conte, Massimo De Marchi, Alessandro Antonio Nacci, Vincenzo Rana, and Donatella Sciuto. Bluesentinel: a first approach using ibeacon for an energy efficient occupancy detection system. In *BuildSys@ SenSys*, pages 11–19, 2014.

[28] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM, 2000.

[29] Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal De Lara. Accurate gsm indoor localization. In *International conference on ubiquitous computing*, pages 141–158. Springer, 2005.

[30] Yin Chen, Dimitrios Lymberopoulos, Jie Liu, and Bodhi Priyantha. Fm-based indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 169–182. ACM, 2012.

[31] Shuyu Shi, Stephan Sigg, and Yusheng Ji. Joint localization and activity recognition from ambient fm broadcast signals. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 521–530. ACM, 2013.

[32] Timothy W Hnat, Vijay Srinivasan, Jiakang Lu, Tamim I Sookoor, Raymond Dawson, John Stankovic, and Kamin Whitehouse. The hitchhiker's guide to successful residential sensing deployments. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 232–245. ACM, 2011.

[33] Alexandru Bălan and Michael Black. The naked truth: Estimating body shape under clothing. *Computer Vision–ECCV 2008*, pages 15–29, 2008.

[34] Robert T Collins, Ralph Gross, and Jianbo Shi. Silhouette-based human identification from body shape and gait. In *Automatic Face and Gesture*

Recognition, 2002. Proceedings. Fifth IEEE International Conference on, pages 366–371. IEEE, 2002.

[35] Timothy W Hnat, Erin Griffiths, Ray Dawson, and Kamin Whitehouse. Doorjamb: unobtrusive room-level tracking of people in homes using doorway sensors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 309–322. ACM, 2012.

[36] Nacer Khalil, Driss Benhaddou, Omprakash Gnawali, and Jaspal Subhlok. Nonintrusive occupant identification by sensing body shape and movement. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, pages 1–10. ACM, 2016.

[37] Avinash Kalyanaraman, Dezhi Hong, Elahe Soltanaghaei, and Kamin Whitehouse. Forma track: tracking people based on body shape. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):61, 2017.

[38] Erin Griffiths, Avinash Kalyanaraman, Juhi Ranjan, and Kamin Whitehouse. An empirical design space analysis of doorway tracking systems for real world environments. *ACM Transactions on Sensor Networks (TOSN)*, 13(4), 2017.

[39] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[40] Chuong V Nguyen, Shahram Izadi, and David Lovell. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 524–530. IEEE, 2012.

[41] Yuhua Zou, Weihai Chen, Xingming Wu, and Zhong Liu. Indoor localization and 3d scene reconstruction for mobile robots using the microsoft kinect sensor. In *Industrial Informatics (INDIN), 2012 10th IEEE International Conference on*, pages 1182–1187. IEEE, 2012.

[42] RJ Riella, P Nohama, and JM Maia. Method for automatic detection of wheezing in lung sounds. *Brazilian Journal of Medical and Biological Research*, 42(7):674–684, 2009.

[43] Nandini Sengupta, Md Sahidullah, and Goutam Saha. Lung sound classification using cepstral-based statistical features. *Computers in biology and medicine*, 75:118–129, 2016.

[44] Germán D Sosa, Angel Cruz-Roa, and Fabio A González. Automatic detection of wheezes by evaluation of multiple acoustic feature extraction methods and c-weighted svm. In *10th International Symposium on Medical Information Processing and Analysis*, volume 9287, page 928709. International Society for Optics and Photonics, 2015.

[45] Ho-Kyeong Ra, Asif Salekin, Hee Jung Yoon, Jeremy Kim, Shahriar Nirjon, David J Stone, Sujeong Kim, Jong-Myung Lee, Sang Hyuk Son, and John A Stankovic. Asthmaguide: an asthma monitoring and advice ecosystem. In *2016 IEEE Wireless Health (WH)*, pages 1–8. IEEE, 2016.

[46] Plamen Bokov, Bruno Mahut, Patrice Flaud, and Christophe Delclaux. Wheezing recognition algorithm using recordings of respiratory sounds at the mouth in a pediatric population. *Computers in biology and medicine*, 70:40–50, 2016.

[47] Luis Mendes, IM Vogiatzis, Eleni Perantoni, Evangelos Kaimakamis, Ioanna Chouvarda, Nicos Maglaveras, Venetia Tsara, C Teixeira, Paulo Carvalho, Jorge Henriques, et al. Detection of wheezes using their signature in the spectrogram space and musical features. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 5581–5584. IEEE, 2015.

[48] Naoki Nakamura, Masaru Yamashita, and Shoichi Matsunaga. Detection of patients considering observation frequency of continuous and discontinuous adventitious sounds in lung sounds. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pages 3457–3460. IEEE, 2016.

