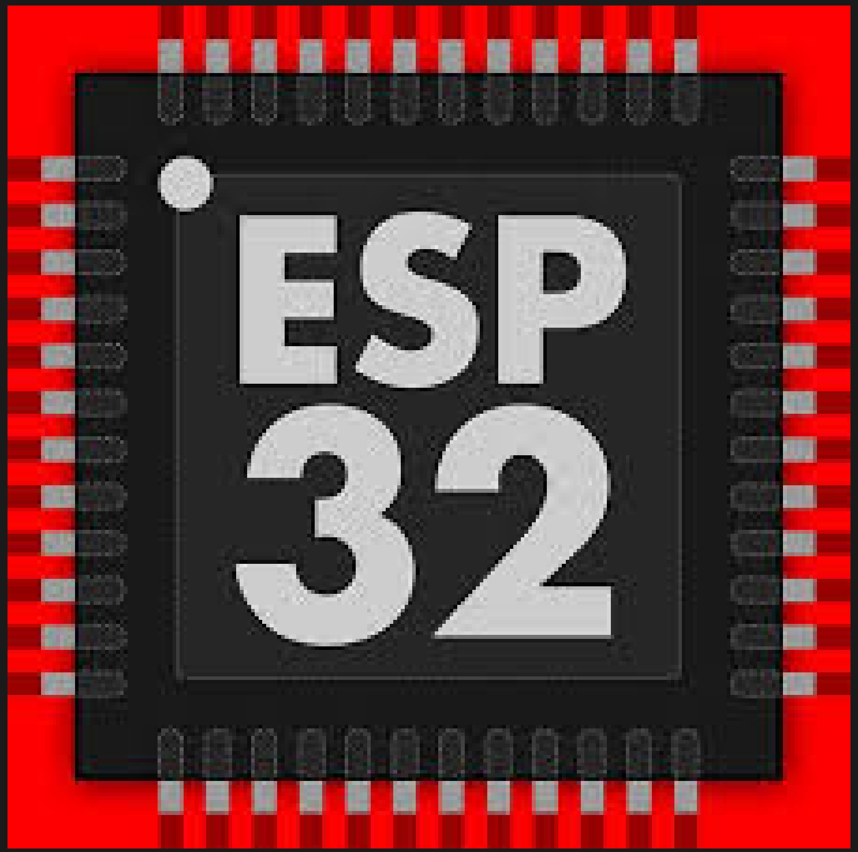
EMBEDDED SYSTEMS WORKSHOP (MONSOON '24)

RESPIRATORY SOUNDS CLASSIFICATION SYSTEM

Professor: Dr. Abhishek Srivastava
TA: Santhoshini Thota

Team Members:
Aditya Dasari
Aditya Nair
Jai Aakash Gopalakrishnan
Sudarshan Nikhil



Development of a System for
Capturing and Classifying Lung Sounds

PROJECT OVERVIEW

OBJECTIVE

- Build a portable system to capture lung sounds, transmit them to a server, and process the audio data for diagnostic use.
- Implement various signal processing techniques, such as bandpass and Gaussian filtering, and generate Mel spectrograms for deeper analysis.

WHY THIS SYSTEM?

- Medical Application: Early detection of respiratory conditions like asthma, bronchitis, and pneumonia through non-invasive audio capture.
- Portable and Scalable: Suitable for low-resource environments or remote patient monitoring.

SYSTEM ARCHITECTURE

OVERVIEW OF THE WORKFLOW

END-TO-END FLOW

AUDIO CAPTURE

INMP441 microphone captures lung sounds using ESP32.

DATA TRANSMISSION

ESP32 sends audio data in 1-second chunks via HTTP to a server.

DATA COMPILATION

The server compiles these 1-second chunks into a raw binary audio file.

AUDIO CONVERSION & PROCESSING

A Python script:

- Converts the raw data into WAV,
- Processes it using bandpass, Gaussian filters, Parametric eq.
- Generates Mel spectrograms for advanced analysis.

Chunked data transfer allows for more manageable data handling, ensuring smooth transmission and reconstruction.

CAPTURING LUNG SOUNDS USING INMP441 AND ESP32

INMP441 MEMS MICROPHONE

- High Fidelity Capture: Suitable for detailed audio capture with a sample rate of 16kHz, making it ideal for lung sounds.
- I2S Interface: Facilitates communication with ESP32 for real-time audio data collection.

