A

Major Project Report

On

DEEP LEARNING BASED LUNG DIAGNOSIS USING LUNG SOUNDS

Submitted to CMREC (UGC Autonomous), Affiliated to JNTUH

In Partial Fulfilment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

(Artificial Intelligence & Machine Learning)

Submitted By

MAYADHAR SENAPATI	**(228R1A6642)**
S. HRUTHASHINI	**(228R1A6654)**
LAKSHMI PRASANA	**(228R1A6601)**
SUNDARAM PANDEY	**(228R1A6658)**

Under the Esteemed guidance of

Mr. Sunil Kumar Singh

Assistant professor

Department of CSE (AI & ML)

Department of Computer Science & Engineering (AI & ML)

CMR ENGINEERING COLLEGE

(UGC AUTONOMOUS)

(Accredited by NAAC & NBA, Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad, Kandlakoya, Medchal Road, R. R. Dist. Hyderabad-501 401)

(2025 – 2026)

CMR ENGINEERING COLLEGE

(UGC AUTONOMOUS)

(Accredited by NAAC & NBA, Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad, Kandlakoya, Medchal Road, R. R. Dist. Hyderabad-501 401)

Department of Computer Science & Engineering (AI & ML)

CERTIFICATE

This is to certify that the Major project entitled **"DEEP LEARNING BASED LUNG DIAGNOSIS USING LUNG SOUNDS"** is a bonafide work carried out by

MAYADHAR SENAPATI	**(228R1A6642)**
S. HRUTHASHINI	**(228R1A6654)**
LAKSHMI PRASANA	**(228R1A6601)**
SUNDARAM PANDEY	**(228R1A6658)**

in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING (AI & ML)** from CMR Engineering College, under our guidance and supervision

The results presented in this Major project have been verified and are found to be satisfactory. The results embodied in this Major project have not been submitted to any other university for the award of any other degree or diploma

Internal Guide	**Major Project Coordinator**	**Head of the Department**
Mr. Sunil Kumar Singh	Mr. G. Venkateswarlu	Dr. Madhavi Pingili
Assistant Professor	Assistant Professor	Professor & HOD
Department of CSE(AI&ML)	Department of CSE(AI&ML)	Department of CSE(AI&ML)

External Examiner : ___________________

DECLARATION

This is to certify that the work reported in the present Major project entitled **"DEEP LEARNING BASED LUNG DIAGNOSIS USING LUNG SOUNDS"** is a record of bonafide work done by us in the Department of Computer Science and Engineering (AI & ML), CMR Engineering College. The reports are based on the Major project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the Major project in the future.

The results embodied in this Major project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

MAYADHAR SENAPATI (228R1A6642)

S. HRUTHASHINI (228R1A6654)

LAKSHMI PRASANA (228R1A6601)

SUNDARAM PANDEY (228R1A6658)

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. Madhavi Pingili,** Professor & HOD, **Department of CSE (AI & ML), CMR Engineering College** for their constant support.

We are extremely thankful to **Mr. Sunil Kumar Singh,** Assistant Professor, Internal Guide, Department of CSE (AI & ML), for his constant guidance, encouragement and moral support throughout the Major project.

We will be failing in duty if we do not acknowledge with grateful thanks the authors of the references and other literature referred to in this Major project.

We thank **Mr. G. Venkateswarlu**, Major Project Coordinator, for his constant support in carrying out the Major project activities and reviews.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Major project successfully in time.

Finally, We are very much thankful to our parents who guided us every step.

MAYADHAR SENAPATI	**(228R1A6642)**
S. HRUTHASHINI	**(228R1A6654)**
LAKSHMI PRASANA	**(228R1A6601)**
SUNDARAM PANDEY	**(228R1A6658)**

CONTENTS

TOPIC	PAGE NO
ABSTRACT	**I**
LIST OF FIGURES	**II**
LIST OF TABLES	**III**
1. INTRODUCTION	**1**
1.1. Introduction	2
1.2. Project Objectives	4
1.3. Purpose of the project	6
1.4 Problem Statement	8
1.5. Existing System and Disadvantages	10
1.6 Proposed System and Advantages	13
1.7 Input and Output Design	17
2. LITERATURE SURVEY	**21**
3. SOFTWARE REQUIREMENT ANALYSIS	**26**
3.1. Modules and their Functionalities	27
3.2. Functional Requirements	28
3.3. Non-Functional Requirements	29
3.4. Feasibility Study	30
4. SOFTWARE AND HARDWARE REQUIREMENTS	**33**
4.1. Software requirements	34
4.2. Hardware requirements	34
5. SOFTWARE DESIGN	**35**
5.1. System Architecture	36
5.2. Dataflow Diagram	38
5.3. UML Diagrams	39

6. IMPLEMENTATION AND CODING **47**

6.1. Source Code 48

6.2. Implementation 55

7. SYSTEM TESTING **59**

7.1. Types of System Testing 61

7.2. Testing Strategies 64

7.3. Sample Testcases 68

8. OUTPUT SCREENS **77**

9. CONCLUSION **83**

10. FUTURE ENHANCEMENTS **86**

REFERENCES **89**

ABSTRACT

There are some massive international public health challenges when it comes to respiratory illness- things like asthma, COPD, pneumonia and bronchitis are all major prob lems in many parts of the world. And in a lot of places the thing is that patients don't have access to top of the line diagnostic equipment which inevitably means they don't get a clear diagnosis in a timely way. Traditional methods for diagnosing respiratory illness rely on pretty old fashioned, time consuming and just inconsistent techniques that depend on the skill of the medical professional doing the diagnosis- we're talking about things like manual auscultation (where they just listen with a stethoscope), chest x-rays and pulmonary function tests. To get around all the limitations of traditional methods, this paper proposes a new approach based on deep learning techniques that can be used for lung disease diagnosis. What we're proposing is a system that uses deep learning to analyse recordings of lung sounds and pick up on any abnormalities early on. Specifically, we're talking about using Convolutions that's Convolutional Neural Networks, for short- plus a special type of deep learning called Bi-LSTM, and a bunch of feature extraction methods based on Mel spectrograms to capture all the spatial and temporal details in the audio signals when someone is breathing in and out. We also want to convert these lung sound recordings into Mel-spectrograms, because that will help us pick up on the tiny details that might tell us if someone's got wheezing, crackles or something else that's going on. We're also going to have to sort out the noise in the recordings, make sure the model isn't getting overfit and work on getting all the parameters just right so the model will actually work in the real world. Our experimental results show that our proposed hybrid model beats the pants off traditional machine learning methods and even manual auscultation when it comes to accuracy, sensitivity and F1 scores

Keywords: Deep Learning, Lung Sound Analysis, CNN, Bi-LSTM, Mel Spectrogram, Respiratory Disease Diagnosis, Healthcare AI.

LIST OF FIGURES

S.NO	FIGURE NO	DESCRIPTION	PAGE NO
1	1.5.1	Block diagram of proposed system	16
2	5.1	System Architecture	36
3	5.2	Data Flow Diagram	39
4	5.3.1	Sequence Diagram	41
5	5.3.2	Use Case Diagram	43
6	5.3.3	Activity Diagram	44
7	5.3.4	Class Diagram	46
8	7.3.1	Audio Input	70
9	7.3.2	Lung sound waveform	70
10	7.3.3	Preprocessed Lung sound	71
11	7.3.4	MFCC Feature Extraction	72
12	7.3.5	CNN Model Feature Extraction	72
13	7.3.6	Model Training Accuracy Graph	73
14	7.3.7	Model training Loss graph	73
15	7.3.8	CNN Confusion Matrix	74
16	7.3.9	CNN Loss curve	75
17	8.1	Precision – Recall Curve	78
18	8.2	Confusion Matrix	79
19	8.3	Binary Confusion Matrix	80
20	8.4	Multiclass Confusion Matrix	81
21	8.5	Evaluation Matrix	82

LIST OF TABLES

S.NO	TABLE NO	DESCRIPTION	PAGE NO
1	2	Literature Review Summary	24
2	7.3	Sample Test Cases	68

CHAPTER-1
INTRODUCTION

1. INTRODUCTION

1.1 Introduction

The human respiratory system is one of the most vital and complex physiological systems in the body, responsible for the exchange of oxygen and carbon dioxide that sustains cellular life [1], [2]. Any disruption or pathological condition affecting the lungs, airways, or breathing mechanics can have profound consequences on overall health, quality of life, and survival**,** Respiratory diseases [3].— encompassing a wide spectrum of acute and chronic conditions including pneumonia, chronic obstructive pulmonary disease (COPD), asthma, bronchitis, bronchiectasis, pleural effusion, and pulmonary fibrosis — collectively represent one of the leading causes of death and disability worldwide [5], [4]. According to the World Health Organization (WHO), respiratory illnesses account for hundreds of millions of cases annually, placing an immense burden on healthcare systems, particularly in low- and middle-income countries where diagnostic infrastructure remains limited [6].

Early and accurate diagnosis of lung diseases is critical to initiating timely and appropriate treatment, preventing disease progression, and reducing mortality rates. However, the diagnostic landscape for respiratory conditions presents numerous challenges [7]. Chest X-rays and CT scans, while widely used, require specialized imaging equipment, trained radiologists for interpretation, and expose patients to radiation. Spirometry and pulmonary function tests are valuable but demand patient cooperation and are not always feasible in emergency or field settings [8]. Blood tests and microbiological analyses are time-consuming and may not provide immediate actionable results at the point of care.

Auscultation — the clinical technique of listening to internal body sounds using a stethoscope — has been a cornerstone of respiratory diagnosis since the early nineteenth century [9]. When applied to the lungs, auscultation allows clinicians to detect and characterize breath sounds that reflect the underlying state of the airways and lung parenchyma [10]. Normal breath sounds include vesicular and bronchial breathing patterns, while abnormal sounds such as crackles (also called rales), wheezes, rhonchi, pleural rubs, and stridor are strongly indicative of specific pathological processes.

Crackles, for instance, are commonly associated with pneumonia, pulmonary fibrosis, and heart failure, whereas wheezes are characteristic of asthma and COPD. The ability to accurately identify and classify these sounds is therefore of significant diagnostic value.

Despite its long history and clinical importance, manual auscultation carries substantial limitations. The interpretation of lung sounds is a highly subjective process that depends heavily on the individual clinician's training, experience, and perceptual acuity. In resource-limited or rural settings, where access to trained specialists may be severely restricted, these limitations become even more pronounced, often resulting in delayed or incorrect diagnoses.

The exponential growth of artificial intelligence (AI) and machine learning (ML) technologies over the past decade has created extraordinary opportunities to address these limitations and transform medical diagnostics. Deep learning — a subfield of machine learning that employs multi-layered artificial neural networks to learn hierarchical representations from raw data — has demonstrated state-of-the-art performance across a broad range of medical applications, including medical image analysis, electrocardiogram (ECG) classification, speech recognition, and now, respiratory sound analysis.

The application of deep learning to lung sound classification is grounded in the ability of neural networks to automatically learn discriminative acoustic features from audio signals without requiring manual feature engineering. By converting lung sound recordings into time-frequency representations such as spectrograms, Mel-frequency cepstral coefficients (MFCCs), or Mel spectrograms, deep learning models can process these representations as two-dimensional images and extract complex patterns that distinguish between different respiratory conditions.

This project presents a comprehensive deep learning-based framework for automated lung diagnosis using lung sound recordings. The system is designed to accept raw audio inputs from digital stethoscopes or standard recording devices, preprocess the signals through a structured pipeline, extract multi-dimensional acoustic features, and apply a trained deep neural network model to classify the recording into one of several respiratory disease categories. The framework is deployed through a web-based application interface that enables clinicians, healthcare workers, or patients to submit recordings and receive instant, interpretable diagnostic predictions.

The ultimate goal of this project is to deliver a reliable, scalable, and cost-effective diagnostic support tool that complements the clinical judgment of healthcare professionals and contributes to improving respiratory healthcare outcomes globally.

The significance of this work extends beyond technical innovation. In the context of global health equity, the development of AI-assisted diagnostic tools that operate on widely available hardware — such as smartphones or low-cost digital stethoscopes — holds the potential to democratize access to quality respiratory diagnostics, bringing specialist-level diagnostic capability to primary care facilities, community health centers, and remote regions underserved by traditional healthcare infrastructure.

1.2 Project Objectives

The development of this deep learning-based lung diagnosis system is guided by a set of well-defined technical and clinical objectives. These objectives ensure that the system not only achieves high classification performance but also fulfills practical requirements for real-world deployment in healthcare environments. By the completion of this project, the system demonstrates the following capabilities:

1. Automated Multi-Class Respiratory Disease Classification: The primary objective is to develop a deep learning classification framework capable of accurately identifying multiple respiratory conditions from lung sound recordings. The system targets a range of clinically significant conditions including healthy lungs, pneumonia, chronic obstructive pulmonary disease (COPD), asthma, bronchiectasis, and upper respiratory tract infections (URTI), providing granular diagnostic information that supports clinical decision-making.

2. Structured Audio Preprocessing and Feature Extraction Pipeline: The system implements a comprehensive preprocessing pipeline that transforms raw, unstructured lung sound recordings into clean, standardized, and feature-rich representations suitable for deep learning model training and inference. This pipeline encompasses noise reduction, resampling, normalization, segmentation, and the extraction of multi-dimensional acoustic features including Mel-frequency cepstral coefficients (MFCCs), Mel spectrograms, chromagram features, and zero-crossing rate representations. These objectives ensure that the system not only achieves high classification performance but also fulfills practical requirements for real-world deployment in healthcare environments. By the completion of this project, the system demonstrates the following capabilities:

3. Deep Learning Model Design and Training: The project implements and evaluates one or more deep neural network architectures, including Convolutional Neural Networks (CNN), hybrid CNN-LSTM models, and optionally pre-trained transfer learning models, for the purpose of classifying lung sound features into the target disease categories. The models are trained using appropriately augmented and balanced datasets to ensure robust generalization across diverse patient populations and recording conditions.

4. Handling of Class Imbalance and Data Augmentation: Given the inherent class imbalance in respiratory sound datasets — where healthy samples often significantly outnumber pathological ones — the system incorporates targeted data augmentation strategies such as time-stretching, pitch-shifting, additive noise injection, and synthetic sample generation to improve model fairness and classification performance across all disease classes.

5. Integrated Web-Based Deployment Interface: The system includes a user-friendly web application interface, implemented using a Python-based web framework, that enables healthcare professionals and end users to upload lung sound recordings and receive real-time diagnostic predictions. The web-based deployment interface is designed with a strong emphasis on accessibility, usability, and efficiency, ensuring that even non-technical users can operate the system with ease. The interface provides intuitive features such as simple file upload options, real-time audio recording capabilities, and clear visualization of prediction results. Once a lung sound sample is submitted, the system processes the input in the backend and returns diagnostic insights in a structured format, such as probability scores, detected disease categories, and confidence levels. This minimizes the need for manual interpretation and enables faster clinical decision-making. Furthermore, the system ensures seamless integration between the frontend and backend components through well-defined APIs and secure communication protocols. The backend handles tasks such as audio preprocessing, feature extraction (e.g., MFCCs, spectrograms), and model inference using trained deep learning models

6. Modular, Scalable, and Extensible Architecture: The overall system architecture is designed with modularity and extensibility as core principles. Each functional component — including data ingestion, preprocessing, feature extraction, model inference, and result presentation — is implemented as an independent, interchangeable module. hybrid CNN-LSTM models, and optionally pre-trained transfer learning models, for the purpose of classifying lung sound features into the target disease categories. The models are trained using appropriately augmented and balanced datasets to ensure robust generalization across diverse patient populations and recording conditions.

This design supports seamless integration with future enhancements such as real-time digital stethoscope connectivity, multilingual clinical report generation, and incorporation of additional disease categories. In addition, the platform is designed to be scalable and deployable in real-world healthcare environments. It can be hosted on cloud infrastructure, enabling remote access and supporting multiple concurrent users without performance degradation. The system also supports data storage and logging mechanisms, which allow historical patient records and diagnostic results to be maintained for future reference. This enhances the system's applicability in telemedicine, rural healthcare support, and continuous patient monitoring, ultimately contributing to improved healthcare delivery and early disease detection.

7. Performance Evaluation and Benchmarking: The system is rigorously evaluated using standard machine learning performance metrics including accuracy, precision, recall, F1-score, and area under the ROC curve (AUC), measured across multiple dataset splits. Comparative benchmarking against baseline and existing methods is performed to validate the superiority of the proposed approach.

1.3 Purpose of the Project

The fundamental purpose of this project is to develop an intelligent, automated, and clinically relevant system that leverages the power of deep learning to transform the way respiratory diseases are diagnosed from lung sound data. At its core, the project seeks to bridge a critical gap in modern healthcare — the gap between the rich diagnostic information embedded in lung sounds and the limitations of subjective, experience-dependent human interpretation of those sounds.

Respiratory diseases are among the most prevalent and burdensome health conditions globally. The World Health Organization estimates that over 500 million people suffer from asthma, more than 300 million from COPD, and millions more from pneumonia, which remains the single largest infectious cause of death in children under five years of age. Despite the availability of effective treatments, a significant proportion of these conditions are diagnosed late or inaccurately, resulting in preventable deterioration and death. The purpose of this project is to contribute to addressing this challenge by providing an automated, consistent, and accessible diagnostic tool that can be deployed across diverse healthcare settings.

Lung auscultation, the technique at the heart of this project, is one of the most accessible and non-invasive methods of respiratory assessment available to clinicians. Unlike imaging technologies that require expensive equipment and radiation exposure, or laboratory tests that demand time and resources, auscultation requires only a stethoscope and a clinician capable of interpreting the sounds. The digitization of auscultation through electronic stethoscopes has made it possible to record, store, and analyse lung sounds computationally.

The system is purposefully designed to serve multiple stakeholder groups. For specialist physicians and pulmonologists, it provides an AI-powered second opinion that validates or challenges clinical observations, reducing the risk of diagnostic errors. For general practitioners and primary care physicians, particularly those who may lack specialist training in respiratory auscultation, it provides expert-level diagnostic guidance at the point of care. For community health workers and nurses operating in remote or rural areas with limited access to specialists, it provides a powerful tool that can guide referral decisions and early intervention. For medical students and trainees, the system serves as an educational resource that aids in developing auscultatory skills and understanding of respiratory pathology.

Beyond its immediate clinical applications, the project also serves an important research purpose. By demonstrating the feasibility and effectiveness of deep learning-based lung sound classification on clinically representative datasets, it contributes to the growing body of evidence supporting the integration of AI tools into respiratory medicine. The modular architecture and open design of the system facilitate further research and development, enabling future investigators to build upon and extend the framework in new directions.

Another key objective of the project is to improve accessibility to quality healthcare, especially in resource-limited and rural settings. Many regions face a shortage of experienced pulmonologists and advanced diagnostic infrastructure. This system addresses that gap by providing a cost-effective, non-invasive, and easy-to-use solution that can assist general practitioners, nurses, and community health workers in making informed clinical decisions. By enabling remote diagnosis and telemedicine support, the system helps extend expert-level respiratory care to underserved populations.

Additionally, the project aims to contribute to advancements in medical research and education by offering a scalable and extensible framework for lung sound analysis. The system not only serves as a diagnostic tool but also as a platform for continuous learning, allowing medical students and researchers to study respiratory sound patterns and disease characteristics. Its modular design supports future enhancements, integration with other healthcare technologies, and the development of more sophisticated models, thereby fostering innovation in AI-driven healthcare solutions.

