IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 August 2022, accepted 3 September 2022, date of publication 12 September 2022, date of current version 19 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3205591

METHODS

Classification of Cough Sounds Using Spectrogram Methods and a Parallel-Stream One-Dimensional Deep Convolutional Neural Network

YO-PING HUANG [1,2,3,4], **(Fellow, IEEE), AND RICHARD MUSHI** [1]

[1] Department of Electrical Engineering, National Taipei University of Technology, Taipei 10608, Taiwan
[2] Department of Electrical Engineering, National Penghu University of Science and Technology, Penghu 88046, Taiwan
[3] Department of Computer Science and Information Engineering, National Taipei University, Taipei 23741, Taiwan
[4] Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan

Corresponding author: Yo-Ping Huang (yphuang@gms.npu.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST108-2221-E-346-006-MY3 and Grant MOST111-2221-E-346-002-MY3; and in part by the Acer Group Research Project under Grant 210D001-1.

ABSTRACT Currently, a subjective method is used to diagnose cough sounds, particularly wet and dry coughs, which can lead to incorrect diagnoses. In this study, novel emergent features were extracted using spectrogram methods and a parallel-stream one-dimensional (1D) deep convolutional neural network (DCNN) to classify cough sounds. The data of this study were obtained from two datasets. We employed the Mel spectrogram, chromagram constant-Q transform, Mel-frequency cepstral coefficient, constant-Q cepstral coefficient, and linear predictive code coefficient to conduct features analysis. The maximum, mean, variance, and standard deviation values of the original spectrogram as well as the maximum first and second derivatives of this spectrogram were extracted and fused to create a single-feature vector. We adopted two types of features: single features and combined features. Each design was restructured according to the magnitude of features with high discrimination power. A parallel-stream 1D-DCNN was developed for classifying cough sounds accurately. We compared the results obtained using the aforementioned network with those obtained using a single-stream 1D-DCNN. We found that the parallel-stream network outperformed the single-stream network for some feature sets. The developed network achieved $F1$ scores of 98.61% and 82.96% for the first and second datasets, respectively. The concatenation of layers at the flattening level resulted in an $F1$ score of 99.30% in dataset one. Moreover, layer merging strategies exhibited a better performance at the second convolutional layer level than at the flattening layer level in many cases.

INDEX TERMS Classification, convolutional neural network, cough sounds, feature extraction, spectrogram methods.

I. INTRODUCTION

Because of the rise in respiratory diseases, increased research attention has been paid to cough sound classification. Coughing is a critical symptom of respiratory diseases [1], [2], and two types of coughs exist: wet and dry coughs [3]. Discriminating between these two types of coughs is crucial [4]; however, many hospitals in developing countries adopt inefficient approaches for diagnosing the cough type.

Advances in computer-assisted technology have enabled the use of audio tools and deep learning (DL) models for reliably classifying wet and dry cough sounds. An audio tool is an electronic device that records and stores sound. Audio tools are used to record sounds in cough research [5], [6] and other related fields. Various audio feature extraction methods have been proposed to detect, analyze, and classify cough sounds. Three types of audio features exist [7], [8], [9]:

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

time-domain features, including zero-crossing, root mean square, and energy envelope; frequency-domain features, including spectral centroid, bandwidth, roll-off, and power spectral density; and time–frequency features, including the Mel spectrogram and Mel-frequency cepstral coefficient (MFCC).

In the present study, time–frequency features were used because they can simultaneously represent signal properties in the time and frequency domains. These features have been used in many studies for classification application [10], [11], [12].

The following questions were addressed in the present study: (1) how are the emergent features extracted using different methods can be combined and input into a DL model? (2) which DL method is appropriate for classifying emergent features? and (3) how are the problems of a small dataset and varying cough signal dimensions overcome in cough research?

The output of the time–frequency method is a two-dimensional (2D) pixel matrix, which can be visualized, and this method is termed as the time–frequency spectrogram method [13]. Several types of features can be extracted from spectrograms to generate one-dimensional (1D) feature vectors. Emergent feature vectors can be extracted from spectrograms along the time axis (column) or frequency axis (row). In our previous study [14], we used one spectrogram method and focused on extracting the maximum cepstral coefficient vector from MFCC row. In this study, we expanded the technique by including five spectrogram methods and their derivatives and then capturing numerous features from spectrograms and their derivatives. The extracted subfeatures include the maximum, mean, variance, and standard deviation. In each spectrogram method, the extracted subfeatures were fused to form a single-feature vector.

Feature vectors obtained in each adopted method were aggregated to obtain diverse feature combinations. Aggregating features and inputting them into a DL model is tedious. In general, a DL model [e.g., a convolutional neural network (CNN)] is affected by the relationships between features in space, especially when two or more features are integrated. In [15], the correlation matrix, clustering, and dendrogram techniques were used to restructure integrated features. A drawback of using the dendrogram technique is the long time required for processing dendrogram data, which leads to errors. Thus, in this paper, we propose methods for restructuring the position of combined features according to their discrimination power and mutual information value.

Studies have transformed cough signals into features such as Mel spectrograms and MFCCs [16], [17] and then have input these features directly into 2D DL models. Thus, cough sounds can be detected using attributes from a spectrogram. These attributes are one-dimensional features and are suitable inputs for 1D DL models, which are highly useful because of their low computational demand, time requirement, and cost.

To the best of our knowledge, a few studies have adopted a 1D DL model for cough detection, and most relevant studies have adopted a single-stream model for cough detection. For example, Baramulari *et al.* [18] classified cough sounds by using a bidirectional long short-term memory model. Hassan *et al.* [19] used a recurrent neural network to detect COVID-19. Amrulloh *et al.* [20] employed a neural network to classify pneumonia and asthma infections. In the present study, we examined the performance of a 1D-CNN, gated recurrent unit (GRU) model, and a neural network for cough detection. We found that the 1D-CNN model outperformed the other two models. Therefore, the 1D-CNN model was used for further analysis in this study.

Insufficient data are a challenge encountered in studies on cough sound [21], [22], as well as a class imbalance problem [23], [24]. A similar problem was encountered in this study. Of the two collected datasets, one contained 118 wet cough sounds and 170 dry cough sounds, and the other contained 389 wet cough sounds and 413 dry cough sounds. These datasets exhibited the class imbalance problem. Therefore, we used the weighted $F1$ score [25] and Matthews correlation coefficient (MCC) [26] as metrics for assessing model performance. Furthermore, the two datasets contained signals with different dimensions. We used a zero-padding system to address the varying dimensions of cough signals. The problem of insufficient data was addressed using the data enhancement technique.

The main contributions of this study are as follows:

1) Sounds of wet and dry coughs were classified using novel features extracted using spectrogram methods and a parallel-stream 1D deep CNN (DCNN).
2) The features extracted using spectrogram methods were analyzed using a novel method to classify wet and dry coughs.
3) Feature structures were designed, and two techniques were developed for restructuring the positions of combined features and were compared to determine the better technique.
4) A parallel-stream 1D-DCNN was developed, and the performance of this CNN was compared with that of a single-stream 1D-CNN. The developed model differs from existing related models [15], [36] in three ways: (1) it does not contain a maximum pooling layer, (2) layer concatenation occurs in its flattening layer, and (3) it contains a few layers as a small network might prevent overfitting [28]. Moreover, the performance benefits of concatenating layers at different levels were examined.
5) Model performance achieved with layer merging strategies at different levels in a parallel-stream network was examined.

The rest of this article is organized as follows. Section II provides an overview of the related research. Section III details the methodology used for constructing the designed system. Section IV describes the proposed DL models. Section V presents the experimental results and a discussion on the results. Finally, Section VI provides the conclusions of this study.

II. RELATED RESEARCH

Studies have reported that the cough type can be identified using audio recordings, feature extraction techniques, and DL models. Therefore, we reviewed research on feature extraction approaches and 1D DL models.

