This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2021.3064237, IEEE Journal of Biomedical and Health Informatics

CNN-MoE based framework for classification of respiratory anomalies and lung disease detection

Lam Pham, Huy Phan, Ramaswamy Palaniappan, Alfred Mertins, Ian McLoughlin

Abstract— **This paper presents and explores a robust deep learning framework for auscultation analysis. This aims to classify anomalies in respiratory cycles and detect diseases, from respiratory sound recordings. The framework begins with front-end feature extraction that transforms input sound into a spectrogram representation. Then, a back-end deep learning network is used to classify the spectrogram features into categories of respiratory anomaly cycles or diseases. Experiments, conducted over the ICBHI benchmark dataset of respiratory sounds, confirm three main contributions towards respiratory-sound analysis. Firstly, we carry out an extensive exploration of the effect of spectrogram types, spectral-time resolution, overlapping/non-overlapping windows, and data augmentation on final prediction accuracy. This leads us to propose a novel deep learning system, built on the proposed framework, which outperforms current state-of-the-art methods. Finally, we apply a Teacher-Student scheme to achieve a trade-off between model performance and model complexity which holds promise for building real-time applications.**

Index Terms—**Respiratory disease, lung auscultation, wheezes, crackles, anomaly detection, data augmentation, mixture of experts.**

I. Introduction

According to the World Health Organization (WHO) [1], respiratory illness, which comprises lung cancer, tuberculosis, asthma, chronic obstructive pulmonary disease (COPD), and lower respiratory tract infection (LRTI), accounts for a significant percentage of mortality worldwide. Indeed, records indicate that around 10 million people currently have tuberculosis (TB), 65 million have COPD, and 334 million have asthma. Notably, the WHO estimates that about 1.4, 1.6 and 3 million people die respectively from TB, lung cancer and COPD annually. To deal with respiratory diseases, early detection is the key factor in enhancing the effectiveness of intervention, including treatment and limiting spread. During a respiratory examination, lung auscultation (listening to the sounds of breathing through a stethoscope) is an important aspect of respiratory disease diagnosis. By listening to respiratory sounds during lung auscultation, experts can recognise adventitious sounds (including *Crackles* and *Wheezes*) during the respiratory cycle. These anomalous respiratory sounds often occur in those who have pulmonary disorders. If automated methods can be developed to detect such anomalous sounds, it will improve the early detection of respiratory disease and enable screening of a wider population than manual screening. Research into the automated detection or analysis of respiratory sounds has some precedents [2], [3], [4], but has drawn increasing attention in recent years as robust machine hearing methods have been developed, leveraging on ever more capable deep learning techniques.

Most existing respiratory sound analysis systems tend to rely upon frame-based feature representations such as Mel-Frequency Cepstral Coefficients (MFCC) [5], [6], borrowed from the Automatic Speech Recognition (ASR) and Speaker Recognition (SR) fields. However, Grønnesby *et al.* [7] found that MFCCs did not represent crackles well. They thus replaced them with five-dimensional feature vectors, comprising four time domain features (variance, range, and sum of simple moving average (coarse and fine)), and one frequency domain feature (spectrum mean). Meanwhile, Hanna *et al.* [8] firstly extracted spectral information from bark-bands energy, Mel-bands energy, MFCCs, rhythm features from beat loudness, harmonicity and inharmonicity features, as well as tonal features such as chords strength and tuning frequency. Next, they computed statistical features including standard deviation, variance, minimum, maximum, median, mean, first derivative, second derivative from those features in addition to mean and variance of the raw signal. This extensive list aimed to maximize the chance of achieving a discriminative feature set. To further explore audio features, Mendes *et al.* [9] went further to propose 35 different types of feature, mainly coming from Music Information Retrieval research. Inspired by the finding that only some features contributed to the final result, Datta *et al.* [10] firstly assessed features such as power spectral density (PSD), STFT and Wavelet spectrograms, MFCCs, and Linear Frequency Cepstral Coefficients (LFCCs). Next, they applied a Maximal Information Coefficient (MIC) [11] to score each feature, selected only the most influencing, before feeding into a classifier to improve performance and reduce complexity. Similarly, Kok *et al.* [6] applied the Wilcoxon Sum of Rank test to indicate which features among MFCCs, Discrete Wavelet Transform (DWT) and a set of time domain features (namely power, mean, variance, skewness and kurtosis of audio signal) mainly affected final classification accuracy. Image processing techniques were then tried by Sengupta *et al.* [12], who employed Local Binary Pattern (LBP) analysis on mel-frequency spectral coefficients (MFSCs) to capture texture information from the MFSC spectrogram,

L. Pham is with Center for Digital Safety & Security, Austrian Institute of Technology, Austria.

L. Pham and R. Palaniappan are with the School of Computing, University of Kent, UK.

H. Phan is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, UK.

A. Mertins is with the Institute for Signal Processing, University of Lübeck, Germany.

I. McLoughlin is with Singapore Institute of Technology, Singapore.

thus obtained an LBP spectrogram. The LBP spectrogram was converted into a histogram presentation before feeding it into a back-end classifier which was shown to outperform the previous MFCC-based methods. In these systems, the stream of audio feature vectors is classified by a range of traditional machine learning techniques. These include Logistic Regression [9], k-Nearest Neighbour (KNN) [7], [12], Hidden Markov Models [5], [13], [14], Support Vector Machines [7], [10], [12], [15] and decision trees [6], [7], [8], [16].

Deep learning techniques have achieved strong and robust detection performance for general sound classification [17], [18]. Feature extraction in state-of-the-art deep learning based systems typically involves generating two-dimensional time-frequency spectrograms that are able to capture both fine grained temporal and spectral information as well as present a much wider time context than single frame analysis. While a variety of spectrogram transformations have been utilised, Mel-based methods such as log-Mel spectra [19], [20], [21] and stacked MFCC features [19], [22], [23], [24], [25], [26] are the most popular ones. Some researchers combined different types of spectrogram, e.g. short-time Fourier transform (STFT) and Wavelet as proposed by Minami *et al.* [27] or optimized S-Transformations in [28]. Although extracting good quality representative spectrograms is very important for the back-end classifier, researchers to date have not explored the settings used in this step deeply – something we aim to contribute in this paper.

Current deep learning classifiers acting on spectrograms for respiratory sound analysis are mainly based on Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), or hybrid architectures. The CNN-based systems span some diverse architectures such as LeNet6 [23], [22], VGG5 [20], two parallel VGG16s [27], and ResNet50 [28]. Inspired by the fact that respiratory indicative sounds such as *Crackle* and *Wheeze* present certain sequential characteristics, RNN-based networks have been developed in order to capture the sequential information. For example, Perna and Tagarelli [24] analysed the use of a Long Short-term Memory (LSTM) network for two tasks of classifying anomalous respiratory sounds and classifying respiratory diseases. By using LSTM and Gated Recurrent Unit (GRU) cells in a RNN-based network, Kochetov *et al.* [26] proposed a novel architecture, namely the Noise Masking Recurrent Neural Network, which aimed to distinguish both noise and anomalous respiratory sounds. Regarding hybrid architectures proposed in [21], [27], a CNN was firstly used to map a spectrogram input to a temporal sequence. Then, LSTM [21] or GRU [27] layers were used to learn sequence structures before classification takes place via fully-connected layers.

Performance comparison of state-of-the-art respiratory sound detection methods presented in [24], [27], [28] indicates that deep learning classifiers are robust and effective. However, some of the deep learning based models have extremely complicated architectures, hindering their implementation within mobile or wearable devices for real-time applications. Clearly, state-of-the-art systems involve ever-increasing model complexity.

