Deep Learning Framework Applied For Predicting Anomaly of Respiratory Sounds

Dat Ngo
Dept. Electrical and Electronic Eng
HCMUT, VNU-HCM
Ho Chi Minh City, Vietnam
datt.ngo.hcmut@gmail.com

Lam Pham
Dept. Electrical and Electronic Eng
HCMUT, VNU-HCM
Ho Chi Minh City, Vietnam
lamd.pham@hcmut.edu.vn

Anh Nguyen
Dept. Electrical and Electronic Eng
HCMUT, VNU-HCM
Ho Chi Minh City, Vietnam
anh.nguyenk2017@hcmut.edu.vn

Ben Phan
Dept. Electrical and Electronic Eng
DUT, UT
Da Nang City, Vietnam
phanben110@gmail.com

Khoa Tran
Dept. Electrical and Electronic Eng
DUT, UT
Da Nang City, Vietnam
tdkhoa1212@gmail.com

Truong Nguyen
Dept. Electrical and Electronic Eng
HCMUT, VNU-HCM
Ho Chi Minh City, Vietnam
truongnguyen@hcmut.edu.vn

***Abstract*—This paper proposes a robust deep learning framework used for classifying anomaly of respiratory cycles. Initially, our framework starts with front-end feature extraction step. This step aims to transform the respiratory input sound into a two-dimensional spectrogram where both spectral and temporal features are well presented. Next, an ensemble of C-DNN and Autoencoder networks is then applied to classify into four categories of respiratory anomaly cycles. In this work, we conducted experiments over 2017 Internal Conference on Biomedical Health Informatics (ICBHI) benchmark dataset. As a result, we achieve competitive performances with ICBHI average score of 0.49, ICBHI harmonic score of 0.42.**

***Keywords*— Respiratory disease, wheeze, crackle, ensemble, C-DNN, autoencoder network**

I. INTRODUCTION

According to statistics of Global Burden of Diseases, Injuries and Risk Factors Study, there is an alarming number of deaths due to chronic respiratory diseases, with the figure increased from 3.32 million in 1990 to 3.91 million in 2017 [1]. Furthermore, it becomes worse when this number is expected to continue go up in the next ten years. However, with the timely development of respiratory research, most respiratory diseases nowadays would be preventable by the early diagnosis. For instance, lung auscultation has been introduced as one of the most inexpensive, noninvasive and time-saving methods for respiratory examination thanks to every respiratory cycle can be heard and detected as whether its sound is normal or not. In particular, to better spread effective prevention and treatment widely for respiratory diseases, a reliable and quantitative diagnosis support method such as Computer-Aided Diagnosis (CAD) system [2] is proposed. This systems is in an attempt of supporting doctors to hear, detect and differentiate automatically between different respiratory sound patterns [3]. Inspired from this, analysing respiratory sound by robust machine learning methods has recently attracted much attention. Particularly, authors in [4] ultilized Mel-frequency cepstral coefficient (MFCC) as a frame-based feature representation to represent lung sounds into featuring vectors. Next, conventional machine learning models such as Hidden Markov Model [4], Support Vector Machine [5], and Decision Tree [6] explored these vectors to classify anomalies of respiratory sounds. On the other hand, some researchers laid an emphasis on further analysis on feature extraction step via two-dimensional spectrogram. This is applied in order to fully represent audio features like an image in both temporal and spectral information, and then classified by more powerful architectures from image processing such as CNN [7], [8] and RNN [9], [10]. Although many machine learning methods participated in this field, here is an inconsistency between dataset and performance comparison among publications. For instance, some authors in [11]–[15] evaluated their systems over unpublished datasets. Furthermore, it is hard to compare performance when systems proposed use different ratio for splitting data, especially patient's objects.

978-0-7381-3196-2/21/$31.00 ©2021 IEEE

To tackle these issues, we evaluate our proposed systems over the 2017 Internal Conference on Biomedical Health Informatics (ICBHI) [16], one of the largest dataset of respiratory sound published. In terms of the system proposed, we approach deep learning based framework. In particular, we use Gammatone filter to generate Gamatonegram spectrogram where both spectral and temporal information are well represented. Next, the spectrogram is explored by an ensemble of C-DNN and Autoencoder networks.