Islam *et al.* [29] employed a deep neural network (DNN) to detect COVID-19. They used time-, frequency-, and time–frequency-domain features for COVID-19 detection and obtained an accuracy of 97.5%. Zhao *et al.* [30] used pig cough sounds to differentiate respiratory diseases. They extracted 39 MFCC features and classified them using a hybrid DNN and hidden Markov model.

Lella *et al.* [31] used a 1D-CNN to detect COVID-19 on the basis of voice, breath, and cough sounds. They input MFCC data into an autoencoder to extract deep features, which were then classified using a single-branch 1D-CNN. The aforementioned authors achieved a classification accuracy of 90%. Amrulloh *et al.* [20] distinguished asthma and pneumonia by using three features: the MFCC, Shannon entropy, and non-Gaussianity score. They fed these features to a DNN and achieved a sensitivity of 89%.

Feng *et al.* [32] detected COVID-19 on the basis of recorded cough sounds. They obtained time- and frequency-domain features from each sound. They authors achieved a maximum classification accuracy of 99.56% with a recurrent neural network. Islam *et al.* [33] used the chromagram feature to detect COVID-19. They compared the performance of a CNN and DNN in COVID-19 detection and found that the CNN had higher accuracy than did the DNN.

III. METHODOLOGY

This section describes the methodology used in this study for constructing the designed system. The system design framework is illustrated in Fig. 1, and it contains seven key parts: dataset collection, data enhancement, zero padding, emergent feature analysis, feature structure design, restructuring of multiple features, and DL.

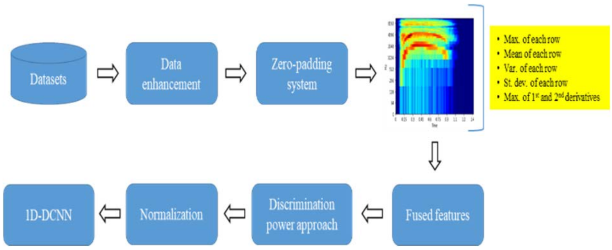


FIGURE 1. System design framework.

Two datasets were used in this study. The first dataset comprised 189 files of cough audio [34]. Cough signals were manually segmented using Audacity software [55] to obtain 170 dry cough sounds and 118 wet cough sounds for this dataset. Similarly, the second dataset comprised 222 recordings of cough audio [35]. Segmentation of cough signals identified 389 wet coughs and 413 dry coughs in this dataset. The details of data collection and enhancement are presented in Table 1. The two datasets were preprocessed using techniques that were similar to those performed in [14], with the only difference being that signals were not resampled in the present study.

TABLE 1. Details regarding the collected datasets and data enhancement.

	Data Type	Seg[a]. QTY	Duration (s)	Enhc[b]. QTY	Total QTY
Dataset 1	Wet(100)	118	40.44	472	590
	Dry(89)	170	64.43	680	850
	Total(189)	288	104.87	982	1440
Dataset 2	Wet(111)	389	196.17	1556	1945
	Dry(111)	413	202.64	1652	2065
	Total(222)	802	398.81	3208	4010

[a]Seg = segmentation, [b]Enhc = enhancement, QTY = quantity. The numbers inside the parentheses indicates the numbers of cough sound samples selected from the specified dataset.

A. DATA ENHANCEMENT

The basic concept of data enhancement in machine learning involves increasing the quantity of training data; however, data enhancement also can be performed to enrich data in a dataset [27]. Data enhancement can be performed using two approaches: image- and audio-based approaches. The audio-based approach was used in the present study. Two strategies were used in this study to enhance the quantity of data: time stretch and pitch shift. In the time stretch method, we stretched the duration of cough signals by factors of 1.07 and 0.5. The factor of 1.07 was used to accelerate a cough signal, and the factor of 0.5 was used to decelerate a cough signal. Pitch shift was performed using factors similar to those used in [27].

The results indicated that after data enhancement, the numbers of dry and wet cough signals in dataset 1 increased from 170 to 850 and from 118 to 590, respectively. Moreover, the numbers of dry and wet cough signals in dataset 2 increased from 413 to 2065 and from 389 to 1945, respectively. Overall, the total number of cough signals increased from 288 to 1440 for the first dataset and from 802 to 4010 for the second dataset.

B. PADDING SYSTEM

A padding system is used to overcome the problem of multivariable bit lengths of cough signals in a dataset. In this study, the bit lengths of the signals with short bit lengths were increased to the maximum value. Thus, a fixed bit length was achieved for all the cough signals (bit length is the size of a signal). Inspired by the random padding technique proposed by Dong *et al.* [36], we created a zero-padding system instead of a random padding system. The procedures for creating a zero-padding system are described in the following text.

Consider the example of dataset 2. First, we determined that the maximum bit length in this dataset was 45 982, approximately 2.08 s, when the sample rate was set as 22050 Hz. Second, we calculated the bit length of each cough signal in the database. Subsequently, we determined the difference (N) between the maximum bit length and the bit length of the signal. Third, the zero () function was used to obtain a zero array of size N. Finally, the append () function was used to copy the current signal sample with N zero values until the maximum bit length was attained.

The experimental results indicated that after padding, the dimensions of each signal in a dataset became the same. The dimensions of signals in datasets 1 and 2 were 29 952 and 45 982, respectively. Fig. 2 displays an example of a cough signal obtained before and after zero padding on dataset 2.

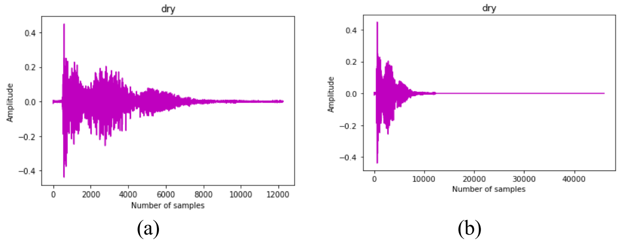


FIGURE 2. Example dry cough waveforms from dataset 2: (a) original waveform and (b) waveform obtained after zero padding.

C. FEATURE EXTRACTION

The following methods were used for feature extraction: the Mel spectrogram, MFCC, chromagram constant-Q transform (CQT), constant-Q cepstral coefficient (CQCC), and linear predictive code coefficient (LPC) [37]. Table 2 presents the parameter settings for the conversion of cough signals into spectrograms and LPCs. In contrast to [38] and [39], we extracted the maximum, mean, variance, and standard deviation values of the original spectrogram as well as the maximum first and second derivatives of the original spectrogram.

TABLE 2. Parameters Setting.

	Parameters	Practical values
General	Window size	882
	Window name	Hann
	Number of FFT	882
	Number of overlapping	220
	Window size	882
	Sampling frequency	22050 Hz
	Maximum frequency	11025 Hz
LPC	Order	26
Chroma	Bins per octave	40
	Number of chroma	20
	Number of overlapping	512
CQCC	Bins per octave	12
	Number of bins	84
	Number of overlapping	512
MFCC	HTK	True
	Number of MFCC	40
MELSPEC	HTK	True
	Number of MEL	64

Essentially, a spectrogram is obtained after four steps: pre-emphasis, framing, windowing, and short-time Fourier transform (STFT). The spectrogram $S(n,k)$ [40] is the squared magnitude of $X\ (n,k)$, which is expressed as follows:

$$X\ (n,k) = \int_{-T}^{T} x\ (\tau) \cdot w(\tau - k) \cdot exp\,(-j2\pi n\tau) d\tau \quad (1)$$

where $x(t)$ is the cough signal, n is the Fourier coefficient, k is the time frame, $w(t)$ is the windowing function, and $X(n,k)$ is the STFT in the complex number.

$$S\ (n,k) = |X\ (n,k)|^2 \quad (2)$$

1) MEL SPECTROGRAM

A Mel spectrogram (MELSPEC) is an auditory system derived by passing a cough signal through an STFT filter and a Mel filter bank [41]. A Mel spectrogram is expressed as follows [42], [43], [44]:

$$M\ (m,k) = \sum_{m} |X\ (n,k)|^2 \cdot \Delta(m) \quad (3)$$

where $M(m,k)$ is the generated Mel spectrogram, and $\Delta(m)$ represents a triangular Mel filter bank with m Mel-frequency bands. The Mel frequency is calculated using the following formula:

$$Mel\ freq = 2595 \times\ log_{10}(1 + \frac{f}{700}) \quad (4)$$

In a Mel spectrogram, cough intensity bands are represented equally in Mel frequencies; thus, capturing different attributes from each frequency band will provide interesting results. Table 3 presents the code procedures used to obtain attributes from a Mel spectrogram.