ESP32 MICROCONTROLLER

- Configuration: Set up as I2S master to capture 16-bit mono audio data.
- Data Handling: Collects audio in 1-second intervals and transmits it to the server.

KEY ADVANTAGES

- Portability: Wireless data transfer and minimal hardware requirements.
- Real-Time Capture: Suitable for continuous monitoring and real-time diagnostics, which can be implemented during the later stages.

DATA TRANSMISSION

Chunked Data Transmission

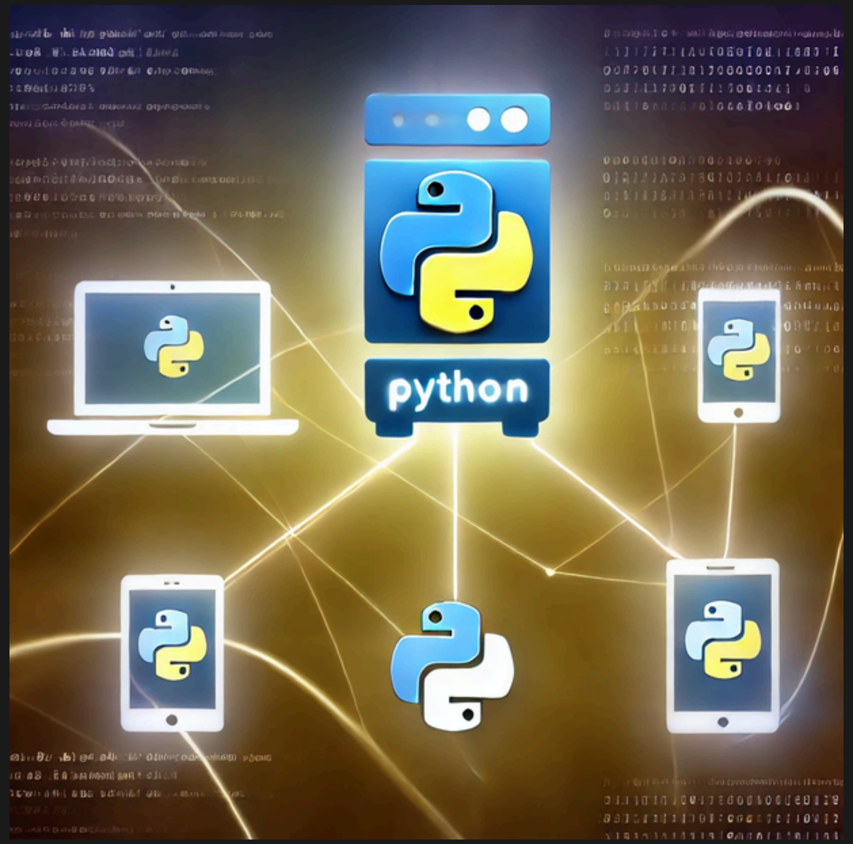
Why Chunks?

- Audio is captured in 1-second chunks to optimize memory and bandwidth usage.

Reliable Transmission:

- Data sent over HTTP POST requests ensures reliable and ordered delivery.