[49] Masataka Himeshima, Masaru Yamashita, Shoichi Matsunaga, and Sueharu Miyahara. Detection of abnormal lung sounds taking into account duration distribution for adventitious sounds. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1821–1825. IEEE, 2012.

[50] Shoichi Matsunaga, Katsuya Yamauchi, Masaru Yamashita, and Sueharu Miyahara. Classification between normal and abnormal respiratory sounds based on maximum likelihood approach. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 517–520. IEEE, 2009.

[51] Sergio Matos, Surinder S Birring, Ian D Pavord, and David H Evans. An automated system for 24-h monitoring of cough frequency: the leicester cough monitor. *IEEE Transactions on Biomedical Engineering*, 54(8):1472–1479, 2007.

[52] SS Birring, T Fleming, S Matos, AA Raj, DH Evans, and ID Pavord. The leicester cough monitor: preliminary validation of an automated cough detection system in chronic cough. *European Respiratory Journal*, 31(5):1013–1018, 2008.

[53] Kevin E Forkheim, David Scuse, and Hans Pasterkamp. A comparison of neural network models for wheeze detection. In *WESCANEX 95. Communications, Power, and Computing. Conference Proceedings., IEEE*, volume 1, pages 214–219. IEEE, 1995.

[54] P Mayorga, C Druzgalski, RL Morelos, OH Gonzalez, and J Vidales. Acoustics based assessment of respiratory diseases using gmm classification. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 6312–6316. IEEE, 2010.

[55] Jianmin Zhang, Wee Ser, Jufeng Yu, and TT Zhang. A novel wheeze detection method for wearable monitoring systems. In *Intelligent Ubiquitous Computing and Education, 2009 International Symposium on*, pages 331–334. IEEE, 2009.

[56] Feng Jin, Farook Sattar, and Daniel YT Goh. Automatic wheeze detection using histograms of sample entropy. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 1890–1893. IEEE, 2008.

[57] Shih-Hong Li, Bor-Shing Lin, Chen-Han Tsai, Cheng-Ta Yang, and Bor-Shyh Lin. Design of wearable breathing sound monitoring system for real-time wheeze detection. *Sensors*, 17(1):171, 2017.

[58] Mohammed Bahoura. Pattern recognition methods applied to respiratory sounds classification into normal and wheeze classes. *Computers in biology and medicine*, 39(9):824–843, 2009.

[59] Amjad Hashemi, Hossein Arabalibiek, and Khosrow Agin. Classification of wheeze sounds using wavelets and neural networks. In *International Conference on Biomedical Engineering and Technology*, volume 11, pages 127–131. IACSIT Press, 2011.

[60] R Jané, S Cortes, JA Fiz, and J Morera. Analysis of wheezes in asthmatic patients during spontaneous respiration. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, volume 2, pages 3836–3839. IEEE, 2004.

[61] K McGuinness, A Kelsall, J Lowe, A Woodcock, and JA Smith. Automated cough detection: a novel approach. *Am J Respir Crit Care Med*, 175:A381, 2007.

[62] Samantha J Barry, Adrie D Dane, Alyn H Morice, and Anthony D Walmsley. The automatic recognition and counting of cough. *Cough*, 2(1):8, 2006.

[63] Vinayak Swarnkar, Udantha R Abeyratne, Yusuf Amrulloh, Craig Hukins, Rina Triasih, and Amalia Setyati. Neural network based algorithm for automatic identification of cough sounds. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1764–1767. IEEE, 2013.

[64] Xiao Sun, Zongqing Lu, Wenjie Hu, and Guohong Cao. Symdetector: detecting sound-related respiratory symptoms using smartphones. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 97–108. ACM, 2015.

[65] Amzon echo. `https://www.resapphealth.com.au/`, 2019.

[66] PP Moschovis, EM Sampayo, P Porter, U Abeyratne, G Doros, V Swarnkar, R Sharan, and JC Carl. A cough analysis smartphone application for diagnosis of acute respiratory illnesses in children. In *A27. PEDIATRIC LUNG INFECTION AND CRITICAL CARE AROUND THE WORLD*, pages A1181–A1181. American Thoracic Society, 2019.

[67] Keegan Kosasih and Udantha Abeyratne. Exhaustive mathematical analysis of simple clinical measurements for childhood pneumonia diagnosis. *World Journal of Pediatrics*, 13(5):446–456, 2017.

[68] Keegan Kosasih, Udantha R Abeyratne, Vinayak Swarnkar, and Rina Triasih. Wavelet augmented cough analysis for rapid childhood pneumonia diagnosis. *IEEE Transactions on Biomedical Engineering*, 62(4):1185–1194, 2014.

[69] Udantha R Abeyratne, Vinayak Swarnkar, Amalia Setyati, and Rina Triasih. Cough sound analysis can rapidly diagnose childhood pneumonia. *Annals of biomedical engineering*, 41(11):2448–2462, 2013.