TABLE 3. Code for extracting emergent features from a Mel spectrogram.

```
Input: cough signals,
Output: a single feature vectors.
1.   for $k$ in range of dataset shape:
2.       Compute Mel spectrogram
3.       Compute first and second derivatives of Mel spectrogram
4.   for $j$ in range of dataset shape:
5.       $MEL = M(m,k)[j]$
6.       $MEL' = M(m,k)'[j]$
7.       $MEL'' = M(m,k)''[j]$
8.       mx = $[np.\max(MEL(i)\ \textbf{for}\ i\ in\ range(MEL.shape[0])]$
9.       mn = $[np.\text{mean}(MEL(i)\ \textbf{for}\ i\ in\ range(MEL.shape[0])]$
10.      var = $[np.\text{var}(MEL(i)\ \textbf{for}\ i\ in\ range(MEL.shape[0])]$
11.      std = $[np.\text{std}(MEL(i)\ \textbf{for}\ i\ in\ range(MEL.shape[0])]$
12.      $mx' = [np.\max(MEL'(i)\ \textbf{for}\ i\ in\ range(MEL'.shape[0])]$
13.      $mx'' = [np.\max(MEL''(i)\textbf{for}\ i\ in\ range(MEL''.shape[0])]$
14.      Fuse the results (8-13).
```

In dataset 1, the Mel spectrogram $M(m,k)$ and its derivatives ($M(m,k)'$ and $M(m,k)''$) were computed for each cough signal. Subsequently, for each computed Mel spectrogram, the vectors of the maximum spectral intensity, mean spectral intensity, spectral intensity variance, and standard deviation of the spectral intensity were extracted from each frequency band. The aforementioned process was also used to extract maximum spectral intensity vectors from the derivatives of a Mel spectrogram. The results obtained from a Mel spectrogram and its derivatives were then fused. The fused feature was a single-feature vector. The resulting shape of a single-feature vector obtained from a Mel spectrogram for dataset 1 was (1440,384), where 1440 is the quantity of data in the dataset, and 384 is the length of a single-feature vector.

2) MEL-FREQUENCY CEPSTRAL COEFFICIENT

The MFCC is one of the most crucial features for speech recognition, and MFCC is based on the power spectrum. The MFCC is typically calculated after passing a cough signal through an STFT filter, a Mel filter bank, and a discrete cosine transform filter [45]. The MFCC is calculated using the following equation:

$$MFCC\,(m,k) = \sum_{z} log(|X\,(n,k)|^2 \cdot \Delta(m)) \cdot cos\left[m(z-\frac{1}{2})\frac{\pi}{z}\right] \qquad (5)$$

where $M(m, k)$ is a matrix in which the row (m) represents the MFCC and the column (k) represents the time frame. In this study, the attributes were acquired from each row of $M\ (m, k)$ matrix. The following text describes how various attributes from the MFCC were captured. For each cough sound, the MFCC and its derivatives were generated.

Vectors of the maximum cepstral coefficient, mean cepstral coefficient, cepstral coefficient variance, and standard deviation of the cepstral coefficient were captured from each row of the MFCC. Moreover, the maximum cepstral coefficient was extracted from each derivative of the MFCC. The results attained from MFCC and its derivatives were combined, and the shape of a single-feature vector obtained from the MFCC was (1440,240).

3) CHROMAGRAM

A chromagram (CHROMA) is a feature used to examine pitch characteristics in music. A chromagram is a pitch class profile and can be used to distinguish different types of cough sounds because the cough signals of different patients have different amplitudes; thus, transforming cough sounds into a chromagram can indicate how cough energy is distributed among different pitch classes. Three types of chromagrams exist [37]: the CHROMA-STFT, CHROMA-CQT, and CHROMA-energy normalized (CHROMA-EN). The CHROMA-STFT is generated through STFT, which includes a linear frequency scale. The CHROMA-CQT is generated using the CQT, which contains a logarithmic frequency scale [46]. The energy is normalized in the CHROMA-EN. The CHROMA-CQT was adopted in the present study.

A CHROMA [47] is usually a 2D matrix with pitch classes in the rows and the time frames in the columns. We intended to determine the features in the pitch classes; therefore, we employed a method similar to MELSPEC. First, the CHROMA-CQT and its derivatives were generated. Second, the maximum, mean, variance, and standard deviation of the CHROMA magnitude as well as the maximum magnitude of the derivatives of the CHROMA-CQT were generated. The overall shape of a single-feature vector developed using the CHROMA-CQT was (1440,120).

4) CONSTANT-Q CEPSTRAL COEFFICIENT

The CQCC was developed for automatic speaker verification. It has also been applied to distinguish between patients with asthma and healthy people [48]. The CQCC is determined using three steps: (1) the CQT is calculated, after which the amplitude of CQT is converted into decibels; (2) the MFCC is used to obtain a 2D CQCC; and (3) emergent features are extracted. The aforementioned steps are described in Table 4. The shape of a single-feature vector obtained from the CQCC was (1440, 240) in this study.

TABLE 4. Code for extracting emergent features from the CQCC.

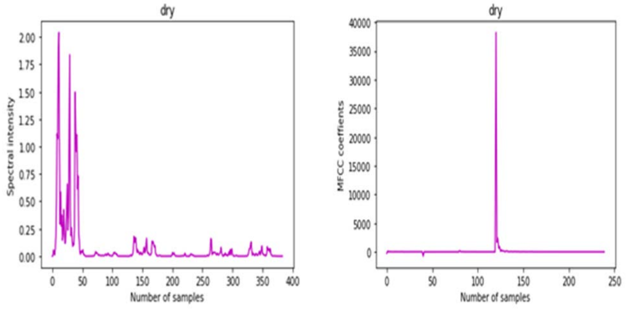
```
Input: cough signals,
Output: a single feature vectors.

1.   for k in range of dataset shape:
2.       Compute Constant-Q transform (CQT)
3.       Compute the magnitude of CQT
4.       Using results in (3) and transform to dB using amplitude to dB conversion
5.       Using results in (4), compute MFCC as a result CQCC
6.       Compute the first and second derivatives of CQCC.
7.   for j in range of dataset shape:
8.       $CQ = CQCC(m,k)[j]$
9.       $CQ' = CQCC(m,k)'[j]$
10.      $CQ'' = CQCC(m,k)''[j]$
11.      $\mathrm{mx} = [np.\max(CQ(i)\ \mathbf{for}\ i\ in\ range(CQ.shape[0])]$
12.      $\mathrm{mn} = [np.\mathrm{mean}(CQ(i)\ \mathbf{for}\ i\ in\ range(CQ.shape[0])]$
13.      $\mathrm{var} = [np.\mathrm{var}(CQ(i)\ \mathbf{for}\ i\ in\ range(CQ.shape[0])]$
14.      $\mathrm{std} = [np.\mathrm{std}(CQ(i)\ \mathbf{for}\ i\ in\ range(CQ.shape[0])]$
15.      $mx' = [np.\max(CQ'(i)\ \mathbf{for}\ i\ in\ range(CQ'.shape[0])]$
16.      $mx'' = [np.\max(CQ''(i)\ \mathbf{for}\ i\ in\ range(CQ''.shape[0])]$
17.      Fuse the results (11-16).
```

5) LINEAR PREDICTIVE CODE COEFFICIENT

The LPC is a vocal tract feature used to characterize the spectral envelope of a speech signal. This coefficient has been used for classifying cough sounds [49], [50], with suitable results. After extracting the LPC [37], [51], its first and second derivatives are calculated. Subsequently, all the computed features are fused to obtain a single LPC feature. The shape of a single-LPC-feature vector generated in this study was (1440,81). Fig. 3 illustrates the features extracted using some spectrogram methods in this study.

