Lung Sound Classification Using Snapshot Ensemble of Convolutional Neural Networks

Truc Nguyen and Franz Pernkopf

Abstract— **We propose a robust and efficient lung sound classification system using a snapshot ensemble of convolutional neural networks (CNNs). A robust CNN architecture is used to extract high-level features from log mel spectrograms. The CNN architecture is trained on a cosine cycle learning rate schedule. Capturing the best model of each training cycle allows to obtain multiple models settled on various local optima from cycle to cycle at the cost of training a single mode. Therefore, the snapshot ensemble boosts performance of the proposed system while keeping the drawback of expensive training of ensembles moderate. To deal with the class-imbalance of the dataset, temporal stretching and vocal tract length perturbation (VTLP) for data augmentation and the focal loss objective are used. Empirically, our system outperforms state-of-the-art systems for the prediction task of four classes (normal, crackles, wheezes, and both crackles and wheezes) and two classes (normal and abnormal (i.e. crackles, wheezes, and both crackles and wheezes)) and achieves 78.4% and 83.7% ICBHI specific micro-averaged accuracy, respectively. The average accuracy is repeated on ten random splittings of 80% training and 20% testing data using the ICBHI 2017 dataset of respiratory cycles.**

Clinical relevance **Lung sound classification, convolutional neural networks, snapshot ensemble.**

I. INTRODUCTION

Lung sounds convey relevant information for pulmonary disorders including adventitious breath sounds such as crackles, wheezes, or both of crackles and wheezes [1]. To facilitate a more objective assessment of the lung sound for diagnosis of pulmonary diseases/conditions, digital recording and processing techniques have been matter of intensive research over past decades. Computational methods for the analysis of lung sounds eliminate several limitations of simple auscultation and offers advantages for medical diagnosis [2]. Computational lung sound analysis (CLSA) requires high accuracy algorithms (including features) for adventitious sound detection and classification, careful evaluation in real-life use scenarios, and portable easy-to-use devices without the necessity of expert interaction.

In recent years, deep learning became one of the main approaches for adventitious sound detection and classification in CLSA. In early computational lung sound research, conventional machine learning methods were used to recognize lung sounds such as self-organizing maps [3], Gaussian mixture models (GMMs) [4], and support vector machines (SVMs) [5]. Recently, classifiers such as CNNs [1], [6], recurrent neural networks (RNNs) [7], [8], or CNNs combined with RNNs [9] using time-frequency representations such as MFCCs and spectrograms belong to the most successful approaches. In addition, data augmentation, transfer learning, and ensemble methods have been explored to enhance performance [1], [9]. The systems were evaluated on non-public datasets such as R.A.L.E. [1] or multi channel lung sound data [10] and public datasets i.e. the ICBHI 2017 dataset [7], [8], [11].

In this paper, we develop a robust lung sound classification system using the ICBHI 2017 database. We extend the audio pre-processing and feature extraction to augment the data for the CNN model. In particular, we perform temporal stretching/compressing and vocal tract length perturbation (VTLP) to counteract the class-imbalance of the data and improve model performance. In the feature processing stage, we propose *sample padding* and *feature splitting*. Both of them improve performance. Furthermore, the snapshot ensemble of CNN models increases the performance at moderate additional training cost. These modifications help to outperform the state-of-the-art systems for respiratory sound classification tasks of two and four categories [7], [8], [12].

II. ICBHI 2017 DATABASE

The ICBHI 2017 database [11] consists of 920 annotated audio samples from 126 subjects. The audio samples were recorded using different stethoscopes. The recording duration ranges from 10s to 90s and the sampling rate ranges from 4000Hz to 44100Hz. Each recording is composed of a certain number of breathing cycles with annotations of the beginning and the ending, and the presence/absence of crackles and/or wheezes. We use the annotations of the database to split audio recordings into respiratory cycles. The cycle duration ranges from 0.2s to 16s and the average cycle duration is 2.7s. The database includes 6898 different respiratory cycles with 3642 normal cycles, 1864 crackles, 886 wheezes, and 506 cycles consisting of both crackles and wheezes.

III. PROPOSED FRAMEWORK

The proposed system includes three key stages shown in Fig. 1. Firstly, the respiratory cycles are pre-processed and divided into short chunks of log-mel spectrograms. Secondly, the features are fed to the CNN model for training. Finally, the class probabilities of all chunks of each cycle are averaged and the argmax determines the class label.

*This work was supported by the Vietnamese - Austrian Government Scholarship and the Austrian Science Fund (12706-N31).

