Abstract

This study investigates various audio preprocessing techniques for classifying lung sounds, particularly on Per-Channel Energy Normalization (PCEN). We combined and augmented two primary datasets to create a comprehensive set of labeled audio clips covering a range of respiratory conditions. Using these datasets, we pursued three classification tasks: disease diagnosis, distinguishing between wheezes, crackles, and normal sounds, and differentiating between normal and abnormal lung sounds. Each dataset was processed using several methods, including log-mel spectrograms, Mel-Frequency Cepstral Coefficients (MFCCs), and PCEN spectrograms. The datasets were then fed into a convolutional neural network (CNN) for training and evaluation. The CNN architecture utilizes 2D convolutional layers to learn frequency-time features while careful data splitting and augmentation minimize overfitting. Our results, which support our hypothesis, demonstrate that PCEN consistently outperformed MFCC and log-mel spectrogram methods across all datasets and evaluation metrics such as loss and accuracy. These results demonstrate PCEN's effectiveness in suppressing background noise and amplifying subtle lung sounds, leading to improved evaluation metrics. The most critical parameters were the smoothing coefficient (T), which controls temporal smoothing for signal stability, and root (r), which manages dynamic range compression to balance soft and loud sounds. While delta (δ) and alpha (α) also contributed significantly, epsilon (ε) had minimal impact. Despite requiring more computational effort and parameter tuning, PCEN's noise suppression and resistance to overfitting make it a powerful tool for automated lung sound diagnostics.

Introduction

Introduction to the Study

This study compares three audio preprocessing methods: log-mel spectrograms, Per-Channel Energy Normalization (PCEN), and Mel-Frequency Cepstral Coefficients (MFCCs). This comparison is conducted across three distinct lung-sound datasets. The first dataset encompasses multiple diagnostic categories, the second focuses on classifying lung sounds into crackles, wheezes, or normal sounds, and the third differentiates between normal and abnormal sounds. Additionally, we created a fourth dataset that does not include any augmentation; it consists solely of downsampled time-split audio data. This dataset allows for an analysis of the effect of data augmentation on PCEN.

Challenges in Lung Sound Diagnostics

Currently, medical lung sound analysis is an integral part of diagnosing and monitoring respiratory conditions such as asthma, bronchitis, pneumonia, and chronic obstructive pulmonary disease (COPD). "Auscultation, which is the process of listening to the internal sounds in the human body through a stethoscope, has been an effective tool for diagnosing lung disorders and abnormalities." [1] This process relies heavily on the physician's expertise. Through a stethoscope, physicians can identify normal breathing sounds, diminished or absent breath sounds, and abnormal sounds such as rales, rhonchi, squawks, stridor, wheezes, and

rubs. However, this process is inherently subjective, relying heavily on the experience and training of the clinician. Studies have shown that even among experienced physicians, there is significant variability in diagnosing respiratory conditions based on lung sound interpretation. [2] This highlights the need for a more objective, data-driven approach to lung sound analysis

In recent years, machine learning and signal processing have emerged as promising tools to improve lung sound classification's diagnostic accuracy and consistency. Central to these approaches is the preprocessing of raw audio data to extract meaningful features that can be fed into classification models. One of the most commonly used techniques in this context is the Mel-Frequency Cepstral Coefficients (MFCCs) method.

An explanation of Mel-Frequency Cepstral Coefficients (MFCCs)

MFCCs work by converting an audio signal into a spectrogram, which is generated through multiple Fast Fourier Transforms (FFTs). The spectrogram represents time on the x-axis, frequency on the y-axis, and magnitude as color intensity. The following steps are performed to create an MFCC representation: First, a mel-scale triangular filter bank is applied to the spectrogram to produce a mel spectrogram, emphasizing frequencies relevant to human hearing. Next, a logarithmic transformation is applied to the mel spectrogram, resulting in a mel-log spectrogram. Finally, the discrete cosine transform (DCT) is applied to extract the Mel-Frequency Cepstral Coefficients, which compactly represent the spectral characteristics of the sound. [3]

MFCCs are a standard feature in audio analysis due to their ability to capture the spectral profile of sound in a manner inspired by human auditory perception. However, despite their effectiveness in controlled environments, MFCCs are sensitive to noise, posing challenges in real-world medical settings. [3] Ambient noise from the environment or the patient can significantly affect the quality of recordings, reducing the accuracy and reliability of diagnostic models. Mel triangular filterbanks lack a robust theoretical basis, introducing additional variability [3]. For example, in our experiment, we used 40 mel triangular filterbanks, a standard number typically used for precise separation of frequencies in machine learning, but some other common numbers include 13 or 128. An example image of MFCC is shown below in Figure 1.

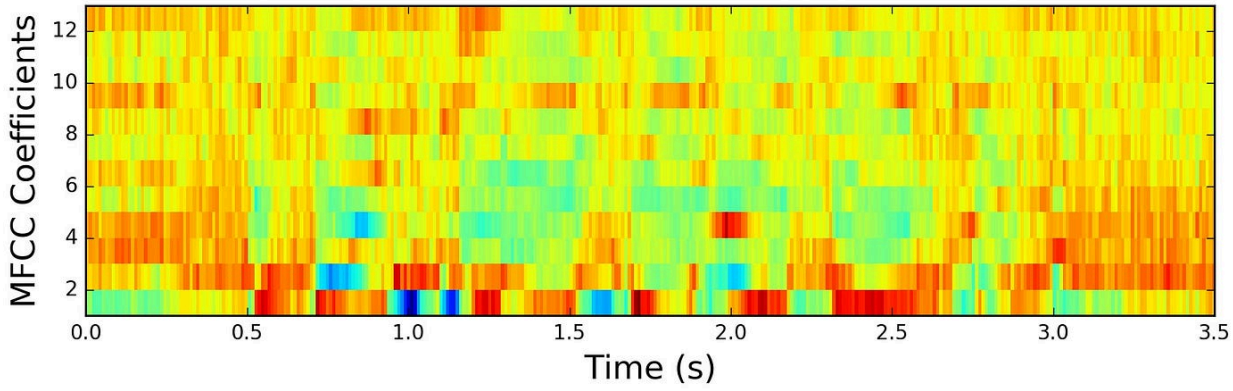


Figure 1: An example of what the mel-frequency cepstrum coefficient (MFCC) looks like. [4]

An explanation of spectrograms

Recent research suggests that spectrogram images derived directly from audio samples offer comparable or superior accuracy in classification tasks. A 2017 study on the classification of lung sounds using convolutional neural networks found that using spectrograms as an alternative to MFCC in a CNN architecture yielded equal, if not better, results across four datasets. “1) healthy versus pathological classification; (2) rale, rhonchus, and normal sound classification; (3) singular respiratory sound type classification; and (4) audio type classification with all sound types. Accuracy results of the experiments were; (1) CNN 86%, SVM 86%, (2) CNN 76%, SVM 75%, (3) CNN 80%, SVM 80%, and (4) CNN 62%, SVM 62%, respectively.” [1] However, spectrogram-based approaches and MFCCs are both highly susceptible to noise, which can negatively impact model performance in noisy environments. These limitations highlight the ongoing need for feature extraction methods that are more robust to noise and better suited for medical applications.

The generation of a spectrogram begins with a wave in the time domain—a signal that shows how amplitude varies over time. A wav (audio) file is divided into overlapping time windows to create the spectrogram. A Fast Fourier Transform (FFT) is applied for each window, converting the signal from the time domain to the frequency domain. This transformation reveals the present frequencies and their strengths within each time window.

The resulting spectrogram displays this information in three dimensions: time on the x-axis, frequency bands on the y-axis, and energy (commonly represented by color). Due to the waves' oscillating nature, the energy values are calculated as amplitude squared to make all values positive and provide a measure of energy at each time-frequency point.

Spectrograms work well with audio classification studies in neural networks due to their structure. By transforming audio into a 2D time-frequency representation, spectrograms create an image-like data point that aligns well with existing Convolutional Neural Network (CNN) architectures. [5] The nature of spectrograms captures high-definition frequency patterns for a short window and the whole audio sample, allowing neural networks to learn hierarchical features more effectively than from raw audio input. (Figure 2, below)

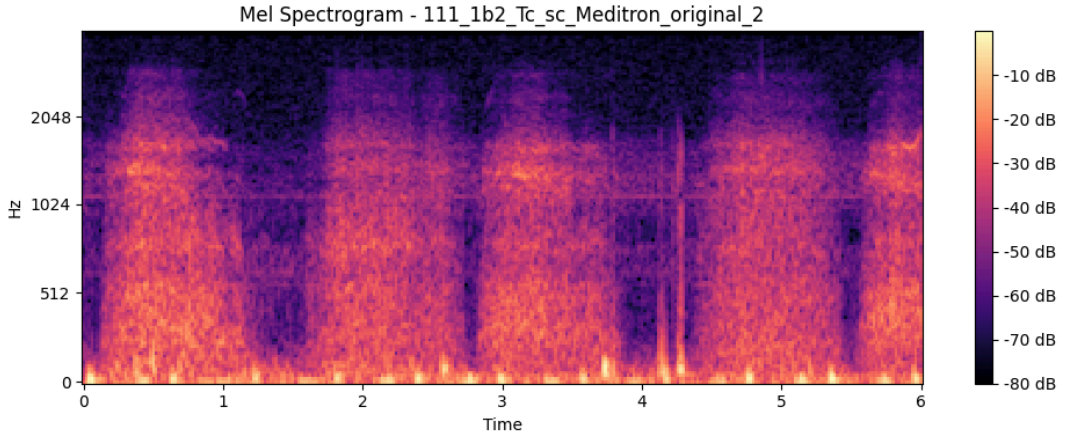


Figure 2: An example of what a spectrogram looks like.

Overcoming Limitations of Traditional Preprocessing Methods

This research seeks to address the limitations of MFCCs and Spectrograms by exploring and comparing them with an alternative preprocessing technique called Per-Channel Energy Normalization (PCEN). PCEN offers several advantages over traditional log-mel spectrograms and MFCCs, including suppression of ambient noise and enhanced sensitivity to transient sounds. Given that lung sounds are often faint and masked by background noise, these features make PCEN a good fit for improving the performance of machine-learning models in this domain.

This study addresses a pressing need in medical diagnostics by providing clinicians and researchers with insights into preprocessing methods that improve classification accuracy and robustness. The findings could contribute to the development of automated tools capable of reliably distinguishing between normal and abnormal lung sounds, supporting more timely and accurate diagnoses while reducing the burden on human expertise.

Machine Learning and CNN

Machine learning enables computers to learn from examples, recognize patterns, and make predictions without relying on explicitly programmed rules. A core type of machine learning model, the convolutional neural network (CNN), is particularly well-suited for analyzing structured data like images or spectrograms. CNNs excel at identifying local spatial or temporal relationships within data, making them a powerful tool for tasks that involve patterns, textures, or structured information. [5]

CNNs consist of multiple layers designed to extract and process increasingly abstract features. Convolutional layers apply filters that highlight specific patterns, such as edges or textures, while pooling layers reduce dimensionality, helping the model focus on the most essential information. [5] This architecture enables CNNs to generalize complex data and capture subtle input variations.

By leveraging CNNs, various types of raw input, including images, audio, and time-series data, can be effectively analyzed. For audio-based tasks, raw signals can be transformed into visual representations like spectrograms or mel-frequency cepstral coefficients (MFCCs), which CNNs can process as if they were images. This capability has proven instrumental in fields like speech recognition, music analysis, and medical diagnostics, where identifying frequency-based patterns is critical for accurate interpretation. [6]

Performance Metrics for Evaluation

This study considers overall accuracy, precision, recall, F1-score, and confusion matrices to evaluate model performance comprehensively. These metrics offer insight into how well each preprocessing method distinguishes between lung sound classes, guiding the selection of an optimal approach for real-world medical applications. Loss measures how well the model's predictions match the ground truth (lower is better). [7] Accuracy is the fraction of correctly classified samples. Precision (correct predictions among those labeled positive) and Recall (correct predictions among all actual positives) illustrate how effectively each model

identifies true positives. [7] The F1 score is a metric that evaluates the balance between precision and recall in a classification task. It is defined as the harmonic mean of precision and recall [7]. AUC_ROC (Area Under the Curve for the Receiving Operator Characteristic) reflects how well the model ranks positive predictions across different classes. [7]

Although accuracy is useful for measuring how often a model correctly predicts a label, it does not capture the severity of errors. In high-stakes domains like medical diagnostics, understanding *how often* the model is right and *how wrong* it can be is crucial. Loss addresses this concern by penalizing large misclassifications more severely. For example, a model might achieve high accuracy in detecting pneumonia by correctly identifying many mild cases yet failing to recognize severe cases, properly assigning them extremely low probabilities. These underdiagnosed severe cases can lead to delayed intervention in a clinical environment, significantly worsening patient outcomes. By placing more weight on such critical errors, loss provides a deeper evaluation of whether the model reliably identifies all risk levels or occasionally makes dangerous mistakes that an accuracy metric would overlook. Thus, focusing on loss offers a clearer picture of a model's reliability under real-world medical constraints.