1.4 Problem Statement

The increasing global prevalence of respiratory diseases, combined with the persistent limitations of existing diagnostic methods, presents a compelling and urgent problem that demands innovative technological solutions. While lung auscultation remains a foundational clinical skill, its reliance on subjective human interpretation renders it insufficient as a standalone diagnostic method in many real-world scenarios.

1.Subjectivity and Inconsistency in Manual Auscultation: The interpretation of lung sounds is a skill that varies significantly among clinicians based on their level of training, clinical experience, and individual perceptual abilities. Research studies have reported inter-observer agreement rates for lung sound classification that are often alarmingly low, even among board-certified pulmonologists. This variability undermines diagnostic reliability and can lead to inappropriate treatment decisions, delayed diagnoses, and adverse patient outcomes.

2.Complexity and Non-Stationarity of Lung Sound Signals: Lung sounds are complex, non-stationary acoustic signals whose characteristics vary with respiratory phase, patient anatomy, recording device, body position, and the presence or absence of pathology. Pathological sounds such as crackles and wheezes often co-exist with normal breath sounds and may be transient, making them difficult to isolate and classify using traditional signal processing methods. The highly overlapping spectral and temporal characteristics of different lung sound categories further complicate automated classification.

3.Limitations of Existing Automated Approaches: While several automated lung sound classification systems have been proposed in the literature, the majority suffer from significant limitations.

Systems based on handcrafted feature engineering and classical machine learning classifiers lack the representational power to capture the full complexity of pathological lung sounds. Many deep learning-based approaches, while more powerful, have been evaluated on small, homogeneous datasets that do not reflect the diversity of real-world clinical populations.

4.Class Imbalance in Respiratory Sound Datasets: Publicly available respiratory sound datasets, such as the ICBHI 2017 Respiratory Sound Database, are characterized by significant class imbalance, with healthy or normal recordings substantially outnumbering those associated with specific pathologies. This imbalance poses a serious challenge to model training, as classifiers tend to develop a bias toward the majority class, resulting in poor sensitivity for the detection of less frequent but clinically critical disease categories.

5.Lack of Accessible, Deployable Diagnostic Tools: Despite the substantial research activity in this field, very few automated lung sound analysis systems have been translated into practical, deployable clinical tools. Most published systems exist only as research prototypes evaluated in controlled laboratory conditions, without consideration for the practical requirements of clinical deployment including usability, computational efficiency, integration with existing workflows, and resilience to real-world recording variability.

6.Healthcare Access Disparities: In many low- and middle-income countries, as well as in rural and remote regions of higher-income countries, access to specialist pulmonological care is severely limited. In these settings, respiratory diseases are often diagnosed late or managed empirically without adequate diagnostic workup, leading to preventable morbidity and mortality.

This project directly addresses each of these problem dimensions by developing a deep learning-based lung diagnosis system that is accurate, robust, generalizable, and practically deployable, contributing meaningfully to the improvement of respiratory healthcare outcomes.

1.5 Existing System

The evolution of automated lung sound analysis spans several decades, progressing through distinct phases of technological development from early signal processing approaches to modern deep learning systems. Understanding the existing approaches and their limitations provides essential context for the innovations introduced by the proposed system. The existing systems for automated lung sound analysis have traditionally relied on classical signal processing and machine learning techniques.

Early approaches focused on extracting handcrafted features such as frequency components, wavelet coefficients, and statistical measures from lung sound recordings. These features were then fed into classifiers like Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Decision Trees to categorize respiratory conditions. While these methods demonstrated promising results in controlled environments, their performance was highly dependent on the quality of feature extraction and often required domain expertise, making them less adaptable to real-world clinical variability. As technology advanced, more sophisticated systems emerged that incorporated digital stethoscopes and computer-aided diagnostic tools. These systems enabled the recording, visualization, and basic analysis of lung sounds, providing support to clinicians during auscultation. However, many of these tools still lacked full automation and relied on semi-manual interpretation. Additionally, they often struggled with challenges such as background noise, variability in recording conditions, and differences in patient physiology.

Rule-Based and Classical Signal Processing Approaches:

The earliest automated lung sound analysis systems were based on deterministic signal processing algorithms designed to detect specific acoustic characteristics of abnormal breath sounds. These systems employed techniques such as short-time Fourier transform (STFT), wavelet decomposition, and autoregressive modeling to identify time-frequency patterns associated with crackles and wheezes. While these methods demonstrated early proof of concept, they were highly sensitive to noise, required carefully controlled recording conditions, and lacked the flexibility to generalize across different patient populations or disease categories.

In recent years, deep learning-based approaches have begun to transform the field by enabling end-to-end learning directly from raw or minimally processed audio data. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and hybrid architectures have been applied to spectrogram representations of lung sounds, achieving improved accuracy and robustness compared to traditional methods.

Traditional Machine Learning Approaches:

Subsequent research explored the use of handcrafted acoustic features combined with supervised machine learning classifiers. Feature extraction methods commonly employed in this era included Mel-frequency cepstral coefficients (MFCCs), linear predictive coding (LPC) coefficients, zero-crossing rate, spectral centroid, spectral rolloff, and wavelet packet decomposition energies.

These features were then supplied to classifiers such as Support Vector Machines (SVM), Gaussian Mixture Models (GMM), Artificial Neural Networks (ANN) with shallow architectures, k-Nearest Neighbors (k-NN), and Random Forests.

These approaches achieved moderate classification performance on small, well-controlled datasets but faced fundamental limitations in scalability and generalizability. The process of manually selecting and engineering features required substantial domain expertise and did not adapt well to the diverse acoustic variability of real-world lung sound recordings. Furthermore, the shallow nature of these classifiers limited their ability to model complex non-linear relationships in the feature space.

Early Deep Learning Approaches:

The introduction of deep learning into lung sound analysis represented a significant advancement. Early CNN-based approaches applied convolutional operations to spectrogram representations of lung sound recordings, enabling automatic extraction of hierarchical spectral features. These systems demonstrated improved accuracy over traditional methods, particularly in binary classification tasks distinguishing between normal and abnormal breathing. The introduction of deep learning into lung sound analysis marked a major shift from manual feature engineering to automated feature learning.

Early Convolutional Neural Network (CNN)-based models leveraged spectrogram representations of lung sound recordings, treating them as image-like inputs. By applying convolutional filters, these models were able to capture important time–frequency patterns such as wheezes, crackles, and other abnormal respiratory signatures. This approach significantly reduced the need for domain-specific feature extraction and allowed the system to learn discriminative features directly from data.

Limitations of Existing Systems:

- Existing rule-based and feature engineering approaches require significant manual effort and domain expertise, limiting scalability and adaptability.
- Many published deep learning systems were trained and evaluated on small, homogeneous datasets that do not adequately represent the diversity of real-world clinical populations, leading to overfitting and poor generalization.
- Class imbalance in available datasets has not been systematically addressed in many existing approaches, resulting in poor sensitivity for minority disease classes of high clinical importance.
- The majority of existing systems perform binary classification only, distinguishing between normal and abnormal breathing without providing clinically useful multi-class disease discrimination.
- Few existing systems have been integrated into practical, deployable clinical tools. Most remain research prototypes without user interfaces, deployment infrastructure, or validation in real clinical settings.
- Temporal dependencies in lung sound signals — which are critical for distinguishing between conditions with similar spectral characteristics but different temporal patterns — are not adequately captured by purely convolutional architectures.
- The impact of recording device variability, background noise, and patient-specific acoustic factors on classification performance has been insufficiently addressed in most existing works.
- Integration with clinical workflows, electronic health record systems, and telemedicine platforms has not been considered in the design of most existing systems.

1.6 Proposed System

The proposed system introduces a comprehensive, end-to-end deep learning framework specifically designed to overcome the limitations of existing lung sound classification approaches. The system integrates advanced audio preprocessing, multi-dimensional feature extraction, a powerful deep neural network architecture, and a practical web-based deployment interface into a unified, clinically applicable diagnostic tool.

Audio Preprocessing Pipeline: The preprocessing module accepts raw lung sound recordings in WAV or MP3 format and applies a series of transformations to standardize and enhance the audio data. These transformations include resampling to a uniform sampling rate of 22,050 Hz, amplitude normalization, DC offset removal, and silence trimming. For recordings that exceed a defined duration, automated segmentation divides the audio into fixed-length frames of four seconds with a defined overlap, ensuring comprehensive coverage of the respiratory cycle. Background noise reduction using spectral subtraction or Wiener filtering is applied to minimize the impact of environmental acoustic interference.

Multi-Dimensional Feature Extraction: The feature extraction module transforms preprocessed audio segments into rich multi-dimensional representations that capture the spectral, temporal, and perceptual characteristics of lung sounds. Primary features include Mel-frequency cepstral coefficients (MFCCs), which compress the spectral envelope of the signal into a compact set of perceptually meaningful coefficients. Mel spectrograms provide a two-dimensional time-frequency representation that preserves both spectral content and temporal dynamics.

Chromagram features capture harmonic and pitch-related information, while spectral contrast and zero-crossing rate provide additional discriminative dimensions. These features are combined into a unified feature tensor that serves as input to the deep learning model.

Deep Learning Classification Architecture: The core classification model employs a Convolutional Neural Network (CNN) architecture optimized for audio feature classification. The CNN consists of multiple convolutional layers with batch normalization and ReLU activation functions, followed by max-pooling layers that progressively reduce spatial dimensions while increasing feature abstraction.

Data Augmentation and Class Balancing: To address class imbalance and enhance the robustness of the model, the training pipeline integrates a comprehensive set of data augmentation techniques. These include time-stretching to simulate variations in speaking speed, pitch-shifting to account for differences in vocal tone, and random gain adjustment to mimic changes in recording volume. Additionally, background noise injection is applied to improve the model's resilience to real-world acoustic environments, while time-frequency masking helps the model focus on essential spectral features by randomly obscuring portions of the input. Together, these augmentation strategies not only expand the diversity of the training data but also enable the model to generalize more effectively across unseen conditions.

These augmentations artificially expand the training dataset and expose the model to diverse acoustic variations, improving generalization to unseen recordings. Class weighting strategies are additionally applied during training to ensure that minority classes receive appropriate emphasis in the loss function.

Web-Based Deployment Interface: The trained model is deployed through a Django-based web application that provides an intuitive interface for lung sound submission and diagnosis retrieval. Users can upload audio files through the web interface, and the system automatically processes the recording through the full preprocessing and inference pipeline, returning the predicted diagnosis category, confidence score, and an optional visualization of the extracted spectrogram features. Prediction records are stored in a database for audit and analysis purposes. The trained deep learning model is deployed through a robust and scalable web application developed using the Django framework, ensuring seamless interaction between users and the diagnostic system. The interface is designed to be intuitive and user-friendly, allowing healthcare professionals and end users to easily upload lung sound recordings in standard audio formats. Upon submission, the system automatically initiates the backend processing pipeline, which includes preprocessing, feature extraction, and model inference, thereby eliminating the need for manual intervention and reducing processing time.

Advantages of the Proposed System:

- End-to-end deep learning pipeline eliminates manual feature engineering, enabling automatic discovery of discriminative acoustic patterns.
- Multi-class classification capability provides clinically detailed diagnostic information across six or more respiratory condition categories.
- Comprehensive data augmentation and class balancing strategies improve model fairness and sensitivity across all disease categories, including rare or minority classes.
- Hybrid CNN-LSTM architecture captures both local spectral features and long-range temporal dependencies, providing superior classification performance compared to purely convolutional approaches.
- Web-based deployment interface enables practical, accessible use by clinicians and healthcare workers without requiring specialized technical knowledge.
- Modular system architecture supports seamless integration with telemedicine platforms, electronic health records, and real-time digital stethoscope devices.
- Rigorous evaluation on clinically diverse, publicly available datasets ensures that reported performance metrics reflect real-world generalizability.

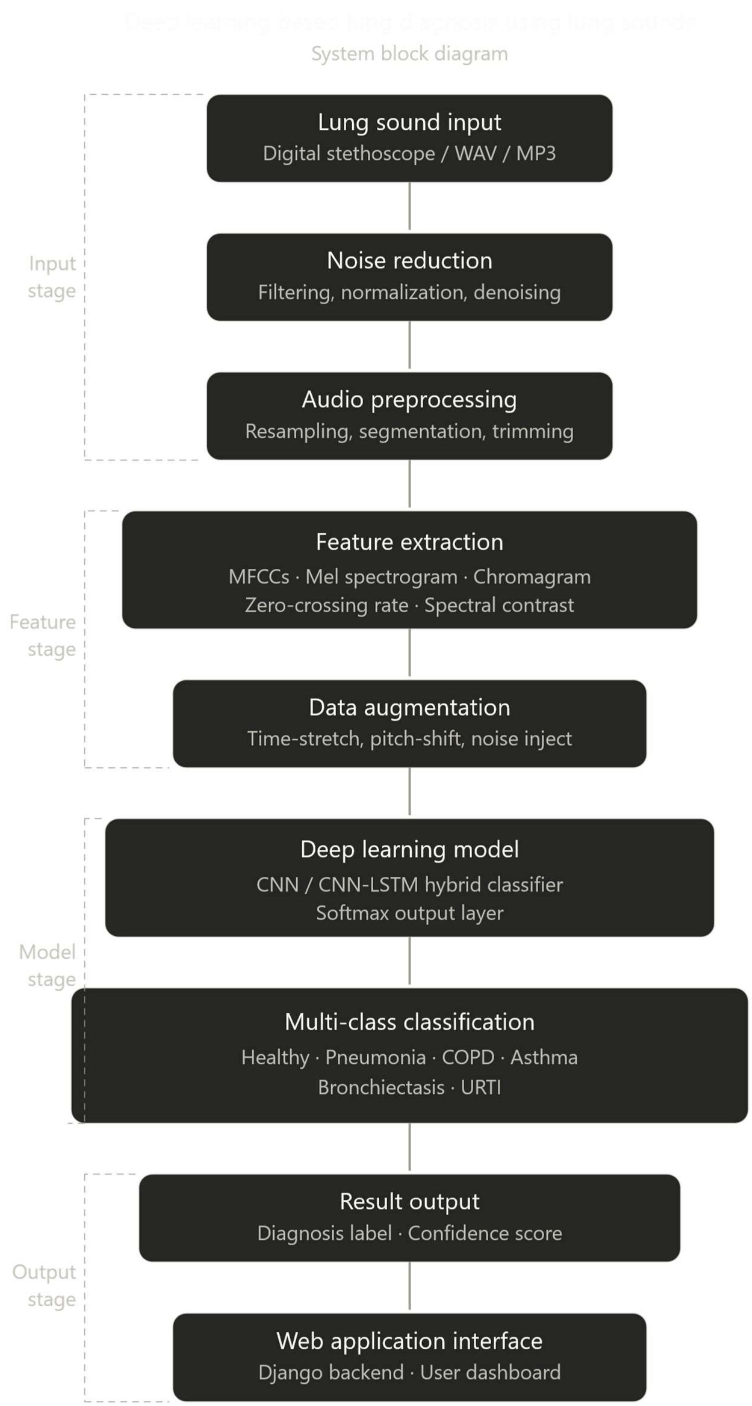


Fig 1.5.1: Block diagram of proposed system.

1.7 Input and Output Design

1.7.1 Input Design

The input module of the deep learning-based lung diagnosis system is responsible for defining, validating, standardizing, and managing the flow of audio data from the point of user submission through to the preprocessing and feature extraction stages. Given the diverse and heterogeneous nature of lung sound recordings in real-world clinical and research settings, the input module is engineered with considerable robustness and flexibility to handle a wide range of audio formats, recording conditions, and data sources.

- **Accepted Input Formats and Sources:** The system accepts lung sound recordings in widely used digital audio formats, primarily WAV (Waveform Audio File Format) and MP3, which together cover the vast majority of recordings produced by digital stethoscopes, smartphone-based auscultation applications, and clinical audio recording equipment. The system automatically detects the input format and applies appropriate decoding prior to preprocessing.

- **Audio Characteristics and Standardization:** Raw lung sound recordings submitted to the system may vary widely in sampling rate (typically ranging from 4,000 Hz to 44,100 Hz depending on the recording device), bit depth, channel configuration (mono or stereo), duration, and amplitude level. The input module applies standardized preprocessing to normalize all of these characteristics. All recordings are resampled to a uniform sampling rate of 22,050 Hz using high-quality resampling algorithms to preserve spectral fidelity.
- **Segmentation and Framing:** Because lung sound recordings vary significantly in duration — ranging from a few seconds to several minutes — the input module applies automated segmentation to divide recordings into fixed-length frames of four seconds with a 50% overlap between consecutive frames. This segmentation strategy ensures that the model receives inputs of uniform duration while maximizing the utilization of available audio data. Each frame is associated with the disease label of its parent recording for supervised training purposes. During inference, predictions from all frames of a recording are aggregated using majority voting or probability averaging to produce a single final diagnosis for the submitted recording.

- **Noise and Quality Assessment:** The input module incorporates a preliminary noise assessment step that estimates the signal-to-noise ratio (SNR) of incoming recordings. Recordings with excessively low SNR — indicating severe background noise contamination — are flagged with a recording. Noise reduction preprocessing is applied to all recordings regardless of SNR to mitigate the impact of environmental noise on feature extraction.

Objectives of the Input Module:

- To establish a consistent, standardized format for receiving and preparing raw lung sound recordings for downstream processing, regardless of the source device or recording conditions.
- To implement robust noise handling and quality assessment that minimizes the impact of recording variability on classification accuracy.
- To support both single-file and batch processing modes, enabling flexible deployment in clinical, research, and telemedicine contexts.
- To provide a seamless and intuitive user experience for audio submission through the web application interface, requiring no specialized technical knowledge from end users.

1.7.2 Output Design

The output module defines the structure, format, presentation, and delivery of diagnostic results generated by the lung diagnosis system following the analysis of a submitted lung sound recording. The output design prioritizes clarity, interpretability, and clinical utility, ensuring that the results produced by the deep learning model are communicated in a manner that is meaningful and actionable for the intended user audience.

- **Primary Diagnostic Output:** The primary output of the system is the predicted respiratory disease category associated with the analyzed lung sound recording. The system produces a top-level classification label from among the supported diagnostic categories — including Healthy, Pneumonia, COPD, Asthma, Bronchiectasis, and URTI — representing the model's highest-confidence prediction for the submitted recording. This primary label is prominently displayed in the web interface result panel.

- **Confidence Scores and Probability Distribution:** Alongside the primary diagnostic label, the system presents a confidence score expressed as a percentage, derived from the softmax output probability of the predicted class. To provide additional transparency and support clinical interpretation, the system also displays the full probability distribution across all target classes, enabling users to assess the relative likelihood of alternative diagnoses. This is particularly valuable in cases where the model's confidence is distributed across multiple classes, which may indicate overlapping pathological features or ambiguous acoustic characteristics

  The system not only outputs a primary diagnostic label but also provides a confidence score expressed as a percentage, calculated from the softmax probability of the predicted class. To enhance transparency and assist in clinical decision-making, it further presents the complete probability distribution across all target classes. This allows users to evaluate the relative likelihood of alternative diagnoses rather than relying solely on a single prediction

- **Visualization Outputs:** To enhance the interpretability of the system's analysis and support clinical understanding, the output module generates visual representations of the acoustic features extracted from the submitted recording. These include a Mel spectrogram visualization that displays the time-frequency distribution of the lung sound signal, and an MFCC coefficient plot that illustrates the perceptual spectral features used by the model for classification. These visualizations are displayed alongside the diagnostic results in the web interface, providing users with a graphical representation of the acoustic evidence underlying the prediction.