II. ICBHI DATASET AND TASK DEFINED

A. ICBHI dataset

The 2017 Internal Conference on Biomedical Health Informatics (ICBHI) [16] is one of the largest annotated dataset of respiratory sounds published. Specifically, it contains 920 audio recordings collected in several years from 126 subjects in two different European countries. The subjects are identified as being healthy or exhibiting one of the following respiratory diseases or conditions such as: COPD, Bronchiectasis, Asthma, upper and lower respiratory tract infection, Pneumonia, Bronchiolitis. All recordings account for the duration of 5.5 hours, comprising 6898 respiratory cycles professionally labeled by respiratory experts. Within each audio recording, four different types of respiratory cycle are denoted as *Crackle*, *Wheeze*, *Both* (*Crackle & Wheeze*), and *Normal* according to the identified onset (i.e. starting time) and offset (i.e. ending time). Furthermore, these cycles show various duration ranging from 0.2 s up to 16.2 s and unbalanced (i.e. 1864 cycles of *Crackle*, 886 cycles of *Wheeze*, 506 cycles of *Both*, and 3642 cycels of *Normal*).

B. Task defined from ICBHI dataset

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

Given by ICBHI dataset, this paper evaluates performance of respiratory anomaly classification among four different cycles (*Crackle*, *Wheeze*, *Both*, and *Normal*). In terms of metric used for evaluating, we follow ICBHI challenge, thus report ICBHI scores as mentioned in [16]. In particular, a confusion matrix of respiratory cycle classified is presented in Table I. Specifically, the letters of *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 inference results. The sums C_t, W_t, B_t and N_t are the total numbers of cycles. Thus, *Sensitivity (SE)* , and *Specitivity (SP)* are firstly computed by,

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

$$Specificity = \frac{N_n}{N_t} \tag{2}$$

Next, ICBHI scores comprising average score (AS) and the harmonic score (HS) are compuated by,

$$AS = \frac{SE + SP}{2} \tag{3}$$

$$HS = \frac{2.SE.SP}{SE + SP} \tag{4}$$

III. DEEP LEARNING BASED FRAMEWORK PROPOSED

The proposed high-level system architecture including two main parts: front-end feature extraction as described in the upper part of Fig. 1 with setting parameters in Table II and back-end deep learning model as shown in the lower part of Fig. 1.

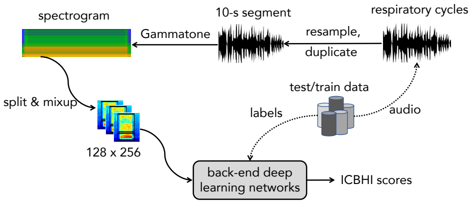

Fig. 1. Deep learning based framework proposed

TABLE II
FEATURE EXTRACTION PARAMETER SETTING

Factors	**Setting**
Re-sample	4000 Hz
Cycle duration	10 s
Spectrogram	Gammatonegram [17]
FFT number	1024
Window size	0.2s
Hop size	0.04s
Patch size	128 × 256
Data augmentation	Random oversampling & Mixup

A. Front-end feature extraction

In particular, we re-sample respiratory cycles to 4000 Hz since frequency banks of abnormal sounds (*Crackle* and *Wheeze*) locate mostly from 60 to 2000 Hz. Consequently, re-sampled respiratory cycles showing different lengths are next duplicated to ensure the same length of 10 seconds. Next, respiratory cycles go through a bandpass filter of 100-2000 Hz to reduce noise. After that, these respiratory sounds are transformed into two-dimensional spectrograms by using Gammatone transformation. To generate Gammatone spectrogram (Gamma), we firstly compute Short-Time Fourier Transform (STFT) as presented below:

Short-Time Fourier Transform (STFT): The STFT spectrogram applies Fourier Transform to extract Frequency content of local section of input signal over short time duration. Let consider $\mathbf{s}(n)$ as digital audio signal with length of N , a pixel value at central frequency f and time frame t of STFT spectrogram $\mathbf{STFT}[F,T]$ is computed as:

$$\mathbf{STFT}[f,t] = \sum_{n=0}^{N-1} \mathbf{s}[n].\mathbf{w}[n]e^{-j2\pi fn} \tag{5}$$

where $\mathbf{w}[n]$ is a window function, typically Hamming. While time resolution (T) of STFT spectrogram is set by window side and hope size, the frequency resolution (F) equals to the number of central frequencies set to 1024. Then, we apply Gammatone filter into STFT spectrogram as described below:

