Deep Feature Learning for Medical Acoustics

Alessandro Maria Poirè, Federico Simonetta, and Stavros Ntalampiras[✉]

LIM – Music Informatics Laboratory, Department of Computer Science,
University of Milano, Milan, Italy
stavros.ntalampiras@unimi.it
https://www.lim.di.unimi.it/

Abstract. The purpose of this paper is to compare different learnable frontends in medical acoustics tasks. A framework has been implemented to classify human respiratory sounds and heartbeats in two categories, i.e. healthy or affected by pathologies. After obtaining two suitable datasets, we proceeded to classify the sounds using two learnable state-of-art frontends – LEAF and nnAudio – plus a non-learnable baseline frontend, i.e. Mel-filterbanks. The computed features are then fed into two different CNN models, namely VGG16 and EfficientNet. The frontends are carefully benchmarked in terms of the number of parameters, computational resources, and effectiveness.

This work demonstrates how the integration of learnable frontends in neuralaudio classification systems may improve performance, especially in the field of medical acoustics. However, the usage of such frameworks makes the needed amount of data even larger. Consequently, they are useful if the amount of data available for training is adequately large to assist the feature learning process.

1 Introduction

Cardiovascular and respiratory diseases are the leading cause of mortality worldwide; it is estimated that in 2019 17.9 million people died due to cardiovascular diseases, representing the first and second cause of death worldwide (32 % of all deaths worldwide), followed by respiratory disease [23,24]. Therefore, considerable efforts have been devoted to research for the improvement of the early diagnosis and routine monitoring of patients with cardiovascular and respiratory diseases. A large portion of the research has focused on the auscultation of respiratory sounds and heart tones. Indeed, these diseases, such as asthma, COPD, pneumonia, heart murmurs, heart valve abnormalities, and arrhythmia, are associated with distinct sound patterns. Such abnormal breathing sounds in the lungs are called adventitious sounds [24]. A similar phenomenon can be observed relatively to abnormal blood flows in the heart, which can also cause characteristics noises.

To the purpose of screening these cardiovascular and respiratory diseases, cardiac auscultation by phonocardiograms (PCG) and pulmonary auscultation

are among the most important tools. The auscultation happens via an electronic stethoscope capable of digitally recording PCGs and respiratory sounds. However, this process is based on the availability of an expert as well as on his degree of competence. Thus, the need to automate the diagnosis process has arisen in recent years, bringing the development of algorithms able of classifying heart or pulmonary sounds. Such algorithms are usually based on Machine Learning technologies with the aim of assisting physicians in the diagnosis of health diseases, as well as providing patients with effective auto-diagnosis tools where physicians are not available [15,16].

Artificial Neural Networks (ANN) comprise the most used approach for the classification of heart and pulmonary sounds [5]. ANNs require discriminating features of the signal as input; usually such features are time-frequency representations of an audio signal, such as spectrograms, Mel spectrograms and Mel-frequency cepstral coefficient (MFCC) [1,9,13]. Recent studies, regarding sound classification in general, show that using Log-Mel spectrograms has significant improvements on the efficiency of the neural network [7,18]. Some studies also adopted Wavelet-based representations, but such features were only little explored compared to FFT-based ones [6].

The goal of this work is the comparison of various frontends, i.e. feature-extraction methods. Indeed, various frontends for neural features extraction were recently proposed in the field of audio signal processing. Specifically, two learnable frontends for audio processing received a large attention – LEAF and nnAudio [4,25]. Both the two frontends allow to compute time-frequency representations specifically crafted for the learning problem. This study aims at assessing if the learned features can improve the efficiency of ANN for Medical Acoustics and therefore we compare them to a standard representation method based on Log-Mel-spectrograms.

The contributions of this work are:

- a binary classification method for respiratory and heartbeat sounds;
- comparison of LEAF and nnAudio frameworks with traditional hand-crafted features for audio processing;
- efficiency and effectiveness benchmarks of different feature extraction strategies using different types of Neural Networks;

To the sake of reproducibility, the source-code used for this work is fully available online[1].