A more serious issue with this research field has been the difficulty of comparing between techniques due to the lack of standardised datasets for evaluation. Most publications evaluate on proprietary datasets that are unavailable to others [9], [10], [13], [19], [25].

In this paper, we tackle the main issues of respiratory sound analysis in the following way;

- We ensure repeatability and ease of comparison by adopting the 2017 Internal Conference on Biomedical Health Informatics (ICBHI) [29] dataset for all experiments. The ICBHI dataset is one of the largest public datasets which includes audio recordings. Using this resource, we will comprehensively analyse factors such as different types of spectrogram, overlapping/non-overlapping windowing, spectrogram patch sizes, and data augmentation to pinpoint their effects on performance.
- From this analysis, we then propose a deep learning framework to target two related tasks of anomaly sound classification and respiratory disease detection. We evaluate two methods of train/test splitting used in the literature (namely random 5-fold cross validation and 60/40 splitting as per the ICBHI challenge's recommendation), and compare against state-of-the-art systems.
- To aid in the trade-off between performance and complexity, we propose a Student-Teacher scheme. Specifically, the best deep learning framework, which is used for the task of respiratory disease detection and requires a large number of trainable parameters, is referred to as the Teacher. We extract classification outputs from the Teacher model and distill these information to train another network with fewer trainable parameters, referred to as the Student. Eventually, we successfully obtain a reduced-size Student network which achieves similar performance as the Teacher.

II. ICBHI DATASET AND OUR TASKS PROPOSED

A. ICBHI dataset

The 2017 ICBHI dataset [29] provides a large database of labelled respiratory sounds comprising 920 audio recordings with a combined duration of 5.5 hours. The recording lengths are uneven, ranging from from 10 to 90 seconds, and were recorded with a wide range of sampling frequencies from 4 kHz to 44.1 kHz. In total, the dataset contains recordings from 128 patients, who are identified as being healthy or exhibiting one of the following respiratory diseases or conditions: COPD, Bronchiectasis, Asthma, upper and lower respiratory tract infection, Pneumonia, Bronchiolitis. These respiratory condition labels are linked to audio recording files. Within each audio recording, four different types of respiratory cycle are presented – called *Crackle*, *Wheeze*, *Both* (*Crackle & Wheeze*), and *Normal*. These cycles, labelled by experts, include identified onset and offset times. The cycles have various recording lengths ranging from 0.2 up to 16.2

seconds, with the number of cycles being unbalanced (i.e. 1864, 886, 506 and 3642 cycles respectively for *Crackle*, *Wheeze*, *Both*, and *Normal*).

B. Main tasks proposed from ICBHI dataset

Given the ICBHI recordings and metadata, this paper targets over two main tasks.

Task 1, respiratory anomaly classification, is separated into two sub-tasks. The first aims to classify four different cycles (*Crackle*, *Wheeze*, *Both*, and *Normal*). The second is to classify the four types of cycle into two groups of *Normal* and *Anomaly* sounds (the latter group consisting of *Crackle*, *Wheeze*, and *Both*). For convenience, we will identify these as Task 1-1 and Task 1-2, respectively.

Task 2, respiratory disease prediction, also comprises two sub-tasks. The first aims to classify audio recordings into three groups of disease conditions: *Healthy*, *Chronic Disease* (i.e. COPD, Bronchiectasis and Asthma) and *Non-Chronic Disease* (i.e. upper and lower respiratory tract infection, Pneumonia, and Bronchiolitis). The second sub-task is for classification into two groups of *Healthy* and *Unhealthy* (comprising the *Chronic* and *Non-Chronic* disease groups combined). We name theses sub-tasks Tasks 2-1 and Task 2-2, respectively. While Tasks 1-1 and 1-2 are evaluated over individual respiratory cycles, Task 2-1 and 2-2 are evaluated over entire audio recordings.

Existing state-of-the-art systems that use the ICBHI dataset follow two different approaches to split the database into training and testing portions. The first [14], [15], [16], [27] follows the ICBHI challenge recommendations [29] to divide the dataset into non-overlapping 60% and 40% portions for training and test subsets, respectively. Notably, this avoids a situation in which audio recordings from one subject are found in both of the subsets. Meanwhile, the second [6], [8], [20], [23], [24] randomly separates the entire dataset into training and test subsets, with different ratios.

To evaluate our proposed framework on each task in this paper, we first separate the ICBHI dataset (6898 respiratory cycles for Task 1 and 920 entire recordings for Task 2) into five folds for cross validation. We then introduce a baseline system upon which we will evaluate the effect of a number of settings and influencing factors. Due to extensive training times, this initial exploration evaluates over one fold. Then, following the initial exploration, we propose two systems; one for the task of anomaly cycle detection (Tasks 1-1 and 1-2) and a second system for respiratory disease detection (Tasks 2-1 and 2-2). We evaluate each of those systems with both the full 5-fold cross validation and 60/40 splitting as specified in the ICBHI challenge's recommendation, and compare against state-of-the-art methods.

C. Evaluation metrics

TABLE I

Confusion matrix of anomaly cycle classification.

	Crackle	Wheeze	Both	Normal
Crackle	C_c	W_c	B_c	N_c
Wheeze	C_w	W_w	B_w	N_w
Both	C_b	W_b	B_b	N_b
Normal	C_n	W_n	B_n	N_n
Total	C_t	W_t	B_t	N_t

TABLE II

Confusion matrix of respiratory disease detection.

	Chronic	Non-chronic	Healthy
Chronic	C_c	NC_c	H_c
Non-chronic	C_{nc}	NC_{nc}	H_{nc}
Healthy	C_h	NC_h	H_h
Total	C_t	NC_t	H_t

As state-of-the-art systems which explore the metrics of *Sensitivity* (Sen.), *Specificity* (Spec.), and *ICBHI score* [24], [29], our proposed baseline and framework variants are also assessed using these metrics. To understand these scores, consider a confusion matrix for Task 1 as presented in Table I. In this case, *C, W, B,* and *N* denote the numbers of cycles of *Crackle*, *Wheeze*, *Both*, and *Normal*, respectively, whereas *c, w, b,* and *n* subscripts indicate the classification results. The sums C_t, W_t, B_t and N_t are the total numbers of cycles. *Sensitivity* is then computed for Task 1-1 (4-class anomaly classification) as follows:

$$Sensitivity = \frac{C_c + W_w + B_b}{C_t + W_t + B_t}, \tag{1}$$

and for Task 1-2 (binary anomaly classification) as:

$$Sensitivity = \frac{C_{c+w+b} + W_{c+w+b} + B_{c+w+b}}{C_t + W_t + B_t}, \tag{2}$$

where $C_{c+w+b} = C_c + C_w + C_b$, $W_{c+w+b} = W_c + W_w + W_b$, and $B_{c+w+b} = B_c + B_w + B_b$. Then we can define

$$Specificity = \frac{N_n}{N_t}. \tag{3}$$

Similarly, consider Task 2's confusion matrix as shown in Table II. In this case, *C, NC* and *H* are the numbers of recordings of the three classes in Task 2. *c, nc* and *h* subscripts indicate the classification results. As before, C_t, NC_t, and H_t are the total numbers of *Chronic*, *Non-chronic*, and *Healthy* recordings, respectively. For Task 2-1, *Sensitivity* is defined as follows:

$$Sensitivity = \frac{C_c + NC_{nc}}{C_t + NC_t}, \tag{4}$$

and for Task 2-2 it reads:

$$Sensitivity = \frac{(C_c + C_{nc}) + (NC_c + NC_{nc})}{C_t + NC_t}. \tag{5}$$

We simply then define

$$Specificity = \frac{H_h}{H_t}. \tag{6}$$

Regarding the *ICBHI score*, this represents an equal trade-off between the two metrics and is computed in the same way for each task – namely averaging the *Sensitivity* and the *Specificity* scores. Furthermore, we also use the other standard metrics of F1 score (macro) [30] and Kappa score [31] for evaluation.