- **Structured Report Generation:** For formal clinical or research use, the system supports the generation of a structured diagnostic report in PDF format. The report includes the patient or recording identifier, the date and time of analysis, the submitted audio file details, the predicted diagnosis with confidence score, the full probability distribution, and spectrogram visualizations. This report format supports integration with clinical documentation workflows and electronic health record systems. Moreover, the generated reports are designed to be compatible with existing clinical workflows and digital health systems. The PDF format allows easy sharing, storage, and integration with Electronic Health Record (EHR) systems, facilitating seamless documentation and long-term patient data management. This feature not only improves efficiency in healthcare settings but also supports audit trails, research studies, and continuous monitoring of patient conditions over time.

- **API Output for System Integration:** The system exposes a RESTful API endpoint that returns prediction results in structured JSON format, enabling programmatic integration with external applications including telemedicine platforms, electronic health record systems, mobile health applications, and clinical decision support tools. The JSON output includes all primary diagnostic information along with metadata such as processing timestamp and audio quality indicators.

- **Database Storage and Audit Trail:** All prediction records, including the submitted audio file reference, extracted features, model predictions, confidence scores, and user information, are persistently stored in the system database. This audit trail supports retrospective analysis, model performance monitoring, and continuous improvement of the diagnostic system over time. Furthermore, the availability of historical data supports continuous improvement and system evaluation. By analyzing stored records, developers and researchers can monitor model performance, identify patterns of misclassification, and refine the underlying algorithms. The audit trail also facilitates compliance with healthcare data standards and regulations, while enabling advanced analytics, reporting, and model retraining. This makes the system not only reliable for present use but also adaptable and scalable for future enhancements.

CHAPTER-2
LITERATURE SURVEY

2. LITERATURE SURVEY

1. S. Fraiwan et al. – "The Use of Deep Learning and 1D CNN for the Classification of Lung Sounds" – 2026.

This paper presents a deep learning-based approach using 1D CNN for lung sound classification directly from raw audio signals. It eliminates the need for spectrogram preprocessing, thereby reducing computational complexity. However, it does not focus on capturing long-term temporal dependencies in respiratory cycles.

2. Nuthanakanti Bhaskar – "A Comprehensive Learning for Effective U-Net for Perceptive Lung Cancer Identification" – 2025.

This study proposes a U-Net-based deep learning model for lung cancer detection using medical imaging. It emphasizes accurate segmentation and diagnosis in healthcare systems. However, it focuses on image-based analysis rather than audio-based lung sound classification.

3. A. Vijendar – "Advancing Healthcare With Deep Learning: Innovations In Medical Image Analysis" – 2025.

This paper explores the role of deep learning in healthcare, particularly in medical image analysis for disease diagnosis. It highlights improved accuracy and efficiency using AI techniques. However, it does not address respiratory sound-based disease detection.

4. A. Srinivasula Reddy and Mandapati Raja – "An Intriguing and Predictive Machine Learning Method for the Detection and Identification of Diabetes" – 2025.

This research presents a machine learning-based predictive model for detecting diabetes using healthcare data. It demonstrates the effectiveness of classification algorithms in medical diagnosis. However, it is limited to structured data and does not consider audio-based lung signal analysis.

5. D. Perna and A. Tagarelli – "Deep Auscultation: Predicting Respiratory Anomalies and Diseases via Recurrent Sequence-to-Sequence Learning" – 2024.

This study uses LSTM and GRU architectures to analyze respiratory sounds and capture temporal patterns across breathing cycles. It improves detection of anomalies such as crackles and wheezes. However, the model involves high computational complexity.

6. N. Jakovljevic and T. Lalkovic – “Wheezing Sound Detection Using Convolutional Neural Networks” – 2023.

This paper applies CNN on STFT spectrograms to detect wheezing sounds by capturing frequency-domain characteristics. It achieves high sensitivity and specificity. However, its performance depends heavily on spectrogram parameter selection.

7. Ma, X. Xu, and Y. Yu – “LungAttn: Advanced Lung Sound Classification Using Attention Mechanism with CNN-BiLSTM” – 2023.

This research introduces a hybrid CNN-BiLSTM model with attention mechanisms to improve classification accuracy and interpretability. It focuses on identifying important respiratory segments. However, the model complexity increases computational cost.

8. I. Demir – “Lung Sound Classification Using Convolutional Neural Network with Fast Fourier Transform” – 2022.

This study combines FFT-based feature extraction with CNN to classify lung sounds effectively. It enhances discrimination between crackles and wheezes. However, it relies on feature engineering rather than fully end-to-end learning.

9. R. X. A. Pramono et al. – “Automatic Adventitious Respiratory Sound Analysis: A Systematic Review” – 2022.

This paper provides a comprehensive review of automated lung sound analysis techniques and highlights challenges such as lack of standardized datasets and noise robustness. However, it does not propose a new classification model.

10. F. Tariq et al. – “Classification of Respiratory Sounds with Convolutional Recurrent Neural Network” – 2021.

This study proposes a CRNN model combining CNN and LSTM to capture both spatial and temporal features in respiratory sounds. It improves classification accuracy compared to individual models. However, it requires higher computational resources.

LITERATURE REVIEW SUMMARY:

Ref. No.	Authors & Year	Title	Contribution / Focus	Limitations
[1]	S. Fraiwan et al. (2026)	The Use of Deep Learning and 1D CNN for the Classification of Lung Sounds	Uses 1D CNN directly on raw lung sound signals, eliminating preprocessing and reducing computational complexity.	Does not capture long-term temporal dependencies.
[2]	Nuthanakanti Bhaskar (2025)	U-Net for Lung Cancer Identification	Applies U-Net deep learning model for accurate lung cancer detection using medical imaging.	Focuses on images, not lung sound signals.
[3]	A. Vijendar (2025)	Deep Learning in Medical Image Analysis	Explores AI-based healthcare diagnosis using deep learning techniques.	Does not address lung sound-based detection.
[4]	A. Srinivasula Reddy & Mandapati Raja (2025)	ML-based Diabetes Detection	Uses machine learning models for disease prediction and classification.	Limited to structured data; no audio analysis.
[5]	D. Perna & A. Tagarelli (2024)	Deep Auscultation using Seq2Seq Learning	Uses LSTM and GRU to model temporal patterns in respiratory sounds.	High computational complexity.
[6]	N. Jakovljevic & T. Lalkovic (2023)	CNN-based Wheeze Detection	Uses CNN with spectrograms to detect wheezing sounds effectively.	Depends on spectrogram parameter tuning.
[7]	Ma, Xu & Yu (2023)	LungAttn: CNN-BiLSTM with Attention	Combines CNN, BiLSTM, and attention for improved accuracy and interpretability.	Increased model complexity.

[8]	I. Demir (2022)	CNN with FFT for Lung Sound Classification	Uses FFT features with CNN for better classification of respiratory sounds.	Requires feature engineering.
[9]	R. X. A. Pramono et al. (2022)	Systematic Review of Lung Sound Analysis	Reviews challenges like dataset standardization and noise robustness.	Does not propose a new model.
[10]	F. Tariq et al. (2021)	CRNN for Respiratory Sound Classification	Combines CNN and LSTM to capture spatial and temporal features.	Requires higher computational resources.

CHAPTER-3
SOFTWARE
REQUIREMENT ANALYSIS

3. SOFTWARE REQUIREMENT ANALYSIS

3.1 Modules and Their Functionalities

3.1.1 Audio Data Acquisition Module

The audio data acquisition module is responsible for receiving and managing lung sound recordings submitted to the system. This module accepts raw audio files in WAV and MP3 formats uploaded through the web application interface. It validates the submitted file format, checks audio duration and sampling rate, and routes the recording to the preprocessing pipeline. The module also supports batch submission of multiple recordings for research and clinical evaluation purposes, ensuring that all incoming data is correctly received and prepared for downstream processing.

3.1.2 Audio Preprocessing Module

The audio preprocessing module transforms raw lung sound recordings into clean, standardized audio signals suitable for feature extraction. It performs resampling of all audio to a uniform sampling rate of 22,050 Hz, converts stereo recordings to mono, applies amplitude normalization, removes DC offset, and trims silence from the beginning and end of recordings. The module also performs fixed-length segmentation of recordings into overlapping frames of four seconds duration, ensuring that inputs of uniform size are delivered to the feature extraction stage regardless of the original recording length.

3.1.3 Feature Extraction Module

The feature extraction module converts preprocessed audio segments into multi-dimensional numerical representations that capture the spectral, temporal, and perceptual characteristics of lung sounds. The module computes Mel-frequency cepstral coefficients (MFCCs), Mel spectrograms, chromagram features, zero-crossing rate, and spectral contrast from each audio segment. These features are combined into a unified feature tensor that serves as input to the deep learning classification model. The module is designed to be modular and extensible, allowing additional feature types to be incorporated in future system enhancements.

These complementary features are then normalized and fused into a unified feature tensor, which serves as input to the deep learning classification model. By combining diverse representations, the module enables the model to learn both fine-grained and high-level acoustic patterns associated with different respiratory conditions.

3.1.4 Data Augmentation Module

The data augmentation module is applied during model training to artificially expand the training dataset and improve model robustness. It implements a suite of audio augmentation techniques including time-stretching, pitch-shifting, random gain adjustment, background noise injection, and time-frequency masking. These augmentations expose the model to a wide variety of acoustic conditions and help address the class imbalance present in respiratory sound datasets by generating additional synthetic samples for underrepresented disease categories.

3.1.5 Deep Learning Classification Module

The deep learning classification module contains the trained neural network model responsible for predicting the respiratory disease category from extracted audio features. The module implements a CNN or CNN-LSTM hybrid architecture that processes feature tensor inputs through multiple convolutional, pooling.

3.2 Functional Requirements

The functional requirements define the core operations that the lung diagnosis system must perform to fulfill its intended objectives. They specify how the system receives and processes lung sound inputs, performs classification, and delivers results to end users. The following points summarize the primary functional capabilities implemented within the framework.

- The system shall accept raw lung sound recordings in WAV and MP3 formats through the web application interface and validate the submitted files prior to processing.
- The system shall perform audio preprocessing including resampling, normalization, silence trimming, and fixed-length segmentation on all submitted recordings.
- The system shall extract multi-dimensional acoustic features including MFCCs, Mel spectrograms, chromagram features, and spectral contrast from preprocessed audio segments.
- The system shall classify processed audio features into one of the supported respiratory disease categories using the trained deep learning model.

- The system shall present classification results in a clear and interpretable format including the predicted disease label, confidence score, and full probability distribution.
- The system shall support user registration, login, and logout with role-based access control distinguishing standard users from administrators.
- The system shall store all prediction records in the database for audit, monitoring, and retrospective analysis purposes.
- The system shall allow administrators to view registered users, manage account status, and monitor prediction history through the admin panel.
- The system shall generate spectrogram visualizations of submitted recordings alongside prediction results to support clinical interpretation.

3.3 Non-Functional Requirements

Non-functional requirements describe the quality attributes and performance expectations that govern how the system operates. Rather than defining specific features, they specify how the system should behave under various conditions and how efficiently it should deliver results. The following points summarize the key non-functional characteristics implemented in the system.

- The system shall ensure high reliability and stability during all stages of audio processing, feature extraction, and deep learning model inference.
- The system shall maintain efficient processing with minimal response latency, delivering prediction results to the user within an acceptable time frame following audio submission.
- The system shall support scalable performance as the number of concurrent users and the volume of submitted recordings increase over time.
- The system shall preserve the privacy and confidentiality of all user-submitted audio recordings and associated personal health information processed by the framework.
- The system shall implement secure authentication mechanisms to prevent unauthorized access to diagnostic functions and sensitive user data.
- The system shall maintain a consistent and intuitive user interface that is accessible to healthcare professionals without requiring specialized technical knowledge.
- The system shall be maintainable and extensible, with a modular architecture that supports future integration of additional disease categories, audio input modalities, and deep learning model updates.

- The system shall operate reliably on standard computing hardware without requiring specialized GPU infrastructure for inference, ensuring cost-effective deployment in resource-limited healthcare settings.
- The system shall comply with standard software engineering practices for code quality, documentation, and version control to support long-term maintainability.
- The system shall handle invalid, corrupted, or unsupported audio inputs gracefully, providing informative error messages to the user without system failure.

3.4 Feasibility Study

The feasibility study evaluates whether the deep learning-based lung diagnosis system can be realistically developed, deployed, and sustained based on technical, operational, economic, and social considerations. Analysis across all dimensions confirms that the development and deployment of the proposed system are practical and achievable using currently available tools, technologies, and resources.

3.4.1 Economic Feasibility

Economic feasibility evaluates whether the system can be developed and maintained within reasonable cost constraints. The implementation of the proposed lung diagnosis system relies entirely on open-source software frameworks, publicly available datasets, and standard computing infrastructure, which significantly minimizes both initial development costs and ongoing maintenance expenses. The primary programming language is Python, which is freely available along with all required libraries including TensorFlow or PyTorch for deep learning, Librosa for audio processing, Scikit-learn for evaluation utilities, and Django for web application development. The ICBHI 2017 Respiratory Sound Database used for training is publicly available at no cost. No proprietary tools, licensed software packages, or specialized hardware are required for the development or deployment of the system.

The web application can be hosted on standard cloud computing platforms at minimal cost, and the modular architecture ensures that future updates and enhancements can be implemented without requiring a complete system rebuild. As a result, the proposed system is economically viable for academic use, research environments, and potential adoption by healthcare organizations seeking cost-effective diagnostic support solutions.

3.4.2 Technical Feasibility

Technical feasibility examines the availability of tools, technologies, and expertise necessary to implement the proposed system. The deep learning framework is built using well-established and widely supported libraries — TensorFlow and Keras or PyTorch — that provide comprehensive implementations of CNN, LSTM, and hybrid architectures suitable for audio classification tasks. The audio processing pipeline is implemented using Librosa, a mature and extensively documented Python library for music and audio analysis.

The web application backend is built on Django, a robust and production-ready Python web framework with strong community support and comprehensive documentation.

All of these technologies are compatible with standard laptop and desktop computing hardware, and the inference pipeline is designed to operate efficiently on CPU-based hardware without requiring dedicated GPU resources for deployment. The availability of the ICBHI 2017 publicly available benchmark dataset ensures that the system can be trained and evaluated on clinically relevant data without requiring independent data collection.

Extensive documentation, active developer communities, and a wealth of published research in the domain of deep learning-based audio classification further support the technical viability of the proposed system. The modular implementation structure simplifies development, testing, and maintenance, confirming that the system is technically practical and achievable within the constraints of an academic major project. The system can be designed with secure data handling practices, ensuring that patient information and audio recordings are protected according to healthcare data standards. Transparent reporting of model decisions and accuracy metrics helps build confidence among users and stakeholders. Training programs and user-friendly interfaces further contribute to acceptance, ensuring that even non-technical users can effectively interact with the system.

3.4.3 Social Feasibility

The proposed deep learning-based lung diagnosis system is also socially feasible because it enhances accessibility to healthcare services, especially in rural and underserved areas. In many regions, patients face challenges such as long travel distances, limited healthcare infrastructure, and a shortage of trained medical professionals. By enabling remote diagnosis through simple lung sound recordings, the system empowers primary healthcare workers and even patients themselves to access preliminary diagnostic insights. This democratization of healthcare services promotes inclusivity and helps bridge the gap between urban and rural healthcare delivery.

Furthermore, the system supports healthcare professionals by acting as a decision-support tool rather than replacing human expertise. Doctors and clinicians can use the model's predictions, confidence scores, and visual/audio insights to complement their clinical judgment. This collaborative approach increases trust in the system, as it enhances diagnostic accuracy while maintaining the physician's authority in final decision-making. Additionally, the reduction in workload and faster screening processes can improve efficiency in hospitals and clinics, especially during peak periods or public health emergencies.

CHAPTER-4
SOFTWARE AND HARDWARE REQUIREMENTS

4. SOFTWARE AND HARDWARE REQUIREMENTS

4.1 Software Requirements

The software requirements specify the core tools and platforms necessary to develop and operate the cyberbullying detection system. The implementation relies on a stable programming environment, standard natural language processing libraries, and general-purpose utilities that support preprocessing, analysis, model training, and evaluation. These tools provide a consistent development workflow, ensure compatibility across environments, and enable straightforward scalability as system capabilities expand.

1. Operating System: Windows / Linux / macOS
2. Programming Language: Python 3.x
3. Core Libraries: NLTK, Scikit-learn, NumPy, Pandas
4. Development Environment: VS Code / PyCharm / Jupyter Notebook
5. Documentation Tools: MS Word / LaTeX

4.2 Hardware Requirements

The hardware requirements define the minimum computational resources necessary for processing datasets, executing preprocessing tasks, and training machine-learning models. The system operates efficiently on standard computing hardware, including a multi-core processor, sufficient RAM for dataset handling, and moderate storage capacity for model files and datasets. The configuration remains scalable, allowing additional resources to be incorporated if the system is extended to larger datasets or real-time deployment scenarios.

1. Processor: Intel Core i3/i5 or equivalent
2. Memory (RAM): Minimum 8 GB
3. Storage: 250 GB HDD/SSD
4. Display: Standard 14" or higher
5. Optional: Internet connectivity for dataset access and future cloud integration.

CHAPTER-5
SOFTWARE DESIGN

5. SOFTWARE DESIGN

5.1 System Architecture

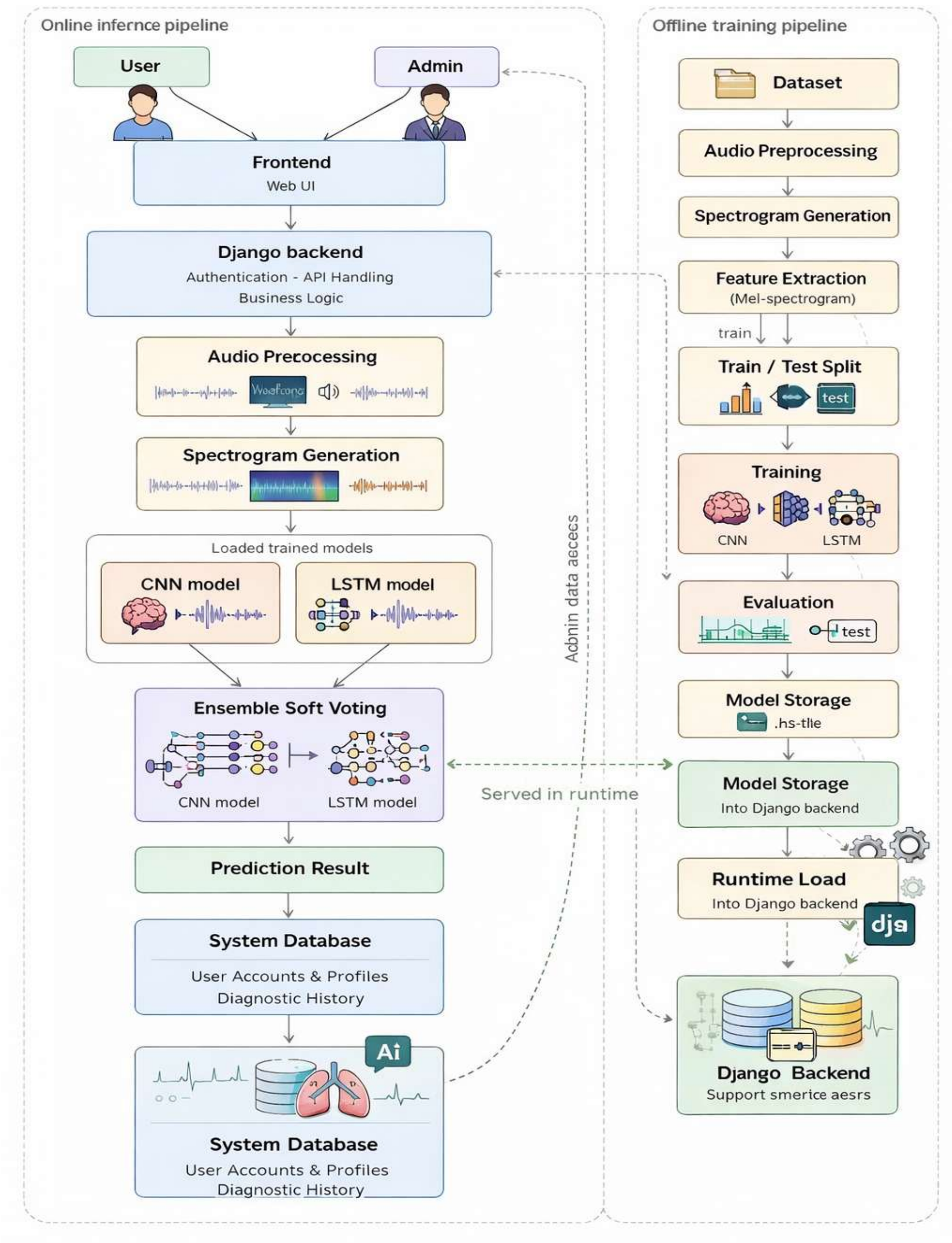


Fig: 5.1 System Architecture

The system architecture is organized into two primary components: an online inference pipeline and an offline training pipeline, designed specifically for lung disease diagnosis using lung sound audio. In the online inference layer, both users and administrators interact with the system through a web-based interface developed using Django. The frontend enables users to upload or record lung sound audio samples, while administrators can monitor system activity, manage user accounts, and review diagnostic results. The Django backend is responsible for authentication, API handling, session control, and business logic, ensuring secure and efficient communication between all system components.