2   The Considered Frontends

As mentioned above, the purpose of this paper is to compare two learnable frontends, LEAF and nnAudio. LEAF and nnAudio are features extractors that, unlike Mel-filterbank, are completely trainable during the neural network training process. Interestingly, all audio features extraction operations, such as filtering, pooling, compression and normalization are learnable.

---

[1] https://github.com/LIMUNIMI/Feature-Learning-Medical-Acoustics.
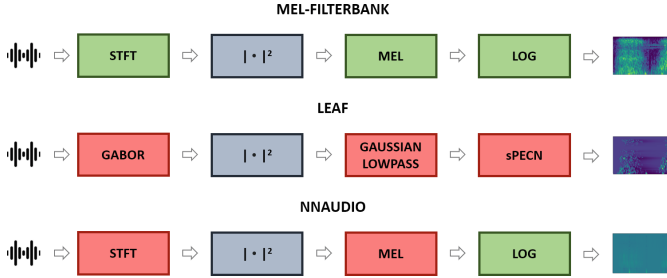
Fig. 1. Breakdown of Mel-filterbanks, LEAF and nnAudio frontends. Orange boxes are fixed, while computations in blue boxes are learnable. Grey boxes represent activation functions. (Color figure online)

Log-Mel spectrograms are the most used time-frequency representation for neural classification tasks in the field of medical acoustics [1,11,12,14,21]; it is for this reason that they have been chosen as a baseline for comparing representations produced by learnable frontends.

The three frontends are depicted in Fig. 1.

2.1 Mel-filterbanks

Mel-filterbank is a fixed frontend that receives waveforms as an input, and produces Log-Mel spectrograms as output. It is fixed because the parameters that control it are non-learnable, that is, they do not change during the training of the network.

A Mel-filterbank is applied to an audio excerpt to obtain the Log-Mel spectrograms. More specifically, we first compute the spectrogram of the audio excerpt using the squared modulus of the short-term Fourier transform (STFT). Then, the spectrogram is passed through a bank of triangular bandpass filters, spaced logarithmically according to the Mel scale. The Mel scale is designed to replicate the non-linearity of human pitch perception. Finally, to reflect the non-linearity of the human loudness sensitivity, the resulting coefficients are passed through a logarithmic compression.

Log-Mel spectrograms have various parameters that should be finely tuned, adding a large number of hyper-parameters to the resulting pipeline and making the designing of Machine Learning methods more complex.

2.2 LEAF

LEAF (LEarnable Audio Frontend) [25] is a neural network-based frontend to extract features such as Mel spectrograms. Being a neural network, it can be trained inside any neural architecture to discover task-specific features, adding only a few parameters to the model. This frontend learns all operations of audio features extraction, from filtering to pooling, compression and normalization.

In the first stage of filtering the sound wave passed through a bank of Gabor bandpass filters followed by a non-linearity. Then, the temporal resolution of the signal is reduced in the "pooling" phase. Finally, the dynamic range is optimized with a compression and/or normalization stage.

2.3   nnAudio

nnAudio (neural network Audio) [4] is a neural network based frontend able to extract Mel spectrograms as features. nnAudio uses convolutional neural networks to perform the conversion from time domain to frequency domain, and it can be trained together with any classifier.

As input, nnAudio receives a waveform from which it extracts the Mel-spectrogram via a learnable process. The frontend first computes the STFT using a Convolutional Neural Network, and then applies a bank of Mel filters. The values of the Mel filter bank are used to initialize the weights of a single-layer fully-connected neural network. Each time step of the STFT is sent in this fully-connected layer initialized with Mel weights. The Mel filter bank therefore must only be created during the initialization of the neural network. All of these weights are trainable.

3   Models

To test the frontends under different conditions, two different well-known CNNs were chosen for the classification phase: *EfficientNet-B0* and *VGG16* [19,20].