Confusion matrices enable detailed analysis of classification performance through their diagonal and off-diagonal elements. Diagonal elements (running top-left to bottom-right) represent correct classifications, while off-diagonal elements indicate misclassifications, where the model predicted a different class than the true label. [7] In medical applications, the consequences of these errors are asymmetric. Misclassifying a healthy patient as diseased (false positive) typically leads to additional testing, while misclassifying a diseased patient as healthy (false negative) could result in delayed treatment and worse health outcomes. For example, misclassifying COPD as healthy (off-diagonal) is more concerning than misclassifying healthy as COPD (also off-diagonal), but on the healthy horizontal, since missed diagnoses pose greater risks than false alarms.

Methodology

Datasets and Data Sources

This study utilized two primary datasets to support its analysis. The first, the ICBHI 2017 Challenge Dataset, comprises 920 annotated audio recordings from 126 individuals, encompassing a total of 5.5 hours and 6,898 respiratory cycles. Among these cycles, 1,864 feature crackles, 886 include wheezes, and 506 exhibits both, with categories such as bronchiectasis, bronchiolitis, COPD, healthy, URTI, and pneumonia represented. [8] However, the diagnostic label LRTI was excluded due to insufficient data. The second dataset, sourced from Kaggle, includes 336 recordings from 112 subjects, consisting of 35 healthy and 77 unhealthy cases. [9] This dataset features categories such as heart failure, lung fibrosis, pleural effusion, health, COPD, and asthma, although bronchitis was excluded due to limited data. The recordings in this dataset were processed using Diaphragm, Bell, or Extended filters.

Derived Datasets

The datasets were combined to create four distinct groupings tailored to various research objectives. The Disease-Diagnosis Dataset includes lung sounds across categories like Bronchiectasis, COPD, and Healthy. The Crackles–Wheezes–Normal Dataset focuses on three classes—crackles, wheezes, and normal—excluding files with both crackles and wheezes to avoid ambiguity. [10] The Normal vs. Abnormal Dataset consolidates all abnormal conditions into one class, with Healthy as the only normal class. Lastly, the Unaugmented Dataset comprises raw WAV files without any data augmentation.

How Per-Channel Energy Normalization (PCEN) Works

Lung sounds often have low amplitude and are susceptible to ambient noise, making careful signal interpretation crucial. Per-Channel Energy Normalization (PCEN) addresses this challenge by starting with a mel spectrogram—a frequency-time representation mapped to the mel scale, reflecting human auditory perception. In this two-dimensional array (rows as frequency bins, columns as time steps), each element represents energy (the square of the amplitude). [11] To distinguish slowly changing background noise from rapidly varying respiratory sounds, PCEN first applies a smoothing function via convolution of the spectrogram (E) with a smoothing kernel (ϕ). [12] The smoothing coefficient, T, dictates how quickly the smoothing envelope responds: lower values of T capture fast signal changes, while higher values emphasize gradual variations. [11] (Figure 3, below)

$$\mathbf{PCEN}(t,f) = \left(\frac{\mathbf{E}(t,f)}{(\varepsilon + (\mathbf{E} \overset{t}{*} \phi_T)(t,f))^{\alpha}} + \delta \right)^{r} - \delta^{r}$$

Figure 3: Equation for PCEN. [9]

Once the spectrogram is smoothed, a gain control exponent, α, selectively compresses energy, reinforcing distinctions between background noise and meaningful sound events. δ provides a bias to ensure quieter sounds do not vanish altogether, while ε (epsilon) is simply a small constant safeguarding against division by zero (though, in practice, the averaging in the smoothing kernel makes this unlikely). Finally, dynamic range compression is accomplished through r, which tunes how forcefully PCEN highlights or suppresses signal amplitudes. When r is closer to zero, PCEN behaves more like a log-mel spectrogram, sharply compressing amplitude ranges to illuminate faint signals. Pushing r toward one reduces compression, mimicking a standard mel spectrogram without scaling. In practice, T and r often have the most significant impact on performance, followed by δ and α. Small parameter adjustments can dramatically alter downstream accuracy, making PCEN a parametric generalization of existing front ends rather than a simple on/off choice. Figure 4 provides an illustration of PCEN, while Figure 5 offers a side-by-side comparison of a mel-log-spectrogram (a) and PCEN spectrogram (b). The color intensity indicates amplitude, highlighting how PCEN effectively suppresses background noise and enhances transient noise for improved clarity

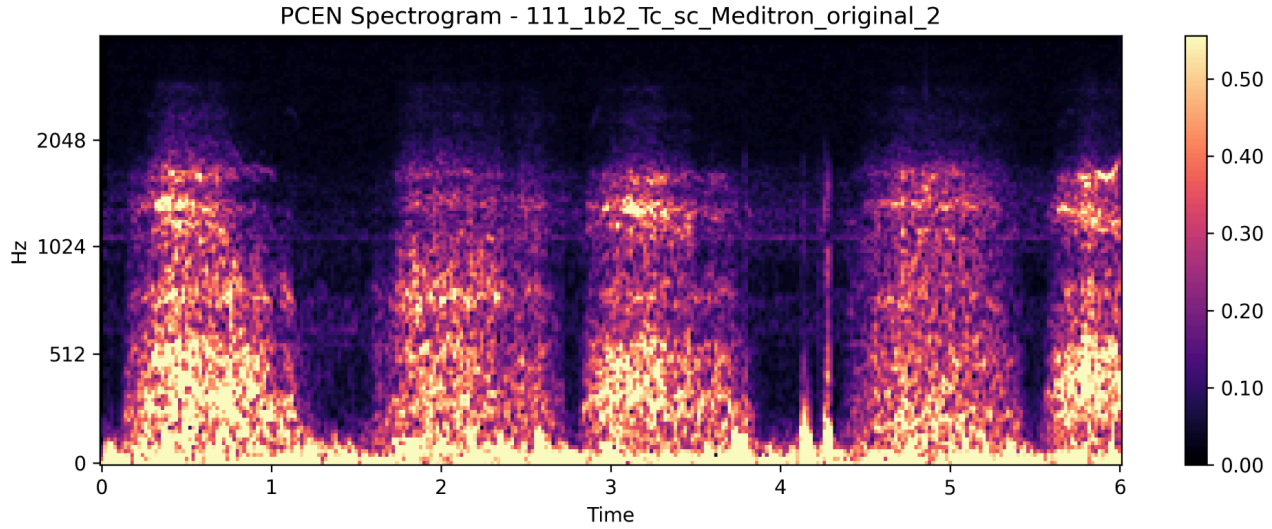
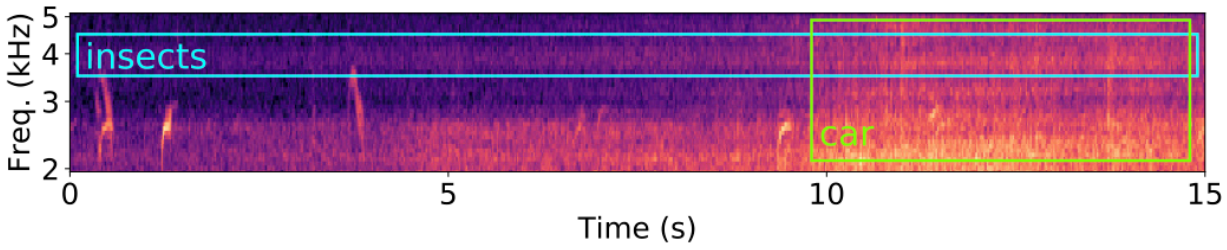
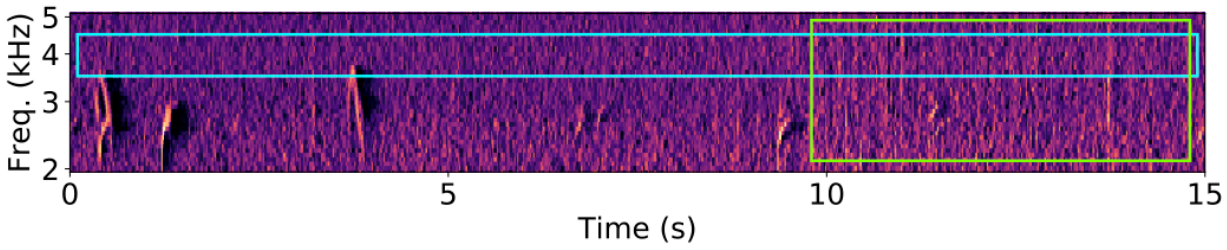


Figure 4: a visual representation of PCEN.



(a) Logarithmic transformation.



(b) Per-channel energy normalization (PCEN).

Figure 5: Side-by-side comparison of mel-log-spectrogram (a) and PCEN spectrogram (b). [11]

Choosing Audio Format

Sampling Rate. The sampling rate of an audio signal, measured in Hertz (Hz) or kilohertz (kHz), determines how often the sound is sampled per second. Higher sampling rates capture greater detail in the audio but require more storage and computational resources. [5] Most lung sounds, such as wheezes and crackles, occupy a frequency range below 8 kHz, making a

sampling rate of 8 kHz sufficient for capturing all diagnostically relevant information. For this study, recordings initially sampled at 44.1 kHz were downsampled to 8 kHz, reducing data size and computational overhead while maintaining essential features.

Lung sounds primarily occupy low- to mid-frequency ranges, with wheezes typically spanning 100 Hz to 500 Hz and crackles between 50 Hz and 2000 Hz, as shown in Figure 6 below. To ensure no important crackles are left out, we assume a range of 0-4000 Hz of relevant sounds for our audio. Given the Nyquist theorem, an 8 kHz sampling rate is sufficient to capture these frequencies, ensuring no diagnostically relevant information is lost during downsampling.

Characteristics of respiratory sounds: Normal, Wheeze, and Crackle [8], [9].

Type	Continuous	Frequency	Pitch	Cause	Disease
Normal	–	100–200 Hz	High (>800 Hz)	–	Asthma, COPD
Wheeze	O	400 Hz	High (>400 Hz)	Airway narrowing, airflow limitation	Asthma, COPD
Crackle	X	60–2000 Hz	Low (<350 Hz)	Explosive opening of small airways (fine crackle) and air bubbles in large bronchi (coarse crackle)	ILD, Bronchiectasis, Pneumonia

Figure 6: Chart displaying typical frequency ranges of wheezes, crackles, and normal. [10]

Channel Format. Channel format specifies whether audio is recorded as mono (single channel) or stereo (two channels). A mono format is ideal for lung sound analysis, as it outputs the same signal to both left and right speakers, avoiding unnecessary spatial information. In this study, all recordings were converted to mono to simplify processing and ensure consistency across the dataset.

Dynamic Range and Bit Depth. Dynamic range represents the amplitude difference between the quietest and loudest sounds in a recording. Lung sounds often have subtle variations in amplitude that are critical for accurate diagnosis. Bit depth, which determines the precision of amplitude measurements, was maintained at 16-bit or 32-bit for this study to ensure these variations were captured. To further emphasize diagnostically relevant features, preprocessing steps such as gain and normalization were applied to manage dynamic range.

Relevance to Lung Sound Analysis. Balancing data efficiency with diagnostic accuracy is paramount for medical applications. Combining an 8 kHz sampling rate, mono format, and appropriate bit depth ensured that lung sound recordings retained all essential information while

optimizing storage and processing efficiency. These preprocessing choices facilitated robust audio feature extraction for subsequent machine learning analysis, enhancing the model's ability to classify respiratory conditions accurately.

Choice in CNN architecture and Overfitting

Machine learning allows models to learn patterns directly from data, minimizing errors between predicted labels (e.g., normal vs. abnormal) and the ground truth. Ideally, models identify general patterns that apply to unseen data, but an overly complex model can overfit by memorizing specific details and noise in the training data. This results in high training accuracy but poor performance on new data, a critical issue in medical applications where generalization across diverse patient populations and conditions is essential.

Overfitting typically occurs when there is a considerable discrepancy between training and validation accuracy or when predictions on unseen data are inconsistent. [6] For instance, in lung sound classification, an overfitted model may pay too much attention to patient-specific features or background noise rather than focusing on the general characteristics of respiratory conditions.

