Lung Sound Recorder and Classification System

Project Report December 2023

Santhoshini Thota

-

Abstract

This project focuses on the development of a comprehensive system for the recording and analysis of lung sounds, aimed at early diagnosis of respiratory diseases. Leveraging an ESP32 microcontroller and the INMP441 I2S microphone module, the system captures bronchovesicular (BV) and vesicular (V) signals, transmitting the recorded data to a server for further processing. The project encompasses various components, including noise reduction, heartbeats removal, generation of Mel spectrograms, and classification using Support Vector Machines (SVMs). The processed data is visualized on a dedicated webpage, offering an intuitive interface for interpreting lung sounds. Achievements include robust noise reduction, heartbeats removal, and accurate classification of lung sounds, giving further space for evolution of the system to identify the type of disease, positioning the project as a valuable tool for early detection and monitoring of respiratory conditions. Future work may explore further enhancements in classification accuracy and system applicability.

Contents

Chapter 1

Overview

1.1 Introduction

To identify health issues related to breathing and lungs, we rely on the lung sounds. So, we need a device that can extract these lung sounds using a microphone to help doctors diagnose illnesses more effectively.

1.2 Objectives

The main Objectives are as follows:

(1) Design a device that can extract lung sounds using stethoscope and microphone AFE along with the temperature readings of the person and integrating them on ESP32.

(2) Sending the recorded wave file to webpage/dashboard for further processing.

1.3 The Flow

The source of this flow is the microphone on the ESP (specialized device with attached stethoscope). Once the audio is recorded, it is then sent (along with the temperature data) to the server (local computer) for further audio processing. There, the audio file is gone through various pre-processing techniques. Then, sent to SVM classifier trained on labelled lung sounds. Results are then displayed on the webpage / dashboard.