SERVER-SIDE COMPILATION



Flask Server for Data Compilation

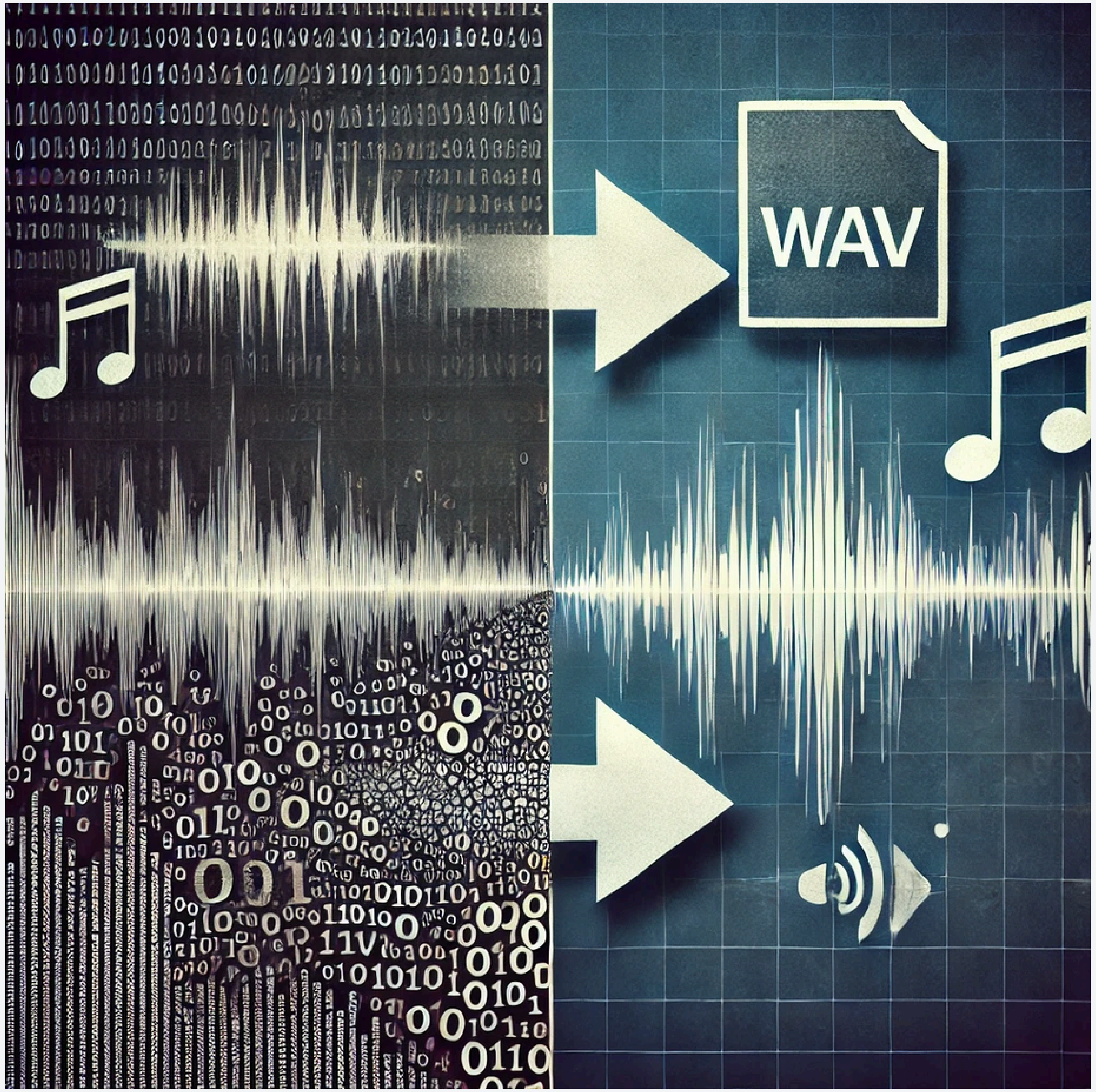
The server listens for incoming data chunks and appends them to a raw binary file, ensuring that no data is lost or reordered.

Ensuring Data Integrity

Sequential Ordering: The server carefully reconstructs the file by appending each chunk in order, ensuring accurate playback.

Turning Raw Data into Audio Ready to be Characterized

PROCESSING & ANALYZING DATA



AUDIO PROCESSING TECHNIQUES

Enhancing Lung Sound Quality Through Filtering

Bandpass Filtering

- Purpose: Focuses on the key frequency range for lung sounds, typically between 100 Hz and 1000 Hz.
- Effect: Removes unwanted noise outside this range, such as low-frequency background hums and high-frequency disturbances.

Gaussian Filter

- Purpose: Smooths the audio signal to reduce random noise and sharp spikes. Complements the use of bandpass filter when used at moderate intensities.
- Effect: Results in a clearer, more consistent signal, crucial for highlighting subtle lung sounds (e.g., wheezing or crackling).

Direct Subtraction:

Direct subtraction removes specific audio components by mathematically subtracting one signal (e.g., heart sounds) from another (e.g., lung sounds). Precise alignment in time and amplitude ensures heart sounds are minimized, leaving a cleaner lung sound signal.

Parametric Equalization

Parametric EQ adjusts specific frequency bands by controlling center frequency, bandwidth (Q factor), and gain. It enhances heart sound frequencies for isolation, making subtraction from the lung recording more effective

OUR LUNG SOUND FILTERING TECHNIQUE

Our filtering technique isolates heart and lung sounds programmatically from stethoscope recordings using signal processing. Heart sounds are extracted via a bandpass filter (20–100 Hz) or parametric EQ, ensuring minimal lung sound leakage. The extracted heart sounds are directly subtracted from the original signal to remove their contribution without distortion. This approach preserves lung sounds for further analysis, optimizing the recording for use in machine learning models like CNNs.