Truc Nguyen and Franz Pernkopf are with the Signal Processing and Speech Communication Lab., Graz University of Technology, Inffeldgasse 16c, A-8010 Graz, Austria/Europe t.k.nguyen@tugraz.at, pernkopf@tugraz.at

978-1-7281-1990-8/20/$31.00 ©2020 IEEE

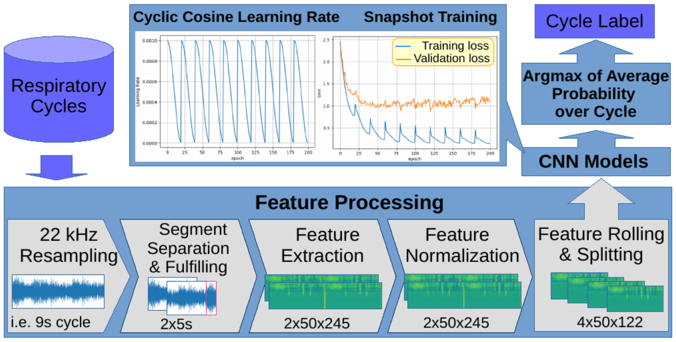

Fig. 1. System Framework.

A. Audio Pre-processing and Feature Extraction

Inspired from the baseline for wheeze and crackle detection [13], we perform five steps for audio pre-processing and feature extraction of the respiratory cycles, namely: resampling, segment separation and fulfilling, feature extraction, feature normalization, and feature rolling and splitting.

1) Resampling: As audio recordings were collected with different sampling rates, respiratory cycles are resampled to 22kHz.

2) Segment Separation and Fulfilling: The duration (length) of respiratory cycles is different while the CNN model requires the same number of samples in each cycle. Therefore, we crop the cycles into one fixed-length segment or more fixed-length segments (without overlap) if the cycle's length exceeds the segment length. Partially filled segments are completed by sampling from the available cycle samples. We call this *sample padding*. This is in contrast to *zero padding* of partially filled segments [13].

We use a fixed length of segments in the range of 3s to 9s and compare the performance of the *sample padding* and *zero padding* techniques (see Section IV.B. Performance).

3) Feature Extraction: As we use convolutional layers for high-level feature extraction, the fixed-length segments of the respiratory cycles are transformed into mel spectrograms. Empirically, we use 512 samples as window size of the fast Fourier transform (FFT) without overlap between the windows. The number of mel frequency bins is chosen as 50. Logarithmic scale is applied to the magnitude of the mel spectrograms.

4) Feature Normalization: We use z-score normalization to scale all log mel spectrogram features.

5) Feature Rolling and Splitting:

- *Rolling feature along temporal axis:* It is used to shift the feature frames with respected to the beginning of the respiratory cycles in a cyclic manner. It helps to simulate real-world conditions where the recording data are not aligned with the respiratory cycles.
- *Splitting feature into chunks:* We split the rolled feature frames into short chunks with the same number of temporal frames before feeding them to the CNN model. We compare the system performance using feature splitting and without feature splitting as used in the baseline [13] (see Section IV.B. Performance).

B. Data Augmentation

Similar as in [13], we use data augmentation in order to balance the training dataset and prevent overfitting.

1) Time stretching: Time stretching increases/reduces the sampling rate of an audio signal without affecting its pitch. It is used to double the number of samples of the wheeze, and both wheeze and crackle classes of the training set. We use a random sampling rate uniformly distributed with $\pm 10\%$ of the original sampling rate.

2) Vocal tract length perturbation (VTLP): VTLP uses a random wrap factor α for each recording and maps the frequency f of the signal bandwidth to a new frequency f' [14]. We select α from a uniform distribution $\alpha \sim \mathcal{U}(0.9, 1.1)$ and set the maximum signal bandwidth to F_{hi} = [3200, 3800]. VTLP is applied directly to the mel filter bank rather than distorting each spectrogram frame. For the original training set and the time stretched data, VTLP is applied to enlarge the dataset for all classes by an additional set of 3642x0.8, 3x1864x0.8, 4x(886x0.8x2), and 7x(506x0.8x2) cycles for normal, crackles, wheezes, and both crackles and wheezes, respectively[1].