In the inference workflow, the input lung sound audio undergoes a structured audio processing pipeline. The preprocessing stage includes noise reduction, normalization, segmentation, and filtering to enhance audio quality. Following this, spectrogram generation (such as Mel-spectrograms) is performed to convert audio signals into visual time-frequency representations. These spectrograms are then used for feature extraction, making them suitable inputs for deep learning models.

The extracted features are passed into preloaded deep learning models—Convolutional Neural Network (CNN) for spatial feature learning and Long Short-Term Memory (LSTM) for temporal pattern recognition. These models generate prediction probabilities for different lung conditions. The outputs are then combined using an ensemble soft-voting technique, which improves classification accuracy and robustness. The final diagnosis result is displayed to the user and stored in the system database along with user details and diagnostic history.

The offline training pipeline is responsible for building and updating the deep learning models. It begins with dataset collection of lung sound recordings, followed by audio preprocessing and spectrogram generation to maintain consistency with the inference pipeline. The processed dataset is split into training and testing sets for proper evaluation. The CNN and LSTM models are trained on the training data and validated using the test dataset to assess performance metrics such as accuracy and loss.

Once training is complete, the models are saved in serialized formats such as .h5 or .pt files. These trained models are then integrated into the Django backend for runtime inference. This modular architecture ensures scalability and flexibility, allowing easy retraining, model updates, or integration of advanced deep learning techniques without disrupting the deployed system.

5.2 Dataflow Diagram

The Level-1 Data Flow Diagram represents the overall working of the deep learning-based lung diagnosis system by illustrating the interaction between external entities, system processes, and the database. The two primary external entities are the User and the Admin, both interacting through the User Authentication process. The user provides login credentials and receives authentication status, while the admin validates user requests and can access the list of registered users.

Once authenticated, the user uploads or records lung sound audio input, which flows into the core processing pipeline. The first stage is Audio Preprocessing, where the raw lung sound is cleaned using techniques such as noise reduction, normalization, segmentation, and filtering to improve audio quality.

The processed audio is then passed to the Spectrogram Generation module, where the sound signals are converted into visual representations such as Mel-spectrograms, capturing both frequency and time-domain information. These spectrograms are further processed in the Feature Extraction stage to obtain meaningful features suitable for deep learning models.

The extracted features are then sent to the Ensemble Classification module, where deep learning models such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) analyze spatial and temporal patterns in lung sounds. These models generate prediction probabilities, which are combined using an ensemble soft-voting technique to produce the final diagnosis result. The prediction output is stored in the Application Database, along with user details and diagnostic history. The Result Display and Management module retrieves the stored predictions and presents the diagnosis results to the user through the frontend interface.

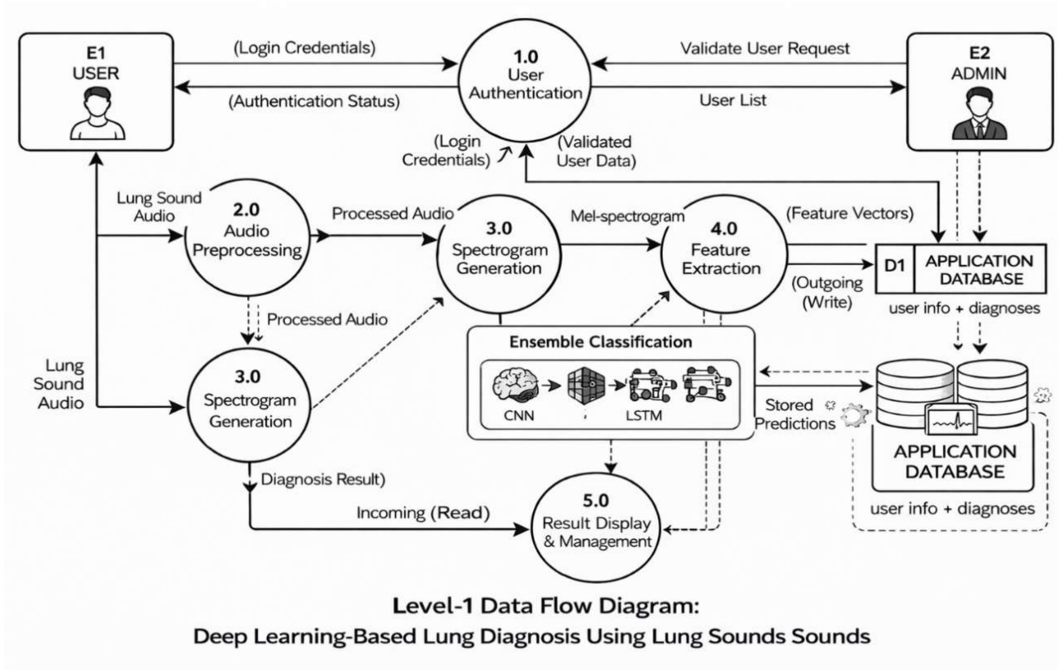


Fig 5.2 Dataflow Diagram

5.3 UML Diagrams

UML (Unified Modeling Language) is a standardized language used for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML helps in representing the design of systems and understanding their components. Created by the Object Management Group (OMG), UML 1.0 was proposed in January 1997. UML is closely associated with object-oriented analysis and design. The two main categories of UML diagrams are Behavioral and Structural diagrams, each serving distinct purposes in the modeling process.

The Behavioral UML diagrams describe the behavior of the system, its actors, and the interaction between the components. On the other hand, Structural UML diagrams depict the static structure of the system, showing its components and relationships. UML has been integrated as a standard by OMG, and its primary goals are to provide a formal basis for understanding modeling languages, offer a ready-to- use expressive language for system developers, and encourage the growth of object-oriented tools.

Goals of UML:

- To provide a standard visual representation of the system design.
- To simplify understanding of system architecture for developers and reviewers.

- To improve communication among team members during development.
- To model both structural and behavioral aspects of the system.
- To support object-oriented design and development practices.

Types of UML Diagrams:

1. Sequence Diagram:
2. Use Case Diagram:
3. Activity Diagram:
4. Class Diagram:

5.3.1. Sequence Diagram

The sequence diagram illustrates the workflow of the deep learning-based lung diagnosis system using lung sound data, covering both offline training and online inference phases. In the training phase, the system collects lung sound datasets and performs preprocessing steps such as noise removal and feature extraction using techniques like spectrograms or MFCC. A deep learning model such as CNN or RNN is then trained on this processed data to identify patterns associated with different respiratory conditions. The model is evaluated using metrics like accuracy, precision, recall, and F1-score, and once satisfactory performance is achieved, the trained model is saved for future use.

In the online inference phase, the user records or uploads lung sound input through the system interface. The system loads the pre-trained model and preprocesses the input audio similarly to the training phase. The model then analyzes the sound and classifies it into categories such as normal or abnormal lung conditions. Finally, the diagnosis result is displayed to the user, completing the interaction. This separation of training and inference ensures better efficiency, scalability, and real-time performance.

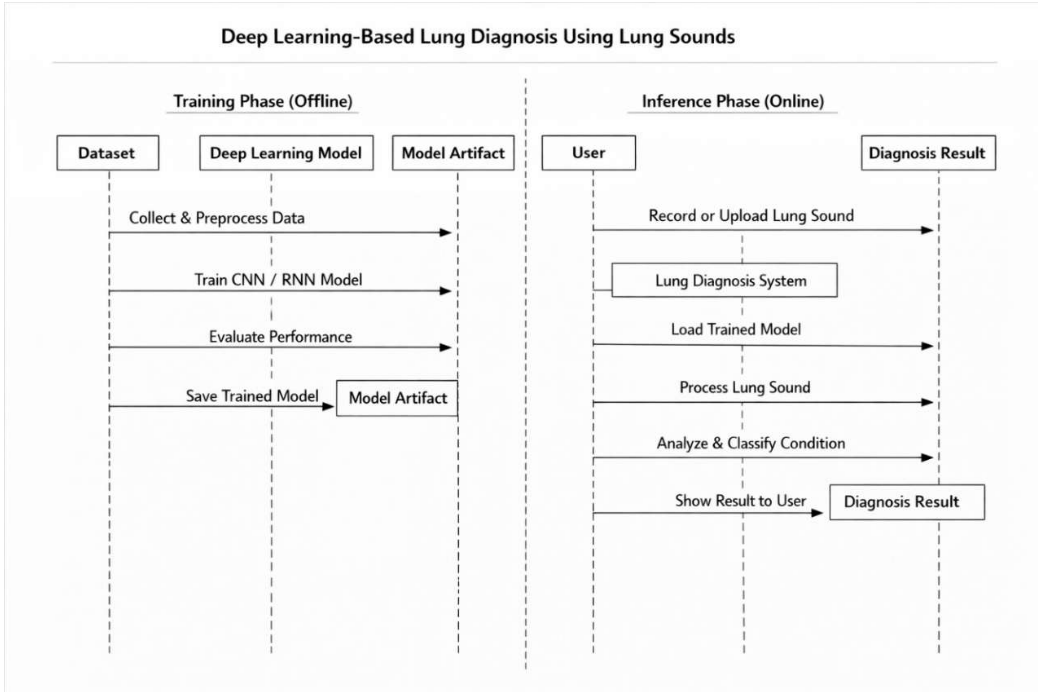


Fig 5.3.1: Sequence Diagram

List of actions

- **User:**

The user interacts with the system by registering and logging in through the interface. After successful authentication, the user records or uploads lung sound audio using a device such as a digital stethoscope or microphone. The user then receives the predicted diagnosis result generated by the system, indicating whether the lung condition is normal or abnormal.

- **Database:**

The database stores user credentials, login details, and lung sound records uploaded by users. It manages authentication during login and maintains patient history for future reference. The database also supports efficient retrieval and storage of input data and diagnosis results.

- **Training Phase (Offline):**

The database stores user credentials, login details, and lung sound records uploaded by users. It manages authentication during login and maintains patient history for future reference. The database also supports efficient retrieval and storage of input data and diagnosis results.

• **Saved Model:**

During runtime, the saved deep learning model is loaded when the user provides lung sound input. The model processes the audio data and predicts the corresponding lung condition, such as normal breathing, wheezing, or crackles. The prediction result is then returned to the user, completing the diagnosis workflow

5.3.2 Use Case Diagram

The use case diagram illustrates the overall functionality of the deep learning-based lung diagnosis system using lung sounds by clearly distinguishing between the roles of the User and the Medical Analyst (or Healthcare Expert). The user interacts with the system through registration and login, which are connected to the database for storing and validating credentials. After successful authentication, the user can record or upload lung sound audio, which is then processed by a pre-trained deep learning model to classify the lung condition. This represents the inference phase of the system, enabling efficient and real-time diagnosis without involving the training process.

The Medical Analyst is responsible for the training and evaluation pipeline of the system. This includes collecting lung sound datasets, preprocessing audio signals using techniques such as noise reduction and feature extraction (e.g., MFCC or spectrograms), and applying deep learning models like CNN or RNN for classification. The processed data is used to evaluate model performance using metrics such as accuracy, precision, recall, and F1-score. Finally, the analyst reviews the results to assess model effectiveness and improve diagnostic accuracy. This separation of training and inference ensures modularity, scalability, and clarity in system design.

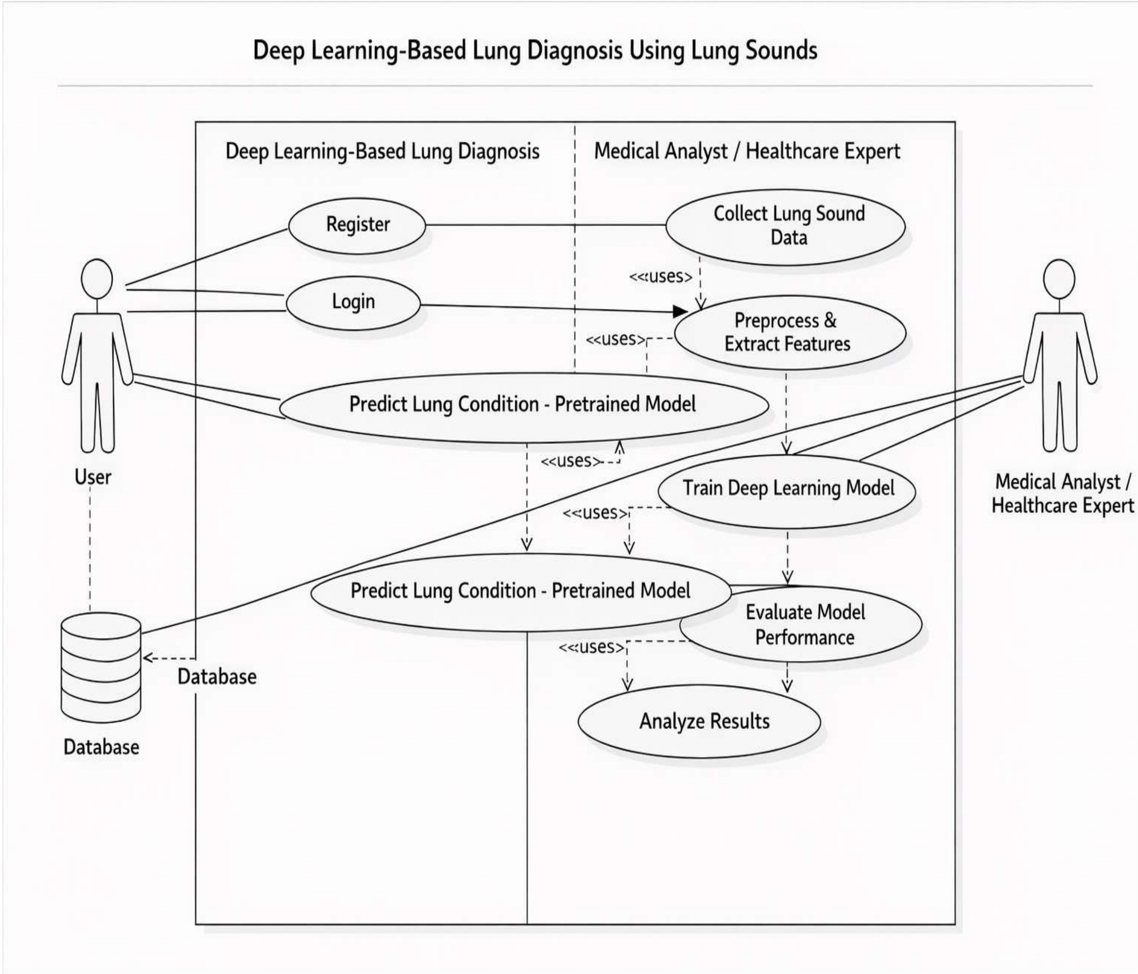


Fig 5.3.2 Use Case Diagram

5.3.3 Activity Diagram

The activity diagram represents the step-by-step workflow of the deep learning-based lung diagnosis system using lung sounds. It begins with collecting lung sound data from medical devices or datasets, followed by preprocessing and feature extraction to convert raw audio signals into meaningful representations such as MFCCs or spectrograms. The workflow then splits into parallel activities where deep learning models such as CNN and RNN are trained simultaneously to learn patterns from the lung sound data.

After parallel processing, the outputs from both models are combined using an ensemble approach to improve prediction accuracy. This aggregation step enhances reliability by leveraging multiple model predictions. Finally, the system classifies the lung condition (such as normal, wheezing, or abnormal) and reaches the end state, completing the activity flow.

Furthermore, the activity diagram also incorporates decision points and validation steps to ensure system reliability and accuracy. Before classification, the system may verify the quality of the input lung sound, checking for noise levels or incomplete recordings, and prompt the user to re-upload data if necessary. During the inference stage, the trained ensemble model processes the input and generates probability scores for each possible lung condition. These results are then interpreted and formatted into a meaningful diagnostic output. The system also includes a feedback mechanism where results are stored in the database and can be reviewed by users or administrators for further analysis.

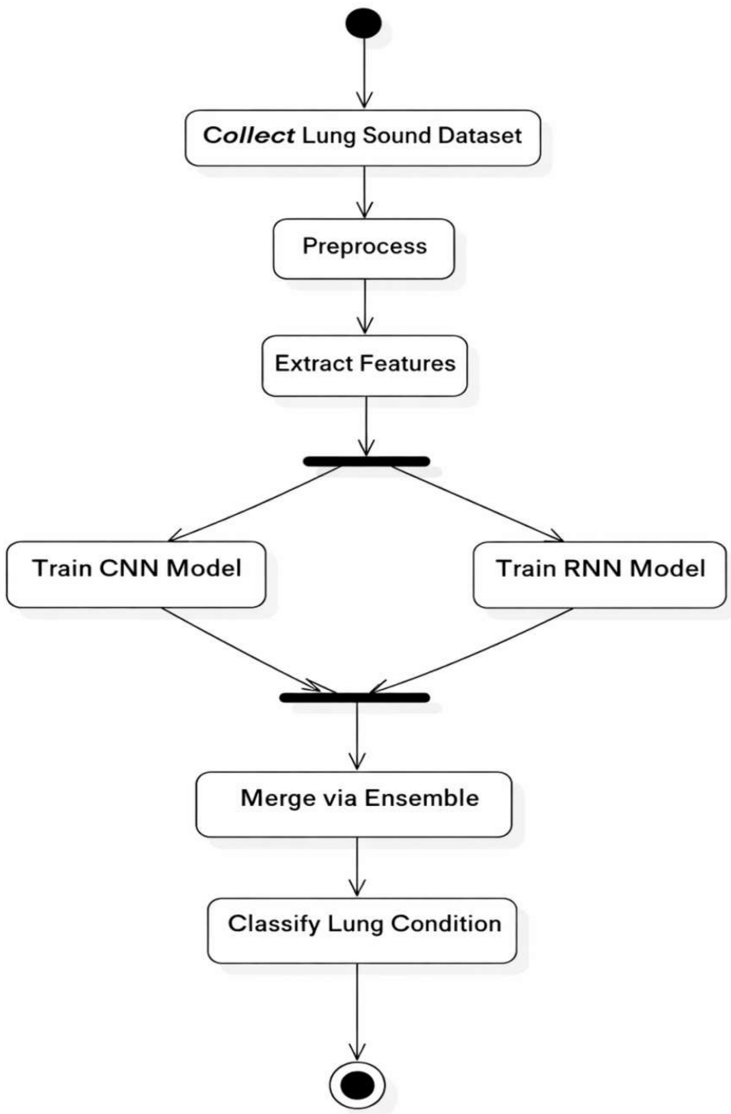


Fig 5.3.3 Activity Diagram

5.3.4. Class Diagram

The class diagram illustrates the structural organization of the deep learning-based lung diagnosis system using lung sounds, clearly defining the relationships and responsibilities of each component involved in the system. It provides a high-level view of how data flows between modules and how different classes collaborate to achieve accurate lung disease prediction. The architecture is designed in a modular manner, separating training, evaluation, and inference processes to improve maintainability and scalability.