A screenshot of the visual plugin used while experimenting on the filter curve

WHY OTHER TECHNIQUES WERE REJECTED

Techniques like noise gating and amplitude-based filtering were explored but found ineffective for separating heart and lung sounds. Noise gating, which sets a threshold to distinguish signals, failed due to the overlapping amplitudes of heart and lung sounds, resulting in inconsistent separation. Similarly, amplitude-based filtering struggled with the complex overlap in frequency spectra, often leaving heart sounds intact or removing vital lung sound components. These limitations made these methods unreliable for cleanly isolating heart sounds without compromising lung sound quality.

MEL SPECTROGRAMS

VISUALIZING FREQUENCY DYNAMICS IN LUNG SOUNDS

WHAT

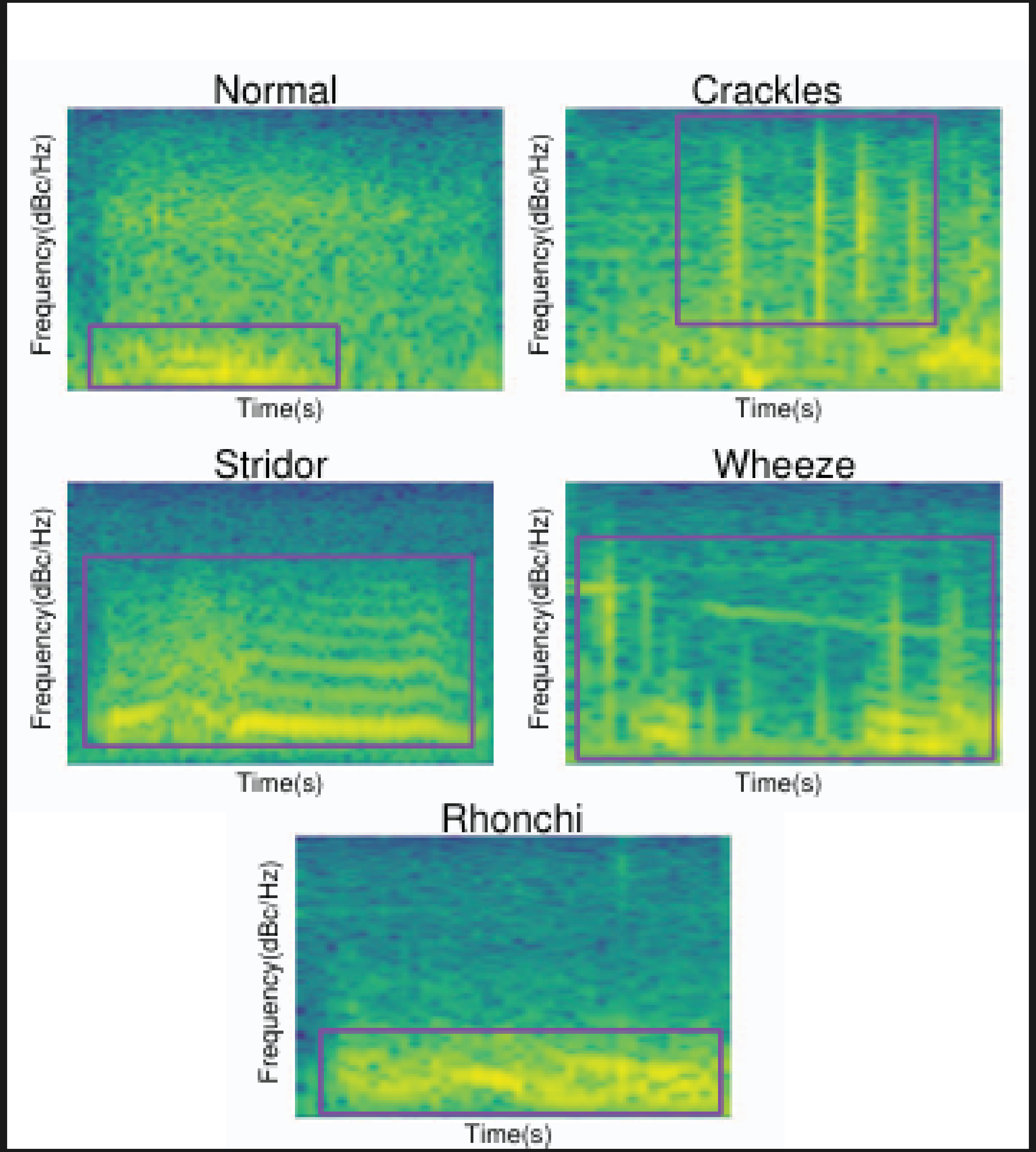
IS A MEL SPECTROGRAM?