C. Neural Network and Relevant Components

1) Convolutional Neural Network: We propose a robust CNN model of 7 convolutional compositions with different number of filters and stride. The input shape of the model is 50xNx1, where N is the number of temporal feature frames (with feature splitting/without feature splitting). For instance, a fixed-length segment of 9s corresponding to a log mel spectrogram size of 50x386 is split into 2 short chunks of 50x193 i.e. N of 193. Consequently, the input shape of the CNN is (50x193x1). N_i is number of temporal frames of the CNN feature maps. Each convolutional composition includes a batch normalization layer (BN) and a convolution layer (Conv2D) using ReLU activations and regularizer L2 (BN-Conv2D-ReLU). This is used as a high-level feature extractor. The stride of 2 is used in the convolutional layer to decrease the spatial dimension of the convolutional outputs, i.e. time-frequency representation, by a factor of 2. It reduces the computational time and complexity for the following layers in the training phase as well as avoids over-fitting. In addition, a global average pooling (GAP) layer is added after the last convolution composition. The GAP layer allows to reduce the number of outputs of the previous layer. After the GAP layer, a fully connected layer of 512 units combined with batch normalization is used for high-level feature extraction. Finally, an output layer of 4 or 2 units using a softmax activation is used to predict the output classes. Table I lists the details of the CNN model.

2) Snapshot Ensemble: A snapshot ensemble allows us to build an ensemble of multiple models at moderate additional training cost [15]. The approach is based on the non-convex nature of neural networks and the ability to converge and escape from local minima using a specific schedule to adjust the learning rate during training. In more detail, a diverse

[1] Factor 0.8 denotes to 80% data of the database for the training set.

TABLE I
DETAILS OF THE CNN MODEL

Layer	Output	Kernel size	Stride
Input layer	50xNx1	-	-
BN+Conv2D+ReLU	50xNx64	3x3	1
BN+Conv2D+ReLU	48xN_1x128	3x3	2
BN+Conv2D+ReLU	23xN_2x128	3x3	1
BN+Conv2D+ReLU	21xN_3x256	3x3	2
BN+Conv2D+ReLU	10xN_4x256	3x3	1
BN+Conv2D+ReLU	8xN_5x512	3x3	2
BN+Conv2D+ReLU	3xN_6x512	3x3	1
BN+GAP	512	-	-
Dense+ReLU+BN	512	-	-
Dense+softmax	4	-	-

set of models is snapshot during a single training run using a cosine cycle learning rate schedule, named cyclic cosine annealing. The optimization converges to a local minima at the end of each cycle along its optimization trajectory. 'Good optimized models' at the end of each cycle are memorized as snapshot models. They are reused as the starting point for the subsequent learning rate cycle instead of a new randomly initialized model.

Cyclic cosine annealing is used as annealing schedule, which relies on the cosine function. It starts at a large learning rate that is rapidly decreased to a minimum value before being drastically increased again. The learning rate α has the form:

$$\alpha(t) = \frac{\alpha_0}{2}\left(\cos\left(\frac{\pi \bmod (t-1), \lfloor T/M \rfloor}{\lfloor T/M \rfloor}\right) + 1\right), \quad (1)$$

where $\alpha(t)$ is the learning rate at epoch t, α_0 is the maximum learning rate, T is the total number of epochs and M is the number of cycles. Mod is modulo operation and $\lfloor . \rfloor$ indicate a floor operation.

The cyclic cosine annealing and losses of the training set and validation set are shown in Fig. 1. We can see that the training loss converges to a local minimum at the end of each cycle. It drastically increases at a large learning rate and gradually decrease until the end of each cycle.

For the snapshot ensemble, the class probabilities of the ensemble are averaged over all snapshot models.

3) Focal loss: The focal loss for multi-class classification is applied instead of the cross-entropy loss because of its ability to deal with difficult samples. The focal loss is also used in case of class imbalance [16]. The loss function is a dynamically scaled cross entropy loss, where the scaling factor decays to zero as confidence in the correct class increases. This scaling factor can automatically down-weight the contribution of easy-to-classify samples during training and rapidly focus the model on samples which are hard to classify [16], i.e. the focal loss (FL) is defined as:

$$FL(p, y) = -\sum_{j=1}^{C}(1 - p_j)^{\gamma} y_j \log(p_j), \quad (2)$$

where p_j is the estimated probability of the model for class j of a sample, y_j is a binary indicator (0 or 1) i.e. y_j = 1 if class j is the correct class. C denotes the number of classes. We select γ = 1.