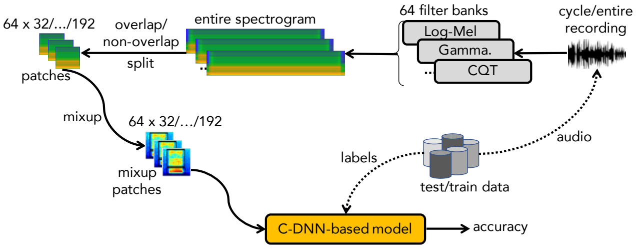


Fig. 1. The high-level architecture and processing pipeline of the proposed framework.

III. High-level framework architecture

A. High-level description

The high-level architecture of the proposed system is described in Fig. 1. The architecture is divided into two main parts: front-end feature extraction (the upper part) and back-end deep learning models (the lower part). In general, respiratory cycles in Task 1 or entire audio recordings in Task 2 are transformed into one or more spectrogram representations. The spectrograms are then split into equal-sized image patches. During training, mixup data augmentation [32], [33] is applied to the patches to generate an expanded set of training data that is fed into a deep learning classifier.

B. Baseline system

From the high-level architecture shown in Fig. 1, it can be seen that a variety of factors in front-end feature extraction could affect the performance of the classifier. These include the type of spectrograms used, the size of image patches and their degree of overlap, and the use of data augmentation. We are thus prompted, in this paper, to investigate the most influencing factors among those listed above. To limit the investigation scope to manageable proportions, we constrain the deep learning architecture assessed and thus propose a C-DNN baseline like VGG-7 [34], defined below.

The main characteristics and settings of this baseline architecture are listed in Table III, while the network architecture is presented in Table IV. During processing, we first re-sample all audio recordings (which, as aforementioned, were recorded with various sample rates) to 16 kHz mono. Since respiratory cycle lengths differ quite widely, we repeat short cycles to ensure that input features for Task 1 have a minimum length of 5 seconds or longer. This is of course unnecessary for Task 2 which uses entire recordings. Next, each cycle (for Task 1) or recording (for Task 2) is transformed into a spectrogram with 64 features per analysis frame. For example, the log-Mel spectrogram is extracted with a window size of 1024 samples, a hop size of 256 samples, and 2048-point FFT, followed by average pooling in the frequency direction to yield a spectrogram with 64 frequency bins. Whichever type of spectrogram is used, the resulting time-frequency output is split into square non-overlapping patches of size 64×64. Since data augmentation is one of factors evaluated, we do not apply this technique to the baseline system.

TABLE III

Baseline system settings.

Factors	Setting
Re-sample	16kHz
Cycle duration (only for Task 1)	5 seconds
Spectrogram	log-Mel
Patch splitting	non-overlapped
Patch size	64 × 64
Data augmentation	None
Deep learning model	C-DNN based architecture

TABLE IV

Baseline C-DNN network architecture

Architecture	Layers	Output
	Input layer (image patch)	64×64
Conv. Block 01	BN - Cv [3×3] @ 64 - ReLU - BN - AP [2×2] - Dr (10%)	32×32×64
Conv. Block 02	BN - Cv [3×3] @ 128 - ReLU - BN - AP [2×2] - Dr (15%)	16×16×128
Conv. Block 03	BN - Cv [3×3] @ 256 - ReLU - BN - Dr (20%)	16×16×256
Conv. Block 04	BN - Cv [3×3] @ 256 - ReLU - BN - AP [2×2] - Dr (20%)	8×8×256
Conv. Block 05	BN - Cv [3×3] @ 512 - ReLU - BN - Dr (25%)	8×8×512
Conv. Block 06	BN - Cv [3×3] @ 512 - ReLU - BN - GAP - Dr (25%)	512
Dense Block	FC - Softmax layer	C

As can be seen from Table IV, the network architecture consists of seven blocks – six are convolutional and one is a dense block. The former blocks comprise batch normalization (BN) layers, convolutional (Cv [kernel size] @ kernel number) layers, rectified linear units (ReLU), average pooling (AP [kernel size]) and global average pooling (GAP) layers, and dropout (Dr (dropout percentage)). The dense block comprises a fully-connected (FC), and a final Softmax layer for classification. C refers to the number of classes, which depends on the specific task being evaluated. That is we train and test two separate C-DNN models with C set to 4 and 3 for Tasks 1 and 2, respectively.

C. Experimental settings for the baseline system

All the systems are implemented using TensorFlow. Network training makes use of the Adam optimizer [35] with 100 training epochs, a mini batch size of 100, and cross-entropy loss:

$$L_{Entropy}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_i \log \hat{\mathbf{y}}_i(\theta) + \frac{\lambda}{2}||\theta||_2^2, \qquad (7)$$

where θ are all trainable parameters, N is batch size, and constant λ is empirically set to 0.0001. $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ denote the ground truth and the network's output, respectively.

The spectrogram of an entire recording or cycle is separated into smaller patches and applied patch-by-patch to the C-DNN model which then returns the predicted probabilities over the classes for each patch. The predicted probabilities of an entire recording or a cycle is computed by averaging over its patches. Let us consider $\mathbf{P}^n = (P_1^n, P_2^n, \ldots, P_C^n)$ as the predicted probabilities obtained from the n^{th} out of N patches. Then, the mean predicted probability of a test sound instance is denoted as $\bar{\mathbf{P}} = (\bar{P}_1, \bar{P}_2, \ldots, \bar{P}_C)$ where

$$\bar{P}_c = \frac{1}{N}\sum_{n=1}^{N} P_c^n \quad \text{for} \quad 1 \le c \le C. \qquad (8)$$

The predicted label $\hat{y}$ is then determined as

$$\hat{y} = \underset{c \in \{1,2,\ldots,C\}}{\text{argmax}} \; \bar{P}_c. \qquad (9)$$

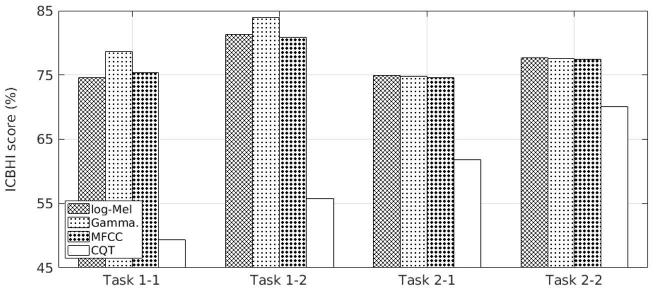


Fig. 2. Comparison of baseline performance using different spectrograms.

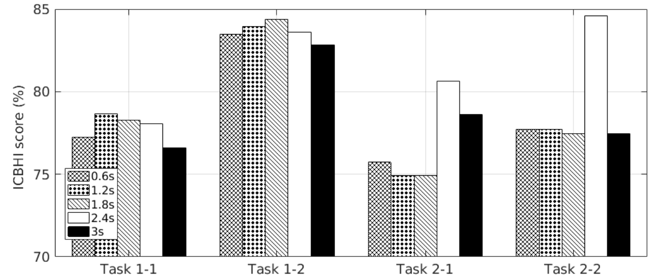


Fig. 3. Performance comparison between different time resolutions on each task.