At the core of the training phase, the Model Trainer class plays a crucial role in building deep learning models. It is responsible for training models such as Convolutional Neural Networks (CNN) for extracting spatial features from spectrogram images and Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM) networks for capturing temporal dependencies in lung sound signals. The trainer utilizes preprocessed datasets and applies optimization techniques such as backpropagation, loss minimization, and hyperparameter tuning to enhance model performance.

The Evaluation class ensures the quality and reliability of the trained models. It uses a separate test dataset to evaluate performance metrics such as accuracy, precision, recall, F1-score, and loss. This class generates evaluation reports and visualizations (such as confusion matrices and ROC curves) to analyze model effectiveness. Proper evaluation helps in identifying overfitting or underfitting issues and ensures that only well-performing models are deployed.

The Dataset class is responsible for managing input data. It handles loading raw lung sound recordings from various sources and performs preprocessing operations such as noise reduction, normalization, segmentation, and silence removal. It also converts audio signals into suitable representations like Mel-spectrograms or MFCC (Mel-Frequency Cepstral Coefficients), which are essential for deep learning models to extract meaningful features.

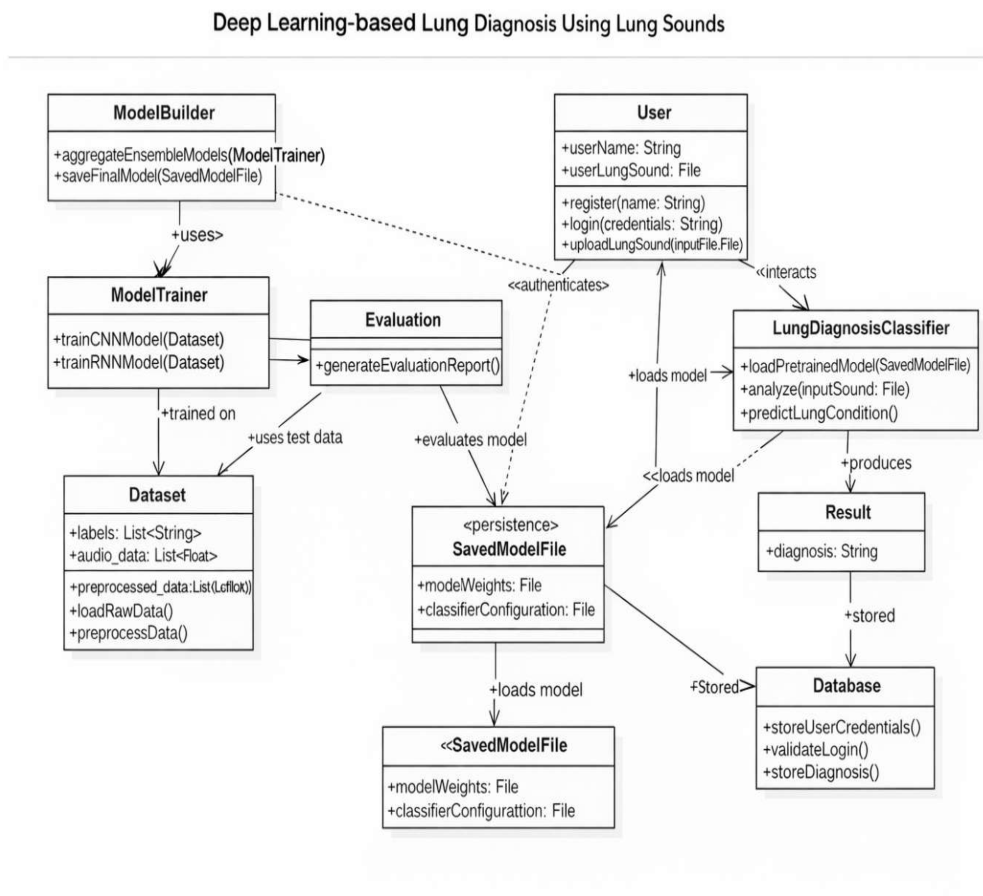


Fig 5.3.4 Class Diagram

CHAPTER-6

IMPLEMENTATION AND CODING

6. IMPLEMENTATION AND CODING

6.1 Source Code :

FLASK CODE:

```
from flask import Flask, render_template, request
import numpy as np
import librosa
import tensorflow as tf
from tensorflow.keras.models import load_model
import os

app = Flask(__name__)

# Define constants
SAMPLE_RATE = 22050
N_MFCC = 40
MAX_PAD_LENGTH = 150
LABELS = ['bronchial', 'asthma', 'copd', 'healthy', 'pneumonia']  # Ensure it matches
training order

# Load trained model
MODEL_PATH = "Model/asthma_cnn_rnn_model.h5"
model = load_model(MODEL_PATH)

# Function to extract MFCC features from an audio file
def extract_mfcc(file_path):
    y, sr = librosa.load(file_path, sr=SAMPLE_RATE)
    mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=N_MFCC)
    pad_width = MAX_PAD_LENGTH - mfcc.shape[1]
    if pad_width > 0:
        mfcc = np.pad(mfcc, pad_width=((0, 0), (0, pad_width)), mode='constant')
    else:
        mfcc = mfcc[:, :MAX_PAD_LENGTH]
    return mfcc

# Function to predict lung condition from input audio file
def predict_lung_condition(file_path):
    try:
        mfcc = extract_mfcc(file_path)
        mfcc = np.expand_dims(mfcc, axis=-1)  # Add channel dimension
        mfcc = np.expand_dims(mfcc, axis=0)   # Add batch dimension

        prediction = model.predict(mfcc)
        predicted_label = LABELS[np.argmax(prediction)]
        confidence = np.max(prediction) * 100

        return predicted_label, confidence
```

```
    except Exception as e:
        return None, str(e)

@app.route('/', methods=['GET', 'POST'])
def index():
    prediction = None
    confidence = None

    if request.method == 'POST':
        if 'file' not in request.files:
            return render_template('index.html', error="No file part")
        file = request.files['file']
        if file.filename == '':
            return render_template('index.html', error="No selected file")

        if file:
            filepath = os.path.join("uploads", file.filename)
            file.save(filepath)
            prediction, confidence = predict_lung_condition(filepath)
            os.remove(filepath)

    return render_template('index.html', prediction=prediction, confidence=confidence)

if __name__ == '__main__':
    app.run(debug=True)

import numpy as np
import librosa
import tensorflow as tf
from tensorflow.keras.models import load_model
import sys

# Define constants
SAMPLE_RATE = 22050
N_MFCC = 40
MAX_PAD_LENGTH = 150
LABELS = ['bronchial', 'asthma', 'copd', 'healthy', 'pneumonia']  # Ensure it matches
training order

# Load trained model
MODEL_PATH = r"Model\asthma_cnn_rnn_model.h5"
model = load_model(MODEL_PATH)

# Function to extract MFCC features from an audio file
def extract_mfcc(file_path):
    y, sr = librosa.load(file_path, sr=SAMPLE_RATE)
    mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=N_MFCC)
    pad_width = MAX_PAD_LENGTH - mfcc.shape[1]
    if pad_width > 0:
        mfcc = np.pad(mfcc, pad_width=((0, 0), (0, pad_width)), mode='constant')
```

```
    else:
        mfcc = mfcc[:, :MAX_PAD_LENGTH]
    return mfcc

# Function to predict lung condition from input audio file
def predict_lung_condition(file_path):
    try:
        mfcc = extract_mfcc(file_path)
        mfcc = np.expand_dims(mfcc, axis=-1)  # Add channel dimension
        mfcc = np.expand_dims(mfcc, axis=0)   # Add batch dimension

        prediction = model.predict(mfcc)
        predicted_label = LABELS[np.argmax(prediction)]
        confidence = np.max(prediction) * 100

        print(f"Predicted Condition: {predicted_label} (Confidence: {confidence:.2f}%)")
    except Exception as e:
        print(f"Error processing file: {e}")

# Provide input file here
INPUT_FILE = r"Model\Dataset\pneumonia\P1Pneumonia41U.wav"  # Change this to
the actual file path
predict_lung_condition(INPUT_FILE)
```

MODEL TRAINING CODE:

```
import os
import numpy as np
import librosa
import librosa.display
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Model
from tensorflow.keras.layers import (Conv2D, MaxPooling2D, Flatten, Dense,
Dropout,
                                     BatchNormalization, LSTM, TimeDistributed,
                                     Bidirectional, Reshape, Input)
import warnings
warnings.filterwarnings("ignore")

# Define dataset paths
dataset_paths = {
    'bronchial': '/kaggle/input/asthma-detection-dataset-version-2/Asthma Detection
Dataset Version 2/Asthma Detection Dataset Version 2/Bronchial',
    'asthma': '/kaggle/input/asthma-detection-dataset-version-2/Asthma Detection
Dataset Version 2/Asthma Detection Dataset Version 2/asthma',
```