IV. EXPERIMENTS

A. Setup

We separate the audio samples using their annotations into a set of respiratory cycles of four classes, namely normal, crackle, wheeze, and both of crackle and wheeze. The data is divided to 20% for testing and 80% for training, the training set is further split into 80% for model training and 20% for validation. The reported performance of the system is the average accuracy of ten independent runs using different data splittings.

We use ICBHI - specific criteria required by the ICBHI Challenge to evaluate the performance. The *sensitivity*, *specificity*, and their average, known as *ICBHI Score* are as follows [8], [11]: $Sensitivity = C_{crackle\_or\_wheeze}/N_{crackle\_or\_wheeze}$ and $Sensitivity = (C_{crackle} + C_{wheeze} + C_{both})/(N_{crackle} + N_{wheeze} + N_{both})$ for 2-class and 4-class classifications, respectively. $Specificity = C_{normal}/N_{normal}$ is similar for both classification tasks. C_s and N_s values denotes the number of correctly recognized instances and the total number of instances, respectively. Representations of $crackle\_or\_wheeze$ of the *sensitivity* measure for the 2-class case refers to all recognized values of the *crackles*, *wheezes*, and *both* classes classified as abnormal lung sounds.

Training the network is carried out by optimizing the focal loss using the Adam optimizer at learning rate of 0.0001 and batch size of 32. The cosine annealing is used for training snapshot models with 20 epochs per cycle. We observe the impact of segment length, sample and zero padding techniques, and feature splitting and no feature splitting for the model inputs. The number of epochs is set to 150 and the optimal model is that with the highest validation accuracy. We use the Glorot uniform initializer for the network weights. Weight decay regularizer is included with a factor of 0.001. Data is shuffled between the epochs. The cycle number of cosine annealing periods is observed in the range of $M \in \{2, ..., 10\}$.

B. Performance

Table II shows classification results for segment splitting, zero padding and sample padding corresponding to various segment lengths. We can see that sample padding for split segments outperforms zero padding in almost all observed segment lengths for both classification tasks. In addition, segment splitting into short chunks works better. We select the setting of the best result on the 4-class classification task for testing with snapshot ensemble i.e. we select 9s segment length split into 2 short chunks using sample padding.

In Table III, we evaluate the snapshot ensemble with different number of cycles M of the cyclic cosine annealing. Although using the same model architecture, different settings of learning rate schedules cause different performance. The table shows that the performance of the snapshot ensemble with different numbers of cycles are mostly on par to the

TABLE II

ICBHI Score Comparison of zero padding, sample padding, and feature splitting

Segment	Zero Padding		Sample Padding		Sample Padding & Splitting		
Length	2-class	4-class	2-class	4-class	Length	2-class	4-class
3s	0.814	0.751	0.823	0.762	2x2s	0.821	0.760
4s	0.817	0.758	0.826	0.769	2x2.5s	0.821	0.762
5s	0.823	0.765	0.825	0.767	2x3s	0.818	0.758
6s	0.821	0.766	0.831	0.773	3x2s	0.818	0.760
7s	0.828	0.767	0.825	0.770	2x3.5s	0.823	0.767
8s	0.825	0.767	**0.832**	**0.774**	2x4s	0.826	0.768
9s	0.827	0.769	0.825	0.768	4x2s	0.826	0.768
					2x4.5s	**0.832**	**0.776**
					3x3s	0.830	0.773

best model of the observed settings in Table II. The snapshot ensemble of 8 cycles achieves the best performance for both the 2-class and 4-class task.

TABLE III

Comparison of different cycle numbers M of cyclic cosine annealing for snapshot ensemble

Annealing	2-class Task			4-class Task		
	Spec.	Sens.	ICBHI-Score	Spec.	Sens.	ICBHI Score
2-cycle	0.869	0.788	0.829	0.869	0.679	0.774
3-cycle	0.870	0.801	0.835	0.870	0.679	0.775
4-cycle	0.865	0.799	0.832	0.865	0.686	0.775
5-cycle	0.864	0.795	0.830	0.864	0.686	0.775
6-cycle	0.860	0.802	0.831	0.860	0.687	0.774
7-cycle	0.868	0.796	0.832	0.868	0.688	0.778
8-cycle	**0.873**	**0.801**	**0.837**	**0.873**	**0.694**	**0.784**
9-cycle	0.871	0.790	0.831	0.871	0.679	0.775
10-cycle	0.868	0.785	0.827	0.868	0.672	0.770