IV. Analysis of influencing factors

We conducted experiments using the baseline system to investigate the impact of various factors on performance.

A. *Influence of spectrogram types*

From our previous work on natural sound datasets [36], [37], [38], we have established that the choice of spectrogram is one of the most important factors that affects final classification accuracy. Therefore, we evaluated the effect of spectrogram types on ICBHI performance for each task. To this end, we maintained all settings as described in Table III but used four spectrogram types: log-Mel spectrogram [39], Gammatone filterbank (Gamma) spectrogram [40], stacked Mel-Frequency Cepstral Coefficients (MFCC) [39], and rectangular Constant Q Transform (CQT) [39] spectrogram. We evaluated each of the spectrogram types on all four subtasks.

The obtained results in terms of ICBHI Score are shown in Fig. 2, revealing that MFCC, log-Mel, and Gamma spectrograms perform competitively, and are much better than CQT for all subtasks. Compared to log-Mel, Gamma spectrogram results in an improvement of 4% for Task 1-1 and 2.7% for Task 1-2. However log-Mel slightly outperforms its Gamma counterpart for Task 2 (0.1% and 0.2% in Tasks 2-1 and 2-2, respectively). MFCC, meanwhile, improves on log-Mel in Task 1-1 (0.8%), but the opposite is seen for all other subtasks (0.4%, 0.3%, and 0.3% lower in Tasks 1-2, 2-1, and 2-2).

These results suggest that Gamma spectrogram is optimal for anomaly cycle classification (Task 1) while log-Mel spectrogram works best for detection of respiratory diseases (Task 2). We thus adopted these two spectrograms in the following experiments for those respective tasks.

TABLE V
Baseline performance loss or gain on each subtask when overlapping spectrogram patches are used.

	Task 1-1	**Task 1-2**	**Task 2-1**	**Task 2-2**
No overlap	**78.6**	**84.0**	74.9	77.2
Overlap	77.8	83.7	**76.6**	**78.6**

B. *Influence of the overlapping degree*

As the spectrogram of an entire cycle or audio recoding is large in temporal dimension and is of variable length, it was split into smaller patches of 64 × 64 before feeding to the back-end deep learning models. In traditional signal processing systems, overlapping analysis windows are used to prevent occlusion of important features in the original data by edge effects. We therefore examined the effect of overlapping or non-overlapping patches on ICBHI performance. Specifically, we contrasted the baseline with non-overlapping patches (the settings in Table III) to the system with patches overlapped by 50% (noting that Gamma and log-Mel are applied on Task 1 and Task 2, respectively). Results shown in Table V reveal that the results obtained for Task 1 are better with non-overlapped patches (78.6% and 84.0% in Task 1-1 and Task 1-2, respectively) while those results for Task 2 are better with overlapped patches (76.6% and 78.6% in Task 2-1 and Task 2-2, respectively). These results can be explained by two potential factors: Firstly different spectrogram types were used in the two tasks, and secondly Task 1 classifies repeated respiratory cycles whereas Task 2 classifies unrepeated recordings.

C. *Influence of time resolution*

The baseline network operates on fixed-size patches where the time span encoded in each patch is defined by its horizontal dimension and sampling rate. Features are presented sequentially, and therefore the time span also defines the temporal resolution of features presented to the classifier. In this section, we explored the effect of different temporal resolution by adjusting patch widths to 0.6 s, 1.2 s, 1.8 s, 2.4 s, and 3.0 s. This is achieved by changing the patch size to be 64×32, 64×64, 64×96, 64×128, and 64×160, respectively, then repeat the experiments for each of them. We note that all settings were reused from Table III with exception that Gamma and log-Mel spectrograms were used for Task 1 and Task 2, respectively. The dimension of the network input layer is increased or decreased to accommodate the differing time resolution.

The obtained results are shown in Fig. 3 for the four subtasks. As can be seen, patch size of 64×64 (i.e. 1.2 s time resolution) works best for Task 1-1 and second best for Task 1-2 (scoring 78.6% and 84.0%, respectively). However a double sized patch of 64×128 (i.e. 2.4 s time resolution) is clearly the best for Tasks 2-1 and Task 2-2 (achieving 80.6% and 84.6%, respectively).

D. *Influence of data augmentation*

Data augmentation (DA) has been shown useful to improve the learning ability of deep learning models in tasks

TABLE VI

PERFORMANCE WITH AND WITHOUT MIXUP DATA AUGMENTATION.

	Task 1-1	**Task 1-2**	**Task 2-1**	**Task 2-2**
Non-mixup	78.6	84.0	74.9	77.2
mixup	**79.8**	**84.7**	**83.5**	**85.4**

TABLE VII

DEEP LEARNING FRAMEWORKS FOR TASKS 1 AND 2.

Factors	**Anomaly cycle classification**	**Respiratory disease detection**
Resample	16kHz	16kHz
Cycle duration	5s	N/A
Spectrogram	Gamma	log-Mel
Patch splitting	non-overlapped	overlapped
Patch size	64 × 64	64 × 128
Data augmentation	Yes	Yes

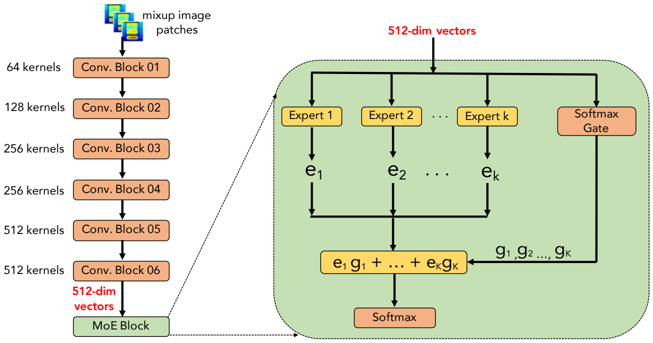


Fig. 4. The proposed CNN-MoE architecture.

involving natural sound classification [37], [36]. We, therefore, applied DA in form of mixup [32], [33] and studied its effect on respiratory sound classification. Let $\mathbf{X}_1$ and $\mathbf{X}_2$ denote two image patches randomly selected from the original image patches with their corresponding labels $\mathbf{y}_1$ and $\mathbf{y}_2$. Mixup DA generates new image patches:

$$\mathbf{X}_{mp1} = \alpha\mathbf{X}_1 + (1-\alpha)\mathbf{X}_2, \tag{10}$$

$$\mathbf{X}_{mp2} = (1-\alpha)\mathbf{X}_1 + \alpha\mathbf{X}_2, \tag{11}$$

$$\mathbf{y}_{mp1} = \alpha\mathbf{y}_1 + (1-\alpha)\mathbf{y}_2, \tag{12}$$

$$\mathbf{y}_{mp2} = (1-\alpha)\mathbf{y}_1 + \alpha\mathbf{y}_2, \tag{13}$$

where $\mathbf{X}_{mp1}$ and $\mathbf{X}_{mp2}$ are the two new image patches obtained by mixing $\mathbf{X}_1$ and $\mathbf{X}_2$ with a mixing coefficient α. By using two types of uniform or beta distribution to generate mixing coefficient α, this doubles the data size and hence, the training time. Note that in Task 1 DA mixes the *Normal* class with one of the other classes (since there is already one mixed class in the dataset, i.e. *Crackle & Wheeze*), whereas it randomly mixes samples of all classes in Task 2. After mixup, the generated patches were shuffled and fed into the C-DNN baseline. Since the labels $\mathbf{y}_{mp1}$ and $\mathbf{y}_{mp2}$ of the resulting patches were no longer one-hot encoded, it was, therefore, necessary to replace the cross-entropy loss by the Kullback-Leibler (KL) divergence loss:

$$L_{KL}(\theta) = \sum_{n=1}^{N} \mathbf{y}_n \log\left\{\frac{\mathbf{y}_n}{\hat{\mathbf{y}}_n}\right\} + \frac{\lambda}{2}||\theta||_2^2. \tag{14}$$

Again, θ denotes the trainable network parameters, λ denotes the ℓ_2-norm regularization coefficient and was set to 0.0001. N is the batch number, $\mathbf{y}_n$ and $\hat{\mathbf{y}}_n$ denote the ground-truth and the network output, respectively.

