AI Engineer/Researcher Take-Home Project - Amplifier Health

Davide Salvi

January 16, 2025

1 Introduction

In this report, we detail the approach taken to address the Take-Home Project for the AI Engineer/Researcher role at Amplifier Health.

The task involved an exploratory analysis of the ICBHI 2017 Challenge Dataset and the development of a system able to classify the data it contains. Since I lacked prior experience with this data domain, I conducted an exploration to identify the most suitable methods to process this kind of audio signals, both regarding model architectures and feature sets. The goal was to identify the best-performing combinations for classifying the audio data. The task was approached from two perspectives:

- **Binary Classification**: Classifying audio into healthy vs. unhealthy categories.
- **Multi-Class Classification**: Directly identifying the specific disease the patient is suffering from.

While the results achieved are promising, there is significant room for improvement. Enhancements could be made both in the data preprocessing pipeline and in refining the model architectures to better capture the characteristics of the audio signals. This report provides an in-depth explanation of the methodology, experiments, and results, along with potential areas for future development.

2 Problem Formulation

The problem we address can be formally defined as follows. Let us consider a discrete-time input signal $\mathbf{x}$, sampled with a sampling frequency f_s and respiratory sound recordings. We consider $\mathbf{x}$ as associated with a class y, which indicates the respiratory conditions of the patient under analysis. Our goal is to develop a detection model $\mathcal{M}$ that takes $\mathbf{x}$ as input and outputs an estimate of its class as

$$\hat{y} = \mathcal{M}(\mathbf{x}). \tag{1}$$

We consider the detection pipeline of $\mathcal{M}$ as divided into two steps, as shown in Figure 1. The first step is the *feature extraction*, followed by the *classification*. During *feature extraction*, the input audio signal is transformed into a feature vector that captures its essential characteristics. Formally, we define the extractor $\mathcal{E}$ such that $\mathbf{f} = \mathcal{E}(\mathbf{x})$, where $\mathbf{f}$ represents the extracted feature vector. In the *classification* step, the feature vector is fed to a CNN-based model, which processes it to perform binary classification, outputting the predicted class of the input signal.

The overall model $\mathcal{M}$ is obtained by the combination of the feature extractor and the classifier as

$$\hat{y} = \mathcal{M}(\mathbf{x}) = \mathcal{C}(\mathcal{E}(\mathbf{x})). \tag{2}$$

Our goal is to investigate how the performance of $\mathcal{M}$ varies when different feature extractors $\mathcal{E}$ are used.

To ensure our findings are generalizable and not limited to a single model, we conduct this analysis using two different CNN-based detectors.

The number of classes y that are considered in the analysis varies depending on how the problem is formulated. In the initial analysis, we simplify the task to a binary classification problem, where the goal is to determine whether the input signal corresponds to a healthy or sick patient. In the subsequent analysis, we increase the number of classes, training the model to identify the specific type of disease affecting the patient.

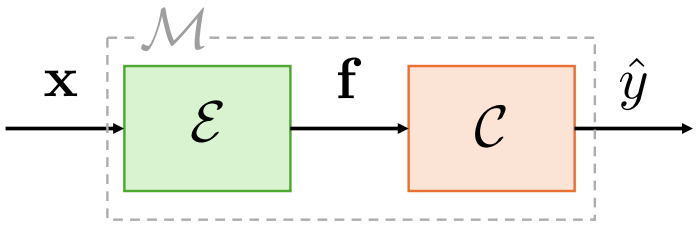

Figure 1: Considered pipeline for the developed system. The extractor $\mathcal{E}$ computes features $\mathbf{f}$ from the input audio $\mathbf{x}$. Features are fed to the classifier $\mathcal{C}$ that estimates the fake singing likelihood $\hat{y}$.

3 Experimental Setup

3.1 Audio Features

As discussed above, our analysis includes various feature sets, as it is the first time I work with this kind of data and I still do not know which is the best way to process them. In particular, I considered four different hand-crafted feature sets that are highly recognized in signal processing, due to their capability to offer compact yet insightful representations of an audio signal's spectrum. These are designed to capture different aspects of the audio signal's characteristics, and each may lead to varying results in the task at hand. The feature sets include: *log-spectrograms*, which represent the logarithm of the audio signal's power spectrum; *mel-spectrograms*, which use an 80-band mel-filterbank to capture frequency components; *LFCC*, extracted with 40 coefficients along with their first and second-order derivatives; and *MFCC*, computed using a mel-filterbank and the same configuration as LFCCs. All the feature sets have been computed using a window length of 25 ms and a hop size of 10 ms.