3.1   EfficientNet

The EfficientNet models are a family of artificial neural networks where the basic building block is the Mobile Inverted Bottleneck Conv Block (MBConv). The Efficient-Net family includes 8 models (from B0 to B7): as the number increases, the complexity of the network increases. The main idea of EfficientNet is to start from one simple, compact and computationally efficient structure, and gradually increasing its complexity. Unlike other CNN models, EfficientNet uses a new activation feature known as Swish, rather than the classic ReLU function. The "lightweight" version of EfficientNet (EfficientNetB0, with ∼4M parameters) has been adopted as first classifier.

3.2   VGG

The VGG16 version of VGG was adopted as another classifier. VGG stands for Visual Geometry Group; It is a standard multi-level CNN architecture. According to the number of layers, the various versions of VGG are named, for example VGG11 has 11 layers, VGG16 has 16 layers, VGG19 has 19 layers and so on.

VGG16 is a deep 16-layer neural network; this means that it is quite large, and has a total of about 138 million parameters. However, its architecture is relatively simple and straightforward.

Table 1. Statistics on the datasets used

		Respiratory sound database [2]	Heartbeat physio-net database [3]
Normal samples	Train set	2732	20461
	Validation set	546	4092
	Test set	364	2728
	Total	**3642**	**27281**
Abnormal Samples	Train set	2442	5284
	Validation set	488	1057
	Test set	326	704
	Total	**3256**	**7045**
Total		**6898**	**34326**

4 Datasets

In order to compare the proposed frontends, tests were performed for two different medical acoustics tasks: anomaly detection in respiratory sounds and in heartbeat recordings. The datasets differ in content and quantity of elements, so that frontends can be tested under different conditions.

Table 1 shows summary statistics about the used datasets.

4.1 Respiratory Dataset

The first database is the Respiratory Sound database [2], created to support the scientific challenge organized at the International Conference on Biomedical Health Informatics - ICBHI 2017 [8].

The database consists of a total of 5.5 h of records containing 6898 respiratory cycles, of which 1864 contain crackles, 886 contain wheezes and 506 contain both crackles and wheezes.

The total number of audio samples was 920, obtained from 126 participants. The recordings were collected using heterogeneous equipment and their duration ranged from 10 to 90 s. For each audio recording, the time-mark list of start and end time of each respiratory cycle is provided. The sampling frequency of the recordings varies, with values of 4 kHz, 10 kHz or 44.1 kHz; in the preprocessing phase they are resampled at 4 kHz. It is currently the largest publicly available respiratory sound database.

The level of noises – cough, speech, heartbeat, etc. – in some breathing cycles is relatively high representing real-life conditions very well. Respiratory cycles were noted by experts, dividing them into four categories: crackles, wheezes, a combination of them, or no adventitious sounds.

In this work we have chosen to use only two labels: normal and abnormal. The normal class covers sounds categorized as non-adventitious, while the abnormal class includes sounds containing crackles, wheezes, or a combination of them. In this way, the two resulting classes are more balanced, accounting 3642 normal sounds and 3256 abnormal cycles.
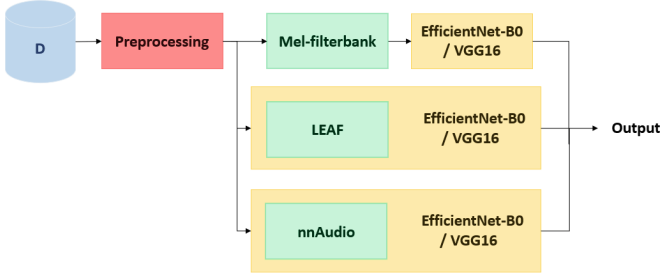
Fig. 2. General scheme of the neural network architecture.

4.2 Heartbeat Dataset

The second database is the Heartbeat Physio-Net Database [3], created specifically for the 2016 PhysioNet Computing in Cardiology (CinC) Challenge [17].