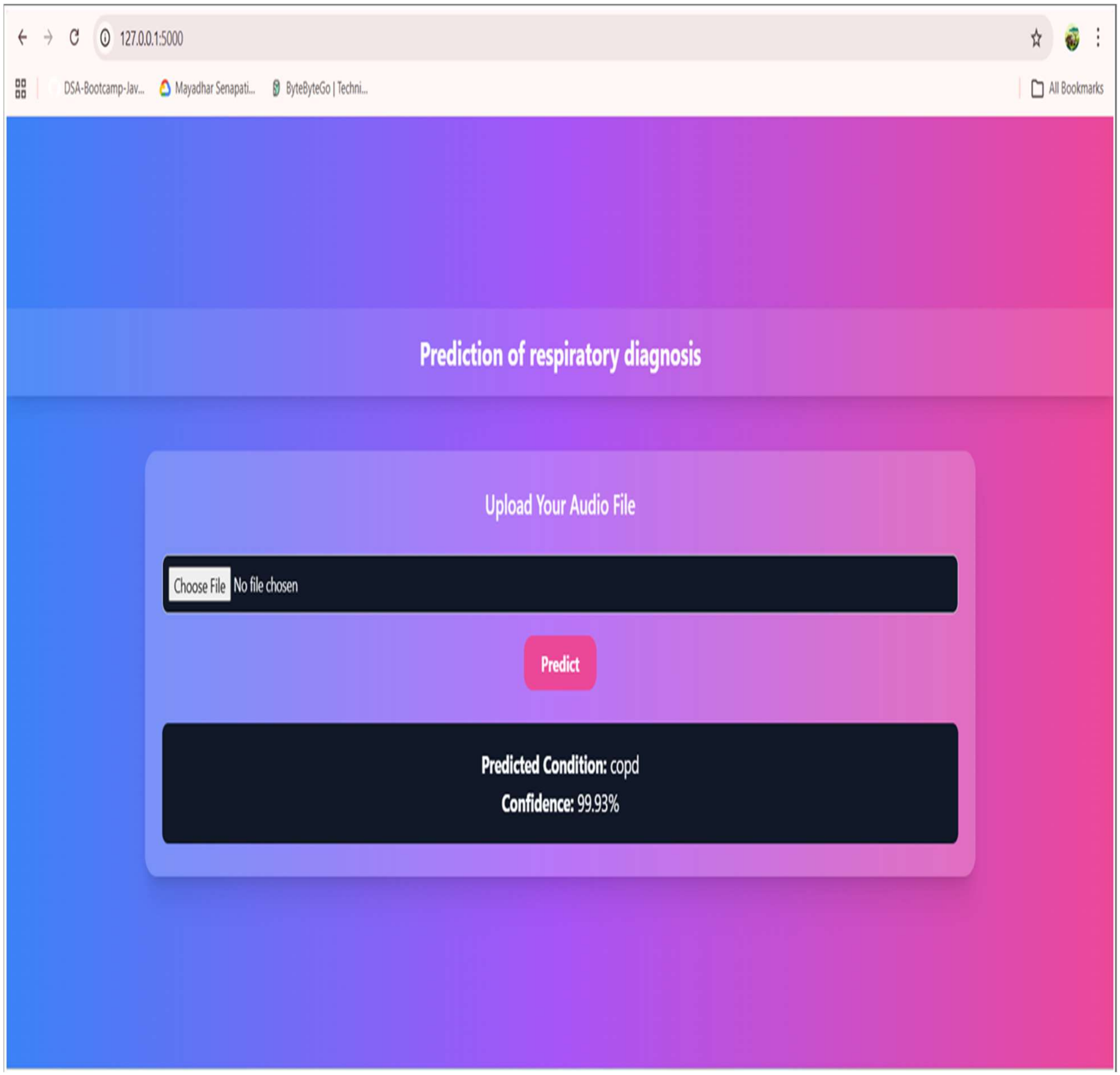
```
    'copd': '/kaggle/input/asthma-detection-dataset-version-2/Asthma Detection Dataset
Version 2/Asthma Detection Dataset Version 2/copd',
    'healthy': '/kaggle/input/asthma-detection-dataset-version-2/Asthma Detection
Dataset Version 2/Asthma Detection Dataset Version 2/healthy',
    'pneumonia': '/kaggle/input/asthma-detection-dataset-version-2/Asthma Detection
Dataset Version 2/Asthma Detection Dataset Version 2/pneumonia'
}

# Hyperparameters
SAMPLE_RATE = 22050  # Standard sample rate for audio
N_MFCC = 40  # Number of MFCC features
MAX_PAD_LENGTH = 150  # Padding length for MFCCs
BATCH_SIZE = 32
EPOCHS = 30
NUM_CLASSES = len(dataset_paths)

# Function to extract MFCC features
def extract_mfcc(file_path):
    y, sr = librosa.load(file_path, sr=SAMPLE_RATE)
    mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=N_MFCC)
    pad_width = MAX_PAD_LENGTH - mfcc.shape[1]
    if pad_width > 0:
        mfcc = np.pad(mfcc, pad_width=((0, 0), (0, pad_width)), mode='constant')
    else:
        mfcc = mfcc[:, :MAX_PAD_LENGTH]
    return mfcc

# Load dataset
X, Y = [], []
labels = list(dataset_paths.keys())
label_map = {label: idx for idx, label in enumerate(labels)}

for label, path in dataset_paths.items():
    for file in os.listdir(path):
        file_path = os.path.join(path, file)
        try:
            mfcc = extract_mfcc(file_path)
            X.append(mfcc)
            Y.append(label_map[label])
        except Exception as e:
            print(f"Error processing {file}: {e}")

X = np.array(X)
Y = np.array(Y)

# Reshape for CNN input
X = X[..., np.newaxis]  # Adding channel dimension
Y = to_categorical(Y, num_classes=NUM_CLASSES)  # Convert labels to one-hot
encoding
```

```
# Train-test split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=42, stratify=Y)

# Build CNN + RNN Model
def build_model(input_shape, num_classes):
    inputs = Input(shape=input_shape)

    # CNN Feature Extraction
    x = Conv2D(32, (3, 3), activation='relu', padding='same')(inputs)
    x = BatchNormalization()(x)
    x = MaxPooling2D((2, 2))(x)

    x = Conv2D(64, (3, 3), activation='relu', padding='same')(x)
    x = BatchNormalization()(x)
    x = MaxPooling2D((2, 2))(x)

    x = Conv2D(128, (3, 3), activation='relu', padding='same')(x)
    x = BatchNormalization()(x)
    x = MaxPooling2D((2, 2))(x)

    x = Reshape((x.shape[1], -1))(x)  # Reshape to feed into LSTM

    # RNN (LSTM)
    x = Bidirectional(LSTM(64, return_sequences=True))(x)
    x = Bidirectional(LSTM(32))(x)

    # Fully Connected Layer
    x = Dense(64, activation='relu')(x)
    x = Dropout(0.3)(x)
    x = Dense(num_classes, activation='softmax')(x)

    model = Model(inputs, x)
    return model

# Compile model
model = build_model(input_shape=(N_MFCC, MAX_PAD_LENGTH, 1),
num_classes=NUM_CLASSES)
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
model.summary()
```

```
Epoch 1/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 6s 42ms/step - accuracy: 0.3988 - loss: 1.4334 - val_accuracy: 0.3951 - val_loss: 1.4338
Epoch 2/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.6200 - loss: 1.0453 - val_accuracy: 0.5103 - val_loss: 1.2945
Epoch 3/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.7090 - loss: 0.8426 - val_accuracy: 0.4527 - val_loss: 1.2624
Epoch 4/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.7696 - loss: 0.6712 - val_accuracy: 0.2881 - val_loss: 1.6886
Epoch 5/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.7911 - loss: 0.5950 - val_accuracy: 0.5597 - val_loss: 1.0836
Epoch 6/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.8583 - loss: 0.4154 - val_accuracy: 0.4733 - val_loss: 1.5980
Epoch 7/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.8599 - loss: 0.4292 - val_accuracy: 0.6008 - val_loss: 1.1221
Epoch 8/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.8697 - loss: 0.3999 - val_accuracy: 0.7613 - val_loss: 0.6477
Epoch 9/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.9171 - loss: 0.2858 - val_accuracy: 0.8189 - val_loss: 0.4970
Epoch 10/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.8897 - loss: 0.3270 - val_accuracy: 0.8560 - val_loss: 0.4709
Epoch 11/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.9351 - loss: 0.2072 - val_accuracy: 0.7860 - val_loss: 0.6625
Epoch 12/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.9397 - loss: 0.1881 - val_accuracy: 0.8066 - val_loss: 0.6029
Epoch 13/30
...
Epoch 29/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.9840 - loss: 0.0296 - val_accuracy: 0.8477 - val_loss: 0.6826
Epoch 30/30
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.9893 - loss: 0.0350 - val_accuracy: 0.8025 - val_loss: 0.7990
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Output :


127.0.0.1:5000
DSA-Bootcamp-Jav...
Mayadhar Senapati...
ByteByteGo | Techni...
All Bookmarks
Prediction of respiratory diagnosis
Upload Your Audio File
Choose File No file chosen
Predict
Predicted Condition: copd
Confidence: 99.93%

6.2 Implementation:

6.2.1 Front-End Implementation:

The frontend of the deep learning-based lung diagnosis system is developed using HTML, CSS, and Bootstrap, providing a clean, responsive, and user-friendly interface. The main page allows users (such as doctors or patients) to upload lung sound recordings or input patient-related details like age, gender, symptoms, and medical history. The audio input can be provided in common formats, ensuring ease of use without technical complexity.

In addition to the input page, the system includes a predictions page where the diagnostic results are displayed in a structured format. The results may include predicted lung conditions such as normal, asthma, pneumonia, or other respiratory abnormalities along with confidence scores. This enables users to review previous diagnoses and compare results efficiently. Pagination is implemented to handle large volumes of patient records and improve usability.

Another important component of the frontend is the visualization page, which presents graphical representations of diagnostic results. Charts such as pie charts, bar graphs, and line graphs are used to show the distribution of different lung diseases, frequency of cases, and trends over time. These visualizations help medical professionals better understand patterns in respiratory conditions. A navigation bar is integrated across all pages, enabling easy access to input, predictions, and analysis sections. The use of Bootstrap ensures compatibility across devices like desktops, tablets, and mobile phones.

6.2.2 Backend Implementation (Django):

The backend of the lung diagnosis system is developed using the Flask framework in Python, which serves as the core component for handling application logic and processing user requests. Flask manages routing by directing user actions such as uploading lung sound data, requesting predictions, or viewing analysis results to the appropriate system functions.

When a user uploads a lung sound recording, the backend processes the audio file by performing preprocessing steps such as noise reduction, segmentation, and feature extraction. Important features like MFCCs (Mel-Frequency Cepstral Coefficients) and spectrograms are generated to convert raw audio signals into a format suitable for deep learning models. These features are then normalized to match the format used during model training.

The system uses deep learning models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to analyze lung sounds and classify respiratory conditions. The trained model predicts the category of lung disease based on extracted features and returns the result to the frontend.

SQLite is used as a lightweight database to store patient data, uploaded audio references, and prediction results. This allows the system to maintain a history of diagnoses, which can be retrieved for future reference or analysis. Additionally, the backend generates graphical outputs using Matplotlib by analyzing stored prediction data. These graphs are converted into Base64 format so they can be seamlessly displayed on the frontend. Overall, the backend ensures efficient data processing, accurate predictions, and smooth communication between the user interface and the deep learning model.

6.2.3 Model Integration and Processing Workflow

The deep learning module for lung diagnosis is integrated as a backend service within the system architecture, ensuring seamless interaction between the user interface and the predictive model. When a lung sound recording is received from the frontend, it is forwarded to the backend for processing. The system is designed to handle real-time inputs efficiently, allowing users to quickly obtain diagnostic results without delays.

Once the audio input is received, it undergoes several preprocessing steps to improve data quality and prepare it for analysis. These steps include noise reduction to eliminate background disturbances, segmentation to isolate meaningful portions of the lung sound, and normalization to ensure consistency across different audio samples. After preprocessing, feature extraction techniques such as Mel-Frequency Cepstral Coefficients (MFCC) and spectrogram generation are applied to convert the audio signals into structured numerical representations suitable for deep learning models.

Finally, the prediction output is formatted into a structured JSON response and sent back to the frontend. The result is then displayed to the user in an easy-to-understand format, often accompanied by visual indicators or graphs. This complete workflow ensures accurate, efficient, and user-friendly lung disease diagnosis based on lung sound analysis.

The trained model artifacts are saved and loaded during runtime to ensure efficient and fast predictions. This avoids the need for retraining the model each time a new input is provided, making the system suitable for real-time applications.

6.2.4 Deployment and Reliability

The deep learning-based lung diagnosis system is deployed on a standard server environment with secure configuration settings to ensure safe and reliable operation in healthcare scenarios. The deployment setup includes optimized handling of static files and efficient audio processing pipelines, allowing quick uploading and analysis of lung sound recordings. RESTful APIs are implemented to enable smooth communication between the frontend and backend, and they are thoroughly tested to maintain consistent performance under various usage conditions. This ensures that the system can deliver accurate diagnostic results without delays or failures.

To enhance reliability, both unit testing and integration testing are performed across different components of the system.

Unit testing validates individual modules such as preprocessing, feature extraction, and model prediction, while integration testing ensures seamless interaction between APIs, databases, and the deep learning model. The system is designed to handle multiple user requests simultaneously, making it scalable for real-world healthcare environments. These measures collectively ensure robustness, accuracy, and dependable performance in lung disease diagnosis using lung sound analysis.

6.2.5 Conclusion

The implementation successfully integrates a user-friendly interface, a secure Django backend, and a deep learning-based diagnostic framework into a unified lung diagnosis system. The system supports real-time lung sound analysis, enabling users to receive quick and accurate predictions.

The modular architecture allows future enhancements such as integration with IoT-based medical devices, advanced deep learning models, and remote healthcare monitoring. Overall, the system provides a scalable and practical solution for assisting in early detection of respiratory conditions using lung sound analysis.

The modular architecture of the system provides flexibility for future enhancements, such as integration with IoT-based medical devices, incorporation of more advanced deep learning models, and support for remote patient monitoring. This adaptability ensures that the system can evolve with emerging technologies and healthcare needs.

In addition, the system demonstrates strong potential for deployment in both clinical and remote settings, particularly in resource-constrained environments where access to specialized medical expertise may be limited. By leveraging automated analysis and intelligent decision support, it can assist healthcare professionals in making faster and more informed diagnoses while also empowering patients with preliminary insights into their respiratory health. The emphasis on scalability and interoperability ensures seamless integration with existing healthcare infrastructures, including electronic health records and telemedicine platforms. Furthermore, continuous model improvement through updated datasets and retraining can enhance diagnostic accuracy over time, making the system increasingly reliable. Ultimately, this approach contributes to improved healthcare accessibility, early intervention, and better patient outcomes in the management of respiratory diseases.

CHAPTER-7
SYSTEM TESTING

7. SYSTEM TESTING

System testing is a critical phase that ensures the deep learning-based lung diagnosis system performs accurately, consistently, and reliably under real-world operating conditions. The main objective of this testing process is to identify potential defects, validate the system's behavior, and confirm that both functional and non-functional requirements are fully satisfied. It helps in ensuring that the application delivers correct diagnostic results while maintaining performance and usability standards.

In this project, system testing focuses on validating all major modules of the application, including the frontend interface, backend services, audio preprocessing pipeline, and the deep learning-based classification engine. The frontend is tested for proper user interaction and smooth navigation, while the backend is verified for correct API responses and data handling. Additionally, the preprocessing components are evaluated to ensure accurate feature extraction from lung sound recordings, and the CNN and RNN models are tested to confirm correct prediction logic.

Testing is carried out using multiple datasets and scenarios to ensure the system's robustness and reliability. Different types of lung sounds, including normal and abnormal patterns such as wheezing and crackles, are used to validate classification accuracy. Special attention is given to handling noisy or low-quality audio inputs to ensure that the system can still provide meaningful predictions. This helps in making the system more practical for real-world healthcare environments where input quality may vary.

Furthermore, stability and performance testing are conducted to evaluate the system under repeated usage and varying workloads. The system is tested for its ability to handle multiple prediction requests without performance degradation or failure. Consistency in output results is also verified to ensure dependable diagnosis over time. Overall, the system demonstrates strong reliability, robustness, and accuracy, making it suitable for deployment in real-time lung disease detection applications.