The risk of overfitting is significantly heightened when the dataset is insufficiently large or diverse, as the model lacks exposure to the full range of variability present in real-world data. In the context of lung sound classification, for example, a limited dataset may disproportionately represent recordings from a narrow range of patients, devices, or environments. Consequently, the model becomes biased toward these specific features, failing to generalize effectively to new patients or recording conditions.

A small dataset also increases the likelihood that the model will treat irrelevant noise or artifacts as meaningful features. [6] For instance, if consistent background noise or microphone artifacts are present in the dataset, the model might inadvertently associate these artifacts with specific labels. This results in high training accuracy but poor performance on unseen data, where these noise artifacts differ. [6] Moreover, many parameters in modern machine-learning models exacerbate this issue. When the number of parameters exceeds the available training examples, the model is prone to overfitting, as it has an excessive capacity to "memorize" the training data rather than learning general patterns.

In summary, insufficient data restricts the model’s ability to learn generalizable patterns, increasing the risk of overfitting.

Segmentation and Augmentation

The initial phase of this project involved comprehensive data preprocessing. Although we started with 920 WAV files, this amount was insufficient to achieve reliable accuracy, especially since all audio clips varied in length (from as short as 15 seconds to over 90 seconds), sampling rate, and channel format (mono vs. stereo). In order to train a machine learning model effectively, all inputs must be uniform in length, sampling rate, and channel format. Because each audio file contained multiple respiratory cycles, we examined each file's

start and end times in a corresponding data frame to determine optimal segmentation. [5]

Since each audio file is essentially an array sampled at a specific rate, multiplying the start or end time by the sampling rate yields the corresponding index in the array. Consistency in segment length is essential; scatter and box plots suggested that 6-second clips were optimal. To handle any file shorter than 6 seconds, we applied zero padding, which adds silent samples (zeros) to reach the desired length without altering the original content. Consequently, we split the larger audio files into overlapping 6-second segments, converted stereo tracks to mono when needed, and normalized amplitude to avoid distortion. However, the dataset remained too small after these steps, causing the model to overfit with poor validation results. This prompted extensive data augmentation.

The 920 original WAV files were augmented using six transformations: pitch shift up, pitch shift down, time stretch (fast), time shrink (slow), and gain adjustment. Data augmentation creates altered versions of existing data, enabling the model to better generalize to new conditions—for instance, simulating loud background noise or subtle pitch variations. Because the transformations require 32-bit NumPy arrays, all audio files were first converted into that format. After generating the six augmented sets plus the original set, we again split them into 6-second segments, resulting in 42,628 audio clips—enough to reduce overfitting significantly.

Despite expanding the dataset size, our initial diagnostic categories were limited to Bronchiectasis, Bronchiolitis, COPD (Chronic Obstructive Pulmonary Disease) , Healthy, URTI (Upper Respiratory Tract Infection), and Pneumonia. While the ICBHI 2017 dataset originally included Asthma and LRTI (Lower Respiratory Tract Infection), these categories were excluded due to insufficient data. To create a dataset more representative of real-world clinical scenarios, we incorporated an additional Kaggle dataset containing 336 sound files labeled with Bronchitis, Heart Failure, Lung Fibrosis, Pleural Effusion, Healthy, COPD, and Asthma. However, we removed Bronchitis from consideration as it had only three WAV files, making it unsuitable for reliable classification. The Kaggle dataset included recordings in three filter modes—Bell, Diaphragm, and Extended—covering frequency ranges of [20–200 Hz], [100–500 Hz], and [50–500 Hz], respectively. To maintain consistency with the ICBHI dataset, we chose the Diaphragm filter (100–500 Hz), as it minimally altered the original spectrum. We then removed the filter by undoing it across all files. Following the removal of filter effects, we subjected the files to the same data augmentation and 6-second segmentation process, ultimately generating 46,576 clips spanning 10 diagnostic categories.

We then developed a specialized dataset focused on wheezes, crackles, and normal lung sounds. Both the Kaggle and ICBHI datasets included files labeled with wheezes and crackles. After an extensive effort to isolate these labels, we augmented and segmented each file into 6-second clips before merging them into a cohesive dataset. Any file containing wheezes and crackles was excluded entirely to avoid ambiguity for the model. This meticulous process yielded 37,248 clips labeled wheezing, crackling, or normal. Subsequently, we created a third dataset distinguishing between abnormal and normal lung sounds. This was achieved by consolidating all abnormal categories into a single folder while retaining Healthy as the “normal”

class. Lastly, we generated an unaugmented dataset by applying the clean split method to the datasets and merging them, resulting in a comprehensive unaugmented dataset for further analysis.

Problems Encountered with Dataset

During testing, we identified a significant issue with our dataset. This realization emerged while analyzing our fourth unaugmented dataset, which unexpectedly outperformed our first augmented dataset for PCEN-based processing despite using identical PCEN parameters. The key distinction between these datasets was the presence of six different augmentations in the first dataset. We systematically removed augmentations to pinpoint which augmentation or augmentations negatively impacted PCEN. We retrained the model, eventually discovering that noise augmentation significantly degraded PCEN performance—a counterintuitive finding given noise augmentation's typical role in improving robustness. Removing noise augmentation improved PCEN results but left PCEN trailing behind spectrogram performance.

We hypothesized that another issue lay in the class imbalance, with COPD accounting for nearly 50% of the dataset, potentially causing the model to overfit this category. To address this, we developed a method to reduce the number of COPD files while maintaining dataset integrity. Our code grouped files by shared identifiers in their filenames, ensuring that original recordings and their augmentations (e.g., time stretch, pitch shift, noise addition) were treated as a single unit. By randomly selecting and retaining 50% of these groups, we reduced the dataset to approximately 3,000 COPD files while preserving diversity and balance among augmentations.

This group-level reduction maintained data consistency and eliminated bias from random individual file sampling. The result was a more balanced dataset suitable for machine learning tasks. We repeated this process multiple times, iteratively refining the COPD representation while addressing the performance limitations caused by noise augmentation and class imbalance.

Next, we applied the same COPD reduction and noise exclusion process to the remaining datasets. For the abnormal vs. normal dataset, this was straightforward: we grouped all abnormal files from the first dataset into a single folder and placed all healthy files into a separate folder. The process for the wheeze and crackle dataset was more complex. We began by running a script to remove any instances of noise augmentation from the dataset. Since COPD files were distributed across three subfolders (wheeze, crackle, and normal), we developed a script to traverse all subfolders and identify files present in the original 30,000-file COPD dataset but not in the newly filtered 3,000-file COPD dataset. These extraneous files were then deleted, ensuring that only the cleaned COPD files were consistently used across all datasets. Creating the fourth unaugmented dataset was simpler by comparison. We identified all files in the first dataset that were not augmented and copied them into a separate folder. This systematic approach ensured that all datasets maintained consistency and integrity, aligning with the updated COPD representation and noise-free requirements.

Finalization of Dataset

Once all datasets were finalized, we created dedicated datasets for evaluation metrics to ensure reliable and unbiased assessments of model performance. Evaluating a model on the same data used for training is generally not recommended because this often leads to overestimating performance. By testing on separate data, we avoid misleading results and gain a realistic understanding of how well the model generalizes to unseen scenarios. The primary reasons for this approach are rooted in key machine-learning principles as follows.

Overfitting Prevention*. During training, a model learns patterns specific to the training data, including potential noise. Evaluating the same data can yield inflated performance metrics, as the model effectively "remembers" the training examples instead of learning generalizable patterns.*

Generalization Capability*. The actual test of a model's quality lies in its ability to perform well on new, unseen data. Setting aside a validation or test set that the model has not encountered ensures we accurately gauge its ability to generalize beyond the training data.*

Reliable Comparison*. A consistent and unbiased benchmark is critical when comparing multiple models or tuning strategies. Using separate test sets for evaluation ensures that comparisons reflect how well the models would handle new data, providing fair and trustworthy results.*

To implement this, we systematically separated approximately 15% of files from each dataset into evaluation sets. The process involved grouping files by patient name to ensure all related data remained intact, shuffling these groups, and selecting 15% to form a dedicated test set. Files from the chosen groups were moved to a parallel directory structure within a new "test" folder, preserving the original class organization. This prevented partial splits of related files and ensured the test sets were representative of the dataset as a whole.

This methodology was applied consistently across all datasets, resulting in eight total datasets: four for training and four for evaluation. By separating training and evaluation data in this manner, we achieved a realistic measure of model performance, reduced overfitting risks, and established a robust framework for comparing models and tuning strategies.

Data Generator

To efficiently manage audio data loading and preprocessing, we implemented a custom data generator based on a class commonly used in machine learning frameworks for handling large datasets (Tensorflow). This method is particularly crucial for working with extensive medical audio datasets, where loading all recordings into memory is impractical due to memory constraints. Instead of storing the entire dataset in RAM, the generator dynamically loads only the audio files required for each training batch. [5] Once a batch is processed, it is released from memory, enabling the model to work seamlessly with large datasets while avoiding excessive memory usage. This approach ensures efficient resource management and scalability for training on high-dimensional audio data. [5]

In practical terms, the generator maintains an array of indices corresponding to file paths and shuffles these indices at the end of each training epoch. The generator loads the corresponding

.wav files for each batch, reads their waveforms, and reshapes them into the desired input structure (e.g., (samples [sample rate * time], 1) for a single audio channel). [5] It converts integer labels to one-hot vectors before returning the (X, Y) pair to the training loop. This mechanism allows the integration of on-the-fly preprocessing steps—such as Per-Channel Energy Normalization (PCEN), windowing, and other transforms—without requiring the entire dataset to be preprocessed in advance. [5]

Beyond its memory advantages, this data generator setup also promotes better model generalization. Because the data is shuffled between epochs, the model does not inadvertently adapt to the order in which sounds are presented. Furthermore, if a class imbalance in a particular dataset is an issue, the generator can easily be adapted to include class-weighting or other sampling strategies that ensure underrepresented classes are seen sufficiently often. Overall, this batch-based, lazy-loading strategy is critical for large-scale lung sound analysis, enabling efficient, flexible, and robust training pipelines.

Algorithm

We implemented a convolutional neural network (CNN) with five 2D convolutional layers to classify our datasets. As feature extractors, CNNs progressively learn increasingly complex patterns from lung sounds by reducing output dimensions at each layer. Pooling layers are included to downsample data, focusing on key features while enhancing computational efficiency. The network begins with an input layer followed by data normalization for stability. A flattening layer then converts the extracted features into a format suitable for a dense layer, which refines the output. The final softmax layer assigns class probabilities. The model was trained using the Adam optimizer, with categorical cross-entropy as the loss function and accuracy as the primary evaluation metric. As illustrated in Figure 7, the model consists of 13 layers applied after the mel-log spectrogram, MFCC, or PCEN input.

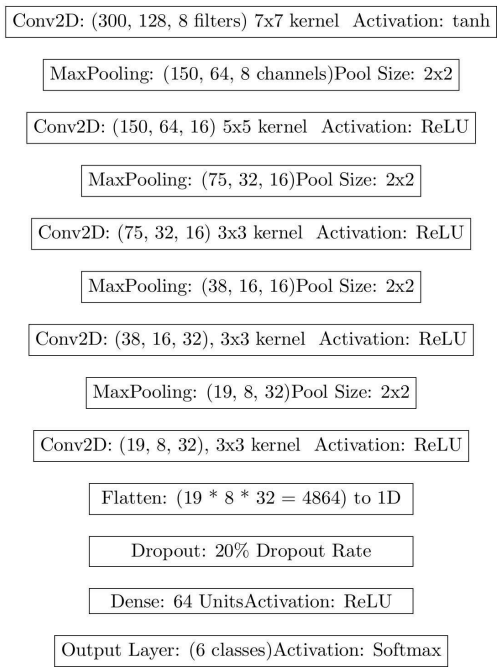


Figure 7: Model layers for all models (varying inputs)

For models involving spectrograms (namely the mel-log spectrogram and PCEN models), we integrated a preprocessing layer powered by the Kapre library. This real-time processing layer generates time-frequency representations directly within the network, eliminating the need to store thousands of spectrogram images and significantly accelerating the training pipeline. [13] We implemented an automatic checkpoint system to mitigate overfitting and save the best model at each epoch based on validation loss. Overfitting typically remained minimal until after around 30 epochs due to the large dataset size. Models were trained for 50 epochs across all three datasets, ensuring robust learning and performance.