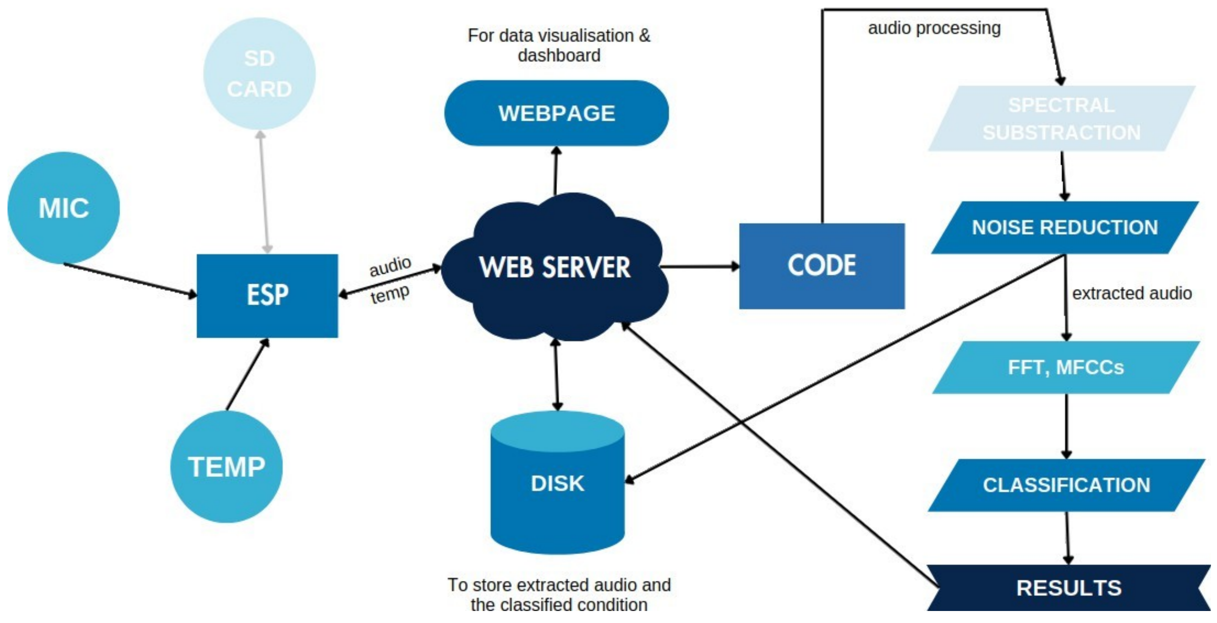


FIGURE 1.1. Plan of Implementation

Chapter 2

Recorder

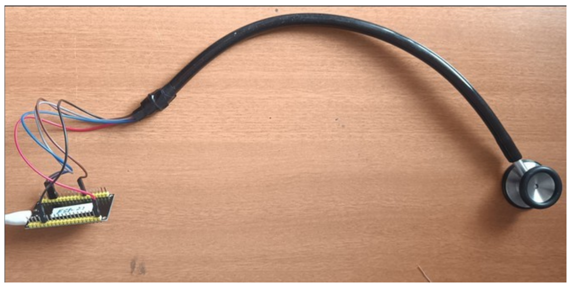
2.1 The Device

In our specialized device, we have connected the INMP441 microphone to the cut-part of a Littmann Stethoscope. This device seamlessly integrates traditional auscultation methods with modern technology, providing healthcare professionals with an advanced tool for comprehensive lung examination.

2.1.1 Microphone

The INMP441 is a high-performance, low-power, digital MEMS (Micro-Electro-Mechanical System) microphone designed for use in various audio applications.

(a) INMP441


(b) Specialized Device

2.2 Recording

This microphone uses I2S interface when reading the recorded bytes from the module. So, to create a .wav file from the flash bytes, we need to create a file on the SPIFFS file system of the ESP32. Then, a header is written to the file, which represents the type of file, sampling rate, sample bits, etc. Those flash bytes from the i2s read essentially get written to that file. Parallel to the recording, we also incorporated a ”File Browser” code [From Arduino Samples], which creates a server that provides the recorded audio in the SPIFFS to the client via http.

We have set the following audio standards for the recording.

- Sample Rate - 44100
- Sample Bits - 16
- Record Time - 15 sec

Chapter 3

Audio Pre-Processing

3.1 MEL spectrograms

MEL is a type of spectrogram that is used to represent the frequency content of audio signals. MEL spectrograms are more sensitive to changes in pitch in the low-frequency range than in the high-frequency range. The vertical axis of the mel spectrogram represents the mel frequency bands, and the horizontal axis represents time. The brightness of each pixel represents the amplitude of the corresponding mel frequency band at a given time.

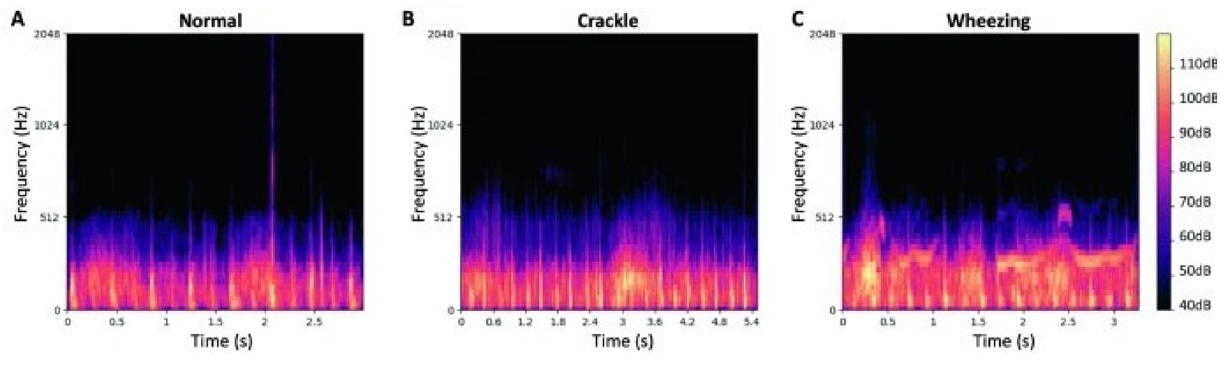


FIGURE 3.1. MEL spectrograms of various lung sounds

3.2 Noise Reduction

3.2.1 Removing noise

The audio denoising algorithm used is Spectral Gating. It is a noise reduction algorithm that works by estimating a noise threshold for each frequency band of an audio signal and then suppressing frequency components below that threshold. This can help to reduce noise and improve the clarity of the signal. The algorithm is implemented in the noisereduce library in Python, which is designed for noise reduction in time-domain signals. The key steps of the algorithm are

(1) Compute the spectrogram of the signal.
(2) Estimate the noise floor.
(3) Compute the noise gate.
(4) Mask the spectrogram.
(5) Compute the inverse spectrogram.