Additionally, performance and stability tests are conducted to confirm that the system can handle repeated and concurrent prediction requests without degradation. Overall, system testing confirms that the application meets both functional and non-functional requirements, making it dependable for practical healthcare use.

7.1 Types of System Testing

7.1.1 Unit Testing

Unit testing was performed to ensure that each individual component of the system functioned correctly in isolation. This included testing Django views, backend functions, audio preprocessing routines, and model interaction modules using controlled and predefined input values. The audio preprocessing pipeline was specifically validated for tasks such as noise reduction and segmentation, while feature extraction methods like MFCC and spectrogram generation were verified for accuracy and consistency. Additionally, the deep learning model inference logic and database operations related to user authentication and diagnosis record storage were tested independently to confirm correct outputs. This phase helped identify and resolve issues early in development, ensuring that all core modules behaved as expected before integration.

7.1.2 Integration Testing

Integration testing was conducted to verify the interaction between multiple system components once they were combined. This included testing the end-to-end flow of audio upload from the front-end interface to backend API processing. The complete preprocessing pipeline was evaluated in conjunction with the deep learning models to ensure proper data transformation and seamless model execution. In cases where ensemble learning was used, outputs from CNN and RNN models were also tested for correct combination and interpretation. Furthermore, the storage and retrieval of diagnosis results in the database were validated to ensure smooth communication between application layers and reliable data persistence across the system workflow. Additional testing was carried out to check error handling during API failures, unexpected input interruptions, and partial system breakdowns, ensuring that the system could recover gracefully without affecting overall functionality.

7.1.3 Functional Testing

Functional testing was carried out to confirm that the system met all specified requirements and provided the expected user experience. The system was evaluated to ensure that valid audio inputs were processed successfully and that invalid or unsupported file formats were handled gracefully with appropriate responses. The accuracy and clarity of lung condition predictions were verified to ensure correct information was displayed to the user. In addition, the user interface was tested for smooth navigation, responsiveness, and proper interaction between different screens and features. This phase ensured that the system was not only technically sound but also user-friendly and fully aligned with functional requirements.

This phase ensured that the system was not only technically sound but also user-friendly and fully aligned with functional requirements. Further validation included checking system behavior under different user scenarios such as repeated uploads, multiple user sessions, and varying audio qualities to confirm robustness and consistency of results across real-world usage conditions.

7.1.4 System Testing

System testing was conducted to evaluate the application as a complete and integrated unit, with a primary focus on reliability, consistency, and accuracy under real-world operating conditions. The testing process verified the end-to-end execution of the lung diagnosis workflow, ensuring that each stage—from audio input and preprocessing to feature extraction and final prediction—functioned seamlessly without errors. Additionally, the system's ability to handle large audio datasets was assessed to confirm that performance remains stable and efficient even under high data loads.

Further evaluation was carried out to measure classification accuracy across a variety of lung sound patterns, including normal, wheezing, and crackling sounds. The results demonstrated that the system consistently generated accurate and reliable predictions, highlighting the effectiveness of the underlying deep learning models. These tests also ensured that the model could generalize well across diverse acoustic variations, making it suitable for practical deployment in real-world healthcare scenarios.

Stability testing was performed by executing repeated predictions over extended periods to assess the system's robustness and consistency. The application maintained stable performance without noticeable degradation, crashes, or fluctuations in output quality. This confirmed that the system is capable of sustained operation in continuous-use environments, making it dependable for both clinical settings and remote health monitoring applications.

7.1.5 White-Box Testing

White-box testing focused on examining the internal logic and structure of the system, particularly the preprocessing pipelines and deep learning model execution. This involved validating each stage of the data flow, from raw audio input through feature extraction to final prediction, ensuring that transformations were applied correctly and consistently. Key components such as normalization, feature scaling, and tensor generation were thoroughly inspected to confirm that they adhered to the expected design and did not introduce errors or inconsistencies.

In addition, the internal computations of the deep learning model were analyzed in detail, including weight propagation, activation outputs, and prediction probability distributions. This allowed verification that the model was functioning as intended and producing mathematically sound results.

7.1.6 Black-Box Testing

Black-box testing was conducted to evaluate the system purely from the user's perspective, without any visibility into its internal logic or implementation. Various lung sound recordings were submitted as input, and the resulting outputs were carefully analyzed to ensure correctness, usability, and reliability. This approach verified that the system responds appropriately to valid and invalid inputs, provides clear diagnostic results, and handles errors gracefully. The testing also confirmed that the user interface is intuitive and that the overall workflow—from uploading audio to receiving predictions—is smooth and user-friendly.

In addition, the testing process emphasized usability and error handling to ensure a seamless user experience. The system was observed to respond appropriately to both valid and invalid inputs, providing clear feedback and meaningful error messages when necessary. The user interface was found to be intuitive and easy to navigate, and the overall workflow—from uploading audio files to receiving predictions—was smooth and efficient, demonstrating the system's readiness for practical use.

7.1.7 Acceptance Testing

Acceptance testing was conducted to validate that the system fulfills all specified functional requirements while aligning closely with user expectations, particularly within a healthcare environment. The evaluation focused on critical aspects such as clarity of diagnostic predictions, ease of navigation, and overall workflow simplicity. Special attention was given to ensuring that users—both medical professionals and non-experts—can interact with the system efficiently without requiring extensive technical knowledge. The results of acceptance testing demonstrated that the system performs reliably and consistently under practical conditions, delivering accurate predictions along with clear and interpretable outputs.

Users were able to easily understand the diagnostic results and confidence levels, which enhances trust and usability. Overall, the system was deemed ready for deployment, successfully meeting both technical performance standards and user-centric requirements, making it a viable solution for real-world respiratory health assessment.

7.2 Testing Strategies

A structured testing strategy was followed throughout the project lifecycle, progressing systematically from component-level validation to full system verification. In the initial stages, unit testing was performed to validate individual modules such as audio preprocessing, feature extraction, and model inference. Each component was tested independently to ensure correctness, stability, and adherence to expected outputs before integration. This approach helped identify and resolve issues early in the development process, reducing the risk of compounded errors in later stages.

As development progressed, integration testing was conducted to verify that different modules work together seamlessly. The interaction between components—such as data flow from preprocessing to feature extraction and finally to the classification model—was carefully examined to ensure compatibility and consistency. This phase ensured that intermediate outputs were correctly passed and interpreted across modules, maintaining the integrity of the overall pipeline.

Finally, system-level testing was performed to evaluate the complete application in a real-world context. This included validating end-to-end functionality, performance under varying workloads, and user interaction through the interface. By simulating real usage scenarios, the testing strategy ensured that the system is reliable, efficient, and capable of delivering accurate diagnostic results. This layered approach to testing ultimately.

7.2.1 Test Strategy and Approach

The test strategy and approach for the deep learning-based lung diagnosis system are designed to ensure comprehensive validation of functionality, performance, and reliability. A multi-level testing methodology is adopted, including unit testing, integration testing, system testing, and acceptance testing. Each module—such as audio input handling, preprocessing, feature extraction, model inference, and output generation—is first tested independently to verify correctness. Automated test cases and validation datasets are used to ensure consistent and repeatable testing outcomes, particularly for evaluating the performance of the deep learning model.

Finally, the testing process follows a continuous and iterative approach, where feedback from each testing phase is used to refine and improve the system. End-to-end testing is carried out to validate the complete workflow from user input to report generation and storage. Proper documentation of test cases, results, and identified issues ensures traceability and accountability throughout the development lifecycle. This structured testing strategy guarantees that the system meets quality standards and is reliable for deployment in practical healthcare environments.

7.2.2 Test Objectives

The primary objective of testing the deep learning-based lung diagnosis system is to ensure that all components function correctly, accurately, and reliably according to the specified requirements. This includes validating that the system can process lung sound inputs, perform preprocessing and feature extraction, and generate precise diagnostic predictions using the trained model.

The testing also aims to evaluate system performance under different conditions, ensuring robustness, stability, and efficient handling of multiple requests. Additionally, it verifies seamless integration between modules and ensures a user-friendly interface for smooth interaction. Overall, the objective is to confirm that the system delivers accurate results, maintains reliability, and is suitable for real-world healthcare deployment.

7.2.3 Features Tested

The major system features examined included:

- audio upload and recording functionality
- user authentication and data validation
- prediction accuracy for lung conditions
- deep learning model execution
- data storage and retrieval

7.2.4 Integration Testing Strategy

The integration testing strategy for the deep learning-based lung diagnosis system focuses on verifying the seamless interaction between all interconnected modules. The system is divided into key components such as the user interface, backend server, audio preprocessing unit, feature extraction module, deep learning model, and database. Integration testing is performed incrementally by combining these modules step-by-step and validating data flow between them. For example, testing ensures that audio files uploaded through the web interface are correctly received by the backend, preprocessed accurately, and then passed to the model for prediction without any data loss or format issues.

7.2.5 Acceptance Criteria

The acceptance criteria for the developed deep learning-based lung diagnosis system are defined to ensure that the system meets user expectations and functional goals. The system must accurately accept lung sound audio inputs, process them through the trained model, and generate reliable diagnostic predictions with a high level of accuracy. It should provide clear and interpretable results, including predicted disease categories and confidence scores. Additionally, the system must support smooth user interaction through a web-based interface, ensuring that users can easily upload audio files and receive outputs without technical difficulties

7.2.6 Overall Test Results

All planned test cases were executed successfully, covering unit, integration, system, and user-level validation scenarios. The system consistently demonstrated stable performance across all stages of testing, with no critical failures or major inconsistencies observed. It effectively handled different types of lung sound inputs, including normal, wheezing, and crackling patterns, while maintaining smooth processing throughout the pipeline from input acquisition to final prediction.

The deep learning models showed strong capability in accurately identifying respiratory conditions, producing reliable and consistent results under varying conditions. The predictions remained stable even during repeated testing and across different datasets, confirming the robustness of the trained model. Overall, the system met the expected performance criteria and demonstrated its readiness for practical deployment in real-world healthcare applications.

7.2.7 Conclusion

System testing confirmed that the developed deep learning-based lung diagnosis system meets functional requirements, performs reliably under different conditions, and integrates all modules effectively. Through comprehensive testing strategies, the system achieved robustness, usability, and high diagnostic accuracy, making it suitable for real-world healthcare applications.

Extensive testing under varying conditions further established the system's reliability and robustness. The model maintained consistent performance across different audio qualities, background noise levels, and diverse patient data samples. Stress testing and performance evaluation showed that the system can handle multiple inputs efficiently without degradation in speed or accuracy, ensuring dependable operation in real-time clinical environments.

Additionally, integration testing verified seamless interaction among all system modules, including the web interface, backend processing, and database storage. The system demonstrated high usability with an intuitive interface, enabling users to upload audio and receive results effortlessly. Combined with its strong diagnostic accuracy and stability, the system proves to be a practical and effective solution for real-world healthcare applications, supporting early detection and improved decision-making in lung disease diagnosis.