Analysis

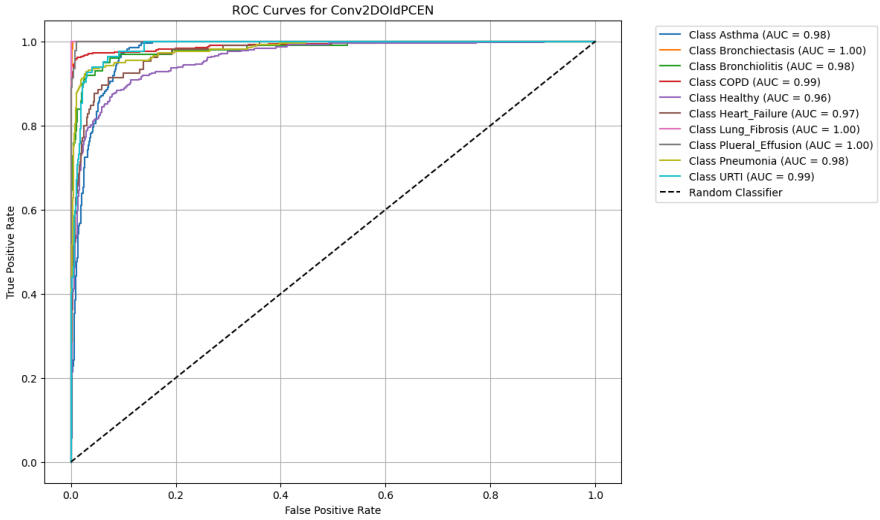
Our results generally proved our hypothesis, with PCEN outperforming other models across all testing datasets. (Table 1, Below)

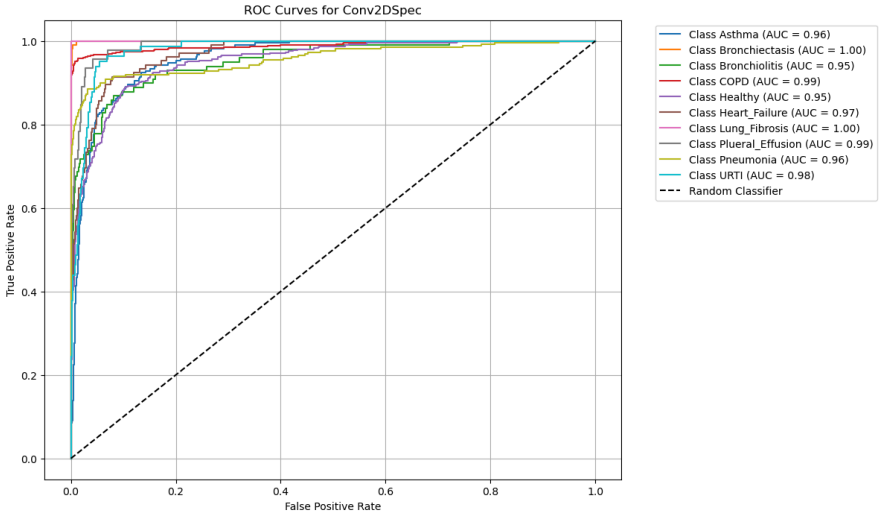
	PCEN PARAMETERS					EVALUATION					
MODEL	alpha	delta	root	smooth	epsilon	LOSS	ACC	PRECISION	RECALL	F1_SCORE	ROC
Disease_PCEN	0.7	5.5	0.85	0.007	1.00E-08	0.5109	0.8502	0.8633	0.8502	0.8519	0.9845
Disease_SPEC	n/a	n/a	n/a	n/a	n/a	0.6718	0.8318	0.8462	0.8318	0.8298	0.9751
Disease_MFCC	n/a	n/a	n/a	n/a	n/a	1.2658	0.7290	0.7333	0.7290	0.7247	0.9421
WheezeCrackleNormal_PCEN	0.75	5.5	0.9	0.007	1.00E-08	0.2019	0.9346	0.9336	0.9346	0.9339	0.9746
WheezeCrackleNormal_SPEC	n/a	n/a	n/a	n/a	n/a	0.2374	0.9288	0.9273	0.9288	0.9273	0.9656
WheezeCrackleNormal_MFCC	n/a	n/a	n/a	n/a	n/a	0.4589	0.8820	0.8782	0.8820	0.8793	0.9022
HealthyUnhealthy_PCEN	0.7	5.5	0.85	0.0075	1.00E-08	0.1674	0.9450	0.9442	0.9450	0.9441	0.9833
HealthyUnhealthy_SPEC	n/a	n/a	n/a	n/a	n/a	0.2374	0.9288	0.9273	0.9288	0.9273	0.9656
HealthyUnhealthy_MFCC	n/a	n/a	n/a	n/a	n/a	0.4589	0.8820	0.8782	0.8820	0.8793	0.9022
Unaugmented_MFCC	0.8	5.5	0.9	0.007	1.00E-08	1.3904	0.6960	0.6764	0.6960	0.6773	0.8440
Unaugmented_SPEC	n/a	n/a	n/a	n/a	n/a	1.6914	0.6080	0.6115	0.6080	0.5942	0.8445
Unaugmented_MFCC	n/a	n/a	n/a	n/a	n/a	2.1132	0.5861	0.5504	0.5861	0.5516	0.7443

Table 1: All metrics and parameters for each Model

Augmented Dataset Disease Classification

For disease classification, PCEN (alpha=0.7, delta=5.5, root=0.85, smoothing=0.007, epsilon=1.00E-08) leads with 0.51 loss, followed by Spectrogram at 0.68 and MFCC at 1.27. This pattern holds across metrics: accuracy (PCEN: 0.85, Spec: 0.83, MFCC: 0.73), F1-score (0.85, 0.83, 0.72), and AUC_ovr scores (0.985, 0.975, 0.942), consistently showing PCEN's superior performance (Figure 4, below).





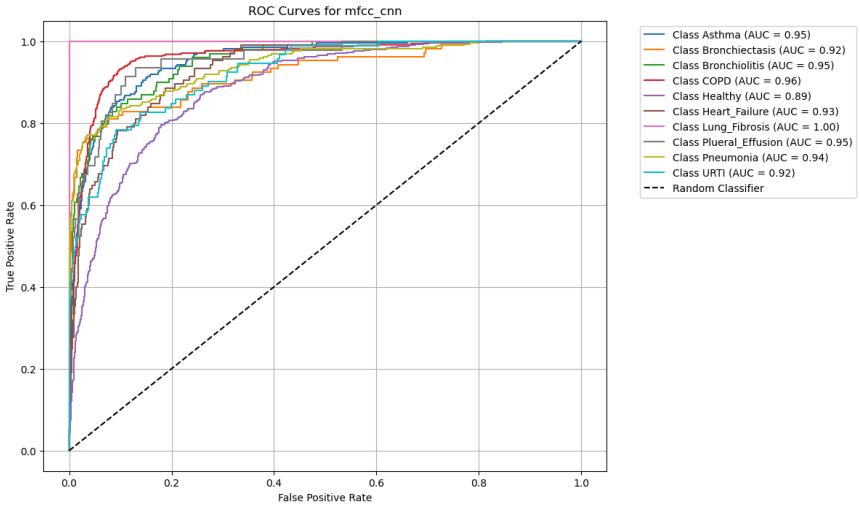
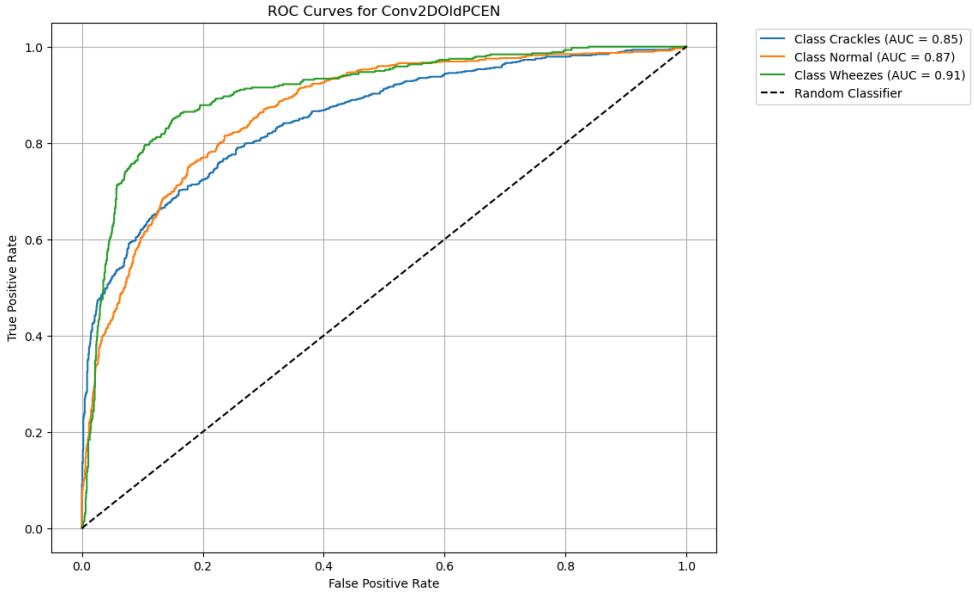


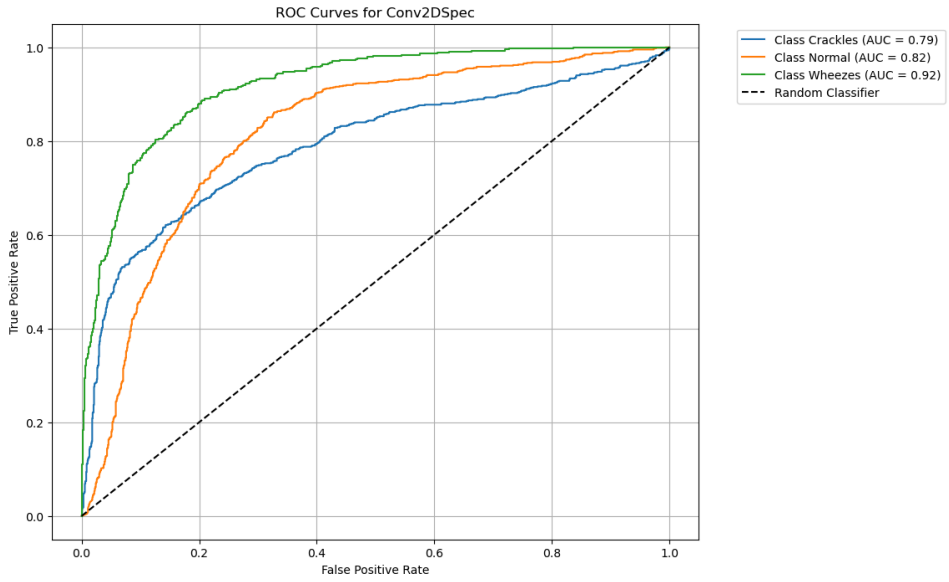
Figure 8: ROC Curves for Augmented Disease Classification

Wheezes, Crackles, Normal

For wheeze/crackle/normal classification, PCEN (alpha=0.75, delta=5.5, root=0.9, smoothing=0.007, epsilon=1.00E-08) maintains the lead with 0.20 loss, followed by Spectrogram at 0.24 and MFCC at 0.46. Performance metrics show similar trends: accuracy (PCEN: 0.934, Spec: 0.929, MFCC: 0.882), F1-score (0.934, 0.927, 0.879), and AUC_ovr (0.975, 0.966, 0.902) (Figure 5, below)



.