Gammatone (GAM): Gammatone filters are designed to model the frequency-selective cochlea activation response of the human inner ear [17], in which filter output simulates the frequency response of the basilar membrane. The impulse response is given by:

$$g[k] = k^{P-1}T^{P-1}e^{-2bkT\pi}cos(2fkT\pi + \theta) \tag{6}$$

where k is time, P is the filter order, T is sampling period, b is filter bandwidth, f is central frequency, θ is the phase of the carrier. The filter bank was then formulated as ERB scale [18] as:

$$ERB = 24.7(4.37.10^{-3}f + 1) \tag{7}$$

To quickly generate Gamma spectrogram, we apply a toolbox developed by Ellis *et al.* [19], namely Gammatone-like spectrogram. Firstly, audio signal is transformed into STFT spectra recently mentioned above. Next, gammatone weighting $\mathbf{COE}[F_{gam}, F]$ is applied on STFT to obtain the Gamma spectrogram.

$$\mathbf{GAM}[F_{gam}, T] = \mathbf{COE}[F_{gam}, F] \times \mathbf{STFT}[F, T] \tag{8}$$

where F_{gam} resolution of GAM spectrogram is Gammatone filter number of 128.

Next, each 10-s spectrogram of one respiratory cycle is thus split into non-overlapped patches of 128×256, likely an image. To deal with unbalanced data issue, we apply two data augmentation techniques on the image patches of 128×256. Firstly, we randomly oversample image patches to make sure that the number of patches per category is equal. Next, the mixup data augmentation [20] is applied to enlarge Fishers criterion (i.e. the ratio of the between- class distance to the within class variance in the feature space) to increase variation of training data. Let consider two original image patches as $\mathbf{X_1}$, $\mathbf{X_2}$ and expected labels as $\mathbf{y_1}$, $\mathbf{y_2}$, new image patches are generated as below equations:

$$\mathbf{X_{mp1}} = \mathbf{X_1}\gamma + \mathbf{X_2}(1-\gamma) \tag{9}$$

$$\mathbf{X_{mp2}} = \mathbf{X_1}(1-\gamma) + \mathbf{X_2}\gamma \tag{10}$$

$$\mathbf{y_{mp1}} = \mathbf{y_1}\gamma + \mathbf{y_2}(1-\gamma) \tag{11}$$

$$\mathbf{y_{mp2}} = \mathbf{y_1}(1-\gamma) + \mathbf{y_2}\gamma \tag{12}$$

where γ is random coefficient from *Beta* distribution, $\mathbf{X_{mp1}}$, $\mathbf{X_{mp2}}$ and $\mathbf{y_{mp1}}$, $\mathbf{y_{mp2}}$ are new image patches and labels generated, respectively. Eventually, the mixup patches are fed into a back-end classifier, report the classification accuracy.

B. Back-end classification

TABLE III
C-DNN NETWORK ARCHITECTURE

Network architecture	Output
CNN	
Input layer (image patch of 128×256)	
Bn - Cv [3×3] - Relu - Bn - Mp [2×2] - Dr (10%)	62×78×64
Bn - Cv [3×3] - Relu - Bn - Mp [2×2] - Dr (15%)	31×39×128
Bn - Cv [3×3] - Relu - Bn - Mp [2×2] - Dr (20%)	16×20×256
Bn - Cv [3×3] - Relu - Bn - Gmp - Dr (25%)	512
DNN	
Input layer (512-dimensional vectors)	
Fl - Relu - Dr (30%)	1024
Fl - Softmax	4

For back-end classification, we propose an ensemble of C-DNN and autoencoder networks in this paper. As regards C-DNN network architecture, it comprises two main parts of CNN and DNN as shown in Table III, likely Lenet-6 [21]. The CNN as the upper part in Table III performs batch normalization (Bn), convolutional (Cv[kernel size]), rectified linear units (Relu), max pooling (Mp[kernel size]) and global max pooling (Gmp), and dropout (Dr (dropout percentage)) layers. Meanwhile, the DNN as shown in the lower part in Table III comprises two fully-connected (Fl), rectified linear units (Relu) and a final Softmax layer for classification.