3.2 Classifiers

We utilize two CNN-based classifiers to assess the impact of different audio features on classification performance. This approach ensures that the results are robust and generalizable across multiple models, enhancing the overall relevance of our findings.

ResNet-18 He et al. 2016. This CNN model encompasses 18 layers that exploit residual connections to effectively address the vanishing gradient problem. We considered the version of this model pre-trained on ImageNet, as it is known for its efficiency in image classification and has often been adapted to audio classification tasks.

LCNN Wu et al. 2018. This is a Light CNN model adapted to the speech deepfake detection task. It operates on 2D features and integrates several convolutional layers with varying kernel sizes and strides to capture different levels of abstraction, together with multi-level feature mapping techniques to enhance feature extraction.

3.3 Dataset

The dataset used in this project is the *ICBHI 2017 Respiratory Sound Database*, originally compiled for the ICBHI 2017 Challenge. It contains 5.5 hours of audio recordings from 126 subjects, annotated by respiratory experts for the presence of crackles, wheezes, or their combination.

We utilize the training and test partitions as these are provided by the challenge organizers. Additionally, a validation set is derived from 25% of the training data, stratifying labels to maintain the class distribution.

3.4 Training setup

We trained and tested the two considered models using the same setup across all the analyzed features, to guarantee the fairest condition possible in their evaluation. We trained the models for 150 epochs using a batch size of 32 samples. We used a CrossEntropy loss function and AdamW as optimizer. The learning rate followed a cosine annealing schedule, starting at 10^{-4} and gradually decaying to 10^{-7}

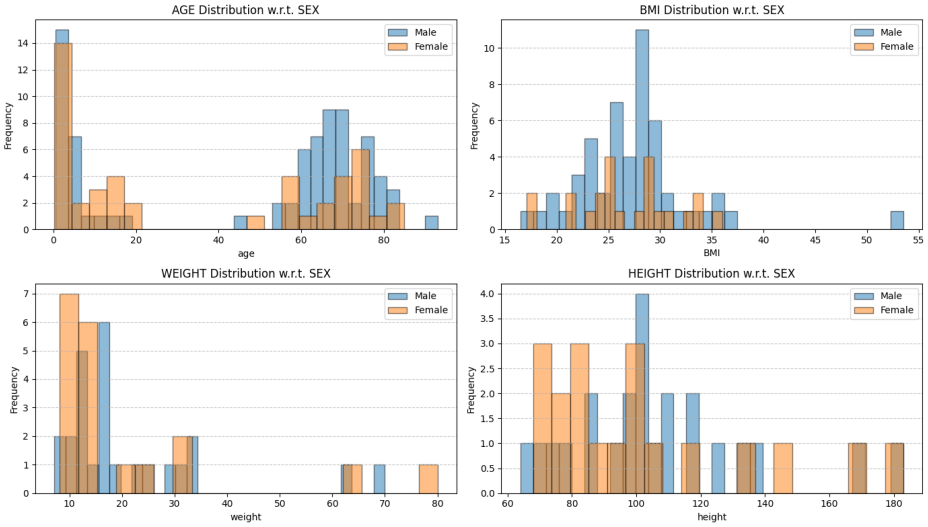

Figure 2: Distribution of the patients across different demographic parameters.

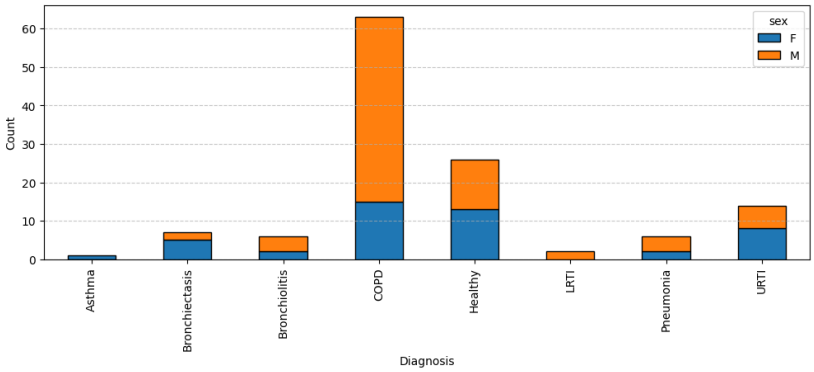

Figure 3: Diagnosis distribution across patients. The values are stacked by sex.

over the course of training. The best-performing model was selected based on the lowest validation loss value.