D. PROPOSED FEATURE STRUCTURES AND METHODS FOR RESTRUCTURING COMBINED FEATURES

This section describes the proposed feature structures and techniques for restructuring combined features. The proposed



(a) (b)

FIGURE 3. Feature extraction results obtained for dry cough signals from dataset 2: (a) MELSPEC and (b) MFCC.

feature structures are single features and combined features. As described in section I, a single feature is a feature obtained after fusing the subfeatures extracted using different spectrogram methods. By contrast, a combined feature is obtained after combining the single features extracted using different spectrogram methods.

The structure of a single feature for dataset 1 is explained as follows:

- The length of a single feature of a Mel spectrogram is 384.

$$f_1 = \{a_1, a_2, a_3, \ldots, a_{384}\} \tag{6}$$

- The length of a single feature of the MFCC is 240.

$$f_2 = \{b_1, b_2, b_3, \ldots, b_{240}\} \tag{7}$$

- The length of a single feature of the CHROMA-CQT is 120.

$$f_3 = \{c_1, c_2, c_3, \ldots, c_{120}\} \tag{8}$$

- The length of a single feature of the CQCC is 240.

$$f_4 = \{d_1, d_2, d_3, \ldots, d_{240}\} \tag{9}$$

- The length of a single feature of the LPC is 81.

$$f_5 = \{e_1, e_2, e_3, \ldots, e_{81}\} \tag{10}$$

Moreover, the feature combination F is expressed as follows:

$$F = \{a_1, .., a_{384}, b_1, .., b_{240}, c_1, .., c_{120}, d_1, .., d_{240}, e_1, .., e_{81}\} \tag{11}$$

The overall structure of a feature combination is expressed as follows:

$$F = \{k_1, k_2, .., k_{1065}\} \tag{12}$$

where; $a, b, c, d, e, k \in \mathbb{R}$

After determining a single-feature set f and feature combination F, we restructured the position of multi-combining features in f and F. The combined features were restructured according to their discrimination power (mean absolute deviation) and mutual information value. The two restructuring techniques are detailed in Table 5. These techniques were analogous, with the difference being that the mean absolute deviation was calculated using (13), whereas the mutual information [52], [53] was calculated using (14).

TABLE 5. Code for the restructuring of combined features.

Input: feature sets in NumPy array
Output: features at different positions.

1. Convert feature sets *f* or *F* to DataFrame.
2. Compute the mean absolute deviation/mutual information value.
3. Determine the indices of each column.
4. Use feature sets f or F to generate the DataFrame with column names using the calculated indices in (3).
5. Sort the DataFrame in ascending order using the column index.

$$MAD(t) = \frac{1}{n} \cdot \sum_{t=1}^{n} |x_t - x_{av}| \tag{13}$$

$$M\,(X, Y) = \iint p(x, y) \ln \frac{p(x, y)}{p\,(x)\,p(y)} dxdy \tag{14}$$

where n is the number of data points, x_t is the value of each data point in a series, x_{av} is an average value of the data, $MAD(t)$ is the mean absolute deviation of the data, $p(x, y)$ is the joint probability of variables x and y, $p(x)$ is the probability of variable x, and $p(y)$ is the probability of variable y.

IV. PROPOSED 1D-DCNN

The main DL architecture used in this study was a parallel-stream 1D-DCNN. The performance of this network was compared with that of a single-stream 1D-DCNN. Both the aforementioned networks exhibited the basic structure of a CNN, which comprises an input layer, a hidden layer, and an output layer. The parallel- and single-stream 1D-DCNNs were constructed using a Keras library and executed in TensorFlow-GPU. The aforementioned networks are described in the following text.

A. SINGLE-STREAM 1D-DCNN

As depicted in Fig. 4, the constructed single-stream 1D-DCNN contained one input layer, three convolutional layers, one flattening layer, one dense layer, and one output layer. The first convolutional layer of this network used the *regularizer* l2 (0.001) kernel. The rectified linear unit activation function was used in all the layers except the last layer, in which the *softmax* activation function was used. Each convolutional layer had a stride of 1 and the same padding. After the dense layer, a 50% dropout was used.

B. PARALLEL-STREAM 1D-DCNN

The architecture of the constructed parallel-stream 1D-DCNN was similar to that of the constructed single-stream 1D-CNN;

however, in contrast to the single-stream 1D-CNN, the parallel-stream network included multiple streams in parallel. Each stream comprised a feature set (input layer), three convolutional layers, one flattening layer, one merged layer, one dense layer, and one output layer (Fig. 4).

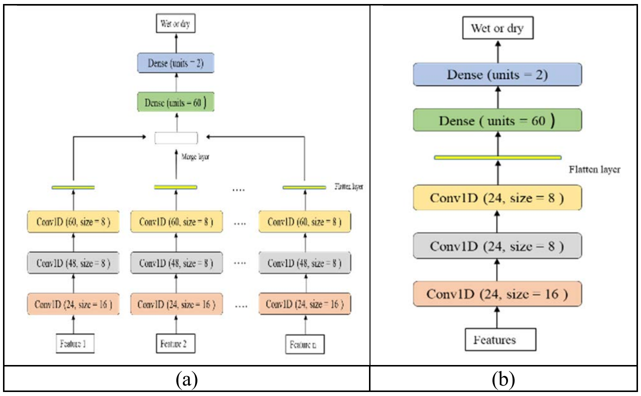


FIGURE 4. Proposed 1D-DCNN: (a) parallel-stream network and (b) single-stream network.

When training the two networks, the number of epochs was set as 50, and the batch size was set as 32. The networks were optimized using the *root mean squared propagation (RMSProp)* optimizer; their loss function was the categorical cross-entropy function; and their performance was evaluated in terms of their accuracy.

C. EXPERIMENTAL SETUP AND EVALUATION METRICS

In the experiments, extracted spectrogram features were normalized using the z-score method. Subsequently, two steps were conducted to split the datasets. First, 80% and 20% of each dataset were randomly divided into a training set and testing set, respectively. Second, 80% and 20% of the training data were further divided into a training set and validation set, respectively. Finally, the $F1$ score and MCC were determined. The confusion matrix was used to evaluate the prediction for each cough category.

The weighted $F1$ score [25] is expressed as follows:

$$\text{F1 score} = \frac{TP}{TP + 0.5(FP + FN)} \times w_i \tag{15}$$

The MCC [54] is expressed as follows:

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{16}$$

where TP indicates the number of true positives, TN indicates the number of true negatives, FP indicates the number of false positives, FN indicates the number false negatives, and w_i is the weight ratio of class i.

D. HARDWARE AND SOFTWARE

The hardware used in this study was a desktop with an Intel CoreTM i7-10700 CPU @2.9 GHz with 16 GB RAM, an Nvidia GeForce GTX 2060 graphics card with 6 GB VRAM, and a 1-TB hard disk drive. Audacity version 3.1.3 was used for signal segmentation in this study. Audacity is a multifunctional tool that enables users to import, edit, export, and record audio files [55]. A Librosa library [37] was used to analyze cough signals through processes such as audio wave loading and spectrogram extraction.