This database contains a total of 3153 heart sound recordings from 764 healthy and pathological patients. The recordings have a duration from 5 to 120 s, obtaining about 25 h of sound material. All audio samples were recorded with a sampling rate of 2 kHz or resampled to the same rate. In the database there are recordings labeled as unsure, i.e. with a very low signal quality. These audio samples were omitted from the test, leaving a total of 2872 recordings for the training, validation and testing phases. All phonocardiograms in the database are categorized into two types: normal and abnormal. Recordings with the normal label come from healthy patients, while those with the abnormal label come from patients with pathologies such as coronary artery disease and heart valve defects (mitral valve prolapse syndrome, mitral regurgitation, aortic regurgitation, aortic stenosis and valve surgery).

As in the Respiratory Database, the data includes not only "clean" heart sounds, but also very noisy recordings, providing an accurate representation of real life conditions.

5 Experiments

This section describes the experiments performed to compare the various frontends and models described in Sects. 2 and 3. The generic workflow is shown in Fig. 2.

5.1 Pre-processing

Pre-processing included segmentation, filtering, resampling, and padding.

For the respiratory database, the samples were segmented following the time-marks annotations indicating each respiratory cycle – see Sect. 4.1. For instance, the audio file `107_2b5_Pr_mc_AKGC417L.wav` is 8.97 s long; it has been segmented according to the indicated time-marks thus producing 4 audio files: `file_1.wav`
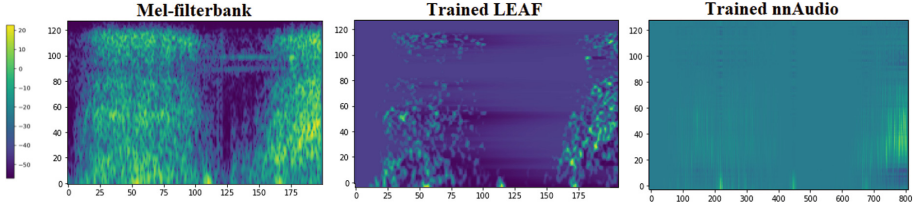
Fig. 3. Features learned on a sample from the Respiratory Sound Database [2] by LEAF and nnAudio compared with classical Mel-spectrograms.

(from 0.077 to 1.411 of the original file), `file_2.wav` (from 1.411 to 3.863), `file_3.wav` (from 3.863 to 6.601) and `file_4.wav` (from 6.601 to 8.97).

For the heartbeat database, instead, the individual audio files of variable length (from 6 to 120 s) were segmented into 2-s files. For instance, a file initially lasting 10 s is split into 5 files of 2 s each. In this way, the total number of samples becomes 34326.

Subsequently, the audio files obtained from the segmentation phase are filtered through a 12th order Butterworth band-pass filter, with cut-off frequencies [120–1800] Hz for the respiratory database, and [25, 400] Hz for the heartbeat database. This eliminates the components of sound caused by coughing, intestinal noises, stethoscope movement and speech.

All audio files were then resampled at 4 KHz (only in the Respiratory dataset, the Heartbeat dataset was already sampled at 4 kHz) and truncated or zero-padded so that they lasted exactly 2 s.

5.2 System Parameterization

To better compare the frontends considered, the same hyper-parameters were used in the tests.

In all three frontends, after various experiments, it was decided to use a window length of 30 ms, with a window stride of 10 ms. The frequency range of the Mel filters was set at [100, 2000] Hz in the tests with respiratory dataset, and [25, 1000] Hz in the tests with heratbeat dataset. 128 Mel-filters were used. Only in the LEAF frontend some parameters have been changed in respect to their factory defaut values; specifically, in the PCEN compression layer, the alpha and root parameters that control the amount of compression applied, respectively alpha = 2, root = 4.