During training, each input signal is processed using a fixed-duration window (either 5 or 10 seconds, depending on the experiment). To enhance variability, the window is randomly selected from within the signal. During test, multiple windows are extracted from each test signal, and the model's predictions for these windows are averaged to improve the robustness and reliability of the final classification.

4 Data analysis

We conducted a demographic analysis of the patients, revealing that the majority (63%) are men, while only 37% are women. Despite this gender disparity, other attributes such as age, BMI, weight, and height are distributed comparably across both sexes (see Figure 2).

In contrast, a significant imbalance is observed in the distribution of diagnosis. Patients with Chronic Obstructive Pulmonary Disease (COPD) make up the largest group, far outnumbering those in other diagnostic categories (see Figure 3).

Given this strong imbalance in disease representation, we began by addressing a simpler case: binary classification. The goal of this initial analysis is to classify whether a patient is healthy or sick based on their breathing recordings. To conduct this experiment, we employed the setup described earlier, using 10-second input windows for the classifier. However, recognizing that the average duration of a single breathing cycle is under 3 seconds (see Figure 4), we also investigated

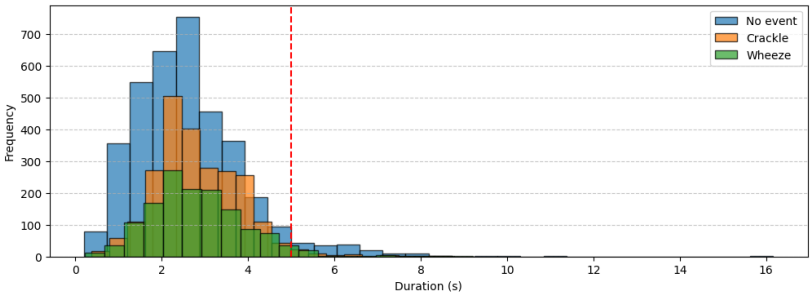

Figure 4: Distribution of the durations of respiratory cycles.

Table 1: EER and AUC values of the ResNet and LCNN models trained on the binary classification problem.

		MelSpec		LogSpec		MFCC		LFCC	
	Window Size	EER	AUC	EER	AUC	EER	AUC	EER	AUC
ResNet	5.0 sec	0.47	0.55	0.53	0.46	0.41	0.59	0.53	0.48
	10.0 sec	0.41	0.65	0.53	0.42	0.35	0.68	0.41	0.60
LCNN	5.0 sec	0.29	0.65	**0.24**	**0.79**	0.35	0.65	0.53	0.39
	10.0 sec	**0.06**	**0.98**	0.47	0.49	**0.29**	**0.75**	**0.24**	**0.88**

the impact of a shorter analysis window on classification performance. Specifically, we considered a 5-second input window, which is longer than 95% of all the respiratory cycles.

5 Classification results

In our first experiment, we evaluate the binary classification performance of the models across all their configurations. We use standard binary classification metrics to assess performance, such as Equal Error Rate (EER) and Area Under the ROC Curve (AUC). Table 1 presents the results of this analysis, while Figure 5 displays the ROC curves for the best-performing classifier, the LCNN trained on 10-second audio segments.

The Mel spectrogram emerges as the best-performing feature set in this experiment, with almost perfect classification performance. Among the models, the LCNN consistently outperforms the ResNet, demonstrating superior performance across various configurations. Finally, the differences between the models trained on 5-second and 10-second audio are highly variable, which may indicate some instability in the training process.

For the multi-class classification experiment, we use accuracy as the evaluation metric. Table 2 shows the performance of all model configurations for this task. As in the binary classification case, the Mel spectrogram is again the top-performing feature set. This may be due to the Mel spectrogram's ability to highlight information in the frequency range most relevant to speech. In contrast to the binary classification experiment, the ResNet model outperforms the LCNN. Figure 6 presents the confusion matrix for the best-performing model, which is the ResNet trained on the Mel spectrogram with 10-second audio windows.

The confusion matrix reveals that COPD is the most accurately identified class, which aligns with expectations, as it is the most populated class in the dataset. Additionally, several classes are absent from the test set, which limits the comprehensive evaluation of the trained models To address this issue, one potential solution would be to employ a more effectively stratified dataset split, as opposed to the division originally proposed by the challenge organizers.

Also, by analyzing the results shown in Table 2, we can notice how, unlike the binary classification analysis, the differences between models trained on 5-second and 10-second windows are minimal. This supports our initial hypothesis, derived from the analysis of Figure 4, that a shorter window can provide a sufficiently comprehensive representation of the respiratory cycles, resulting in acceptable performance while reducing computational costs.