TABLE IV

ICBHI challenge comparison

Task	Method	Spec.	Sens.	ICBHI Score	Param.
4-class	MNRNN [7]	0.74	0.56	0.65	-
4-class	STFT+wavelet [12]	0.83	0.55	0.69	-
4-class	LSTM [8]	0.84	0.64	0.74	-
4-class	Kaggle Baseline [13]	0.832	0.665	0.748	42M
4-class	**Our system (CNN model)**	**0.861**	**0.691**	**0.776**	4.9M
4-class	**Our system (SE-8cycle)**	**0.873**	**0.694**	**0.784**	39M
2-class	LSTM [8]	-	-	0.81	-
2-class	Kaggle Baseline [13]	0.832	0.796	0.814	42M
2-class	**Our system (CNN model)**	**0.861**	**0.804**	**0.832**	4.9M
2-class	**Our system (SE-8cycle)**	**0.873**	**0.801**	**0.837**	39M

Table IV shows the comparison of our models to state-of-the-art systems for both classification tasks. We can see that our best system achieves a significantly better performance. Our results and results in [13] are averaged over 10 independent training/testing splittings. The number of trained parameters of the proposed CNN model is around 10 times smaller compared to the baseline model in [13].

V. CONCLUSIONS AND FUTURE WORK

We propose *sample padding* and *feature splitting* for feature pre-processing. Both of these techniques improve our classification performance. Furthermore, the snapshot ensemble of the CNNs enhance performance of the algorithm at moderate additional training cost. In addition, data augmentation and the focal loss objective are used to increase the model performance. Our systems outperform other state-of-the-art lung sound classification systems for the 4-class and 2-class tasks and achieve the best performance at 78.4% and 83.7% ICBHI score, respectively.

Future work focuses on other deep learning methods for classification of respiratory adventitious sounds and diseases. Furthermore, we plan to collect more clinical data for the multi-channel lung sound database in [10] and extend the multi-channel processing framework to this database.

References

[1] D. Bardou, K. Zhang, and S. M. Ahmad, "Lung sounds classification using convolutional neural networks," *Artificial intelligence in medicine*, vol. 88, pp. 58–69, 2018.

[2] Arati Gurung, Carolyn G Scrafford, James M Tielsch, Orin S Levine, and William Checkley, "Computerized lung sound analysis as diagnostic aid for the detection of abnormal lung sounds: a systematic review and meta-analysis," *Respiratory medicine*, vol. 105, no. 9, pp. 1396–1403, 2011.

[3] LP Malmberg, K Kallio, S Haltsonen, T Katila, and ARA Sovijärvi, "Classification of lung sounds in patients with asthma, emphysema, fibrosing alveolitis and healthy lungs by using self-organizing maps," *Clinical Physiology*, vol. 16, no. 2, pp. 115–129, 1996.

[4] M. Bahoura, "Pattern recognition methods applied to respiratory sounds classification into normal and wheeze classes," *Computers in biology and medicine*, vol. 39, no. 9, pp. 824–843, 2009.

[5] P. Bokov, B. Mahut, P. Flaud, and C. Delclaux, "Wheezing recognition algorithm using recordings of respiratory sounds at the mouth in a pediatric population," *Computers in biology and medicine*, vol. 70, pp. 40–50, 2016.

[6] 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. 32845–32852, 2019.

[7] 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*. Springer, 2018, pp. 208–217.

[8] D. Perna and A. Tagarelli, "Deep auscultation: Predicting respiratory anomalies and diseases via recurrent neural networks," in *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 2019, pp. 50–55.

[9] L. Shi, K. Du, C. Zhang, H. Ma, and W. Yan, "Lung sound recognition algorithm based on vggish-bigru," *IEEE Access*, vol. 7, pp. 139438–139449, 2019.

[10] 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 *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 356–359.

[11] BM Rocha, D Filos, L Mendes, I Vogiatzis, E Perantoni, E Kaimakamis, P Natsiavas, A Oliveira, C Jácome, A Marques, et al., "A respiratory sound database for the development of automated classification," in *Precision Medicine Powered by pHealth and Connected Health*, pp. 33–37. Springer, 2018.

[12] Gorkem S., Sezer U., and Yasemin P. K., "An automated lung sound preprocessing and classification system based on spectral analysis methods," 2018.

[13] "https://www.kaggle.com/eatmygoose/cnn-detection-of-wheezes-and-crackles," 2019.

[14] Na. Jaitly and G. E. Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, 2013, vol. 117.

[15] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q Weinberger, "Snapshot ensembles: Train 1, get m for free," *arXiv preprint arXiv:1704.00109*, 2017.

[16] T.Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.