V. RESULTS AND DISCUSSION

A. COMPARISON OF DIFFERENT 1D DL MODELS

A simple exploratory study was conducted to compare the performance of a 1D-CNN model, 1D-DNN model, and GRU model. Each of these models comprised only one input layer and one output layer. The 1D-DCNN model comprised two convolutional layers and one flattening layer. For this model, the kernel size was 10, the number of filters was 32, and the number of units in the dense layer was 64. The 1D-DNN model contained two dense layers with 32 units each. The GRU model contained two recurrent neural network layers and a flattening layer. It had the same number of units as did the 1D-DNN model. As presented in Table 6, the 1D-DCNN, 1D-DNN, and GRU models achieved $F1$ scores of 97.21%, 96.52%, and 96.51%, respectively. Because the 1D-DCNN model exhibited the highest $F1$ score, this model was selected for further exploration.

TABLE 6. ***F1*** **score and MCC values obtained for different 1D DL models.**

Network	Feature set	F1-score (%)	MCC
1D-DCNN	CHROMA+LPC+MFCC	97.21	0.942
1D-DNN	CHROMA+LPC+MFCC	96.52	0.928
GRU	CHROMA+LPC+MFCC	96.51	0.928

B. COMPARISON OF THE METHODS USED FOR RESTRUCTURING COMBINED FEATURES

Two methods were used in this study for restructuring combined features, and the results obtained with these methods are presented in Table 7. The restructuring method based on discrimination power outperformed that based on the mutual information value. The $F1$ score obtained with the method based on discrimination power was 1.4% and 1.39% higher than that based on the mutual information value for two feature sets. Therefore, the restructuring method based on discrimination power was selected for further analysis.

C. RESULTS OBTAINED FOR THE SINGLE-STREAM 1D-DCNN FOR DIFFERENT DATASETS

Single-feature vectors extracted from all the spectrograms were aggregated, and the method based on discrimination power was used to restructure combined features. Subsequently, these features were input to a single-stream 1D-DCNN. The classification results of this network were

TABLE 7. Results obtained for the two methods used in this study for restructuring combined features.

Methods	Feature sets	F1-score (%)	MCC
Discrimination power	CHROMA+LPC+MFCC	97.57	0.949
	MFCC+LPC+MELSPEC+CHROMA	98.26	0.964
Mutual information	CHROMA+LPC+MFCC	96.17	0.920
	MFCC+LPC+MELSPEC+CHROMA	96.87	0.935

TABLE 8. Classification results obtained with the single-stream 1D-DCNN for different feature sets and datasets.

Model/ Dataset	Feature sets	F1-score (%)	MCC
Single-stream /Dataset 1	MFCC+LPC	97.22	0.942
	MELSPEC+LPC	95.82	0.913
	CHROMA+LPC+MFCC	97.57	0.949
	MFCC+LPC+MELSPEC	98.26	0.964
	CHROMA+LPC+MELSPEC	95.48	0.906
	MFCC+LPC+MELSPEC+CHROMA	98.26	0.964
	MFCC+LPC+MELSPEC+ CHROMA+CQCC	97.91	0.957
Single-stream /Dataset 2	MFCC+LPC	81.24	0.626
	MELSPEC+LPC	81.92	0.638
	CHROMA+LPC+MFCC	80.63	0.613
	MFCC+LPC+MELSPEC	83.53	0.67
	CHROMA+LPC+MELSPEC	71.65	0.433
	MFCC+LPC+MELSPEC+CHROMA	80.28	0.605
	MFCC+LPC+MELSPEC+ CHROMA+CQCC	80.02	0.601

evaluated using the $F1$ score and MCC. Table 8 presents the performance of 1D-DCNN for various feature sets and the two datasets. As presented in Table 8, the results obtained for the feature sets differed according to the dataset.

For dataset 1, the MFCC + LPC + MELSPEC and MFCC + LPC + MELSPEC + CHROMA feature sets exhibited the same classification results. The $F1$ score and MCC for the two feature sets were 98.26% and 0.964, respectively. In addition, lower performance was achieved for the MELSPEC + LPC and CHROMA + LPC + MELSPEC feature sets than for the aforementioned two feature sets. The $F1$ scores and MCCs of the MELSPEC + LPC and CHROMA + LPC + MELSPEC feature sets differed by 0.34% and 0.007, respectively. For dataset 2, the classification results obtained for the MFCC + LPC + MELSPEC feature set were superior to those obtained for the other feature sets. An $F1$ score of 83.53% and an MCC of 0.67 were obtained for the aforementioned feature set. Poor classification results were obtained for the CHROMA + LPC + MELSPEC feature set, with the $F1$ score and MCC being 71.65% and 0.433, respectively.

The classification results obtained for dataset 2 were inferior to those obtained for dataset 1. To determine the reason for the inferior classification performance for dataset 2, we performed a quick review of previous studies conducted using this dataset. Poor classification results were obtained for dataset 2 in [56] and [57].

D. RESULTS OBTAINED FOR THE PARALLEL-STREAM 1D-DCNN FOR DIFFERENT DATASETS

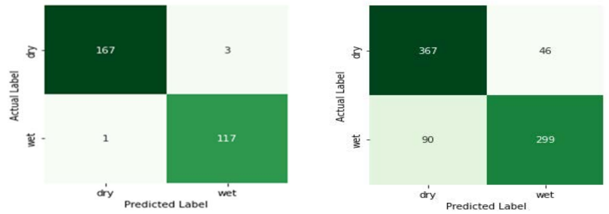
In the parallel-stream 1D-DCNN, every single-feature vector was simultaneously fed to different inputs, and the features extracted from parallel streams were subsequently concatenated to form a merged layer. The resulting features were passed through a dense layer before being classified at the output layer (Fig. 4). The classification results obtained for the parallel-stream network are presented in Table 9.

TABLE 9. Classification results obtained with the parallel-stream 1D-DCNN for different feature sets and datasets.

Model/ Dataset	Feature sets	F1-score (%)	MCC
Parallel-stream /Dataset 1	MFCC+LPC	98.61	0.971
	MELSPEC+LPC	95.12	0.899
	CHROMA+LPC+MFCC	97.22	0.943
	MFCC+LPC+MELSPEC	97.91	0.957
	CHROMA+LPC+MELSPEC	93.37	0.863
	MFCC+LPC+MELSPEC+CHROMA	98.61	0.971
	MFCC+LPC+MELSPEC+ CHROMA+CQCC	97.91	0.957
Parallel-stream /Dataset 2	MFCC+LPC	81.88	0.638
	MELSPEC+LPC	82.28	0.645
	CHROMA+LPC+MFCC	81.39	0.628
	MFCC+LPC+MELSPEC	81.27	0.625
	CHROMA+LPC+MELSPEC	68.93	0.378
	MFCC+LPC+MELSPEC+CHROMA	81.78	0.635
	MFCC+LPC+MELSPEC+ CHROMA+CQCC	82.96	0.663

For dataset 1, the best classification results were obtained for the MFCC + LPC and MFCC + LPC + MELSPEC + CHROMA feature sets. The $F1$ score and MCC for these feature sets were 98.61% and 0.971, respectively. An $F1$ score of 97.91% was obtained for the MFCC + LPC + MELSPEC and MFCC + LPC + MELSPEC + CHROMA + CQCC feature sets. Moreover, the worst classification results were observed for the CHROMA + LPC + MELSPEC feature set, with the $F1$ score being 93.37% and the MCC being 0.863. Fig. 5(a) shows the confusion matrix of the feature set for which the best classification results were obtained. Dataset 1 contained 170 dry cough sounds and 118 wet cough sounds. The classification results of the proposed parallel-stream network contained three false positives and one false negative. For dataset 1, the aforementioned network predicted 168 cough signals as dry coughs and 120 cough signals as wet coughs.