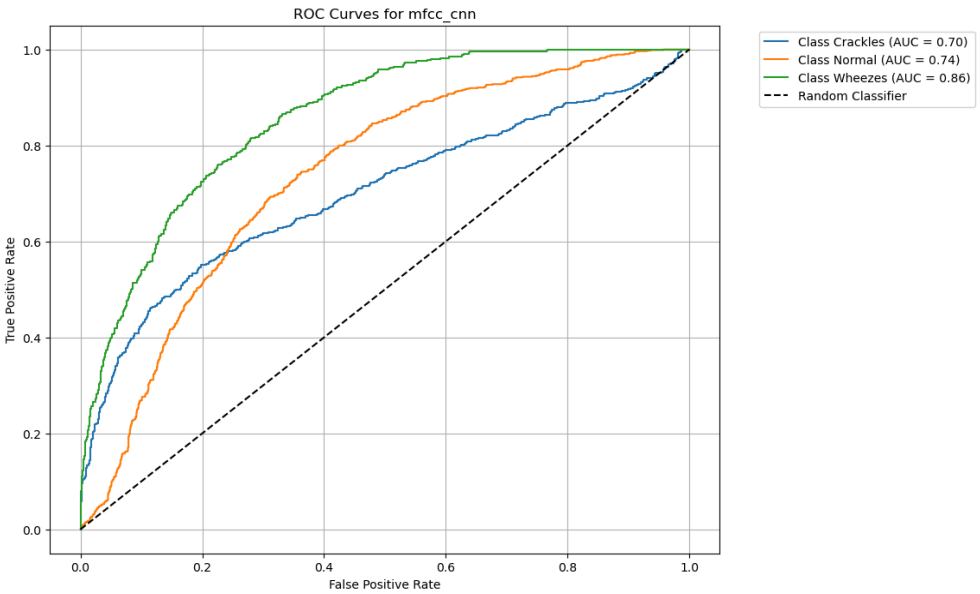
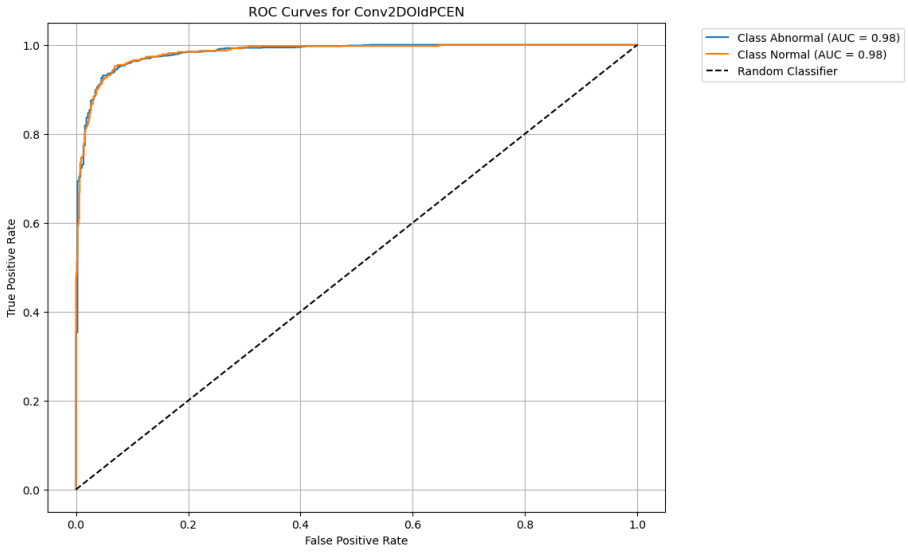


Figure 9: ROC Curves for Augmented Wheeze/Crackle/Normal Classification

Abnormal Normal Classification

For abnormal/normal classification, PCEN (alpha=0.8, delta=5.5, root=0.85, smoothing=0.0075, epsilon=1.00E-08) demonstrates superior performance with 0.167 loss, compared to Spectrogram's 0.237 and MFCC's 0.459. This pattern holds across all metrics: accuracy (PCEN: 0.945, Spec: 0.929, MFCC: 0.882), F1-score (0.944, 0.927, 0.879), and AUC_ROC scores (0.983, 0.966, 0.902). The consistent advantage across all metrics highlights PCEN's robust ability to distinguish normal from abnormal respiratory patterns. The dataset excels in this task due to the relative simplicity of abnormal vs. normal classification compared to other classification challenges. (Figure 6, below)



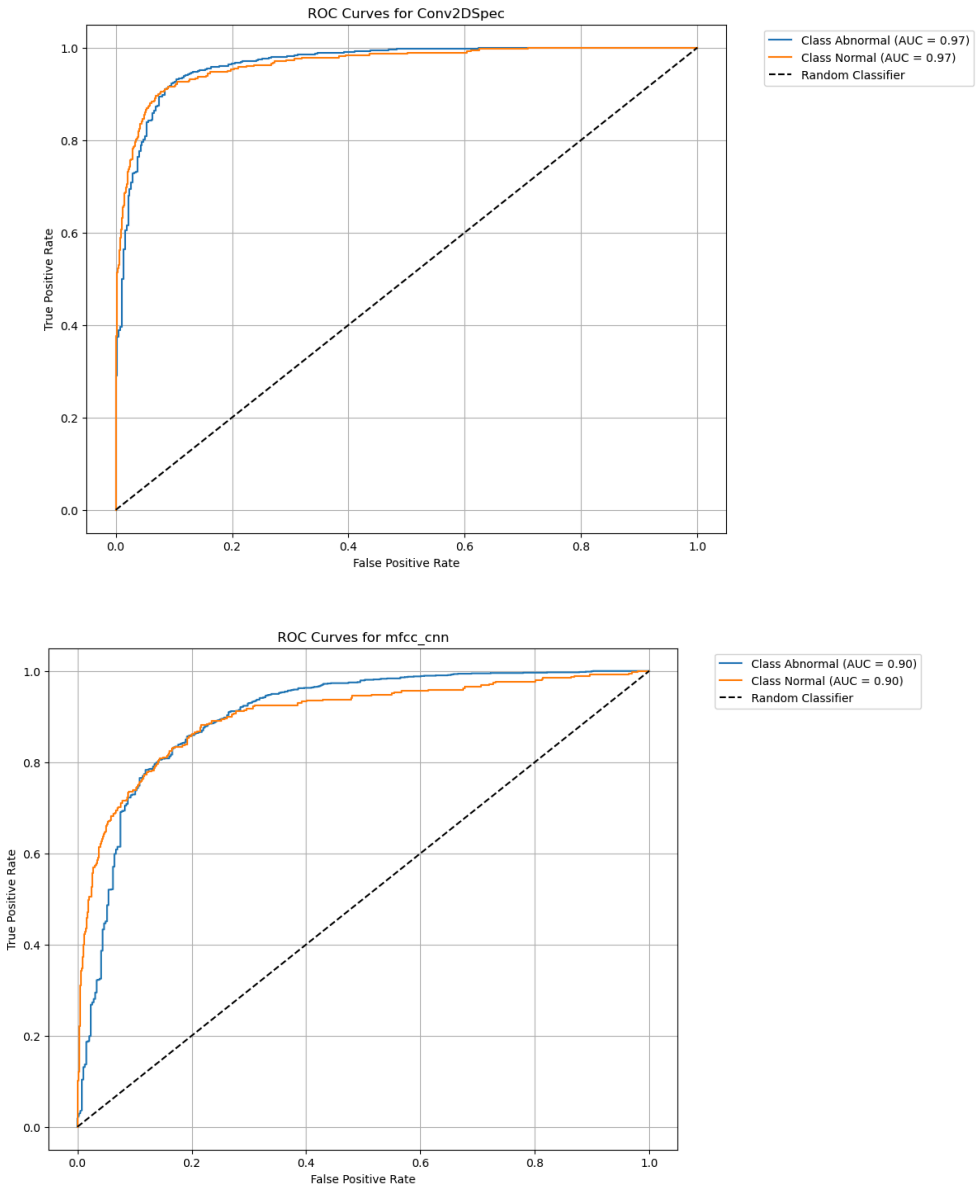
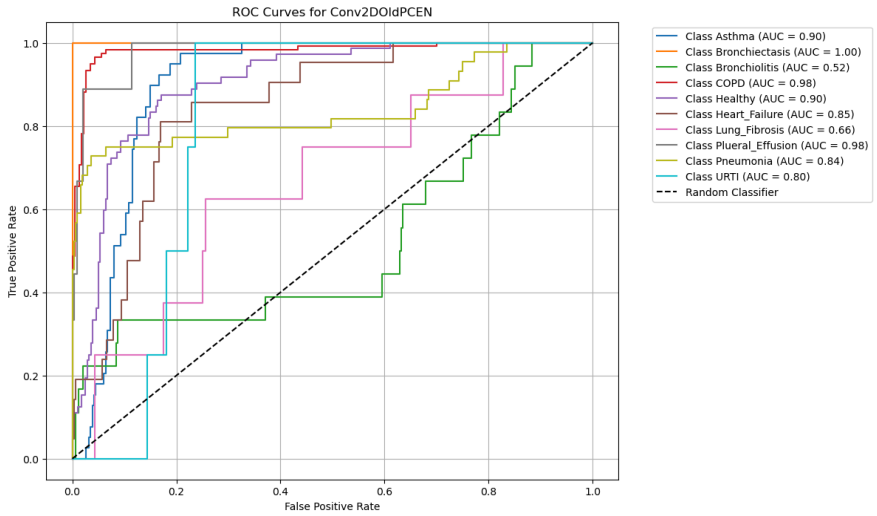
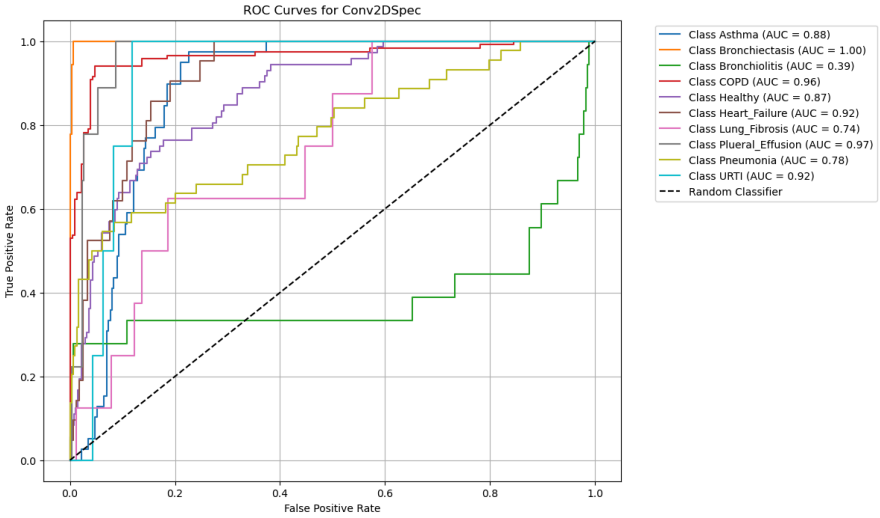


Figure 10: ROC Curves for Augmented Abnormal/Normal Classification

Unaugmented

For unaugmented data, PCEN (alpha=0.8, delta=5.5, root=0.9, smoothing=0.007, epsilon=1.00E-08) maintains advantage with 1.39 loss versus Spectrogram's 1.69 and MFCC's 2.11. Similar patterns in accuracy (0.70, 0.61, 0.59), F1-score (0.68, 0.59, 0.55), and AUC ROC (0.844, 0.844, 0.744), though mel-spectrogram matches PCEN's AUC ROC performance (Figure 7, below).





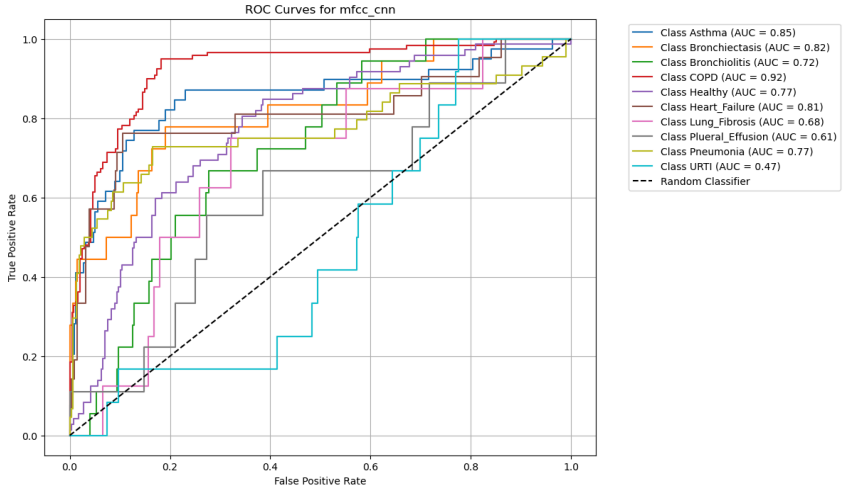


Figure 11: ROC Curves for Unaugmented Disease Classification

Errors Encountered

During our experiment, we encountered some errors when optimizing PCEN performance. While training the models, we observed that PCEN consistently produced lower validation loss (loss on data that the model has not seen) compared to the mel-spectrogram and MFCC approaches. However, when we evaluated the models on the entire dataset using our performance metrics, PCEN's performance fell short of expectations. We attributed this discrepancy to two main factors. First, we recognized that our mel-spectrogram and MFCC approaches were likely overfitting, characterized by a large gap between validation loss and loss when testing on previously seen data. Overfitting creates artificially high performance when a model is tested on the same data that it is trained on. Mel-spectrogram and MFCC models may have been learning to exploit raw amplitude patterns and background noise characteristics in addition to the lung sounds present in the training data, creating higher performance metrics when tested on the full dataset. This is made even more clear when considering that PCEN models are likely less susceptible to “memorizing” background artifacts because they isolate the important parts of a spectrogram. To mitigate this and achieve the most accurate evaluation metrics, we tested our models on the 15% evaluation dataset they had never seen before.