TABLE IV
ENCODER-DECODER NETWORK ARCHITECTURE

Block	Network architecture	Output
	Encoder	
	Input layer (image patch of 128×256)	
Conv. Block 01	Bn - Cv [3×3] - Relu - Bn - Mp [2×2] - Dr (10%)	64×128×64
Conv. Block 02	Bn - Cv [3×3] - Relu - Bn - Mp [2×2] - Dr (15%)	32×64×128
Conv. Block 03	Bn - Cv [3×3] - Relu - Bn - Mp [2×2] - Dr (20%)	16×32×256
Conv. Block 04	Bn - Cv [3×3] - Relu - Bn - Gmp - Dr (25%)	512
	Decoder	
	Input layer (512-dimensional vectors)	
Full. Block 01	Fl - Relu	32768
Reshape 01	Reshape	8×16×256
DeCv. Block 01	DeCv [3×3] - Relu	16×32×128
DeCv. Block 02	DeCv [3×3] - Relu	32×64×64
DeCv. Block 03	DeCv [3×3] - Relu	64×128×32
DeCv. Block 04	DeCv [3×3] - Relu	128×256×1
Reshape 02	Reshape	128×256

TABLE V
MLP-BASED NETWORK ARCHITECTURE

Block	Network architecture	Output
	Input layer (512-dimensional vectors)	
Full. Block 02	Fl - Relu - Dr (50%)	1024
Full. Block 03	Fl - Relu - Dr (50%)	1024
Full. Block 04	Fl - Softmax	4

In terms of autoencoder network architecture proposed, it shows more complicated with two training phases as showed in Fig. 2. At the first phase, we present an Encoder-Decoder architecture which is used to extract embedding vectors containing condensed information. Next, the embeddings are fed into a MLP based network architecture for classifying into four categories in the second phase. As shown in Table IV and the upper part of Fig. 2, Encoder part of Encoder-Decoder architecture comprises four Conv. Blocks each which performs layers as same as C-DNN network architecture. The Encoder helps to compress input image patch into condensed vectors, referred to as embeddings. Meanwhile, Decoder firstly use a fully-connected layer to decompress embeddings, thus apply four DeCv. Blocks (i.e. each DeCv. Block comprises a deconvolutional (DeCv[kernel size]) layer and a rectified linear unit layer (Relu)) to re-construct the input image patch). Notably, batch normalization (Bn), max pooling (Mp[kernel size]) and global max pooling (Gmp) and dropout (Dr (dropout percentage)) layers are not applied in Decoder. For MLP-based network architecture as shown in Table V and the lower part of Fig. 2, it is configured by two fully-connected layers (Fl). The first fully-connected layer follow by a ReLu and a Dr. Meanwhile, a Softmax layer is used after the second fully-connected layer for classification.

C. Experimental setting

Given by ICBHI dataset, we follow ICBHI challenge setting, thus divide into Training and Test subsets with the ratio of 60% and 40% respectively as shown in Table VI. Notably, the splitting proposed prevents the present of object's audio recordings on both Training and Test subsets.