Using the settings in Table III with Gamma spectrogram for Task 1 and log-Mel spectrogram for Task 2, we can assess the improvement over the baseline in each subtask due to mixup data augmentation. Results shown in Table VI indicate that mixup data augmentation substantially improves the ICBHI score by 8.6% and 8.2% on Tasks 2-1 and 2-2, respectively. However, modest improvements are seen in Task 1.

V. ENHANCED DEEP LEARNING FRAMEWORK

From the analysis of influencing factors presented above, we propose two systems. One for Task 1 anomaly cycle classification, and the other for Task 2 respiratory disease detection; both summarised in Table VII. In this section we propose to enhance the performance of the C-DNN architecture by incorporating a mixture-of-experts (MoE) technique into the DNN part of the network, leading to the CNN-MoE architecture.

A. CNN-MoE network architecture

According to the C-DNN architecture entailed in Table IV, the first six convolutional blocks are used to map the image patch input to condensed and discriminative embeddings, often referred to as high-level features. The features are then classified by a dense block comprising a fully-connected layer and Softmax. On the basis that the embedding may contain more information than a single fully-connected layer can unlock, we replace the dense block with a mixture-of-experts (MoE) block as shown in Fig. 4. The MoE block architecture [41] has multiple experts connected to a gate network. The gate network is in charge of deciding which expert is applied to which input region. In our context, the 512-dimensional embedding from the final global average pooling layer (GAP) is presented simultaneously to all experts. The output from all experts are then gated and the combined output is presented to Softmax for classification. In our system, each expert comprises a fully-connected layer and a ReLU activation function. Each expert input dimension, as we have noted, is 512, and the output dimension from each is the number of classes *C*. The gate network is implemented by a Softmax Gate – an additional fully-connected layer with Softmax activation function and the gating dimension equals to the number of experts. Let $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K \in \mathbf{R}^C$ denote the output vectors from the K experts, and $g_1, g_2, \ldots, g_K$ denote the outputs of the gate network where $g_k \in \mathbf{R}$ and $\sum g_k = 1$. The predicted output is then given by

$$\hat{\mathbf{y}} = \text{softmax}\left\{\sum_{k=1}^{K} g_k \mathbf{e}_k\right\}. \tag{15}$$

The proposed systems, as defined in Table VII, are trained with KL-divergence loss [42] (due to the use of mixup data

TABLE VIII
PERFORMANCE COMPARISON BETWEEN THE C-DNN BASELINE AND THE PROPOSED CNN-MOE FRAMEWORKS (**C-DNN SCORES/ CNN-MOE SCORES**) OVER 5-FOLD CROSS VALIDATION (THE UPPER PART) AND ICBHI CHALLENGE'S DATA SPLIT (THE LOWER PART)

Tasks	Spec.	Sen.	ICBHI score	F1 score	Kappa score
1-1	85.6/86.6	70.4/71.3	77.4/78.5	72.2/72.6	66.8/66.8
1-2	85.6/86.6	81.9/82.6	84.1/84.4	84.8/85.0	69.0/70.0
2-1	75.3/86.7	91.1/94.5	84.7/90.5	72.6/77.2	65.8/73.0
2-2	75.3/86.7	96.0/97.9	86.2/92.0	76.8/82.4	61.6/65.0
1-1	60.0/72.4	25.3/21.5	43.3/47.0	31.0/32.0	8.2/11.2
1-2	60.0/72.4	45.1/37.5	53.3/54.1	53.0/55.0	5.2/10.3
2-1	64.7/71.0	92.3/98.1	78.5/84.0	64.0/74.0	62.7/77.0
2-2	64.7/71.0	93.5/98.2	78.7/84.1	81.0/84.0	61.3/67.1

augmentation) and use the same experimental settings as the previous experiments with the C-DNN baseline.

B. Performance comparison

Comparing the C-DNN baseline to the proposed CNN-MoE: We evaluated the efficacy of the applied MoE technique (experimentally using K=10 experts) compared to the C-DNN baseline and report the performance of both in Table VIII (Of note, both the systems followed the settings in Table VII, with the back-end classifier being either C-DNN or CNN-MoE there are thus eight systems in total, two C-DNNs and two CNN-MoEs for each kind of data split).

The results in Table VIII clearly indicate that the CNN-MoE systems outperforms the C-DNN baseline. Although we see only marginal gains over the C-DNN baseline in Task 1, CNN-MoE results in improvement with a margin as large as 6% absolute in terms of ICBHI score with both the data splits, 5-fold cross validation and ICBHI challenge's data split, in Task 2. Notably, the large gap in Task 1 performance between 5-fold cross validation (the upper part of Table VIII) and ICBHI data split (the lower part of Table VIII) reveals that the performance of the tasks strongly depends on the subjects. To confirm that CNN-MoE shows similar performance to the C-DNN baseline in Task 1-1 but outperforms it in Task 2-1, we further conducted a paired t-test over 5-fold cross validation with the null hypothesis "the performance does not differ between C-DNN and CNN-MoE"). Let consider $\mathbf{d} = \{d_1, d_2, \ldots, d_5\}$ as a set of 5 differences of the ICBHI scores obtained by CNN-MoE and C-DNN over 5 folds evaluated. The t-score was computed as $\bar{d}/\sqrt{0.2\sigma^2}$, where $\bar{d}$ and σ^2 denote the sample mean and sample variance of $\mathbf{d}$, respectively. t-scores of 1.26 and 10.66 are obtained for Task 1-1 and Task 2-1 with p-values of 0.13 and 0.0002, respectively. Thus we can confirm the similar performance in Task 1-1 and significant improvement of CNN-MoE over the C-DNN baseline in Task 2-1.

Comparing to state-of-the-art systems: We next contrasted the proposed framework to state-of-the-art systems. For each task, we evaluate every system twice – once with the ICBHI challenge train/test split, and once with the random split (as described in Section II-B). Considering the first data splitting method, Table IX shows the performance obtained by the proposed framework and state-of-the-art published systems (where available). We note that the proposed framework lies second in Task 1-1 evaluation. Our results for other subtasks were listed in Table VIII. Only Task 2-2 is found in the literature (for the ICBHI data split) [21], achieving 72%, which is surpassed by 84% obtained by our system.

TABLE IX
COMPARISON AGAINST STATE-OF-THE-ART SYSTEMS WITH ICBHI CHALLENGE SPLITTING (HIGHEST SCORES IN **BOLD**).

Task	Method	Spec.	Sen.	Score
1-1, 4-category	DT [16]	0.75	0.12	0.43
1-1, 4-category	HMM [14]	0.38	**0.41**	0.39
1-1, 4-category	SVM [15]	0.78	0.20	0.47
1-1, 4-category	CNN-RNN [27]	**0.81**	0.28	**0.54**
1-1, 4-category	**Our system**	0.68	0.26	0.47

TABLE X
PERFORMANCE COMPARISON BETWEEN THE PROPOSED SYSTEM AND STATE-OF-THE-ART SYSTEMS FOLLOWING THE RANDOM DATA SPLIT (HIGHEST SCORES ARE HIGHLIGHTED IN **BOLD**).