[70] Yusuf A Amrulloh, Udantha R Abeyratne, Vinayak Swarnkar, Rina Triasih, and Amalia Setyati. Automatic cough segmentation from non-contact sound recordings in pediatric wards. *Biomedical Signal Processing and Control*, 21:126–136, 2015.

[71] Vinayak Swarnkar, Udantha R Abeyratne, Anne B Chang, Yusuf A Amrulloh, Amalia Setyati, and Rina Triasih. Automatic identification of wet and dry cough in pediatric patients with respiratory diseases. *Annals of biomedical engineering*, 41(5):1016–1028, 2013.

[72] Roneel V Sharan, Udantha R Abeyratne, Vinayak R Swarnkar, and Paul Porter. Automatic croup diagnosis using cough sound recognition. *IEEE Transactions on Biomedical Engineering*, 66(2):485–495, 2018.

[73] Zeya Chen, Mohsin Y. Ahmed, Asif Salekin, and John A. Stankovic. Arasid: Artificial reverberation-adjusted indoor speaker identification dealing with variable distances. In *Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks*, EWSN '19, pages 154–165, USA, 2019. Junction Publishing.

[74] Toshiya Nakakura, Yasuyuki Sumi, and Toyoaki Nishida. Neary: conversation field detection based on similarity of auditory situation. In *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, page 14. ACM, 2009.

[75] Chen Yu, Paul M Aoki, and Allison Woodruff. Detecting user engagement in everyday conversations. *arXiv preprint cs/0410027*, 2004.

[76] Pierluigi Casale, Oriol Pujol, and Petia Radeva. Face-to-face social activity detection using data collected with a wearable device. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 56–63. Springer, 2009.

[77] Chenren Xu, Sugang Li, Gang Liu, Yanyong Zhang, Emiliano Miluzzo, Yih-Farn Chen, Jun Li, and Bernhard Firner. Crowd++: unsupervised speaker count with smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 43–52. ACM, 2013.

[78] Emiliano Miluzzo, Nicholas D Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B Eisenman, Xiao Zheng, and Andrew T Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350. ACM, 2008.

[79] Tanzeem Khalid Choudhury. *Sensing and modeling human networks*. PhD thesis, Massachusetts Institute of Technology, 2004.

[80] Datong Chen, Jie Yang, Robert Malkin, and Howard D Wactlar. Detecting social interactions of the elderly in a nursing home environment. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 3(1):6, 2007.

[81] Amzon echo. `https://www.amazon.com/all-new-amazon-echo-speaker-with-wifi-alexa-dark-charcoal/dp/B06XCM9LJ4`, 2018.

[82] Nicholas D Lane, Petko Georgiev, and Lorena Qendro. Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 283–294. ACM, 2015.

[83] Md Abu Sayeed Mondol, Ifat Afrin Emi, and John A Stankovic. Medrem: an interactive medication reminder and tracking system on wrist devices.

[84] Google home. `https://motherboard.vice.com/en_us/article/53dz8x/people-are-complaining-that-amazon-echo-is-responding-to-ads-on-tv`, 2018.

[85] Microsoft kinect. `https://developer.microsoft.com/en-us/windows/kinect`, 2018.

[86] Jont B Allen and David A Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

[87] Eric A Lehmann and Anders M Johansson. Prediction of energy decay in room impulse responses simulated with an image-source model. *The Journal of the Acoustical Society of America*, 124(1):269–277, 2008.

[88] Heiner Löllmann, Emre Yilmaz, Marco Jeub, and Peter Vary. An improved algorithm for blind reverberation time estimation. In *Proceedings of International Workshop on Acoustic Echo and Noise Control (IWAENC)*, pages 1–4, 2010.

[89] Real time analyzer. `https://www.noisemeters.com/product/real-time-analyzer/options.asp`, 2018.

[90] Intel realsense. `https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html`, 2018.

[91] Zed 3d camera. `https://www.stereolabs.com/`, 2018.

[92] Orbbec. `https://orbbec3d.com/`, 2018.

[93] Vicovr. `https://vicovr.com/`, 2018.

[94] John Shackleton, Brian VanVoorst, and Joel Hesch. Tracking people with a 360-degree lidar. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 420–426. IEEE, 2010.

[95] Paul Lamere, Philip Kwok, Evandro Gouvea, Bhiksha Raj, Rita Singh, William Walker, Manfred Warmuth, and Peter Wolf. The cmu sphinx-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, volume 1, pages 2–5, 2003.

[96] Javier Ramırez, José C Segura, Carmen Benıtez, Angel De La Torre, and Antonio Rubio. Efficient voice activity detection algorithms using long-term speech information. *Speech communication*, 42(3-4):271–287, 2004.