- Converts the audio signal into a 2D visual representation, showing how sound frequencies change over time.
- Why use Mel Scale: Emphasizes lower frequencies (50 Hz to 2500 Hz), where lung sounds typically occur, improving the analysis of respiratory signals.

WHY

USE A MEL SPECTROGRAM?

- Enhanced Analysis: Captures both frequency and time-domain dynamics, making it easier to differentiate between types of lung sounds (e.g., wheezes, crackles, and rhonchi).
- Input for Machine Learning Models: Mel spectrograms can be used as input for machine learning models to classify different respiratory conditions.



Source : Deep Learning Based Portable Respiratory Sound Classification System by Adithya Sunil Edakkadan and Abhishek Srivastava

DATA PREPROCESSING AND FEATURE EXTRACTION

Steps Involved:

- Audio Loading: Standardizes audio to 15 seconds with zero-padding for shorter clips.
- Fourier Transform: Uses STFT to transform audio into the frequency domain.
- Mel Scale Transformation: Converts the frequency scale to the Mel scale, mimicking human auditory perception and emphasizing lower frequencies.
- Logarithmic Scaling: Converts amplitude values to decibels (dB) for perceptual relevance.
- Mel spectrograms display time (x-axis), frequency bands (y-axis), and amplitude intensity (color).

MODEL ARCHITECTURE AND TRAINING

Architecture

- Convolutional Layers (32, 64, 128 filters) with ReLU for feature extraction.
- Max-Pooling to reduce dimensions and highlight key features.
- Dropout Layers (25%-50%) to prevent overfitting.
- Fully Connected Layer (256 units) consolidates features.
- Output Layer with Sigmoid activation for multi-label classification: wheeze, stridor, rhonchi, crackles.

Training

- Audio files converted to Mel spectrograms and normalized with StandardScaler.
- Trained for 20 epochs (batch size: 32).
- Validation performed on a separate test dataset.
- Outputs saved: model (.h5) and scaler (.pkl).

PREDICTION PROCESS

Steps for Prediction:

1. **Audio Preprocessing**: Convert audio to Mel spectrogram.
2. **Feature Normalization**: Use saved scaler from training.
3. **Model Prediction**: Output probabilities for lung conditions.
4. **Thresholding**: Apply 0.5 threshold to classify conditions.

- **Example**:
  Input: steth_20190626_15_12_23.wav.
  Output: Detected conditions: [‘wheeze’, ‘rhonchi’].