Task	Method	train/test	Spec.	Sen.	Score
1-1, 4-category	Boosted DT [8]	60/40	0.78	0.21	0.49
1-1, 4-category	CNN [23]	80/20	0.77	0.45	0.61
1-1, 4-category	CNN-RNN [21]	5 folds	0.84	0.49	0.66
1-1, 4-category	LSTM [24]	80/20	0.85	0.62	0.74
1-1, 4-category	**Our system**	5 folds	**0.90**	**0.68**	**0.79**
1-2, 2-category	Boosted DT [8]	60/40	0.78	0.33	0.56
1-2, 2-category	LSTM [24]	80/20	-	-	0.81
1-2, 2-category	CNN [20]	75/25	-	-	0.82
1-2, 2-category	**Our system**	5 folds	**0.90**	**0.78**	**0.84**
2-1, 3-category	CNN [23]	80/20	0.76	0.89	0.83
2-1, 3-category	LSTM [24]	80/20	0.82	**0.98**	0.90
2-1, 3-category	**Our system**	5 folds	**0.86**	0.95	**0.91**
2-2, 2-category	Boosted DT [8]	60/40	0.85	0.85	0.85
2-2, 2-category	CNN [23]	80/20	0.78	0.97	0.88
2-2, 2-category	RUSBoost DT [6]	50/50	**0.93**	0.86	0.90
2-2, 2-category	LSTM [24]	80/20	0.82	**0.99**	0.91
2-2, 2-category	**Our system**	5 folds	0.86	0.98	**0.92**

Table X compares the performance obtained by our system with previously published results that used the random train/test splitting method. For Tasks 1-1 and 1-2, the proposed framework clearly outperforms other systems quite consistently. Meanwhile, for Task 2-1 and 2-2, the proposed method also outperforms other systems in terms of overall ICBHI score, but not necessarily simultaneously for both subcomponents, specificity or sensitivity.

C. Discussion

Comparing Tables IX and X, it is notable that those systems following the ICBHI data split (i.e. recordings from the same patient are never found in both train/test subsets) exhibit considerably lower performance than those following random splitting. This indicates that the ICBHI dataset presents a high dependence on patient characteristics, which likely make respiratory cycle classification challenging in practice.

However, all the results obtained by the proposed framework for Tasks 2-1 and 2-2 (with both splitting methods) exceed 84%. These results for recording-based classification of lung disease – which is highly related to the overall aim of lung disease detection – provide a strong indicator of the robustness of the proposed framework. As does the fact that the same proposed framework is capable of performing well for all subtasks.

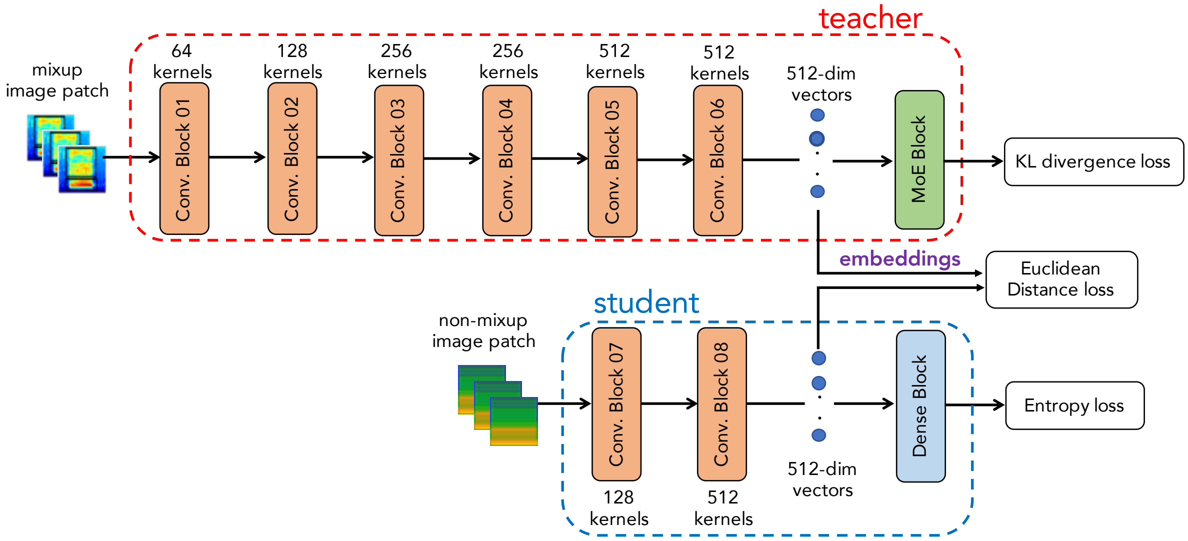


Fig. 5. Architecture of the Student-Teacher scheme.

TABLE XI
THE STUDENT NETWORK ARCHITECTURE.

Architecture	Layers	Output
	Input layer (image patch)	64×128
Conv. Block 07	Cv [3×3] @ 128 - ReLU - AP [4×4]	16×32×128
Conv. Block 08	Cv [3×3] @ 512 - ReLU - GAP	512
Dense Block	FC - Softmax layer	3

VI. STUDENT-TEACHER SCHEME FOR RESPIRATORY DISEASE DETECTION

A. The proposed student-teacher arrangement

Recent works on sound scene and sound event detection reported the effectiveness of Teacher-Student learning schemes [43], [44]. Among other advantages, these schemes offer a trade-off between model size and performance. Since the complexity of our best model based on the proposed MoE framework may be a barrier to future real-time implementation, we explore whether a student-teacher scheme can be used to train a network with much lower complexity (Task 2). The proposed solution, as shown in Fig. 5, comprises two networks, namely the Teacher and the Student. The teacher network re-uses the high-performance CNN-MoE architecture introduced in Section V-A. On the other hand, the student network features a compact architecture, comprising two convolutional blocks (identified *Conv. Block 07* and *Conv. Block 08* in the figure) and a dense block as shown in Table XI (note that the student network does not apply batch normalisation, dropout, or mixup data augmentation).

Training the Teacher-Student network is separated into two phases. First, the Teacher is trained as usual. Afterwards, the Teacher's embedding is distilled to the Student's embedding to assist in the Student's learning process. We will also empirically investigate the influence of this knowledge distillation on the student network's performance. With the presence of this knowledge distillation, training the student network, therefore, aims to minimize two losses: (1) the Euclidean distance between the teacher and student embedding, and (2) the standard cross-entropy loss on the student's classification output. The combined loss function is therefore,

$$L(\theta) = L_{Entropy}(\theta) + \gamma L_{Euclidean}(\theta), \qquad (16)$$

Here, the hyperparameter γ is empirically set to 0.5 to balance the two constituent losses. θ represents the trainable parameters of the student network. Other hyper-parameters and settings are inherited from Section III-C.

B. Experimental results using the Teacher-Student scheme

The experimental results obtained by the student network in comparison with the teacher network are shown in Table XII. On the one hand, it can be seen that without knowledge distilled from the teacher network, the small-footprint student network obtains a substantially low specificity score, although it maintains a very good sensitivity. This observation is consistent with the overall ICBHI score and can be explained by the simplicity of the network which results in low learning capacity. On the other hand, distilling knowledge from the teacher significantly boosts the student performance, yielding specificity, sensitivity, and ICBHI scores that are very competitive to those of the teacher network – even though the student network is much smaller and simpler.