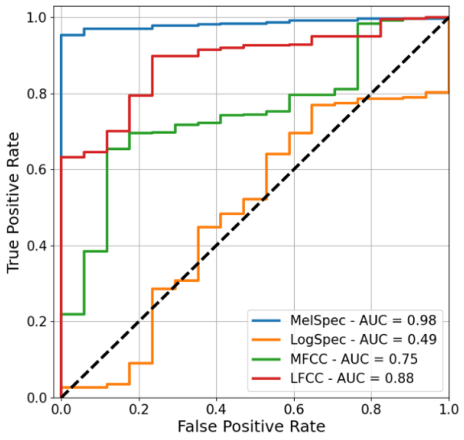

Figure 5: ROC curve of the LCNN model train on signals of 10 seconds with different features as input.

Table 2: Accuracy values of the ResNet and LCNN models trained on the multi-class classification problem.

	Window Size	MelSpec	LogSpec	MFCC	LFCC
ResNet	5.0 sec	0.93	0.92	0.93	0.92
	10.0 sec	**0.95**	0.93	0.92	0.91
LCNN	5.0 sec	0.89	0.74	0.66	0.86
	10.0 sec	0.88	0.75	0.86	0.83

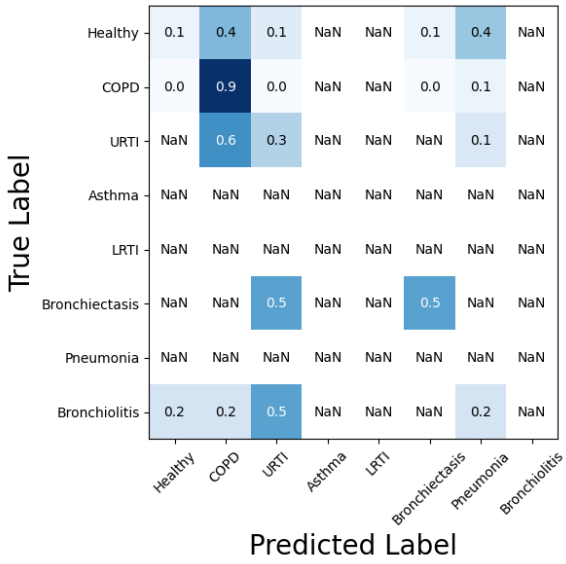

Figure 6: Confusion matrix showing the performance of the ResNet model trained on MelSpectrograms from 10-second signals.

6 Conclusions and possible future works

In this assignment, we explored the classification of respiratory sounds from the ICBHI 2017 Challenge dataset using various features and architectures. We tackled the problem by considering it as a binary and multi-class classification task. The key findings include:

- **Feature Set Performance**: Mel spectrograms consistently outperformed other feature sets (LogSpec, MFCC, and LFCC) in both binary and multi-class classification tasks. This indicates that Mel spectrograms capture more relevant information, particularly in the frequency range associated with speech, which aids in distinguishing respiratory conditions.
- **Model Performance**: There is not a model that is clearly outperforming the other. LCNN generally outperformed ResNet in the binary classification task, while the opposite happened in the multi-class classification scenario.
- **Window Size Impact**: The comparison of 5-second and 10-second input windows revealed minimal differences in performance. This suggests that shorter windows provide a sufficiently detailed representation of the respiratory cycles, leading to reduced computational costs without sacrificing accuracy.

The primary challenge I encountered while addressing this task was my limited knowledge of the data I had to work with and the best methods I could use to process them, both at the audio feature level and the model architectures. To overcome this issue, I adopted a broad and exploratory approach in my analysis, trying to be as flexible as possible. The results obtained have provided a foundation for building more advanced models and have offered valuable insights that I could exploit to work again on this task.

In future experiments, we could investigate deeper the proposed analysis, working on the data processing, model architecture and training setup. For instance, the dataset exhibits several limitations, including a significant class imbalance and an insufficient division between training and test sets. Addressing these issues by implementing better partitioning or incorporating data augmentation could significantly enhance model performance. From a training setup perspective, we observed considerable instability in some of the proposed training processes. A more thorough analysis of the model architectures could help stabilize the training process, particularly for binary classification tasks, and potentially improve overall classification accuracy. Lastly, several metadata were not incorporated into our implementation due to time constraints. For instance, expert annotations and information related to specific sound features like crackles and wheezes were excluded. Future iterations of this study could explore the impact of integrating these additional data points to assess their effect on classifier performance.

References

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wu, Xiang et al. (2018). "A light CNN for deep face representation with noisy labels". In: *IEEE Transactions on Information Forensics and Security* 13.11, pp. 2884–2896.