Advantages/Drawbacks of PCEN

One key advantage of PCEN lies in its customizability to different acoustic environments. During our trials, we altered the parameters T (temporal smoothing) and r (root compression) via a trial-and-error approach. We observed that small changes—on the order of 0.1—could shift the evaluation loss by up to 0.2. This significant variation underscores how sensitive PCEN can be to tuning, allowing researchers to dial in settings that best suit their data. Moreover, had we employed an automated method such as gradient descent to optimize these parameters, PCEN might have performed even better. We plan to integrate such automated parameter tuning in future experiments to maximize the benefits of PCEN’s adaptive noise suppression and dynamic range adjustment.

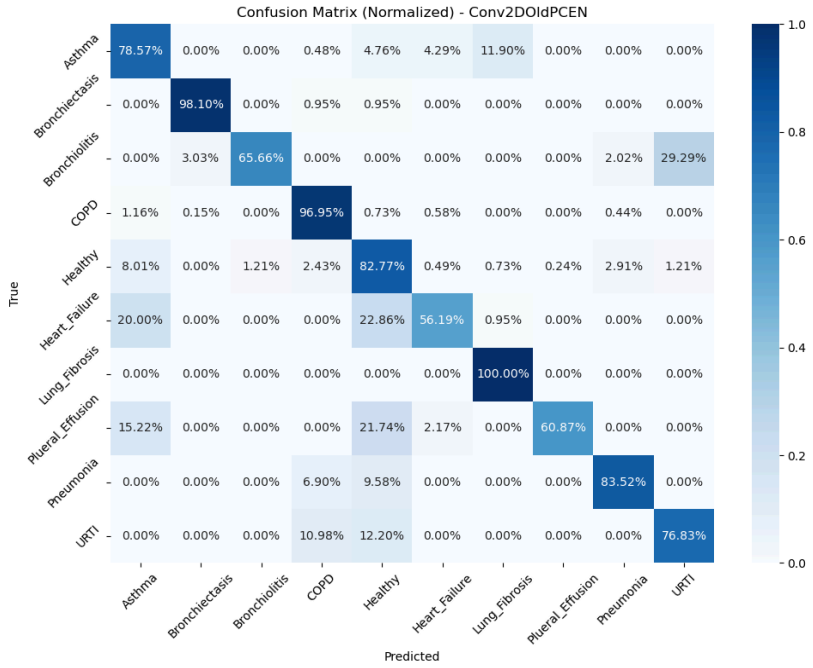
On the other hand, the drawback of PCEN is that it can take longer to train and test due to additional calculations needing to be performed everywhere on the spectrogram. During this project, PCEN could take up to 4 times longer to fully train 50 epochs. Training might have taken even longer if this were coupled with an additional trainable layer to optimize PCEN parameters.

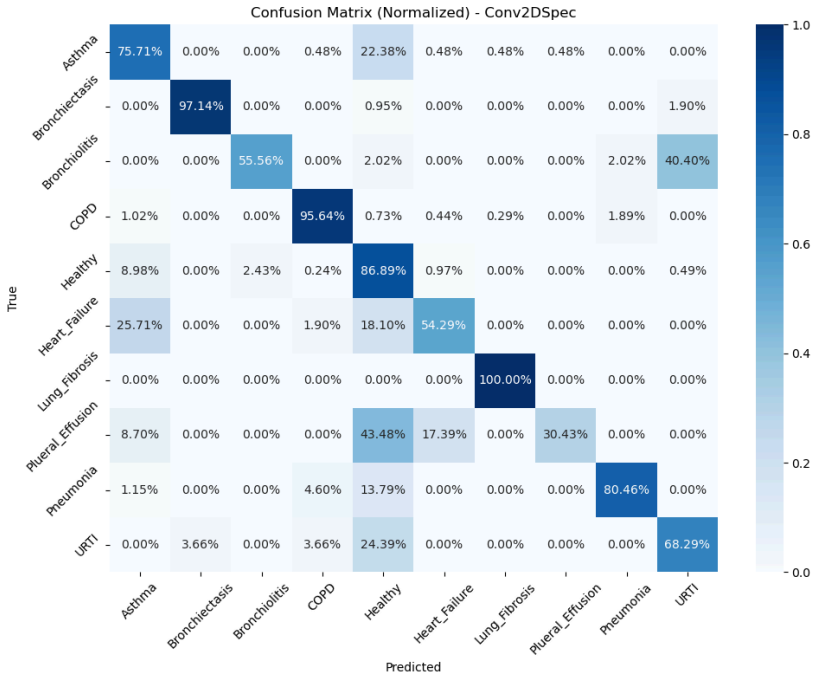
Confusion Matrices

Our confusion matrices for the evaluation set of our models also displayed PCEN’s superiority, especially in medical contexts. When diagnosing a patient, a false negative (classifying an unhealthy condition as healthy) is far worse than a false positive since with the former, a life-threatening condition could go unnoticed, while with the latter, more varied testing will be able to determine whether or not there is actually a condition. So, for our experiment, we want our model to minimize the classification of real diseases as healthy.

For the data-augmented disease classification models (10x10 matrices), PCEN demonstrates superior performance compared to spectrogram and MFCC approaches, with particularly strong results in correctly identifying critical conditions like COPD (96.95%) and Pneumonia (83.52%).

PCEN shows the lowest rate of misclassifying disease states as healthy at 8.01%, compared to Spectrogram's 8.98% and MFCC's substantial 11.11%, which is crucial for minimizing false negatives in medical applications. The model particularly excels at distinguishing between similar respiratory conditions, with asthma-to-healthy misclassification at just 22.38%, significantly better than MFCC's 38.46%. (Figure 12, below)





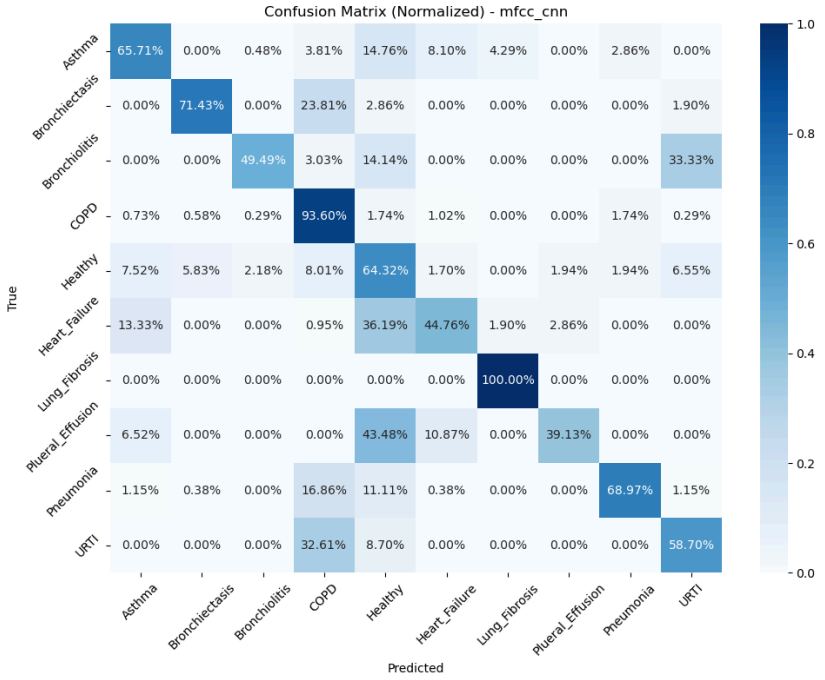
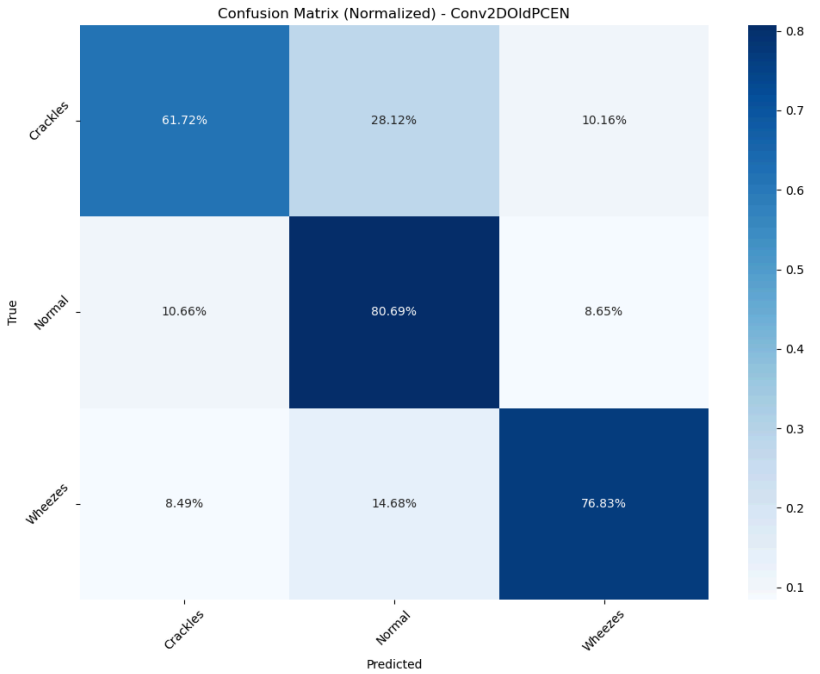


Figure 12: Confusion Matrices for Augmented Data Disease Classification

For the wheeze/crackle/normal classification models (3x3 matrices), PCEN achieves the highest accuracy with 76.83% correct wheeze classification and 80.69% normal classification. Critically, PCEN shows the lowest rate of misclassifying wheezes or crackles as normal breathing at 14.68%, compared to Spectrogram's 17.43% and MFCC's 31.04%. This indicates PCEN's superior ability to detect abnormal breathing patterns that require medical attention. (Figure 13, below)