3.2.2 Amplification

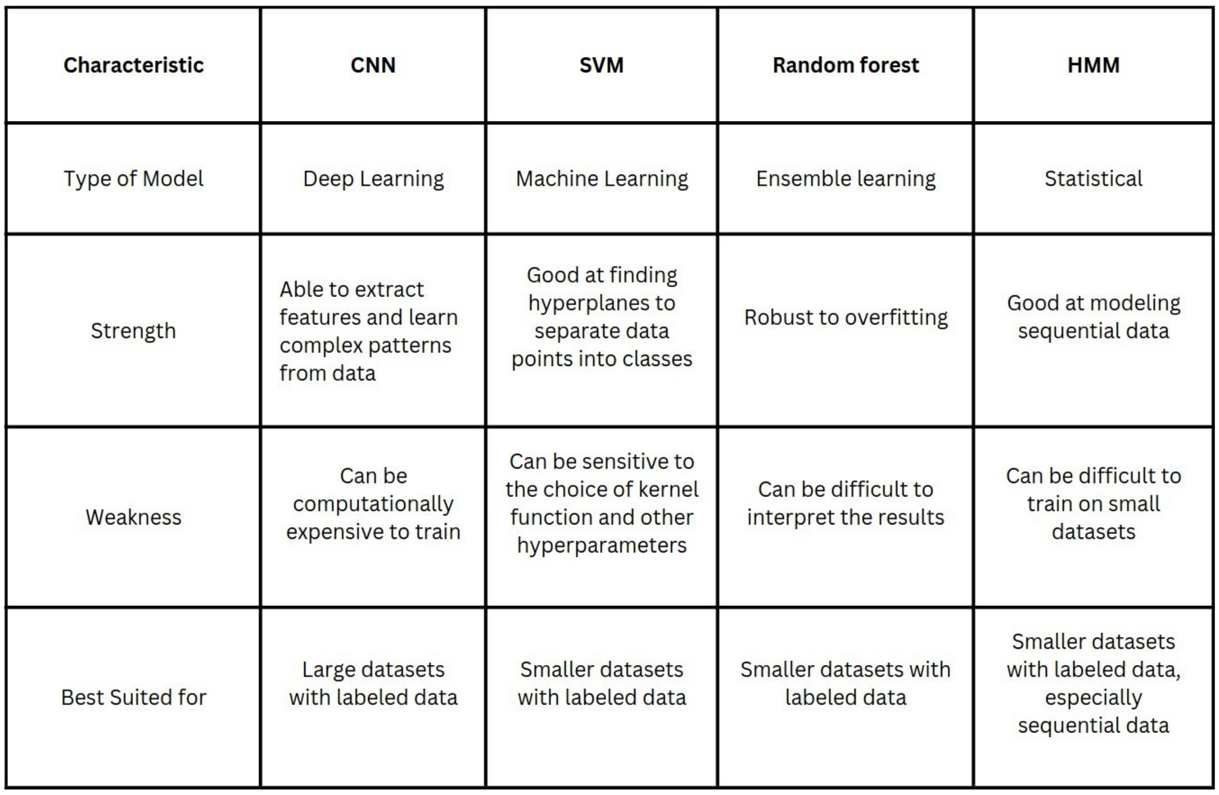
The denoised audio is then amplified by an amplification factor in decibels. This is implemented using the soundfile library. The soundfile library is a comprehensive audio processing library in Python that facilitates various operations, including reading and writing audio files, extracting audio metadata, and performing audio manipulations

3.2.3 Removing heart beats

Heartbeats can introduce interference in lung sound recordings. So we use high pass filter to remove the heartbeats. The high-pass filter operates by allowing signals with frequencies above a certain threshold (the cutoff frequency) to pass through, while attenuating frequencies below this threshold. The cutoff frequency selected to target the characteristic frequency range of heartbeats.