For dataset 2, the best classification results were obtained for the MFCC + LPC + MELSPEC + CHROMA + CQCC feature set. The $F1$ score and MCC for this feature set were 82.96% and 0.663, respectively. The worst classification



(a) (b)

FIGURE 5. Confusion matrix of the feature sets for which the proposed parallel-stream 1D-DCNN exhibited the best classification results: (a) dataset 1 and (b) dataset 2.

results were obtained for the CHROMA + LPC + MELSPEC feature set, with the $F1$ score being 68.93% and the MCC being 0.378. Fig. 5(b) displays the confusion matrix of the feature set for which the best classification results were obtained. Dataset 2 contained 413 and 389 dry and wet cough sounds, respectively. The proposed network predicted 46 dry cough signals as wet cough signals and 90 wet cough signals as dry cough signals.

E. COMPARISON OF THE PERFORMANCE OF THE SINGLE-STREAM 1D-DCNN AND PARALLEL-STREAM 1D-DCNN

The performance, execution times, and number of parameters of the single-stream 1D-DCNN and parallel-stream 1D-DCNN were compared. For dataset 1, the parallel-stream network exhibited the best classification results for two feature sets, with the $F1$ score being 98.61% and the MCC being 0.971. Nevertheless, the single-stream network exhibited excellent classification results for most feature sets. In addition, when using all five features (MFCC + LPC + MELSPEC + CHROMA + CQCC), the single- and parallel-stream networks categorized cough sounds in 386.18 and 481.08 s, respectively. For dataset 2, the parallel-stream network exhibited superior classification results to the single-stream network for most feature sets. Moreover, when using all five features, the single and parallel-stream networks classified cough sounds in 864.21 and 1445.87 s, respectively (Fig. 6).

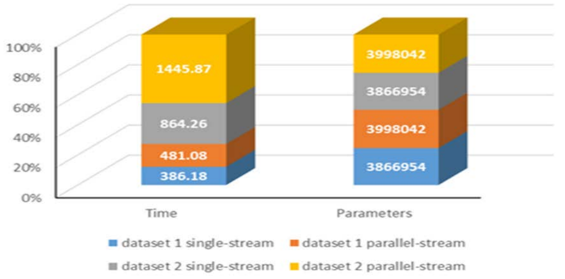


FIGURE 6. Execution time of and training parameters required by the constructed parallel- and single-stream networks for different datasets.

Overall, the single-stream network required fewer training parameters than did the parallel-stream network (3 866 954 vs. 3 998 042). The results presented in Tables 8 and 9 indicate that (1) training multiple parallel networks concurrently does not guarantee excellent classification results, but selection of input features are important, and that (2) the simultaneous aggregation of many features in a single-stream network does not result in high performance.

F. CLASSIFICATION RESULTS OBTAINED UNDER THE CONCATENATION OF LAYERS AT DIFFERENT LEVELS

We examined whether the classification performance of the constructed parallel-stream network could be improved by concatenating layers at different levels. We trained the proposed parallel-stream network [Fig. 4(b)] and then modified the network by performing concatenation at the second and third convolutional layers. The MFCC + CQCC feature set was selected as the input of the aforementioned network because of the similarities in the dimensions of these features. Better classification results were obtained when concatenating layers at the flattening level than when concatenating layers at other levels. When concatenating layers at the flattening level, the $F1$ score was 99.30%, and the MCC was 0.985 (Table 10).

TABLE 10. Classification results obtained when concatenating layers at different levels using dataset 1.

Level	F1-score (%)	MCC
Second convolutional	98.26	0.964
Third convolutional	97.57	0.949
Flattening	99.30	0.985

G. CLASSIFICATION RESULTS OBTAINED WHEN USING DIFFERENT STRATEGIES FOR MERGING LAYERS

We compared the classification results obtained with the proposed parallel-stream network when using different strategies for merging layers, such as addition, multiplication, maximization, and concatenation. Layer merging involves combining two or more models or layers. Four layer merging strategies were adopted at different levels in this study (Table 11). At the flattening level, excellent classification

TABLE 11. Classification results obtained when adopting different layer merging strategies at different levels using dataset 1.

Level	Strategy	F1-score (%)	MCC
Second convolutional	Addition	97.92	0.957
	Multiplication	97.21	0.942
	Maximum	98.61	0.971
	concatenation	98.26	0.964
Flattening	Addition	95.50	0.909
	Multiplication	96.15	0.922
	Maximum	97.23	0.944
	concatenation	99.30	0.985

results were obtained when using the concatenation strategy. Moreover, better classification results were obtained at the second convolutional level than at the flattening level for three of the four adopted strategies (i.e., the addition, multiplication, and maximization strategies). The maximization strategy exhibited the best results ($F1$ score of 98.61%) at the second convolutional level, whereas the concatenation strategy exhibited the best results ($F1$ score of 99.30%) at the flattening level.

VI. CONCLUSION

In this study, cough sounds were classified into wet and dry coughs through the analysis of the emergent features extracted from spectrograms and a parallel-stream 1D-DCNN. Two datasets were used in this study. Data enhancement was conducted to increase the quantity of data in each dataset, and each cough signal was then padded using a zero-padding system to generate cough signals with fixed dimensions. After using the zero-padding system, the cough signals of datasets 1 and 2 had fixed dimensions of 29 952 and 45 982, respectively.

Numerous attributes were extracted from each row of original spectrograms and their derivatives and then fused. The MELSPEC, CHROMA, MFCC, CQCC, and LPC spectrograms were used in this study. We obtained two types of attributes: single attributes and combined attributes. We then examined the performance of two techniques designed in this study to restructure combined attributes. One of these techniques was based on features with high discrimination power, whereas the other was based on features with high mutual information values. The approach based on discrimination power exhibited 1.4% and 1.39% higher $F1$ scores than did the approach based on the mutual value when two feature sets were input to the proposed parallel-stream 1D-DCNN.

Before developing the proposed parallel-stream 1D-DCNN, we conducted a simple study on three models: 1D-DCNN, GRU model, and 1D-DNN. The 1D-DCNN outperformed the other two models; thus, this model was adopted for further analysis in this study. We compared the performance of the proposed parallel-stream 1D-DCNN with that of a single-stream 1D-DCNN.

We then compared the parallel-stream network and a single-stream network. For dataset 1, the highest $F1$ score exhibited by the parallel-stream network was 98.61%. However, the parallel-stream network outperformed the single-stream network for only a few feature sets. Moreover, the confusion matrix of the feature set for which the best classification performance was obtained indicated that the numbers of false positives and false negatives were low.

For dataset 2, the best classification results were obtained for the MFCC + LPC + MELSPEC + CHROMA + CQCC feature set ($F1$ score of 82.96% and MCC of 0.663). The confusion matrix of this feature set contained large numbers of false positives and false negatives. The number of false negatives was almost twice that of false positives. In some cases, the parallel-stream network exhibited excellent classification performance but required a long classification time and numerous training parameters.

We also compared the classification performance achieved when concatenating layers at different levels. When the MFCC + CQCC feature set was used, better classification results were obtained when concatenating layers at the flattening level ($F1$ score of 99.30%) than at other levels.

Finally, the classification performance of the proposed parallel-stream network was examined under four layer merging strategies: addition, multiplication, concatenation, and maximization. These strategies were implemented at two levels: the second convolutional level and flattening level. Better classification results were obtained for all the layer merging strategies, except the concatenation strategy, at the second convolutional level than at the flattening level. The best classification results were obtained with the maximization and concatenation strategies at the second convolutional and flattening levels, respectively.