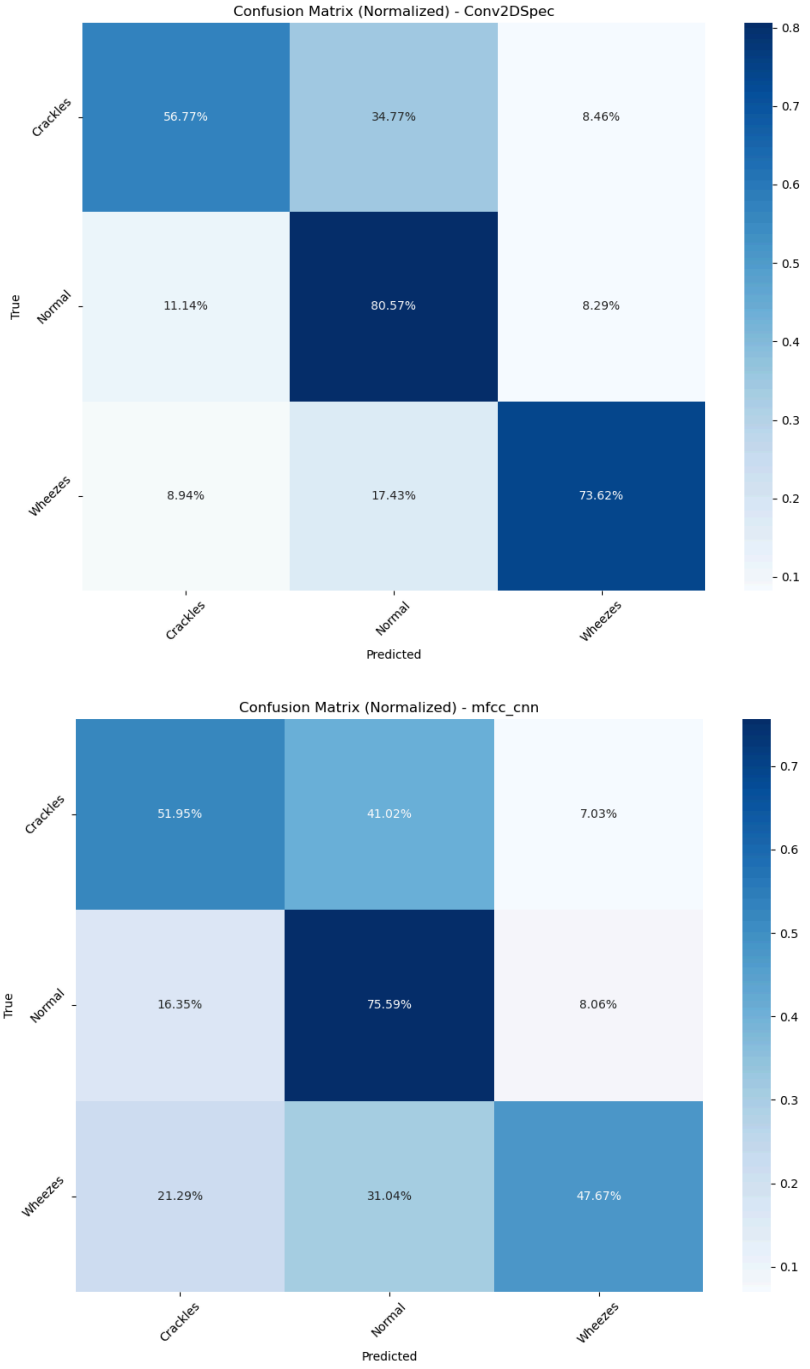
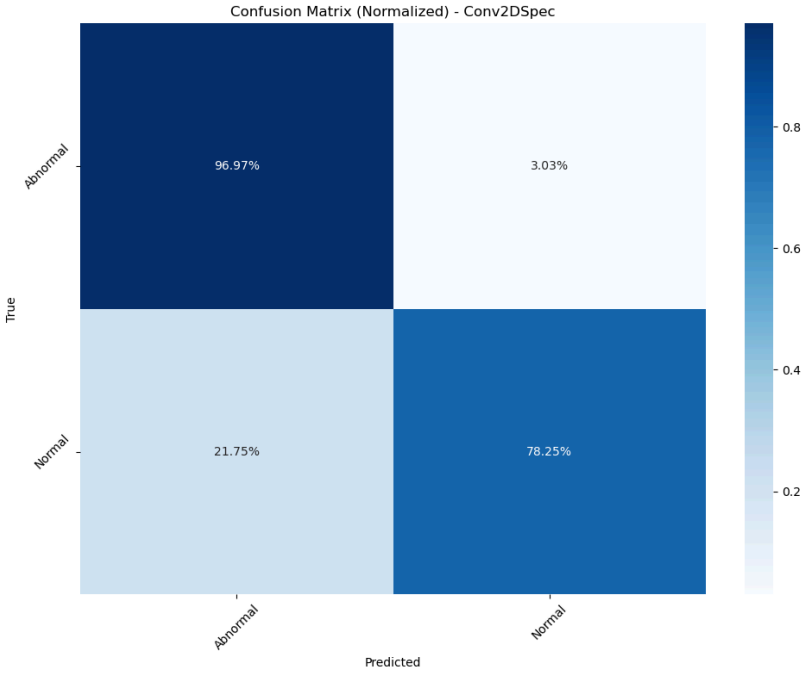
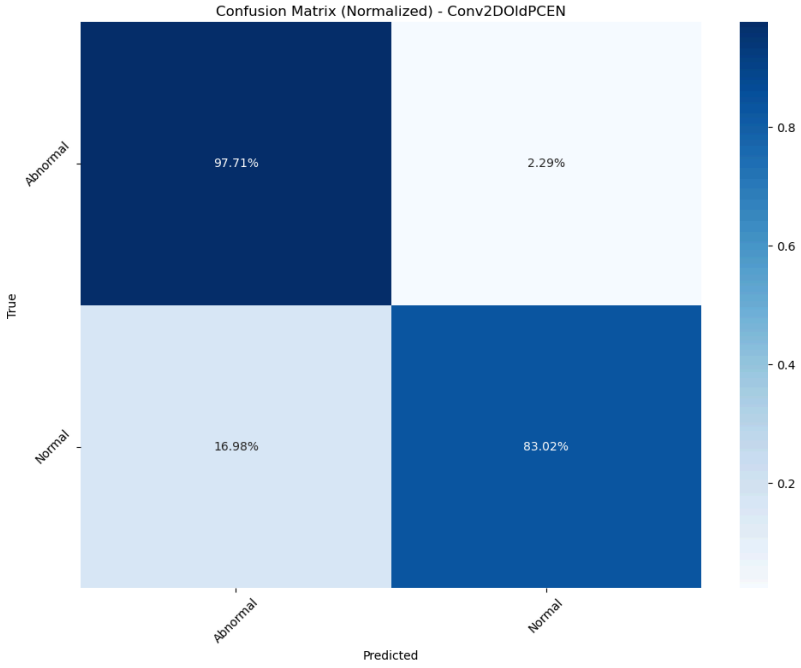


Figure 13: Confusion Matrices for Augmented Data Wheeze/Crackle/Normal Classification

In the binary abnormal/normal classification models (2x2 matrices), PCEN slightly outperforms other approaches with 96.60% abnormal detection accuracy and crucially maintains the lowest false normal rate at 17.77% compared to Spectrogram's 21.75% and MFCC's 32.38%. This superior performance in avoiding false negatives is particularly important in a medical context where missing an abnormal condition could have serious consequences. (Figure 14, below)


Confusion Matrix (Normalized) - Conv2DSpec
96.97%
3.03%
21.75%
78.25%
Abnormal
Normal
True
Abnormal
Normal
Predicted
0.8
0.6
0.4
0.2


Confusion Matrix (Normalized) - Conv2DOldPCEN
97.71%
2.29%
16.98%
83.02%
Abnormal
Normal
True
Abnormal
Normal
Predicted
0.8
0.6
0.4
0.2

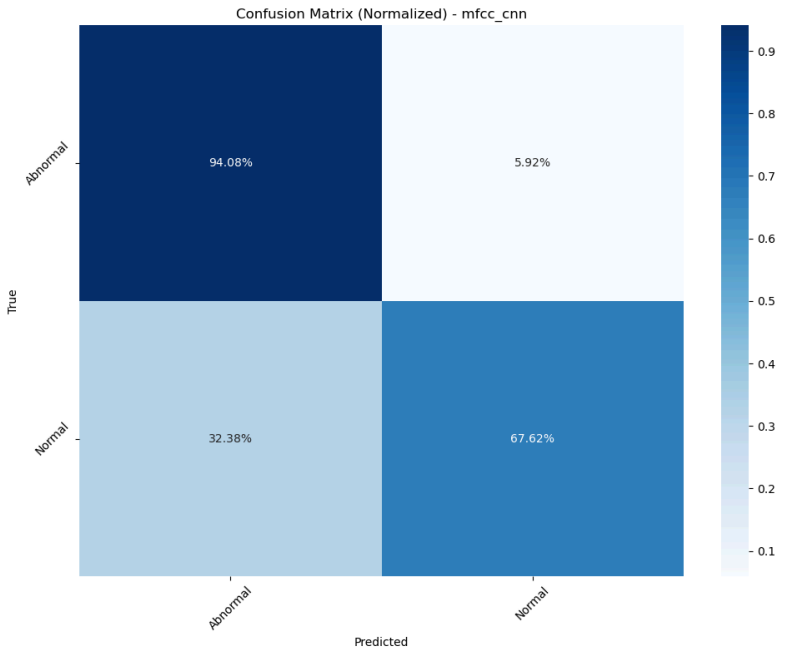
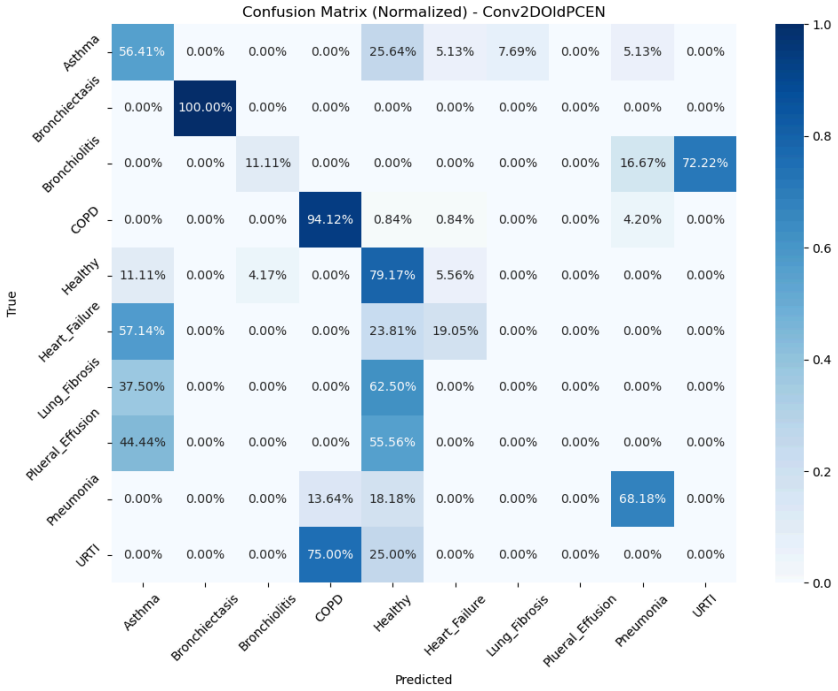


Figure 14: Confusion Matrices for Augmented Data Abnormal/Normal Classification

In the unaugmented disease classification models, all models show degraded performance, but PCEN maintains its relative advantage. However, the false healthy rates increase significantly across all models. PCEN misclassifies 25.64% of disease cases as healthy, while Spectrogram reaches 48.72%, and MFCC shows a concerning 63.89% false healthy rate. This substantial difference highlights the critical importance of data augmentation for reliable medical diagnostics. (Figure 15, below)



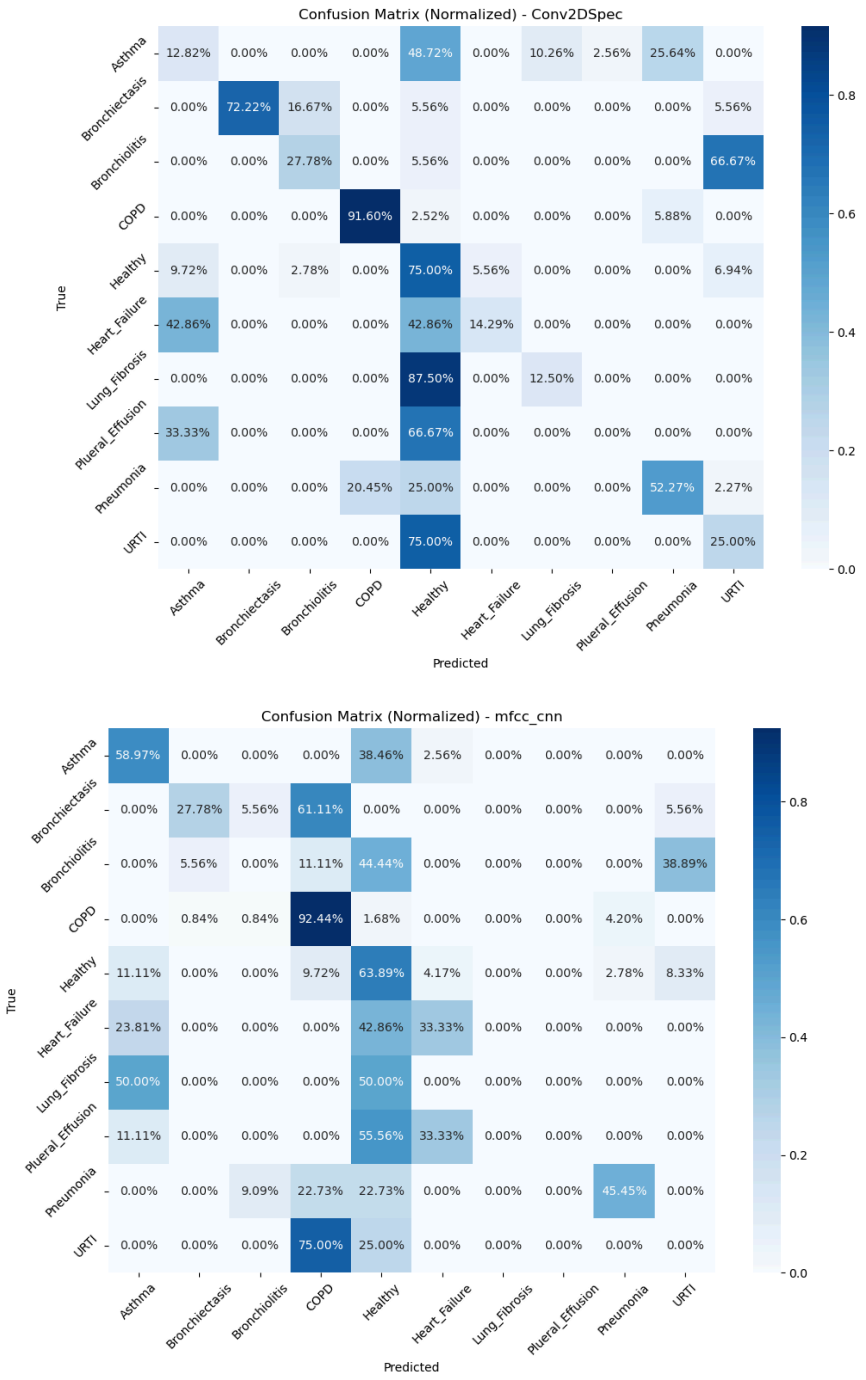


Figure 15: Confusion Matrices for Unaugmented Data DiseaseClassification

Data Augmentation

Looking at the impact of data augmentation, all models showed substantial performance improvements when trained on augmented data. PCEN's performance improved dramatically,

with loss decreasing from 1.39 to 0.51, accuracy increasing from 69.6% to 85.0%, and F1-score improving from 0.68 to 0.85. Similar improvements were seen in the log-spectrogram model, with loss reducing from 1.69 to 0.68, accuracy increasing from 60.8% to 82.9%, and F1-score improving from 0.59 to 0.83. The MFCC model also benefited, with loss decreasing from 2.11 to 1.27, accuracy improving from 58.6% to 72.9%, and F1-score increasing from 0.55 to 0.72. AUC_ovr scores showed significant gains across all models, with PCEN improving from 0.844 to 0.985, spectrogram from 0.844 to 0.975, and MFCC from 0.744 to 0.942. These consistent improvements across all metrics and models demonstrate the substantial value of data augmentation in enhancing model performance.