Details of the model footprint are shown in Table XIII. We can see that the Teacher uses six convolutional layers, eleven fully-connected layers and twelve BN layers that together contribute to a large model size with 4.5×10^6 trainable parameters. Meanwhile, the Student only uses two convolutional layers and one fully-connected layer, requiring only 0.6×10^6 parameters, approximately one-seventh of the Teacher's. The model footprints also scale in terms of computational cost of multiply-accumulate (MAC) operations during inference. While an inference process on the Teacher costs 44,886 kMAC operations, the Student only costs 9,513 kMACs (the MAC operation computation for a deep learning network is presented in [45]). The inference process for a 20

TABLE XII
CLASSIFICATION PERFORMANCE COMPARISON BETWEEN TEACHER AND STUDENT WITH AND WITHOUT KNOWLEDGE DISTILLING.

		Five-fold random split			**ICBHI split**		
Task		**Spec.**	**Sen.**	**ICBHI Score**	**Spec.**	**Sen.**	**ICBHI Score**
2-1, 3-category	Teacher	0.86	0.95	0.91	0.71	0.98	0.84
	Student w/o knowledge distill	0.43	0.94	0.68	0.41	0.97	0.69
	Student w/ knowledge distill	0.86	0.90	0.88	0.71	0.98	0.84
2-2, 2-category	Teacher	0.86	0.98	0.92	0.71	0.98	0.84
	Student w/o knowledge distill	0.43	0.99	0.71	0.41	0.99	0.70
	Student w/ knowledge distill	0.86	0.96	0.91	0.71	0.98	0.84

TABLE XIII
MODEL FOOTPRINT COMPARISON BETWEEN TEACHER AND STUDENT

Features	**Teacher**	**Student**
Trainable Convolutional Layers	6	2
Trainable Fully-connected Layers	11	1
Batch normalization	12	0
Number of trainable parameters	4.5×10^6	0.6×10^6
Number of MAC operations	44,886 K	9,513 K

second long recording in Task 1-1, conducted by a Tesla P100 GPU, takes 0.5 s; nearly ten times longer than the 0.045 s required for the Student's inference process.

VII. CONCLUSION

This paper has presented a robust deep learning framework for the analysis of respiratory anomalies and detection of lung diseases from lung auscultation recordings. Extensive experiments were conducted with different architectures and experimental settings using the ICBHI dataset, and two defined tasks related to that. The proposed system was evaluated against existing state-of-the-art methods, outperforming them for most of the challenge tasks. Furthermore, to facilitate implementation in real-time systems, a Teacher-Student learning scheme was explored to significantly reduce model complexity while still achieving very high accuracy. The final experimental results validate the application of deep learning for the timely diagnosis of respiratory diseases, bringing this research area one step closer to clinical applications. In future, we aim to explore model compression with pruning and quantisation to further try and reduce complexity, before implementing the simplified detector in an embedded device.

REFERENCES

[1] World Health Organization, "The global impact of respiratory diseases (second edition)," 2017. [Online]. Available: https://www.who.int/gard/publications/The_Global_Impact_of_Respiratory_Disease.pdf

[2] H. Polat and İ. Güler, "A simple computer-based measurement and analysis system of pulmonary auscultation sounds," *Journal of medical systems*, vol. 28, no. 6, pp. 665–672, 2004.

[3] R. J. Riella, P. Nohama, R. F. Borges, and A. L. Stelle, "Automatic wheezing recognition in recorded lung sounds," in *Proc. EMBC*, 2003, pp. 2535–2538.

[4] S. Reichert, R. Gass, C. Brandt, and E. Andrès, "Analysis of respiratory sounds: state of the art," in *Proc. CCRPM*, 2008, pp. CCRPM–S530.

[5] T. Okubo, N. Nakamura, M. Yamashita, and S. Matsunaga, "Classification of healthy subjects and patients with pulmonary emphysema using continuous respiratory sounds," in *Proc. EMBC*, 2014, pp. 70–73.

[6] X. H. Kok, S. A. Imtiaz, and E. Rodriguez-Villegas, "A novel method for automatic identification of respiratory disease from acoustic recordings," in *Proc. EMBC*, 2019, pp. 2589–2592.

[7] M. Grønnesby, J. Solis, E. Holsbø, H. Melbye, and L. Bongo, "Feature extraction for machine learning based crackle detection in lung sounds from a health survey," *arXiv preprint arXiv:1706.00005*, 2017.

[8] G. Chambres, P. Hanna, and M. Desainte-Catherine, "Automatic detection of patient with respiratory diseases using lung sound analysis," in *Proc. CBMI*, 2018, pp. 1–6.

[9] L. Mendes, I. M. Vogiatzis, E. Perantoni, E. Kaimakamis, I. Chouvarda, N. Maglaveras, J. Henriques, P. Carvalho, and R. P. Paiva, "Detection of crackle events using a multi-feature approach," in *Proc. EMBC*, 2016, pp. 3679–3683.

[10] S. Datta, A. D. Choudhury, P. Deshpande, S. Bhattacharya, and A. Pal, "Automated lung sound analysis for detecting pulmonary abnormalities," in *Proc. EMBC*, 2017, pp. 4594–4598.

[11] D. Reshef, Y. Reshef, H. Finucane, S. Grossman, G. McVean, P. Turnbaugh, E. Lander, M. Mitzenmacher, and P. Sabeti, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011.

[12] N. Sengupta, M. Sahidullah, and G. Saha, "Lung sound classification using local binary pattern," *arXiv preprint arXiv:1710.01703*, 2017.

[13] D. Oletic, M. Matijascic, V. Bilas, and M. Magno, "Hidden markov model-based asthmatic wheeze recognition algorithm leveraging the parallel ultra-low-power processor (pulp)," in *IEEE Sensors Applications Symposium*, 2019, pp. 1–6.

[14] N. Jakovljević and T. Lončar-Turukalo, "Hidden markov model based respiratory sound classification," in *Precision Medicine Powered by pHealth and Connected Health*, 2018, pp. 39–43.

[15] G. Serbes, S. Ulukaya, and Y. Kahya, "An automated lung sound preprocessing and classification system based onspectral analysis methods," in *Precision Medicine Powered by pHealth and Connected Health*, 2018, pp. 45–49.

[16] B. M. Rocha *et al.*, "An open access database for the evaluation of respiratory sound classification algorithms," *Physiological measurement*, vol. 40, no. 3, p. 035001, 2019.

[17] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *Proc. ICASSP*, 2015, pp. 559–563.

[18] I. McLoughlin, Y. Song, L. D. Pham, H. Phan, P. Ramaswamy, and L. Yue, "Early detection of continuous and partial audio events using CNN," in *Proc. INTERSPEECH*, 2018, pp. 3314–3318.

[19] L. Shi, K. Du, C. Zhang, H. Ma, and W. Yan, "Lung sound recognition algorithm based on vggish-bigru," *IEEE Access*, vol. 7, pp. 139 438–139 449, 2019.

[20] R. Liu, S. Cai, K. Zhang, and N. Hu, "Detection of adventitious respiratory sounds based on convolutional neural network," in *Proc. ICIIBMS*, 2019, pp. 298–303.

[21] J. Acharya and A. Basu, "Deep neural network for respiratory sound classification in wearable devices enabled by patient specific model tuning," *IEEE transactions on biomedical circuits and systems*, vol. 14, no. 3, pp. 535–544, 2020.

[22] M. Aykanat, Ö. Kılıç, B. Kurt, and S. Saryal, "Classification of lung sounds using convolutional neural networks," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, pp. 1–9, 2017.