Chapter 4

Classification



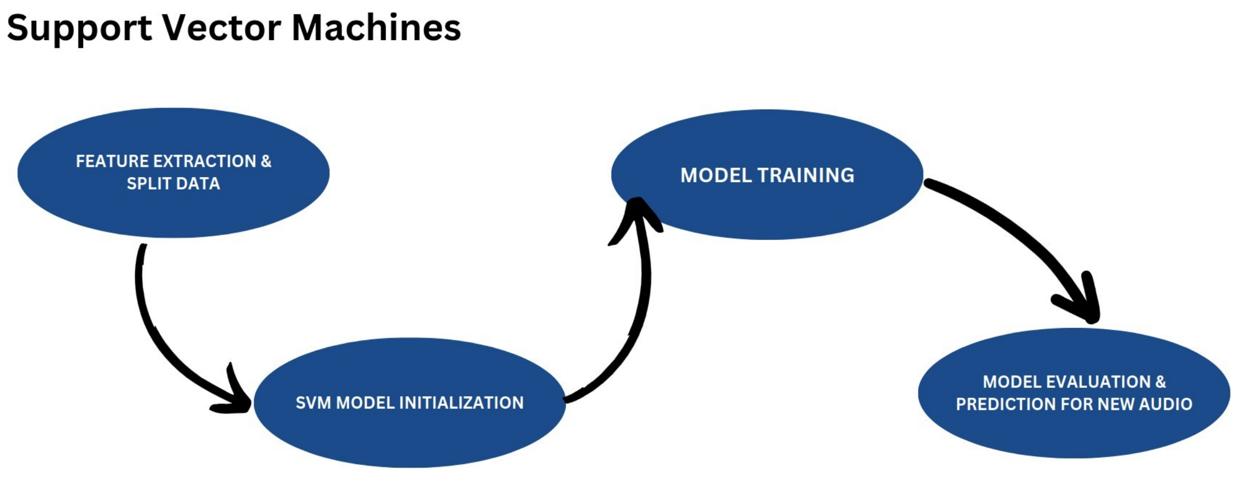
Characteristic	CNN	SVM	Random forest	HMM
Type of Model	Deep Learning	Machine Learning	Ensemble learning	Statistical
Strength	Able to extract features and learn complex patterns from data	Good at finding hyperplanes to separate data points into classes	Robust to overfitting	Good at modeling sequential data
Weakness	Can be computationally expensive to train	Can be sensitive to the choice of kernel function and other hyperparameters	Can be difficult to interpret the results	Can be difficult to train on small datasets
Best Suited for	Large datasets with labeled data	Smaller datasets with labeled data	Smaller datasets with labeled data	Smaller datasets with labeled data, especially sequential data

FIGURE 4.1. Analysis

4.1 Using SVM for Classification

Support Vector Machines (SVM) are employed for audio classification. SVM is a powerful machine learning algorithm used for classification and regression tasks.

4.1.1 Algorithm Explanation



Support Vector Machines
FEATURE EXTRACTION & SPLIT DATA
SVM MODEL INITIALIZATION
MODEL TRAINING
MODEL EVALUATION & PREDICTION FOR NEW AUDIO

FIGURE 4.2. SVM

Audio files are organized into different categories based on the extracted feature. After splitting the data into training and testing sets, a Support Vector Machine (SVM) classifier with a linear kernel is trained. A prediction is made for a new audio file, categorizing it as wheeze, crackle, both, or normal.

The SVM algorithm involves the following steps:

(1) **Feature Extraction**

(2) **Data Preparation**

(3) **Model Training**

(4) **Model Evaluation**

4.1.2 Audio Classification Script in Python

Scikit-learn library to implement a machine learning model for audio classification. The classification is designed to categorize audio into specific classes: wheeze, wheeze-crackle, crackle, or normal.

4.1.3 Feature Extraction

The extract features function, employing Librosa, captures essential audio features such as Mel-frequency cepstral coefficients (MFCCs), spectral centroid, spectral bandwidth, and zero-crossing rate.

4.1.4 Data Organization

Audio files are organized into distinct categories, extracts features from each file, and assigns corresponding labels based on the classification criteria.

4.1.5 Model Training and Evaluation

Data is divided into training and testing sets, a Support Vector Machine (SVM) classifier with a linear kernel is trained. The script evaluates the accuracy of the model on the test set.

4.1.6 Prediction for New Audio

A prediction is made for a new audio file categorizing it as either wheeze, wheeze-crackle, crackle, or normal.