[97] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[98] Kenichi Kumatani, John McDonough, and Bhiksha Raj. Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors. *IEEE Signal Processing Magazine*, 29(6):127–140, 2012.

[99] Steve Renals and Pawel Swietojanski. Neural networks for distant speech recognition. In *Hands-free Speech Communication and Microphone Arrays (HSCMA), 2014 4th Joint Workshop on*, pages 172–176. IEEE, 2014.

[100] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass. Highway long short-term memory rnns for distant speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5755–5759. IEEE, 2016.

[101] Moataz El Ayadi, Mohamed S Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, 2011.

[102] Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter F Sendlmeier, and Benjamin Weiss. A database of german emotional speech. In *Interspeech*, volume 5, pages 1517–1520, 2005.

[103] Marco Jeub, Magnus Schafer, and Peter Vary. A binaural room impulse response database for the evaluation of dereverberation algorithms. In *Digital Signal Processing, 2009 16th International Conference on*, pages 1–5. IEEE, 2009.

[104] Florian Eyben, Martin Wöllmer, and Björn Schuller. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462. ACM, 2010.

[105] Yi-Wei Chen and Chih-Jen Lin. Combining svms with various feature selection strategies. *Feature extraction*, pages 315–324, 2006.

[106] Takuya Yoshioka, Xie Chen, and Mark JF Gales. Impact of single-microphone dereverberation on dnn-based meeting transcription systems.

In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 5527–5531. IEEE, 2014.

[107] Andreas Schwarz and Walter Kellermann. Coherent-to-diffuse power ratio estimation for dereverberation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(6):1006–1018, 2015.

[108] M. R. Schroeder. New method of measuring reverberation time. *The Journal of the Acoustical Society of America*, 37(6):1187–1188, 1965.

[109] Webmd. `https://www.webmd.com/lung/copd/news/20170929/respiratory-disease-death-rates-have-soared`.

[110] Cdc copd. `https://www.cdc.gov/nchs/fastats/copd.htm`.

[111] Cdc asthma. `https://www.cdc.gov/nchs/fastats/asthma.htm`.

[112] Anthony J Guarascio, Shauntá M Ray, Christopher K Finch, and Timothy H Self. The clinical and economic burden of chronic obstructive pulmonary disease in the usa. *ClinicoEconomics and outcomes research: CEOR*, 5:235, 2013.

[113] Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian Müller, and Shrikanth S Narayanan. The interspeech 2010 paralinguistic challenge. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[114] Md Mahbubur Rahman, E Nemati, Viswam Nathan, and Jilong Kuang. Instantrr: Instantaneous respiratory rate estimation on context-aware mobile devices. In *Proceedings of the 13th EAI International Conference on Body Area Networks*, 2018.

[115] Ronald W Schafer. What is a savitzky-golay filter?[lecture notes]. *IEEE Signal processing magazine*, 28(4):111–117, 2011.

[116] Javier Hernandez, Daniel McDuff, and Rosalind W Picard. Biowatch: estimation of heart and breathing rates from wrist motions. In *PervasiveHealth*, pages 169–176. IEEE, 2015.

[117] Rummana Bari, Roy J Adams, Md Mahbubur Rahman, Megan Battles Parsons, Eugene H Buder, and Santosh Kumar. rconverse: Moment by moment conversation detection using a mobile respiration sensor. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 2(1):2, 2018.

[118] Shahriar Nirjon, Robert F Dickerson, Philip Asare, Qiang Li, Dezhi Hong, John A Stankovic, Pan Hu, Guobin Shen, and Xiaofan Jiang. Auditeur: A mobile-cloud service platform for acoustic event detection on smartphones. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 403–416. ACM, 2013.

[119] Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. Recent developments in opensmile, the munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 835–838. ACM, 2013.

[120] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[121] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[122] Roland Neumann and Fritz Strack. " mood contagion": the automatic transfer of mood between persons. *Journal of personality and social psychology*, 79(2):211, 2000.

[123] Richard G Stefanacci. How big an issue is depression in assisted living. *Assisted Living Consult*, 4(4):30–35, 2008.

[124] Qiang Li, Shanshan Chen, and John A Stankovic. Multi-modal in-person interaction monitoring using smartphone and on-body sensors. In *2013 IEEE International Conference on Body Sensor Networks*, pages 1–6. IEEE, 2013.

[125] Jangwon Kim, Sungbok Lee, and Shrikanth S Narayanan. An exploratory study of manifolds of emotional speech. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5142–5145. IEEE, 2010.

[126] Audacity. `https://www.audacityteam.org/`.

[127] Emiliano Miluzzo, Cory T Cornelius, Ashwin Ramaswamy, Tanzeem Choudhury, Zhigang Liu, and Andrew T Campbell. Darwin phones: the evolution of sensing and inference on mobile phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 5–20. ACM, 2010.