In the VGG16 classifier, two dropout layers with a value of 0.5 have been added between the last two fully connected layers. The dropout layer prevents the co-adaptation of a neural network, disabling some nodes of the network during the training phase with a specific probability (0.5 in this case). In EfficientNetB0 the drop connect rate parameter has also been changed, setting it to 0.5.

The training was carried out considering a period of 300 epochs for the heartbeat test, and 200 epochs for the respiratory test. The batch size was set at 64,
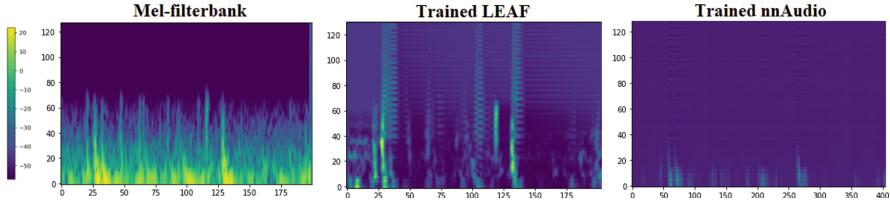
Fig. 4. Features learned on a sample from the heartbeat physio-net database [3] by LEAF and nnAudio compared with classical Mel-spectrograms.

while the learning rate as set at $1e-5$, and ADAM was chosen as weight-update algorithm.

We empirically found an optimal split size using 75% of the dataset for the train set, 15% for the validation set, and 10% for the test set.

Examples of features learned on the two datasets are shown in Figs. 3 and 4.

6   Results

In order to compare the proposed frontends, two tests with two different datasets were formulated: "Test 1 - Respiratory" and "Test 2 - Heartbeat". The datasets differ in content and quantity of elements, so the frontends can be tested under different conditions.

The evaluation metrics used in this study are balanced accuracy, True Positive Rate (TPR) and True Negative Rate (TNR):

$$BalancedAccuracy = \frac{TPR + TNR}{2}$$
$$TPR = \frac{TruePositive}{TruePositive + FalseNegative} \tag{1}$$
$$TNR = \frac{TrueNegative}{TrueNegative + FalsePositive}$$

Table 2. McNemar p-values corrected with Bonferroni-Holm method.

	Mel-LEAF	Mel-nnAudio	LEAF-nnAudio
Resp. Eff.	0.2010	0.0500	0.5263
Resp. VGG	0.3998	0.0366	0.3998
Heart. Eff.	9.8515e–07	0.4194	2.6648e–05
Heart. VGG	1.1219e–03	9.1734e–09	7.1666e–03

6.1   Test 1 - Respiratory

As shown in Tables 3 and 4, VGG16 always achieves better results than EfficientNet. However, using VGG16, the differences between the three frontends are larger, acheiving 6% of difference in accuracy between nnAudio and Mel-filterbank.

Surprisingly, we found that with VGG16 the baseline method outperforms the learnable frontends, proving the well-design of old Log-Mel spectrograms compared to newer neural network frameworks.

When EfficientNet is used, instead, a small difference emerges that awards the learnable frontends, especially nnAudio; however, McNemar test with Bonferroni-Holm correction finds no statistically significant difference among the three.

Specific p-values are shown in Table 2.

6.2   Test 2 - Heartbeat

Even in this scenario VGG16 was better than EfficientNet in all the tests. Nevertheless and contrarily to the respiratory task, the Mel-filterbank was surpassed by both nnAudio and LEAF.

Tables 5 and 6 show that LEAF achieves the better accuracy using both VGG16 and EfficientNet. However, when using EfficientNet, the best TNR was achieved by nnAudio. Note that TNR is particularly important in first-screening diagnosis, because low TNR is associated with a high false negative rate, meaning that false negatives are common. When a false negative prediction happens, the therapeutic intervention may be delayed with catastrophic consequences.

Specific p-values are shown in Table 2.