In the future, we will develop a transfer learning algorithm that can execute the spectrogram methods used in this study. The performance of this algorithm will then be compared with the parallel-stream network designed in this study. Moreover, a novel method will be adopted to increase the quantity of data, and this method will be compared with the data enhancement method adopted in this study.

REFERENCES

[1] C. Infante, D. B. Chamberlain, R. Kodgule, and R. R. Fletcher, "Classification of voluntary coughs applied to the screening of respiratory disease," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 1413–1416.

[2] S. Ranjani, V. Santhiya, and A. Jayapreetha, "A real time cough monitor for classification of various pulmonary diseases," in *Proc. 3rd Int. Conf. Emerg. Appl. Inf. Technol.*, Nov. 2012, pp. 102–105.

[3] A. N. Belkacem, S. Ouhbi, A. Lakas, E. Benkhelifa, and C. Chen, "End-to-end AI-based point-of-care diagnosis system for classifying respiratory illnesses and early detection of COVID-19: A theoretical framework," *Frontiers Med.*, vol. 8, pp. 1–14, Mar. 2021.

[4] H. Chatrzarrin, A. Arcelus, R. Goubran, and F. Knoefel, "Feature extraction for the differentiation of dry and wet cough sounds," in *Proc. IEEE Int. Symp. Med. Meas. Appl.*, May 2011, pp. 162–166.

[5] L. Kvapilova, V. Boza, P. Dubec, M. Majernik, J. Bogar, J. Jamison, J. C. Goldsack, D. J. Kimmel, and D. R. Karlin, "Continuous sound collection using smartphones and machine learning to measure cough," *Digit. Biomarkers*, vol. 3, no. 3, pp. 166–175, Dec. 2019.

[6] U. R. Abeyratne, V. Swarnkar, A. Setyati, and R. Triasih, "Cough sound analysis can rapidly diagnose childhood pneumonia," *Ann. Biomed. Eng.*, vol. 41, no. 11, pp. 2448–2462, Nov. 2013.

[7] N. Hogan and R. W. Mann, "Myoelectric signal processing: Optimal estimation applied to electromyography—Part I: Derivation of the optimal myoprocessor," *IEEE Trans. Biomed. Eng.*, vol. BME-27, no. 7, pp. 382–395, Jul. 1980.

[8] A.-C. Tsai, T.-H. Hsieh, J.-J. Luh, and T.-T. Lin, "A comparison of upper-limb motion pattern recognition using EMG signals during dynamic and isometric muscle contractions," *Biomed. Signal Process. Control*, vol. 11, pp. 17–26, May 2014.

[9] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, "Classification of the myoelectric signal using time-frequency based representations," *Med. Eng. Phys.*, vol. 21, nos. 6–7, pp. 431–438, Jul. 1999.

[10] A. Z. B. Sha'ameri and T. J. Lynn, "Spectrogram time-frequency analysis and classification of digital modulation signals," in *Proc. IEEE Int. Conf. Telecommun. Malaysia Int. Conf. Commun.*, May 2007, pp. 113–118.

[11] K. Wang, J. Li, S. Zhang, Y. Qiu, and R. Liao, "Time-frequency features extraction and classification of partial discharge UHF signals," in *Proc. Int. Conf. Inf. Sci., Electron. Electr. Eng.*, Apr. 2014, pp. 1231–1235.

[12] G. Yu and J.-J. Slotine, "Audio classification from time-frequency texture," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 1677–1680.

[13] J. Niu, Y. Shi, M. Cai, Z. Cao, D. Wang, Z. Zhang, and X. D. Zhang, "Detection of sputum by interpreting the time-frequency distribution of respiratory sound signal using image processing techniques," *Bioinformatics*, vol. 34, no. 5, pp. 820–827, Mar. 2018.

[14] R. Mushi, Y.-P. Huang, and T.-W. Chang, "Automatic classification of wet and dry cough sounds," in *Proc. Int. Conf. Syst. Sci. Eng.* Taichung, Taiwan, May 2022, pp. 1–5.

[15] H. Gunduz, "Deep learning-based Parkinson's disease classification using vocal feature sets," *IEEE Access*, vol. 7, pp. 115540–115551, 2019.

[16] Q. Zhou, J. Shan, W. Ding, C. Wang, S. Yuan, F. Sun, H. Li, and B. Fang, "Cough recognition based on mel-spectrogram and convolutional neural network," *Frontiers Robot. AI*, vol. 8, pp. 1–7, May 2021.

[17] M.-J. Son and S.-P. Lee, "COVID-19 diagnosis from Crowdsourced cough sound data," *Appl. Sci.*, vol. 12, pp. 1–12, Feb. 2022.

[18] B. T. Balamurali, H. T. Hee, S. Kapoor, O. H. Teoh, S. S. Teng, K. P. Lee, D. Herremans, and J. M. Chen, "Deep neural network-based respiratory pathology classification using cough sounds," *Sensors*, vol. 21, no. 6, pp. 1–20. Aug. 2021.

[19] A. Hassan, I. Shahin, and M. B. Alsabek, "COVID-19 detection system using recurrent neural networks," in *Proc. Int. Conf. Commun., Comput., Cybersecurity, Informat. (CCCI)*, Nov. 2020, pp. 1–5.

[20] Y. Amrulloh, U. Abeyratne, V. Swarnkar, and R. Triasih, "Cough sound analysis for pneumonia and asthma classification in pediatric population," in *Proc. 6th Int. Conf. Intell. Syst., Modeling Simulation*, Feb. 2015, pp. 127–131.

[21] M. Cohen-McFarlane, R. Goubran, and F. Knoefel, "Novel coronavirus cough database: NoCoCoDa," *IEEE Access*, vol. 8, pp. 154087–154094, 2020.

[22] R. V. Sharan, U. R. Abeyratne, V. R. Swarnkar, and P. Porter, "Automatic croup diagnosis using cough sound recognition," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 2, pp. 485–495, Feb. 2019.

[23] N. M. Shreesha, "Deep learning anomaly detection methods to passively detect COVID-19 from audio," in *Proc. IEEE Int. Conf. Digit. Health (ICDH)*, Chicago, IL, USA, Sep. 2021, pp. 114–121.

[24] A. Serrurier, C. Neuschaefer-Rube, and R. Röhrig, "Past and trends in cough sound acquisition, automatic detection and automatic Classification: A comparative review," *Sensors*, vol. 22, no. 8, pp. 1–30, Apr. 2022.

[25] A. B. Nassif, I. Shahin, M. Bader, A. Hassan, and N. Werghi, "COVID-19 Detection systems using deep learning algorithms based on speech and image data," *Mathematics*, vol. 10, no. 4, pp. 1–24, Feb. 2022.

[26] D. Chicco, M. J. Warrens, and G. Jurman, "the Matthews correlation coefficient (MCC) is more informative than cohen's Kappa and brier score in binary classification assessment," *IEEE Access*, vol. 9, pp. 78368–78381, 2021.

[27] Z. Mushtaq and S. Su, "Environmental sound classification using a regularized deep convolutional neural network with data augmentation," *Appl. Acoust.*, vol. 167, pp. 107389–107401, Oct. 2020.

[28] T.-H. Tan, Y.-T. Lin, Y.-L. Chang, and M. Alkhaleefah, "Sound source localization using a convolutional neural network and regression model," *Sensors*, vol. 21, no. 23, p. 8031, Dec. 2021.

[29] R. Islam, E. Abdel-Raheem, and M. Tarique, "A study of using cough sounds and deep neural networks for the early detection of COVID-19," *Biomed. Eng. Adv.*, vol. 3, pp. 1–12, Jun. 2022.

[30] J. Zhao, X. Li, W. H. Liu, Y. Gao, M. G. Lei, and H. Q. Tan, "DNN-HMM based acoustic model for continuous pig cough sound recognition," *Int. J. Agricult. Biol. Eng.*, vol. 13, no. 3, pp. 186–193, 2020.