MODEL EVALUATION AND STRENGTHS

Evaluation Metrics

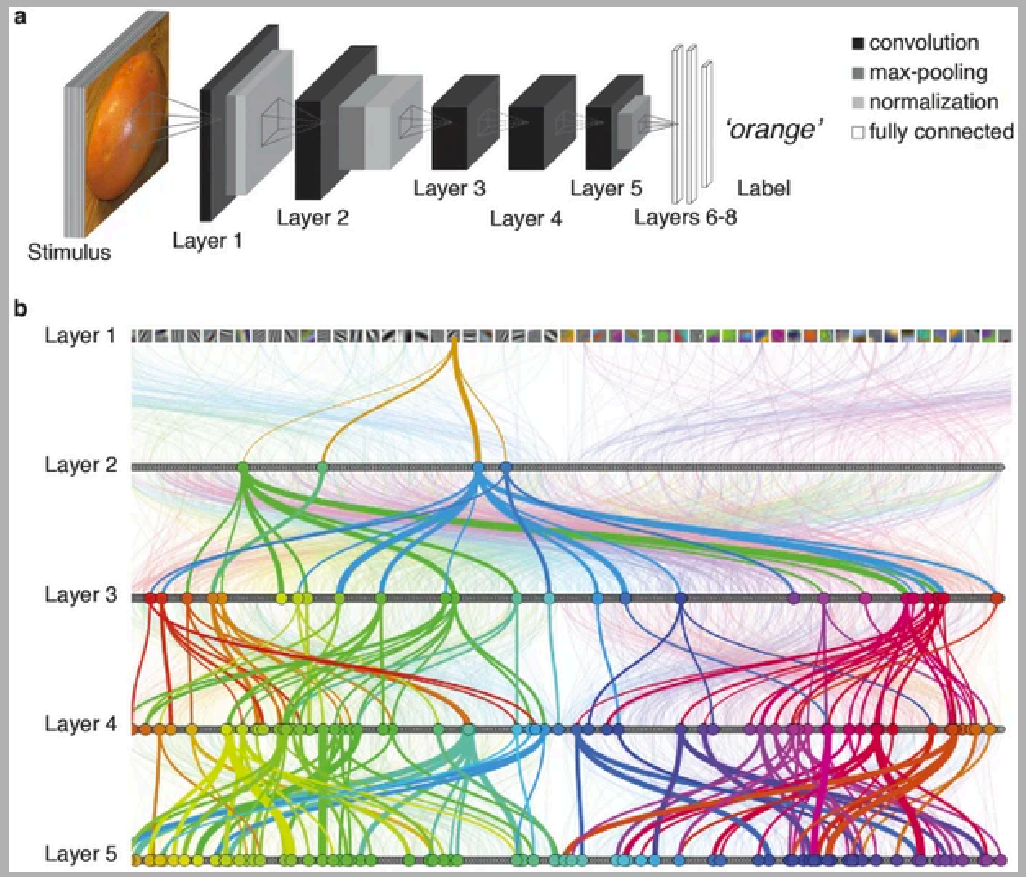
- Standard metrics used: accuracy, precision, loss, and F1 score.
- Multi-label classification allows simultaneous prediction of multiple conditions.

Performance Highlights

- Effective in detecting wheeze, stridor, rhonchi, and crackles.
- Robust handling of overlapping respiratory sounds.