Chapter 5

Classification Results / Dashboard

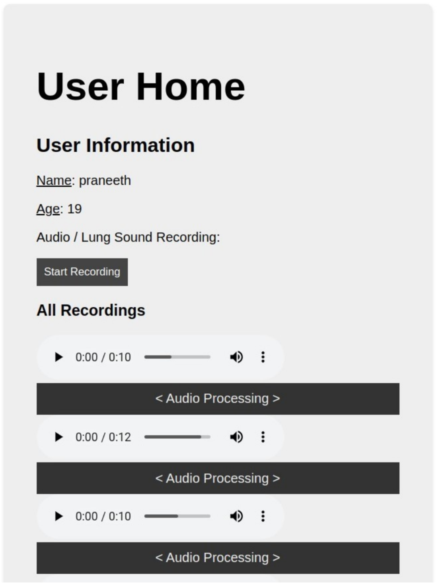
All the Pre-processing and Classification are done on a local Flask server. Based on classification results on the type of Lung sound and temperature data from the ESP, further assessment of the disease can be initiated. After the classification phase, Results will be returned to the user, and displayed on the Web Interface.

5.1 User Interface

A Web Interface has been developed where communication with the flask server is established via http and web sockets (for Live Audio Processing). With this user interface, one can create ”patient entry”, then, record their lung sounds, finally, the processing and classification results are displayed.

5.1.1 Visual Representation

Once the audio file is received by flask server, it is then sent to the requested user. Then, from there, each processing & Classification stage are made available to the user. Audio files at each stage are visualized with the MEL spectrograms. The audio file can be taken through each stage from the UI. The processed audio files can be found in the ”static” folder of the codebase. Finally, Classification results are displayed on the web interface


User Home

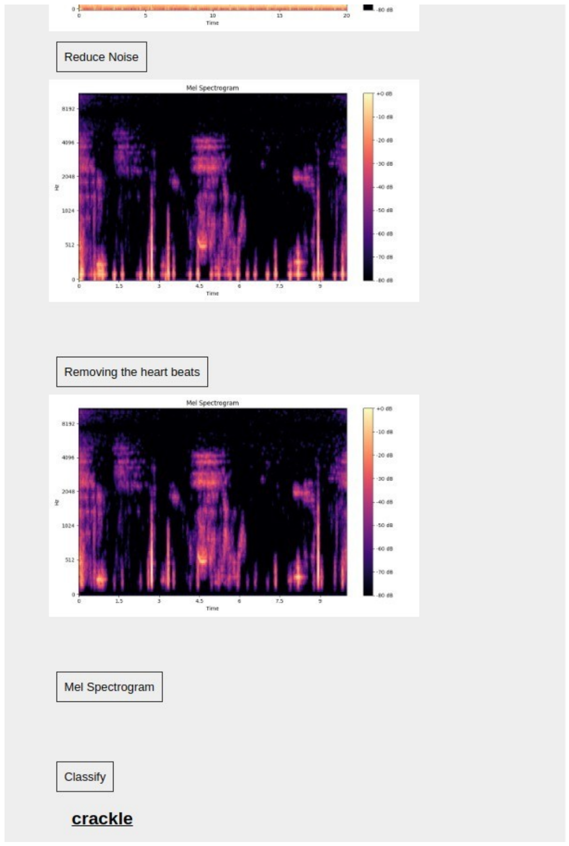
User Information

Name: praneeth

Age: 19

Audio / Lung Sound Recording:

(a) Web Interface with recorded audios



(b) Visual Representation at each stage

5.1.2 Database storing patients

We administer a database that securely manages confidential data pertaining to patients and their respiratory sound recordings. Authorized medical professionals possess access privileges exclusively to information concerning patients under their care.

Bibliography

https://www.sciencedirect.com/science/article/abs/pii/S0933365717302051

datasets:
https://doi.org/10.17632/jwyy9np4gv.3
https://data.mendeley.com/datasets/jwyy9n
p4gv/3

Amplification:
https://g.co/bard/share/1bd18dfcf4a1

Recording lung sounds with mic:
https://www.atomic14.com/2020/09/12/esp32-audio-input.html
https://github.com/0015/ThatProject/tree/master/ESP32$_M$
ICROPHONE

Lung Sound Auscultation
https://youtu.be/2NvBk61ngDY?si=nOZtcsPfdxRB2E11