6.3   Overall

Comparing the results obtained from the two tests Test 1 and Test 2, we note that the best scores were achieved in Test 2 (Heartbeat), with the LEAF frontend. We theorize that LEAF performed better in Test 2 than Test 1 due to the size of the phonocardiogram database, which is much larger than the respiratory sound database.

Table 3. Comparison results using VGG16 as classifier on the ICBHI dataset [2]. Only Mel-filterbank and nnAudio accuracies show a statistical significance ($p \sim 0.04$ using McNemar with Bonferroni-Holm correction).

%	Balanced accuracy	TPR	TNR
Mel-filterbank	**80.21**	**81.42**	**79.01**
LEAF	77.47	80.87	74.07
nnAudio	74.12	77.86	70.37

Table 4. Comparison results using EfficientNet-B0 as classifier on the ICBHI dataset [2]. Only Mel-filterbank and nnAudio accuracies show a small significance ($p \sim 0.05$ using McNemar with Bonferroni-Holm correction).

%	Balanced accuracy	TPR	TNR
Mel-filterbank	61.05	62.84	**59.26**
LEAF	61.40	66.94	55.86
nnAudio	**61.74**	**68.85**	54.63

Table 5. Comparison results using VGG16 as classifier on the heart beats dataset [3]. All p-values between accuracies are <<0.05 (McNemar with Bonferroni-Holm correction)

%	Balanced accuracy	TPR	TNR
Mel-filterbank	90.71	**96.22**	85.20
LEAF	**92.29**	95.29	**89.30**
nnAudio	91.41	93.64	89.16

Moreover, the different balancing of the two databases is probably the reason why TPR and TNR are more distant in Test 2 than in Test 1. Indeed, the Respiratory database has the most balanced classes compared to the Heartbeat database – see Table 1.

In general, the 3 frontends learn different features, as shown in Figs. 3 and 4. Namely, nnAudio learns more sparse representations that focus on low frequencies. On the contrary, LEAF learns representations less sparse and well distributed across the frequency space. Compared with classical Mel-filterbanks, both of them seems to learn specific characteristics that are relevant for the classification. We theorize that LEAF learned features work by extracting discriminative local descriptors in the time-frequency space similarly to audio fingerprint algorithms [10,22]. nnAudio, instead, extracts blurred regions that are likely less characteristics of the single excerpt. Moreover, LEAF manages to handle both positive and negative values, while nnAudio's activation functions only return non-negative values, thus deleting possibly useful information.

Table 6. Comparison results using EfficientNet-B0 as classifier on the heart beats dataset [3]. All comparisons of accuracies revealed statical significance except between Mel-filterbank and nnAudio (McNemar test with Bonferroni-Holm correction).

%	Balanced accuracy	TPR	TNR
Mel-filterbank	81.12	90.92	71.33
LEAF	**84.36**	**95.40**	73.31
nnAudio	83.51	92.52	**74.50**

7   Conclusion

This work has shown how the integration of learnable frontends in classification systems with convolutional neural networks can improve results in the field of medical acoustics. The tests carried out show that learnable frontends are particularly useful when there is a sufficient amount of available data (Test 2), while using small data-sets (Test 1) prevent them from learning accurate features to surpass the classic hand-crafted methods.

The proposed method therefore stands as a valid alternative to traditional feature extraction methods as long as they are used in contexts with a large amount of data available.