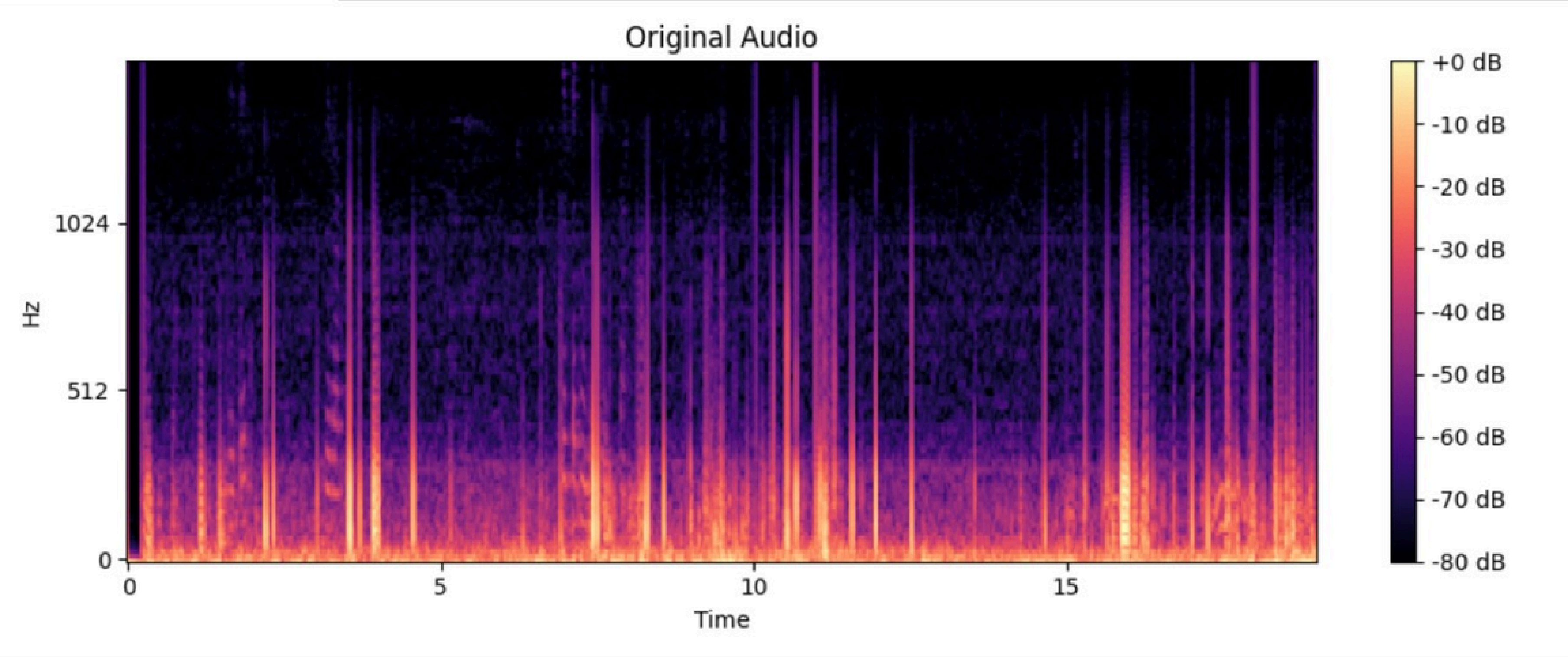
Strengths

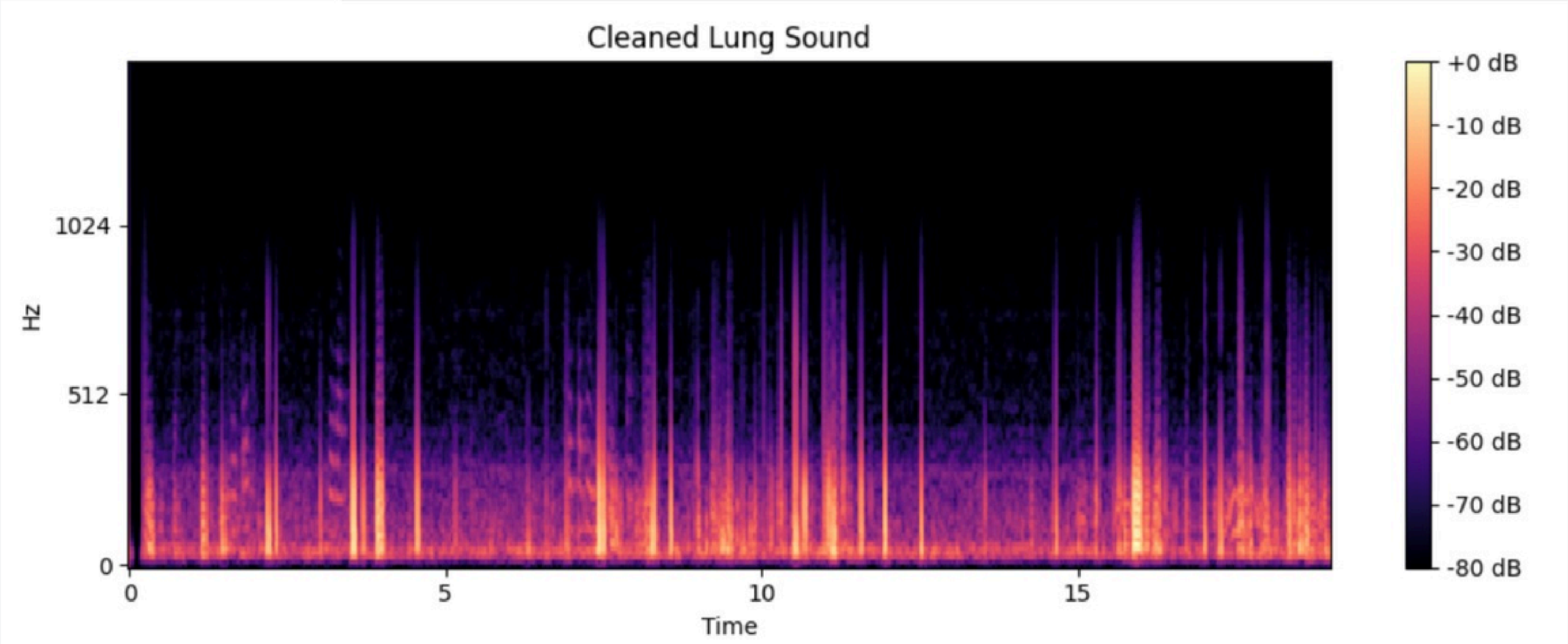
- **Uses Mel spectrograms for feature-rich input.**
- **Handles diverse respiratory conditions with high reliability.**
- **Applicable for real-time diagnostic systems in healthcare.**