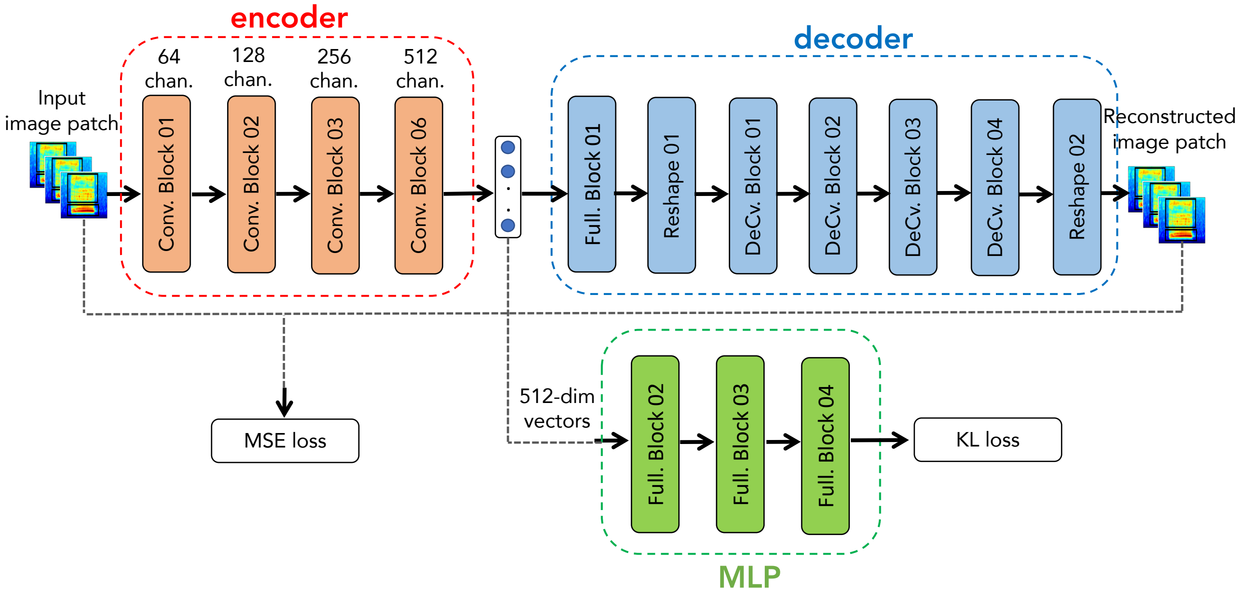

Fig. 2. Block-level architecture of Autoencoder network.

TABLE VI
ICBHI DATASET SPLITTING

	Training Set	Test Set
Wheezes	501	385
Crackles	1215	649
Both	363	143
Normal	2063	1579

In terms for back-end network architectures proposed, we adopt Tensorflow framework and set learning rate to 0.0001, a batch size of 50, epoch number of 100, and Adam method [22] for learning rate optimization. As using mixup data augmentation, the labels are not one-hot format. Therefore, we use Kullback-Leibler (KL) divergence loss [23] in C-DNN and MLP-based networks instead of the standard cross-entropy loss as shown in Eq. (14) below:

$$Loss_{KL}(\Theta) = \sum_{n=1}^{N} \mathbf{y}_n \log(\frac{\mathbf{y}_n}{\hat{\mathbf{y}}_n}) + \frac{\lambda}{2}||\Theta||_2^2, \qquad (13)$$

where $Loss_{KL}(\Theta)$ is KL-loss function, Θ describes the trainable parameters of the network trained, λ denote the ℓ_2-norm regularization coefficient experimentally set to 0.0001, N is the batch size, $\mathbf{y_n}$ and $\hat{\mathbf{y}}_\mathbf{n}$ are the ground-truth and the network recognized output, respectively. To train Encoder-Decoder network, we use Mean Squared Error (MSE) loss to compare original image patches (input of Encoder) to reconstructed image patches (output of Decoder) as below,

$$Loss_{MSE} = \frac{1}{2N}\sum_{n=1}^{N}(\mathbf{X}_n - \hat{\mathbf{X}}_n)^2 \qquad (14)$$

where $\mathbf{X}_n$ and $\hat{\mathbf{X}}_n$ are input image patch and reconstructed image patch, respectively.

D. Late fusion strategy

As the C-DNN model work on patch level, the probability of an entire spectrogram is computed by averaging of all patches' probabilities. Let consider $\mathbf{P^n_{C-DNN}} = (\mathbf{k^n_1}, \mathbf{k^n_2}, ..., \mathbf{k^n_C})$, with C being the category number and the n^{th} out of N patches fed into learning model, as the probability of a test sound instance, then the mean classification probability is denoted as $\mathbf{\bar{p}_{C-DNN}} = (\bar{k}_1, \bar{k}_2, ..., \bar{k}_C)$ where,

$$\bar{k}_c = \frac{1}{N}\sum_{n=1}^{N} k^n_c \quad for \quad 1 \leq c \leq C \qquad (15)$$

As the Autoencoder model work on patch level, the probability of an entire spectrogram is computed by averaging of all patches' probabilities. Let consider $\mathbf{P^n_{MLP}} = (\mathbf{m^n_1}, \mathbf{m^n_2}, ..., \mathbf{m^n_C})$, with C being the category number and the n^{th} out of N patches fed into MLP-based network, as the probability of a test sound instance, then the mean classification probability is denoted as $\mathbf{\bar{p}_{MLP}} = (\bar{m}_1, \bar{m}_2, ..., \bar{m}_C)$ where,

$$\bar{m}_c = \frac{1}{N}\sum_{n=1}^{N} m^n_c \quad for \quad 1 \leq c \leq C \qquad (16)$$

To evaluate the ensemble of C-DNN and Autoencoder, we propose three late fusion schemes, namely *Max*, *Mean*, and *Mul* fusions.