[23] D. Perna, "Convolutional neural networks learning from respiratory data," in *Proc. BIBM*, 2018, pp. 2109–2113.

[24] D. Perna and A. Tagarelli, "Deep auscultation: Predicting respiratory anomalies and diseases via recurrent neural networks," in *Proc. CBMS*, 2019, pp. 50–55.

[25] E. Messner, M. Fediuk, P. Swatek, S. Scheidl, F. Smolle-Juttner, H. Olschewski, and F. Pernkopf, "Crackle and breathing phase detection in lung sounds with deep bidirectional gated recurrent neural networks," in *Proc. EMBC*, 2018, pp. 356–359.

[26] K. Kochetov, E. Putin, M. Balashov, A. Filchenkov, and A. Shalyto, "Noise masking recurrent neural network for respiratory sound clas-

sification," in *International Conference on Artificial Neural Networks*, 2018, pp. 208–217.

[27] K. Minami, H. Lu, H. Kim, S. Mabu, Y. Hirano, and S. Kido, "Automatic classification of large-scale respiratory sound dataset based on convolutional neural network," in *Proc. ICCAS*, 2019, pp. 804–807.

[28] H. Chen, X. Yuan, Z. Pei, M. Li, and J. Li, "Triple-classification of respiratory sounds using optimized s-transform and deep residual networks," *IEEE Access*, vol. 7, pp. 32 845–32 852, 2019.

[29] B. Rocha, D. Filos, L. Mendes, Vogiatzis *et al.*, "A respiratory sound database for the development of automated classification," in *Precision Medicine Powered by pHealth and Connected Health*, 2018, pp. 33–37.

[30] C. J. Rijsbergen, *Information Retrieval: Uncertainty and Logics*. Butterworth-Heinemann, 1979.

[31] C. C. Berry, "The kappa statistic," *Journal of the American Medical Association*, 268(18):2513, 1992.

[32] K. Xu, D. Feng, H. Mi, B. Zhu, D. Wang, L. Zhang, H. Cai, and S. Liu, "Mixup-based acoustic scene classification using multi-channel convolutional neural network," in *Pacific Rim Conference on Multimedia*, 2018, pp. 14–23.

[33] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," in *ICLR*, 2018.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[36] L. Pham, I. Mcloughlin, H. Phan, and R. Palaniappan, "A robust framework for acoustic scene classification," in *Proc. INTERSPEECH*, 2019, pp. 3634–3638.

[37] L. Pham, I. McLoughlin, H. Phan, R. Palaniappan, and Y. Lang, "Bag-of-features models based on C-DNN network for acoustic scene classification," in *Proc. AES*, 2019.

[38] L. Pham, I. McLoughlin, H. Phan, R. Palaniappan, and A. Mertins, "Deep feature embedding and hierarchical classification for audio scene classification," in *Proc. IJCNN*, 2020, pp. 1–7.

[39] McFee, Brian, R. Colin, L. Dawen, D. P., M. Matt, B. Eric, and N. Oriol, "librosa: Audio and music signal analysis in python," in *Proc. 14th Python in Science Conference*, 2015, pp. 18–25.

[40] D. Ellis, "Gammatone-like spectrogram," 2009. [Online]. Available: http://www.ee.columbia.edu/dpwe/resources/matlab/gammatonegram

[41] E. Garmash and C. Monz, "Ensemble learning for multi-source neural machine translation," in *Proc. COLING*, 2016, pp. 1409–1418.

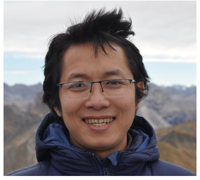
[42] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[43] H. Heo, J. Jung, H. Shim, and H. Yu, "Acoustic scene classification using teacher-student learning with soft-labels," *arXiv preprint arXiv:1904.10135*, 2019.

[44] L. Gao, H. Mi, B. Zhu, D. Feng, Y. Li, and Y. Peng, "An adversarial feature distillation method for audio classification," *IEEE Access*, vol. 7, pp. 105 319–105 330, 2019.

[45] M. Taghavi and M. Shoaran, "Hardware complexity analysis of deep neural networks and decision tree ensembles for real-time neural data classification," in *9th International IEEE/EMBS Conference on Neural Engineering*, 2019, pp. 407–410.

Lam Pham received the Bachelor of Engineering, and Master of Science degree in Electronics-Telecommunication Engineering from Ho Chi Minh City University of Technology in 2009 and 2012, respectively. Currently he is being PhD in University of Kent, UK and also working as Data Scientist in Center for Digital Saftey & Security, Austrian Institute of Technology (AIT), Austria. His research interests include machine hearing and signal processing with a research focus on Acoustic Scene Classification (ASC) and Acoustic Event Detection (AED).



Huy Phan received the M.Eng. degree from Nanyang Technological University, Singapore, in 2012, and the Dr.-Ing. degree in computer science from the University of Lübeck, Lübeck, Germany, in 2017. From 2017 to 2018, he was a Postdoctoral Research Assistant with the University of Oxford, Oxford, United Kingdom. From 2018 to 2020, he was a Lecturer at the University of Kent, Kent, United Kingdom. In April 2020, he joined Queen Mary University of London, London, United Kingdom, where he is a Lecturer in Artificial Intelligence in the School of Electronic Engineering and Computer Science. His research interests include machine learning and signal processing with a special focus on audio and biosignal analysis. In 2018, he was awarded the Bernd Fischer award by the University of Lübeck for the best PhD thesis.

Ramaswamy Palaniappan is currently a Reader in the School of Computing, University of Kent and heads the Data Science Research Group. His current research interests include signal processing and machine learning for electrophysiological applications. To date, he has written two text books in engineering and published over 200 peer-reviewed articles (with over 3000 citations). He serves in editorial boards for several international journals. He also serves in the prestigious Peer Review College for UK Research Councils and many other international grant funding bodies. He has supervised more than half a dozen postgraduate students to completion and has more than two decades of multi-disciplinary teaching experience in computer science and engineering (electrical and biomedical) disciplines. His pioneering work on revolutionary new areas of brain-computer interfaces and emerging biometrics has not only received international awards and recognition by the scientific community but also from the media and public.

Professor Alfred Mertins received his Dipl.-Ing. degree from the University of Paderborn, Germany, in 1984, the Dr.-Ing. degree in Electrical Engineering and the Dr.-Ing. habil. degree in Telecommunications from the Hamburg University of Technology, Germany, in 1991 and 1994, respectively. From 1986 to 1991 he was a Research Assistant at the Hamburg University of Technology, Germany, and from 1991 to 1995 he was a Senior Scientist at the Microelectronics Applications Center Hamburg, Germany. From 1996 to 1997 he was with the University of Kiel, Germany, and from 1997 to 1998 with the University of

Western Australia. In 1998, he joined the University of Wollongong, where he was at last an Associate Professor of Electrical Engineering. From 2003 to 2006, he was a Professor in the Faculty of Mathematics and Science at the University of Oldenburg, Germany. In November 2006, he joined the University of Lübeck, Germany, where he is a Professor and Director of the Institute for Signal Processing. His research interests include speech, audio, and image processing, wavelets and filter banks, pattern recognition, and medical imaging.

Professor Ian McLoughlin completed his PhD in Electronic and Electrical Engineering at the University of Birmingham, UK in 1997. He has worked for over 10 years in the R&D industry and 19 years in academia, on three continents. He is a Fellow of the IET, a Chartered Engineer and has been a full Professor since 2012. He has written many papers and hand holds several patents in this domain, and is the author of the Cambridge University Press reference text on speech and audio processing.