Generalization of PCEN Parameter Optimization Across Datasets

The optimization of PCEN parameters across diverse datasets provides critical insights into their generalizability and adaptability for medical diagnostic tasks. By evaluating PCEN on the comprehensive diagnostic dataset containing ten classes—comprising all WAV files utilized in sub-datasets—we analyzed the impact of parameter tuning on model evaluation metrics. Notably, epsilon (1e-8) was excluded from detailed analysis, as changing it has a negligible impact on model performance. Below is the table showing the trials required to get to the optimized parameters for the disease classification dataset. (Table 2, below)

PCEN DISEASE MODEL	PCEN PARAMETERS					EVALUATION					
	alpha	delta	root	smooth	epsilon	LOSS	ACC	PRECISION	RECALL	F1_SCORE	ROC
Trial 1	0.94	2.5	0.47	0.01	1.00E-08	0.8339	0.7956	0.8062	0.7956	0.7942	0.9712
Trial 2	0.94	2.5	0.65	0.01	1.00E-08	0.6305	0.8398	0.8439	0.8398	0.8374	0.9797
Trial 3	0.94	1.5	0.65	0.01	1.00E-08	0.6688	0.8269	0.8345	0.8269	0.8264	0.9774
Trial 4	0.8	1.5	0.65	0.01	1.00E-08	0.6252	0.8219	0.8335	0.8219	0.8232	0.9783
Trial 5	0.8	1.5	0.65	0.02	1.00E-08	0.6638	0.8254	0.8429	0.8254	0.8254	0.978
Trial 6	0.8	3.5	0.65	0.02	1.00E-08	0.6212	0.8204	0.8405	0.8204	0.8209	0.9812
Trial 7	0.8	3.5	0.75	0.01	1.00E-08	0.6021	0.8398	0.8488	0.8398	0.8397	0.98
Trial 8	0.8	3.5	0.75	0.007	1.00E-08	0.5897	0.8373	0.8572	0.8373	0.8405	0.9808
Trial 9	0.8	3.5	0.75	0.003	1.00E-08	0.6343	0.8428	0.8538	0.8428	0.8414	0.9764
Trial 10	0.8	4.5	0.85	0.007	1.00E-08	0.5565	0.8423	0.8504	0.8423	0.8412	0.9802
Trial 11	0.7	5.5	0.85	0.007	1.00E-08	0.5316	0.8507	0.8679	0.8507	0.8518	0.9851
Trial 12	0.6	6.5	0.85	0.007	1.00E-08	0.6206	0.8239	0.842	0.8239	0.8241	0.9804

Table 2: Data

The optimized parameters for the multi-class disease classification dataset were achieved in trial 11—alpha = 0.70, delta = 5.5, root = 0.85, T = 0.0070, epsilon = 1e-8—were designed to maximize classification accuracy and minimize loss. These parameters, while fine-tuned for

disease classification, also served as a reliable baseline for other datasets, requiring only minor adjustments to accommodate specific task characteristics.

Certain parameters exhibited stability across all datasets. Delta, with a value of 5.5, effectively balanced amplitude variability, ensuring robust detection of faint and subtle respiratory patterns. The root parameter, ranging from 0.85 to 0.90, preserved nuanced audio details across tasks, improving signal retention critical for accurate classification. Meanwhile, epsilon (1e-8) remained constant across all datasets, acting as a safeguard against division-by-zero errors without influencing the overall performance.

Despite this overarching consistency, some datasets required specific parameter adjustments to optimize results. For datasets involving wheezes and crackles, a slightly higher alpha (0.75) and root (0.90) emphasized transient features, enhancing the model's ability to differentiate these sounds effectively. The temporal resolution (T = 0.0070) remained unchanged, striking a balance between capturing rapid changes and maintaining stability in audio signals.

For the normal-abnormal dataset, alpha was further increased to 0.80, paired with a temporal resolution of 0.0075. These adjustments smoothed audio variations more aggressively, aiding in the broader classification task of separating normal from abnormal respiratory patterns. Root remained at 0.85, ensuring a consistent balance between signal compression and the preservation of critical details.

The unaugmented dataset also benefited from tailored adjustments, with alpha set to 0.80 and root increased to 0.90. These parameters effectively preserved signal details while managing noise in the simpler, raw input. Temporal sensitivity was maintained at T = 0.0070, ensuring reliable performance across tasks. The clean dataset mirrored the disease classification dataset, reflecting similar requirements for multi-class diagnostic accuracy.

These findings highlight PCEN’s adaptability to diverse datasets and tasks. Core parameters such as delta, epsilon, and root provided a consistent foundation, while minor adjustments to alpha and T optimized performance for specific challenges, such as distinguishing transient wheezes or addressing broader classification tasks. This adaptability demonstrates PCEN's potential as a versatile tool in medical diagnostics.

Final Analysis

PCEN consistently outperforms mel spectrograms and MFCCs across all classification tasks due to its adaptive normalization properties and robustness to volume variations. Combining automatic gain control (through alpha and delta parameters) and dynamic range compression (through root) allows PCEN to better handle the variable amplitudes and background noise common in respiratory recordings while preserving important temporal dynamics. Log-mel spectrograms, though providing reasonable frequency content representation, are more sensitive to amplitude variations and background noise. MFCCs perform the worst because they discard phase information and temporal dynamics through their discrete cosine transform step, and solely compression through logarithmic means is less suited

for capturing subtle spectral variations in respiratory sounds. From a statistical standpoint, while spectrogram approaches remain competitive, PCEN's strong and consistent performance across the dataset, combined with its flexibility in handling diverse amplitude dynamics, shows its superiority for signal digitization in audio classification tasks and suggests potential for further improvement through gradient-based optimization of its parameters, ultimately advancing the reliability of automated lung sound diagnostics.

Conclusion

This study examined the effectiveness of different audio preprocessing methods—log-mel spectrograms, MFCCs, and PCEN—for classifying lung sounds, focusing on PCEN's adaptability and performance. Using data from the ICBHI 2017 Challenge and Kaggle, we created a comprehensive and augmented dataset to address three classification tasks: identifying specific respiratory diseases, distinguishing wheezes, crackles, and normal sounds, and separating normal from abnormal lung sounds. Preprocessing steps included audio augmentation, segmentation into uniform-length clips, and feature extraction through spectrogram-based methods. These were then evaluated using a convolutional neural network (CNN) designed to capture detailed time-frequency features. Data augmentation and careful validation techniques were used to mitigate overfitting and ensure robust model performance.

PCEN demonstrated remarkable adaptability and noise suppression, significantly enhancing model performance. By reducing background noise and amplifying active sounds, PCEN consistently outperformed MFCCs and log-mel spectrograms regarding loss and accuracy. The tunability of PCEN emerged as a central strength, with small adjustments in parameters such as T (temporal smoothing) and r (root compression) yielding substantial performance gains. This underscores the importance of automated parameter optimization, such as gradient-based tuning, to harness PCEN's potential fully. Despite its computational overhead and tuning complexities, PCEN's consistent performance across diverse datasets—including augmented and unaugmented data—validates its scalability and reliability in real-world clinical settings.

The study's three datasets offer distinct practical applications. The Abnormal/Normal Classification Dataset is suitable for general health screenings, enabling quick identification of respiratory issues. The Disease Classification Dataset provides detailed diagnostic capabilities for conditions like COPD and pneumonia, while the Wheeze/Crackle/Normal Classification Dataset supports symptom-specific diagnostic assistance, aligning sound patterns with potential respiratory conditions.

Future Outlook

Looking forward, incorporating gradient descent optimization to fine-tune PCEN parameters such as T, r, α, and δ within frameworks like PyTorch could unlock even greater performance gains. This dynamic approach would enable models to learn optimal settings for specific datasets, enhancing noise suppression and feature extraction. Further research will expand PCEN's use to other medical domains, such as heart or vocal sound analysis, potentially broadening its clinical impact.

Exploring advanced neural architectures like transformers and integrating PCEN into real-time mobile or wearable device systems can facilitate remote health monitoring, especially in resource-limited settings. By combining advanced preprocessing, parameter optimization, and robust modeling, this research lays the foundation for scalable, reliable, and effective diagnostic tools that enhance clinical workflows and improve patient outcomes worldwide.

References

[1] Aykanat, Murat, et al. “Classification of Lung Sounds Using Convolutional Neural Networks - EURASIP Journal on Image and Video Processing.” *SpringerLink*, Springer International Publishing, 11 Sept. 2017, link.springer.com/article/10.1186/s13640-017-0213-2#Sec2.

[2] Hoffman, Matthew. “Lung Diseases Overview.” Edited by Carmelita Swiner, *WebMD*, WebMD, 3 Dec. 2022, www.webmd.com/lung/lung-diseases-overview.

[3] Abdul, Zrar Kh., and Abdulbasit K. Al-Talabani. “Mel Frequency Cepstral Coefficient and Its Applications: A Review.” *IEEE Xplore*, IEEE Journals & Magazine, ieeexplore.ieee.org/abstract/document/9955539/. Accessed 21 Jan. 2025.

[4] Deruty, Emmanuel. “Intuitive Understanding of Mfccs.” *Medium*, Medium, 15 Dec. 2022, medium.com/@derutycsl/intuitive-understanding-of-mfccs-836d36a1f779.

[5] Fraiwan, M., et al. “Recognition of Pulmonary Diseases from Lung Sounds Using Convolutional Neural Networks and Long Short-Term Memory - Journal of Ambient Intelligence and Humanized Computing.” *SpringerLink*, Springer Berlin Heidelberg, 3 Apr. 2021, link.springer.com/article/10.1007/s12652-021-03184-y#Sec1.

[6] Adams, Seth. “Code for YouTube Series: Deep Learning for Audio Classification.” *GitHub*, Microsoft, github.com/seth814/Audio-Classification. Accessed 20 Jan. 2025.

[7] Scikit. “3.4. Metrics and Scoring: Quantifying the Quality of Predictions.” *Scikit-Learn*, scikit-learn.org/stable/modules/model_evaluation.html. Accessed 20 Jan. 2025.

[8] “ICBHI 2017 Challenge.” *ICBHI Challenge*, bhichallenge.med.auth.gr/ICBHI_2017_Challenge. Accessed 20 Jan. 2025.

[9] Möbius. “A Dataset of Lung Sounds.” *Kaggle*, 16 Dec. 2021, www.kaggle.com/datasets/arashnic/lung-dataset?resource=download.

[10] Choi, Youngjin, and Hongchul Lee. “Interpretation of Lung Disease Classification with Light Attention Connected Module.” *Biomedical Signal Processing and Control*, U.S. National Library of Medicine, July 2023, pmc.ncbi.nlm.nih.gov/articles/PMC9978539/.

[11] Lostanlen, Vincent, et al. *Per-Channel Energy Normalization: Why and How*, www.justinsalamon.com/uploads/4/3/9/4/4394963/lostanlen_pcen_spl2018.pdf. Accessed 20 Jan. 2025.

[12] Lostanlen, Vincent. “Self-Calibrating Acoustic Sensor Networks with per-Channel Energy Normalization.” *Accueil - Archive Ouverte HAL*, 17 Oct. 2021, hal.science/hal-03381500.

[13] Choi, Keunwoo, et al. “Kapre: On-GPU Audio Preprocessing Layers for a Quick Implementation of Deep Neural Network Models with Keras.” *arXiv.Org*, 19 June 2017, arxiv.org/abs/1706.05781.