The probability of combination with *Max* strategy $\mathbf{p_{f-max}}$ is obtained by,

$$\mathbf{p_{f-max}} = max(\mathbf{\bar{p}_{C-DNN}}, \mathbf{\bar{p}_{MLP}}) \qquad (17)$$

The probability of combination with *Mean* strategy $\mathbf{p_{f-mean}}$ is obtained by,

$$\mathbf{p_{f-mean}} = \frac{\bar{\mathbf{p}}_{\mathbf{C-DNN}} + \bar{\mathbf{p}}_{\mathbf{MLP}}}{2} \tag{18}$$

The probability of combination with *Mul* strategy $\mathbf{p_{f-mul}}$ is obtained by,

$$\mathbf{p_{f-mul}} = \frac{\bar{\mathbf{p}}_{\mathbf{C-DNN}} \cdot \bar{\mathbf{p}}_{\mathbf{MLP}}}{2} \tag{19}$$

Eventually, the predicted result is decided by,

$$\hat{y} =_{c \in \{1,2,...,C\}} \bar{p}_c. \tag{20}$$

IV. Experimental results and discussion

As details shown in Table VII, it can be seen that Autoencoder is better than C-DNN in terms of SE score, improving by 0.02. By contrast, the SP score of Auto-encoder reduces by 0.01 compared to C-DNN. As a result, both networks achieve the same AS score of 0.47. Meanwhile, HS scores present 0.41 and 0.43 for C-DNN and Autoencoder, respectively.

As regards comparison among three late fusion methods, Mean fusion achieves the highest performances with SP score of 0.69, SE score of 0.30, and AS/HS scores of 0.49/0.42. Compared to individual C-DNN or Autoencoder model, although ensemble methods make SE scores reduce a little, they help to improve SP scores significantly.

Compare to the state-of-the-art systems as shown in Table VIII (note that we only compare to systems which follow splitting ratio of 60/40 defined by ICBHI challenge), while our system's SP ranks fourth position, SE score achieves the top three. In terms of only using single spectrogram, our system achieves very competitive AS/HS scores of 0.49/0.42 that is top two after the systems proposed in [24].

TABLE VII
Performance of C-DNN, Autoencoder, and their fusions (highest scores in **bold**)

Systems	**SP**	**SE**	**AS/HS Scores**
C-DNN	0.63	0.31	0.47/0.41
Autoencoder	0.62	0.33	0.47/0.43
Max fusion	0.67	0.30	0.48/0.42
Mean fusion	**0.69**	**0.30**	**0.49/0.42**
Mul fusion	**0.69**	0.29	0.49/0.41

TABLE VIII
Compare our systems (the lower part) against state-of-the-art systems with ICBHI challenge splitting (highest scores in **bold**).

Features	**Classifiers**	**SP**	**SE**	**AS/HS Scores**
MFCC	Decision Tree [25]	0.75	0.12	0.43/0.15
MFCC	HMM [26]	0.38	**0.41**	0.39/0.23
STFT+Wavelet	SVM [27]	0.78	0.20	0.47/0.24
Gammatonegram	CNN-MoE	0.68	0.26	0.47/0.37
log-Mel	CNN-RNN [24]	0.69	0.30	0.50/**0.46**
Scalogram	CNN-RNN [24]	0.62	0.37	0.50/**0.46**
log-Mel+Scalogram	CNN-RNN [24]	**0.81**	0.28	**0.54**/0.42
Gammatonegram	**C-DNN**	0.63	0.31	0.47/0.41
Gammatonegram	**Autoencoder**	0.62	0.33	0.47/0.43
Gammatonegram	**C-DNN+Autoencoder**	0.69	0.30	0.49/0.42

V. Conclusion

We have just presented a deep learning based framework which is used for classifying respiratory sound. The exploration of Gammatone transformation and an ensemble of C-DNN and Autoencoder networks achieves significant performances of 0.49 and 0.42 in terms of ICBHI average and harmmonic scores over ICBHI benchmark dataset that are very competitive to the state-of-the-art systems.