MEL SPECTROGRAM - LUNGS

1) BEFORE FILTERING



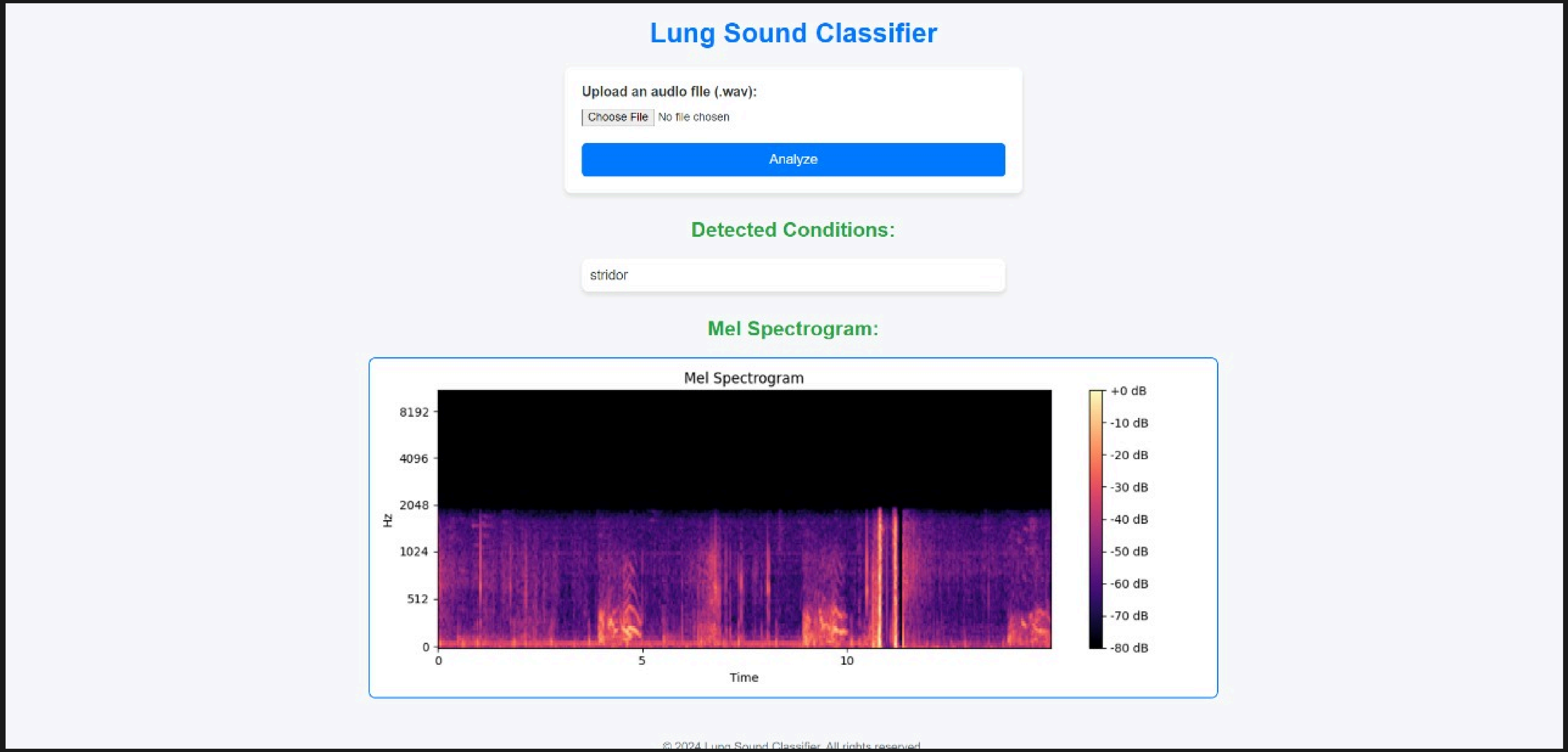


2) AFTER FILTERING

LeBr-onboard Systems

OUR GUI

Our GUI lets the user input their lung audio and process it through the click of a button abstracting away the filtering and CNN model. It then accurately returns the result of the detection of the user's lungs and the corresponding Mel Spectrogram.



REFERENCES

- Deep Learning Based Portable Respiratory Sound Classification System by Adithya Sunil Edakkadan and Abhishek Srivastava
- Lung Sound Classification with CNN
- HF_Lung V1 Lung sounds database - https://gitlab.com/techsupportHF/HF_Lung_V1

LeBr-onboard Systems

THANK YOU

Team Members:
Aditya Dasari
Aditya Nair
Jai Aakash Gopalakrishnan
Sudarshan Nikhil

04 December 2024