[31] K. K. Lella and A. Pja, "Automatic COVID-19 disease diagnosis using 1D convolutional neural network and augmentation with human respiratory sound based on parameters: Cough, breath, and voice," *AIMS Public Health*, vol. 8, no. 2, pp. 240–264, 2021.

[32] K. Feng, F. He, J. Steinmann, and I. Demirkiran, "Deep-learning based approach to identify COVID-19," in *Proc. SoutheastCon*, Mar. 2021, pp. 1–4.

[33] R. Islam, E. Abdel-Raheem, and M. Tarique, "Early detection of COVID-19 patients using chromagram features of cough sound recordings with machine learning algorithms," in *Proc. Int. Conf. Microelectron. (ICM)*, Dec. 2021, pp. 82–85.

[34] V. P. Singh, J. M. S. Rohith, Y. Mittal, and V. K. Mittal, "IIIT-S CSSD: A cough speech sounds database," in *Proc. 22nd Nat. Conf. Commun. (NCC)*, Mar. 2016, pp. 1–6.

[35] L. Orlandic, T. Teijeiro, and D. Atienza, "The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms," *Sci. Data*, vol. 8, no. 1, pp. 1–10, Jun. 2021.

[36] X. Dong, B. Yin, Y. Cong, Z. Du, and X. Huang, "Environment sound event classification with a two-stream convolutional neural network," *IEEE Access*, vol. 8, pp. 125714–125721, 2020.

[37] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *Proc. 14th Python Sci. Conf.*, Jul. 2015, pp. 18–24.

[38] R. V. Sharan and T. J. Moir, "Audio surveillance under noisy conditions using time-frequency image feature," in *Proc. 19th Int. Conf. Digit. Signal Process.*, Aug. 2014, pp. 130–135.

[39] H. Chen and Z. Zhang, "Hybrid neural network based on novel audio feature for vehicle type identification," *Sci. Rep.*, vol. 11, no. 1, pp. 1–10, Apr. 2021.

[40] S. L. Ullo, S. K. Khare, V. Bajaj, and G. R. Sinha, "Hybrid computerized method for environmental sound classification," *IEEE Access*, vol. 8, pp. 124055–124065, 2020.

[41] J. Xie, K. Hu, M. Zhu, J. Yu, and Q. Zhu, "Investigation of different CNN-based models for improved bird sound classification," *IEEE Access*, vol. 7, pp. 175353–175361, 2019.

[42] X. Su, X. Hao, Z. Wang, Y. Liu, H. Xu, T. Liu, G. Gao, and Feilong, "Learning an adversarial network for speech enhancement under extremely low signal-to-noise ratio condition," in *Proc. 26th Int. Conf. (ICONIP)*. Sydney, NSW, Australia, Dec. 2019, pp. 12–15.

[43] M. Dörfler, T. Grill, R. Bammer, and A. Flexer, "Basic filters for convolutional neural networks applied to music: Training or design?" *Neural Comput. Appl.*, vol. 32, no. 4, pp. 941–954, Feb. 2020.

[44] N. Peng, A. Chen, G. Zhou, W. Chen, W. Zhang, J. Liu, and F. Ding, "Environment sound classification based on visual multi-feature fusion and GRU-AWS," *IEEE Access*, vol. 8, pp. 191100–191114, 2020.

[45] F. Haytham. (Apr. 2016). *Speech Processing for Machine Learning: Filter Banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's in-Between*. Accessed: May 29, 2022. [Online]. Available: https://haythamfayek.com

[46] A. Lerch, *An Introduction to Audio Content Analysis Applications in Signal Processing and Music Informatics*, 1st ed. Hoboken, NJ, USA: Wiley, 2012, pp. 23–24.

[47] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*, 1st ed. New York, NY, USA: Springer, 2015, pp. 123–125.

[48] B. T. Balamurali, H. I. Hee, O. H. Teoh, K. P. Lee, S. Kapoor, D. Herremans, and J.-M. Chen, "Asthmatic versus healthy child classification based on cough and vocalized /α:/ sounds," *J. Acoust. Soc. Amer.*, vol. 148, no. 3, pp. 253–259, Sep. 2020.

[49] S. J. Barry, A. D. Dane, A. H. Morice, and A. D. Walmsley, "The automatic recognition and counting of cough," *Cough*, vol. 2, no. 1, pp. 1–9, Sep. 2006.

[50] S. Matos, S. S. Birring, I. D. Pavord, and D. H. Evans, "Detection of cough signals in continuous audio recordings using hidden Markov models," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1078–1083, Jun. 2006.

[51] W. S. M. Sanjaya, D. Anggraeni, and I. P. Santika, "Speech recognition using linear predictive coding (LPC) and adaptive neuro-fuzzy (ANFIS) to control," in *Proc. Int. Conf. Comput. Sci. Eng.* Bandung, Indonesia, Jul. 2017, pp. 1–11.

[52] M. A. Sulaiman and J. Labadin, "Feature selection based on mutual information," in *Proc. IEEE Int. Conf. Asia*, Aug. 2015, pp. 1–6.

[53] K. Rajab and F. Kamalov, "Finite sample based mutual information," *IEEE Access*, vol. 9, pp. 118871–118879, 2021.

[54] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews correlation coefficient metric," *PLoS ONE*, vol. 12, no. 6, pp. 1–17, 2017.

[55] Audacity Team (2021). *Audacity(R): Free Audio Editor and Recorder [Computer Application]. Version 3.1.3*. Accessed: Feb. 1, 2022. [Online]. Available: https://audacityteam.org

[56] J. Leirgulen, M. Nuris-Souguet, C. Levy-Fidel, and L. Orlandic, *Dry vs Wet Coughs Automatic Classification Using the COUGHVID Dataset*, document Corpus ID 236949737, Semantic Scholar, 2021.

[57] D. Celik, N. Mainusch, and X. O. Jurgens. *Cough Classifier: CS-433 Machine Learning Project 2*. Accessed: Apr. 1, 2022. [Online]. Available: https://www.epfl.ch

YO-PING HUANG (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, TX, USA.

He was a Professor and the Dean of Research and Development, the Dean of the College of Electrical Engineering and Computer Science, and the Department Chair with Tatung University, Taipei, Taiwan. He is currently the President with the National Penghu University of Science and Technology, Penghu, Taiwan. He is also the Chair Professor with the Department of Electrical Engineering, National Taipei University of Technology, Taipei, where he served as the Secretary General. His current research interests include deep learning modeling, intelligent control, fuzzy systems design and modeling, and rehabilitation systems design. He is a fellow of IET, CACS, and TFSA. He received the 2021 Outstanding Research Award from the Ministry of Science and Technology (MOST), Taiwan. He serves as the IEEE SMCS VP for Conferences and Meetings, and the Chair of the IEEE SMCS Technical Committee on Intelligent Transportation Systems. He was the IEEE SMCS BoG, the President of the Taiwan Association of Systems Science and Engineering, the Chair of IEEE SMCS Taipei Chapter, the Chair of the IEEE CIS Taipei Chapter, and the CEO of the Joint Commission of Technological and the Vocational College Admission Committee, Taiwan.

RICHARD MUSHI received the B.S. degree in electrical engineering from the University of Dares Salaam, in 2002, and the M.S. degree in telecommunication engineering from The University of Dodoma, in 2012. He is currently pursuing the Ph.D. degree in electrical engineering and computer science with the National Taipei University of Technology, Taipei, Taiwan.

Since 2009, he has been working with St. Augustine University of Tanzania, as a Tutorial Assistant, then promoted to an Assistant Lecturer. His research interests include artificial intelligence in healthcare specialized in cough sound analysis, analysis of various kind of sounds, data mining, and time series prediction.