References

[1] Xiaochen Li, Xiaopei Cao, Mingzhou Guo, Min Xie, and Xiansheng Liu, "Trends and risk factors of mortality and disability adjusted life years for chronic respiratory diseases from 1990 to 2017: systematic analysis for the global burden of disease study 2017," *bmj*, vol. 368, 2020.

[2] Kunio Doi, "Computer-aided diagnosis in medical imaging: historical review, current status and future potential," *Computerized medical imaging and graphics*, vol. 31, no. 4-5, pp. 198–211, 2007.

[3] A Kandaswamy, C Sathish Kumar, Rm Pl Ramanathan, S Jayaraman, and N Malmurugan, "Neural classification of lung sounds using wavelet coefficients," *Computers in biology and medicine*, vol. 34, no. 6, pp. 523–537, 2004.

[4] Hitoshi Yamamoto, Shoichi Matsunaga, Masaru Yamashita, Katsuya Yamauchi, and Sueharu Miyahara, "Classification between normal and abnormal respiratory sounds based on stochastic approach," in *Proc. 20th International Congress on Acoustics*, 2010.

[5] Morten Grønnesby, Juan Carlos Aviles Solis, Einar Holsbø, Hasse Melbye, and Lars Ailo Bongo, "Feature extraction for machine learning based crackle detection in lung sounds from a health survey," *arXiv preprint arXiv:1706.00005*, 2017.

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

[7] Murat Aykanat, Özkan Kılıç, Bahar Kurt, and Sevgi Saryal, "Classification of lung sounds using convolutional neural networks," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, pp. 65, 2017.

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

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

[10] Kirill Kochetov, Evgeny Putin, Maksim Balashov, Andrey Filchenkov, and Anatoly Shalyto, "Noise masking recurrent neural network for respiratory sound classification," in *International Conference on Artificial Neural Networks*, 2018, pp. 208–217.

[11] Luis Mendes, Ioannis M Vogiatzis, Eleni Perantoni, Evangelos Kaimakamis, Ioanna Chouvarda, Nicos Maglaveras, Jorge Henriques, Paulo Carvalho, and Rui Pedro Paiva, "Detection of crackle events using a multi-feature approach," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 3679–3683.

[12] Shreyasi Datta, Anirban Dutta Choudhury, Parijat Deshpande, Sakyajit Bhattacharya, and Arpan Pal, "Automated lung sound analysis for detecting pulmonary abnormalities," in *2017 39th Annual International Conference of the Ieee Engineering in Medicine and Biology Society (Embc)*. IEEE, 2017, pp. 4594–4598.

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

[14] Lukui Shi, Kang Du, Chaozong Zhang, Hongqi Ma, and Wenjie Yan, "Lung sound recognition algorithm based on vggish-bigru," *IEEE Access*, vol. 7, pp. 139438–139449, 2019.

[15] Elmar Messner, Melanie Fediuk, Paul Swatek, Stefan Scheidl, Freyja-Maria Smolle-Juttner, Horst Olschewski, and Franz Pernkopf, "Crackle and breathing phase detection in lung sounds with deep bidirectional gated recurrent neural networks," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 356–359.

[16] BM 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*, pp. 33–37. 2018.

[17] Roy D Patterson, "Auditory filters and excitation patterns as representations of frequency resolution," *Frequency selectivity in hearing*, 1986.

[18] Brian R Glasberg and Brian CJ Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing research*, vol. 47, no. 1-2, pp. 103–138, 1990.

[19] Daniel PW Ellis, "Gammatone-like spectrograms," *web resource: http://www. ee. columbia. edu/dpwe/resources/matlab/gammatonegram*, 2009.

[20] Justin Salamon and Juan Pablo Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[21] Diego Perna, "Convolutional neural networks learning from respiratory data," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018, pp. 2109–2113.

[22] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[24] 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.

[25] B. M. Rocha, D. Filos, L. Mendes, G. Serbes, S. Ulukaya, Y. P. Kahya, N. Jakovljevic, T. L. Turukalo, I. M. Vogiatzis, E. Perantoni, et al., "An open access database for the evaluation of respiratory sound classification algorithms," *Physiological measurement*, vol. 40, no. 3, pp. 035001, 2019.

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

[27] Gorkem Serbes, Sezer Ulukaya, and Yasemin P Kahya, "An automated lung sound preprocessing and classification system based on spectral analysis methods," in *Precision Medicine Powered by pHealth and Connected Health*, pp. 45–49. Springer, 2018.