References

1. Acharya, J., Basu, A.: Deep neural network for respiratory sound classification in wearable devices enabled by patient specific model tuning. IEEE Trans. Biomed. Circuits Syst. (2020)
2. Rocha, B.M., et al.: An open access database for the evaluation of respiratory sound classification algorithms. Physiol. Meas. **40**, 035001 (2019)
3. Liu, C., et al.: An open access database for the evaluation of heart sound algorithms. Physiol. Meas. **37**, 2181 (2016)
4. Cheuk, K.W., Anderson, H., Agres, K., Herremans, D.: nnaudio: An on-the-fly GPU audio to spectrogram conversion toolbox using 1d convolutional neural networks. IEEE Access (2020)
5. Clifford, G.D., et al.: Classification of normal/abnormal heart sound recordings: the physionet/computing in cardiology challenge 2016. PhysioNet (2016)
6. Serbes, G., Ulukaya, Khaya, Y.P.: An Automated Lung Sound Preprocessing and Classification System Based on Spectral Analysis Methods. Precision Medicine Powered by pHealth and Connected Health, Springer, Singapore (2018). https://www.springerprofessional.de/en/an-automated-lung-sound-preprocessing-and-classification-system-/15232246
7. Huzaifah, M.: Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. arXiv:1706.07156v1 (2017)
8. ICBHI: ICBHI 2017 challenge. https://bhichallenge.med.auth.gr/ICBHI_2017_Challenge. Accessed 10 2021
9. Kamarulafizam, I., Salleh, S., Jamaludin, M.N.: Heart sound analysis using MFCC and time frequency distribution. In: 3rd Kuala Lumpur International Conference on Biomedical Engineering 2006, pp. 402–405 (2007)
10. Ke, Y., Hoiem, D., Sukthankar, R.: Computer vision for music identification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005) vol. 1, pp. 597–604 (2005)
11. Meghanani, A., Anoop, C.S., Ramakrishnan, A.G.: An exploration of log-mel spectrogram and MFCC features for Alzheimer's dementia recognition from spontaneous speech. In: IEEE Spoken Language Technology Workshop (SLT) (2021). https://doi.org/10.1109/SLT48900.2021.9383491
12. Mukherjee, U., Pancholi, S.: A visual domain transfer learning approach for heartbeat sound classification (2021). https://doi.org/10.48550/ARXIV.2107.13237

13. Nakamura, N., Yamashita, M., Matsunaga, S.: Detection of patients considering observation frequency of continuous and discontinuous adventitious sounds in lung sounds. In: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2016)
14. Nguyen, T., Pernkopf, F.: Lung sound classification using snapshot ensemble of convolutional neural networks. In: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC) (2020). https://doi.org/10.1109/EMBC44109.2020.9176076
15. Ntalampiras, S.: Collaborative framework for automatic classification of respiratory sounds. IET Signal Process. **14**(4), 223–228 (2020). https://doi.org/10.1049/iet-spr.2019.0487
16. Ntalampiras, S.: Identification of anomalous phonocardiograms based on universal probabilistic modeling. IEEE Lett. Comput. Soc. **3**(2), 50–53 (2020). https://doi.org/10.1109/LOCS.2020.3014306
17. PhysioNet: Classification of heart sound recordings: The physionet/computing in cardiology challenge 2016. https://physionet.org/content/challenge-2016/1.0.0/. Accessed 10 2021
18. Purwins, H., Li, B., Virtanen, T., Schluter, J., Chang, S.Y., Sainath, T.: Deep learning for audio signal processing. IEEE J. Sel. Topics Signal Process. **13**(2), 206–219 (2019). https://doi.org/10.1109/jstsp.2019.2908700
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556v6 (2015)
20. Tan, M., Le, Q.V.: EfficientNet: rethinking model scaling for convolutional neural networks. arXiv:1905.11946v5 (2020)
21. Ul Haq, H.F.D., et al.: Efficientnet optimization on heartbeats sound classification. In: 2021 5th International Conference on Informatics and Computational Sciences (ICICoS) (2021). https://doi.org/10.1109/ICICoS53627.2021.9651818
22. Wang, A.: An industrial strength audio search algorithm. In: ISMIR (2003)
23. WHO: The top 10 causes of death. https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death. Accessed 03 2022
24. WHO: World statistics on cardiovascular disease. https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds). Accessed 03 2022
25. Zeghidour, N., Teboul, O., de Chaumont Quitry, F., Tagliasacchi, M.: Leaf: a learnable frontend for audio classification. In: ICLR (2021)