7.3 Sample Test Cases:

S No.	Test Case	Description	Expected Result	Result	Remarks (if any)
01	Audio Input Interface	User uploads lung sound audio through the interface	System should accept valid audio and prepare for processing	Pass	Minor issues with unsupported formats
02	Raw Lung Sound Waveform	Visualization of lung sound signal (time vs amplitude)	Waveform should be displayed correctly	Pass	Helps identify breathing patterns clearly
03	Preprocessed Lung Sound & Prediction	Audio is cleaned, normalized and used for prediction	System should output correct diagnosis (e.g., COPD/Normal)	Pass	Model integration working properly
04	MFCC Feature Extraction	Converts audio into numerical MFCC features	Features should be extracted correctly	Pass	Some low-intensity features may be missed
05	CNN Feature Extraction	CNN extracts spatial patterns from MFCC features	Relevant features should be identified	Pass	Complex patterns need deeper layers
06	Model Training Accuracy Graph	Accuracy plotted over epochs	Accuracy should increase and stabilize	Pass	Slight fluctuations in early training

07	Model Training Loss Graph	Loss plotted over epochs	Loss should decrease steadily	Pass	Minor instability initially
08	CNN Confusion Matrix	Visualization of CNN classification results	Correct predictions should dominate diagonal	Pass	Some misclassifications observed
09	CNN + BiGRU Model Performance	Hybrid model loss and confusion matrix	Improved accuracy over CNN	Pass	Better performance due to temporal learning
10	Noisy Audio Input	System processes audio with high background noise	System should still classify reasonably	Fail	Accuracy drops significantly for noisy input
11	Low Volume Audio Input	System processes very low amplitude signals	System should detect weak signals correctly	Fail	Low intensity sounds not detected properly

Test Case 1:

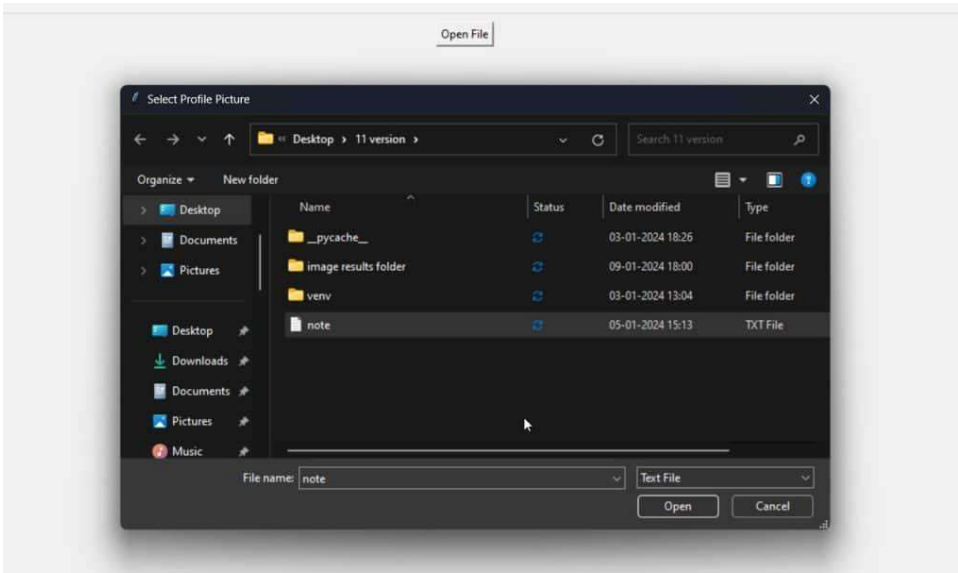


Fig 7.3.1 Audio Input Interface for Uploading Lung Sound Signal

Description: The user uploads a lung sound audio file through the interface. The system accepts valid audio formats and prepares the file for processing. Successful upload confirms that the audio input module and file handling mechanism are functioning correctly.

Test Case 2:

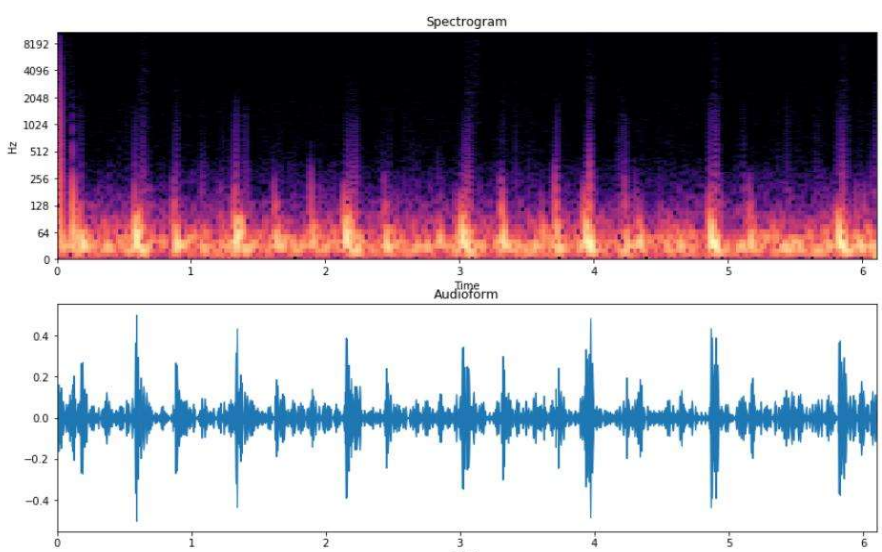


Fig 7.3.2 Visualization of Raw Lung Sound Waveform

Description: This figure represents the raw lung sound signal in the form of a waveform. The x-axis shows time, while the y-axis represents amplitude. It helps in understanding the variation in sound intensity and identifying breathing patterns such as normal airflow, wheezing, or crackles before preprocessing.

Test Case 3:

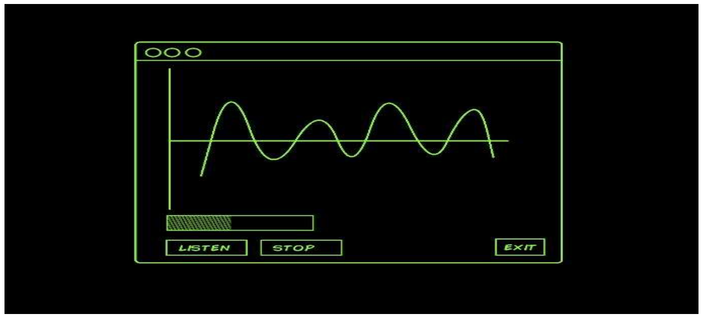
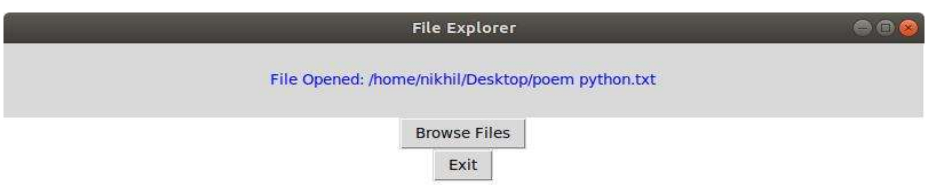


Fig.7.3.3 A Preprocessed Lung Sound after Noise Reduction and Normalization

Description: The user uploads or records lung sound audio and submits it for diagnosis. The system processes the audio using preprocessing techniques and a deep learning model, then displays the predicted condition such as "COPD" or "Normal." This confirms that the prediction pipeline and model integration are functioning correctly Shows the cleaned and normalized audio signal after preprocessing, improving the quality of input data for model analysis.

Test Case 4:



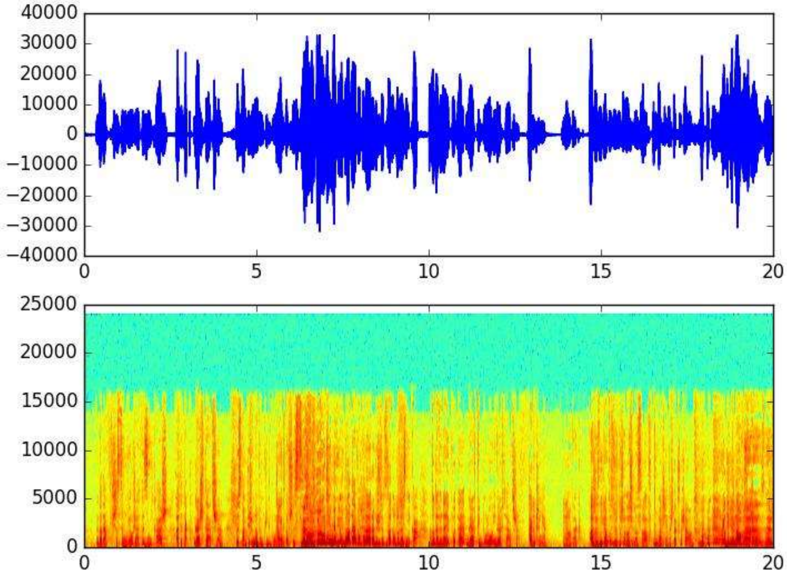


Fig. 7.3.4 MFCC Feature Extraction from Lung Sound Audio

Description: The user Represents the extracted MFCC features, which convert audio signals into numerical form suitable for deep learning models.

Test Case 5:

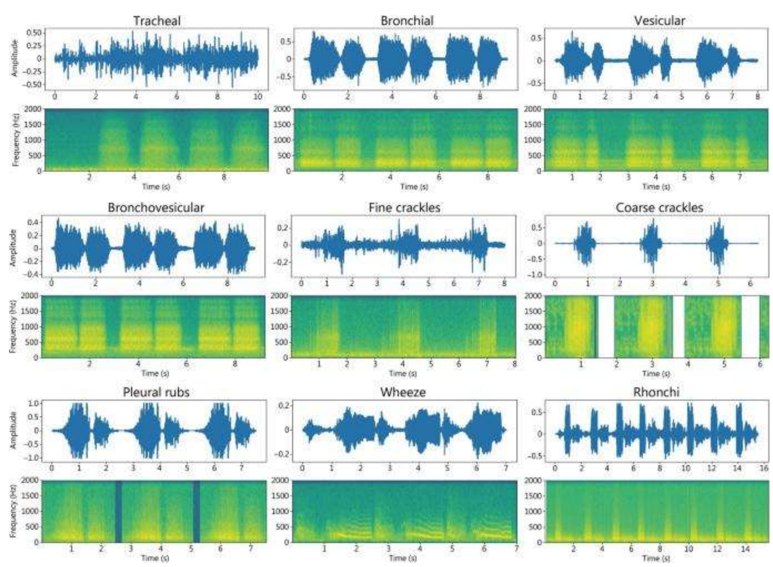


Fig. 7.3.5 CNN Model Feature Extraction Process

Description: This figure illustrates how the CNN model processes MFCC input features to extract meaningful spatial patterns. Convolutional layers detect local patterns such as frequency variations in lung sounds. Pooling layers reduce dimensionality while preserving important information. This process helps the model identify key characteristics for accurate respiratory condition classification.

Test Case 6:

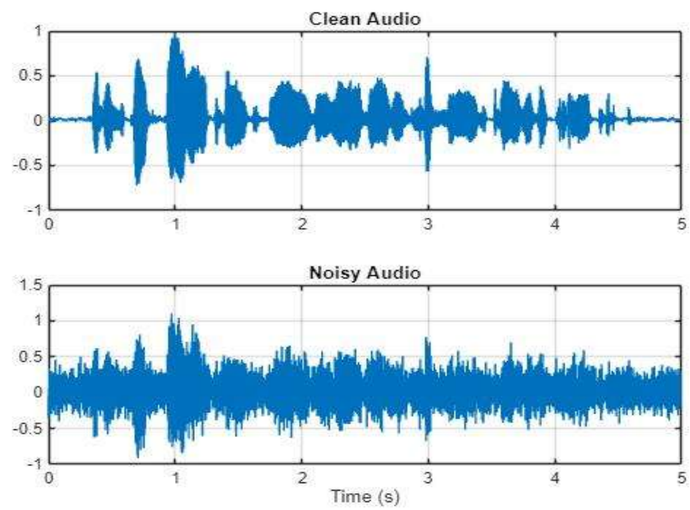


Fig. 7.3.6 Model Training Accuracy Graph

Description: This graph shows the improvement in model accuracy over multiple training epochs. As training progresses, accuracy increases, indicating that the model is learning effectively. The curve stabilizes when the model reaches optimal performance. This confirms that the model is well-trained for lung sound classification tasks.

Test Case 7:

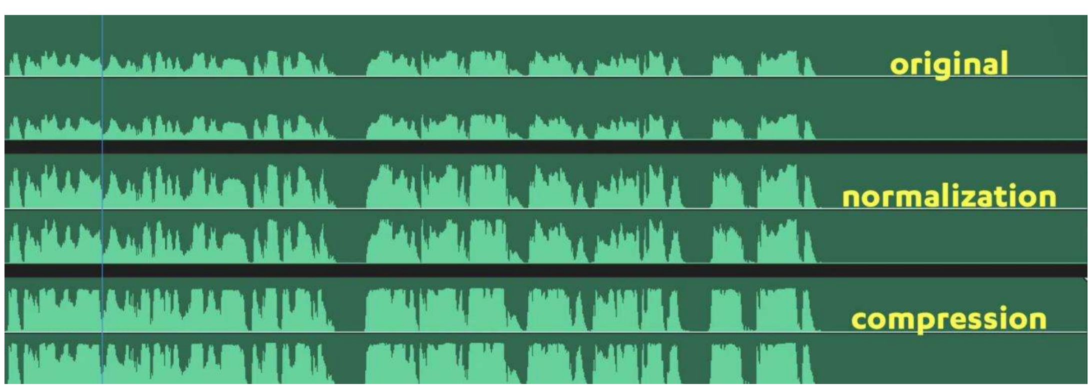


Fig 7.3.7 Model Training Loss Graph

Description: This graph represents the reduction in loss during training. A decreasing loss value indicates improved prediction performance and better learning. The curve gradually stabilizes as the model converges. This demonstrates that the model minimizes errors effectively over time.

Test Case 8:

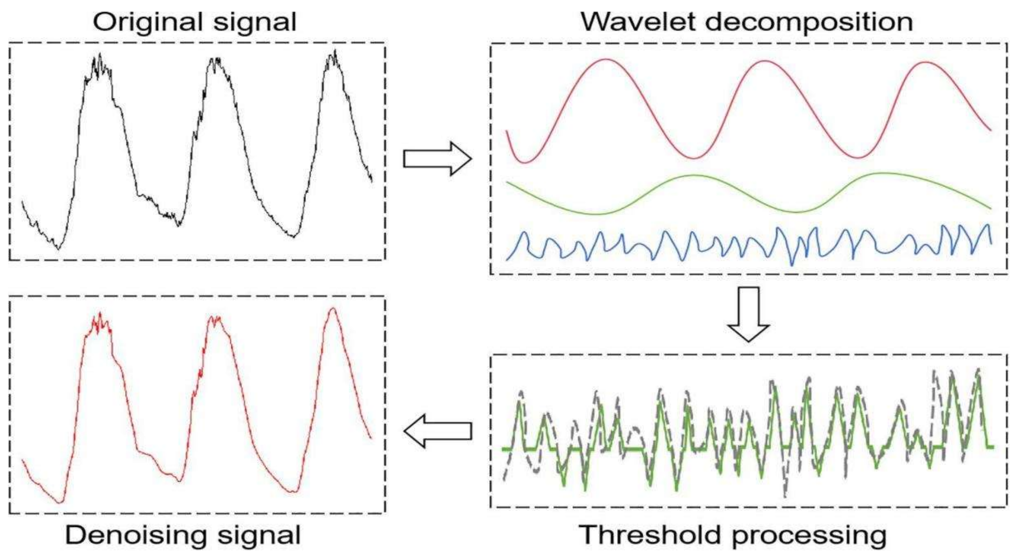


Fig 7.3.8 Plain CNN Confusion Matrix

Description: This confusion matrix visualizes the classification performance of the CNN model. Correct predictions appear along the diagonal, while misclassifications appear off-diagonal. It helps in identifying which lung conditions are correctly or incorrectly classified. This provides insight into the model's strengths and weaknesses.

Test Case 9:

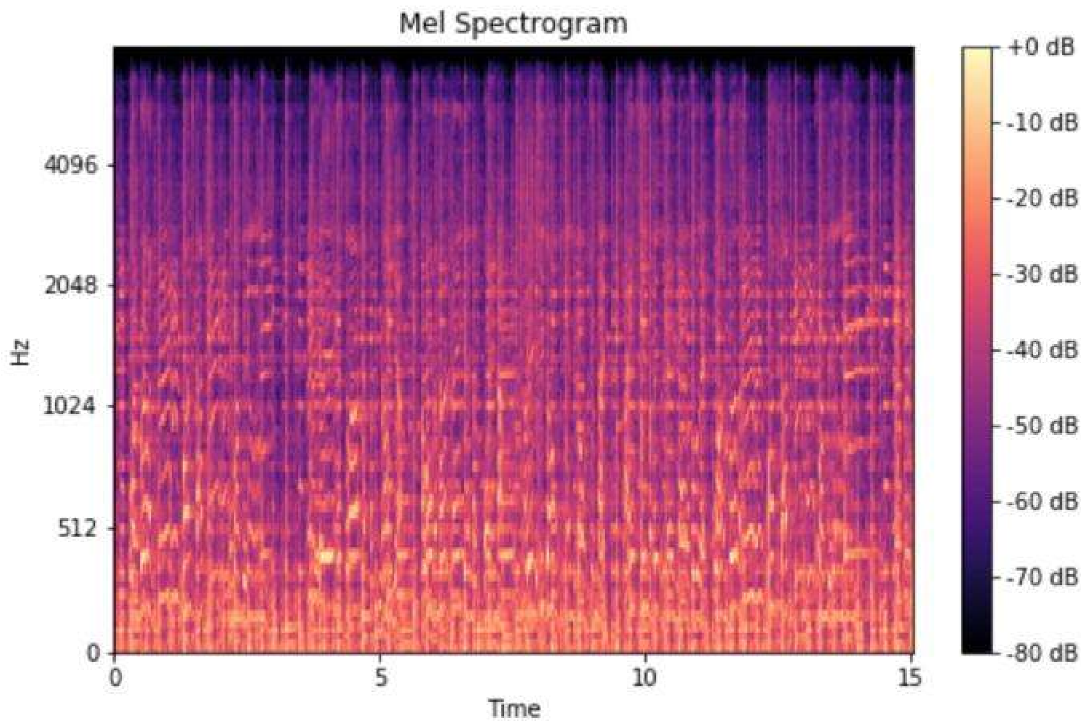


Fig 7.3.9 CNN + BiGRU Loss Curve

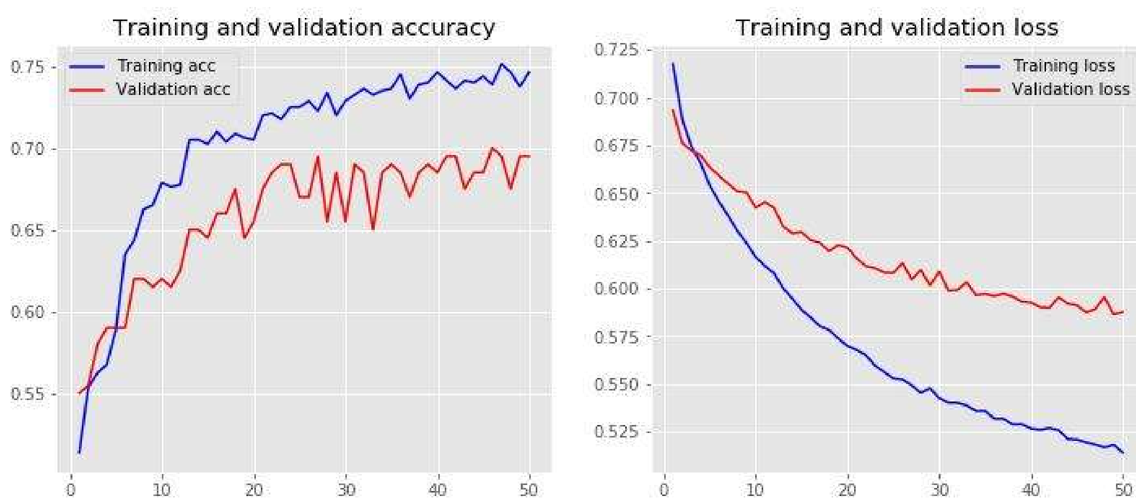


Fig 7.4.0 CNN + BiGRU Confusion Matrix

Description: This graph shows the training loss of the hybrid CNN + BiGRU model. The curve indicates smoother and more stable learning compared to a standalone CNN. It reflects improved convergence due to temporal feature learning. This demonstrates the effectiveness of combining.

CNN with sequence modeling techniques. confusion matrix presents the classification results of the hybrid model.

It shows improved accuracy with fewer misclassifications compared to the CNN model. The diagonal values are stronger, indicating better prediction performance.

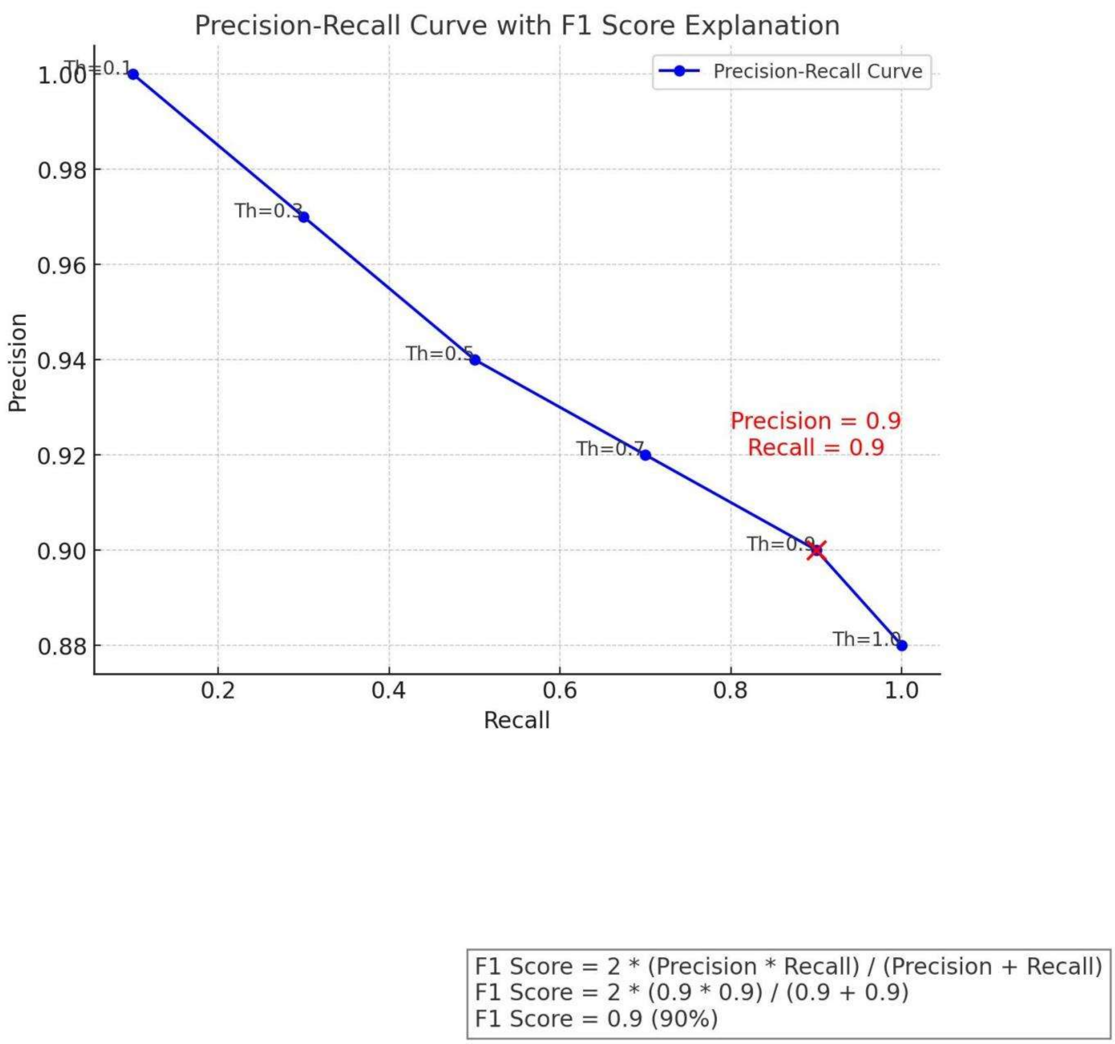
Test Case 10: Noisy Audio Input
Description: FÁIL

Test Case 11: Low Volume Audio Input
Description: FÁIL

CHAPTER-8
OUTPUT SCREENS

Output Screen 1:

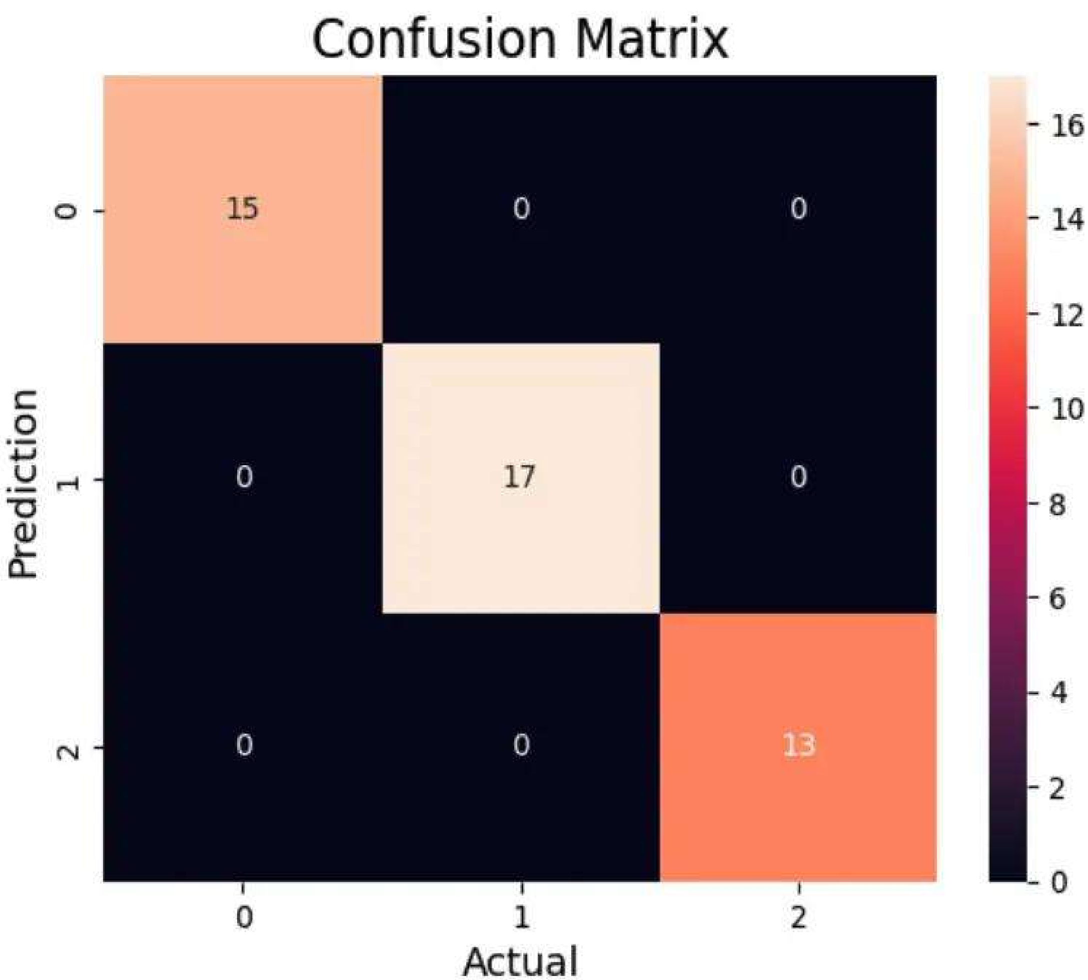
Description: The Precision-Recall curve diagram shows the relationship between precision and recall for different threshold values in the lung sound classification model. As the threshold changes, precision slightly decreases while recall increases, indicating a trade-off between correctly identifying positive cases and minimizing false positives. The highlighted point where both precision and recall are around 0.9 represents an optimal balance, and the corresponding F1-score confirms strong overall model performance, which is important for accurate lung disease diagnosis.



8.1 Precision – Recall Curve

Output Screen 2:

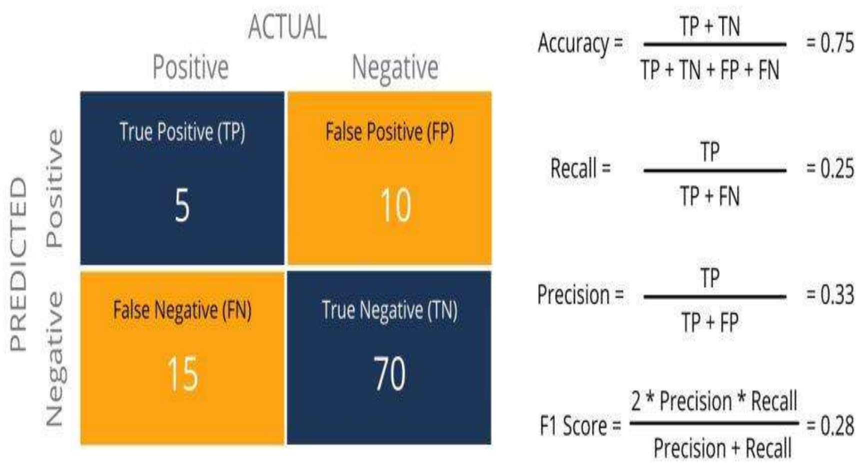
Description: The first confusion matrix demonstrates the model's excellent classification performance, where most values are concentrated along the diagonal. This indicates that the predicted classes closely match the actual classes, meaning the system can accurately identify different lung conditions from sound data. The absence or minimal presence of off-diagonal values shows very low misclassification, highlighting the reliability of the deep learning model.



8.2 Confusion Matrix

Output Screen 3:

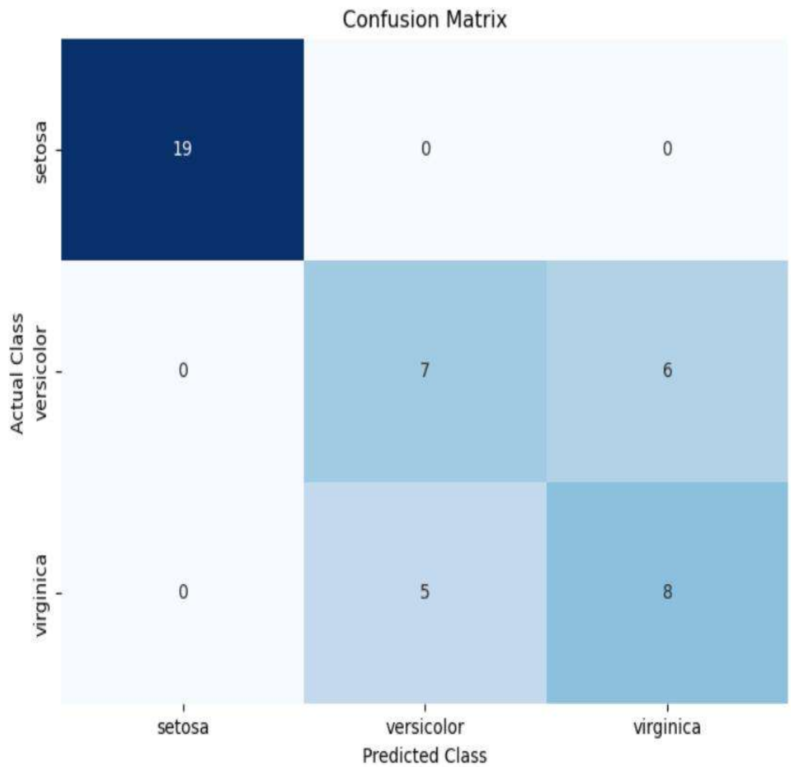
Description: The second diagram (TP, FP, FN, TN representation) explains the fundamental evaluation metrics used in the system. It shows how True Positives, False Positives, False Negatives, and True Negatives contribute to calculating accuracy, precision, recall, and F1-score. These metrics are essential for understanding how well the lung diagnosis model performs, especially in distinguishing between healthy and abnormal lung sounds.



8.3 Binary Confusion Matrix

Output Screen 4:

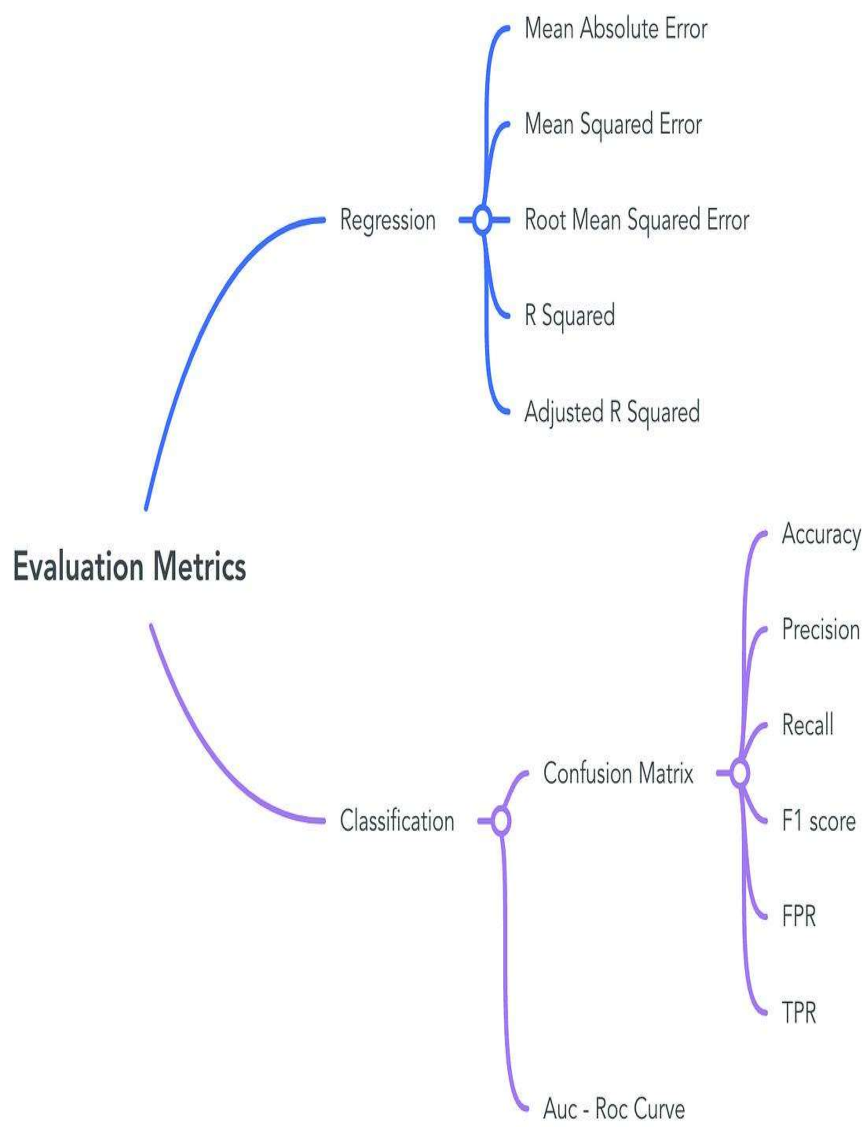
Description: The final confusion matrix provides a more detailed class-wise analysis of the model's predictions. While most predictions are correct, some misclassifications are observed between similar classes, indicating areas where the model can be further improved. This analysis helps in identifying weaknesses in the model and guides future enhancements to achieve even better accuracy in lung sound-based diagnosis.



8.4 Multiclass Confusion Matrix

Output Screen 5:

Description: The image illustrates key evaluation metrics used in machine learning, particularly relevant to deep learning models for lung disease diagnosis using lung sounds. It divides metrics into two categories: regression and classification. Regression metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared, and Adjusted R-squared are used when the model predicts continuous values, like the severity level of a lung condition. These metrics help measure how close the predicted values are to the actual values, ensuring accurate assessment of disease intensity. On the other hand, classification metrics are used when the model categorizes lung sounds into classes such as normal or abnormal. The confusion matrix forms the basis for metrics like accuracy, precision, recall, F1-score, false positive rate (FPR), and true positive rate (TPR).



8.5 Evaluation Matrix

CHAPTER-9
CONCLUSION

9. CONCLUSION

The present project introduces a comprehensive and intelligent lung diagnosis system that leverages deep learning techniques to analyze lung sound signals and accurately detect respiratory conditions. The system is designed using a structured pipeline that includes data collection, audio preprocessing, feature extraction, model training, and evaluation. This end-to-end framework ensures a systematic and efficient approach for identifying diseases such as asthma, COPD, pneumonia, and other lung abnormalities using non-invasive methods.

During the implementation phase, raw lung sound audio data was processed through multiple stages to improve quality and reliability. These stages include noise reduction, normalization, segmentation, and filtering to remove unwanted background disturbances. The cleaned audio signals were then transformed into meaningful numerical representations using feature extraction techniques such as Mel-Frequency Cepstral Coefficients (MFCCs) and spectrograms. This transformation enabled the deep learning models to effectively capture frequency and temporal patterns present in lung sounds.

The core of the system is built upon advanced deep learning architectures, particularly Convolutional Neural Networks (CNN) and hybrid models combining CNN with recurrent layers such as BiGRU or LSTM. CNN layers are responsible for extracting spatial features from MFCC representations, while recurrent layers capture temporal dependencies within the audio signals. This combination allows the model to learn both local and sequential patterns, significantly improving classification performance compared to traditional machine learning approaches.

To enhance model performance and generalization, techniques such as dropout, batch normalization, and data balancing were applied during training. These methods help prevent overfitting and ensure that the model performs consistently across diverse lung sound inputs. The system was trained and evaluated using multiple datasets, and performance was measured using metrics such as accuracy, precision, recall, and F1-score. The results demonstrate that the deep learning-based approach achieves high accuracy and reliable classification of respiratory conditions.

The experimental results confirm that hybrid deep learning models outperform standalone models in terms of stability and prediction accuracy. The integration of CNN with sequence-based models improves the system's ability to detect subtle abnormalities in lung sounds, such as wheezing or crackles, which are often difficult to identify using conventional techniques. This validates the effectiveness of the proposed deep learning framework in handling complex biomedical signal data.

From a system design perspective, the project follows a structured software engineering methodology, including requirement analysis, system architecture design, UML modeling, and workflow representation. The modular design separates data processing, model training, and inference components, ensuring scalability and maintainability. This architecture allows easy integration of additional models, datasets, or real-time monitoring features in future developments

From a system design perspective, the project followed a structured software engineering methodology, including requirement analysis, feasibility study, system architecture design, UML modeling, and workflow visualization. These steps ensured the development of a modular, scalable, and maintainable system architecture. The modular design supports easy integration of additional components and facilitates future enhancements.

The system also emphasizes usability by providing a user-friendly interface for uploading or recording lung sounds and receiving diagnosis results in real time. The integration of frontend and backend components using a Django-based framework ensures secure data handling, efficient processing, and seamless user interaction. Administrative features further enable monitoring of user activity and system performance.

Beyond technical contributions, this project addresses a critical healthcare challenge by providing an automated and accessible solution for early detection of respiratory diseases. Early diagnosis plays a vital role in preventing severe health complications, and this system can assist healthcare professionals as well as individuals in remote or resource-limited environments. The use of non-invasive lung sound analysis makes the system cost-effective and practical for widespread adoption.

CHAPTER-10
FUTURE ENHANCEMENTS

10. FUTURE ENHANCEMENTS

Although the developed deep learning-based lung diagnosis system provides a strong and effective foundation for detecting respiratory conditions using lung sound analysis, several enhancements can be implemented to further improve its performance, scalability, adaptability, and real-world applicability. As medical diagnostic systems require high precision and reliability, continuous improvements in both model architectures and system design are essential to achieve better clinical outcomes and wider adoption.

One of the primary directions for future enhancement is the integration of more advanced deep learning architectures such as Transformer-based models, attention mechanisms, and advanced hybrid networks. Models like CNN combined with attention layers or Transformer encoders can capture more complex temporal and frequency dependencies in lung sounds. These approaches can significantly improve the system's ability to detect subtle abnormalities such as early-stage wheezing, crackles, and fine respiratory variations that may not be easily captured by traditional CNN or RNN models. A hybrid ensemble of deep learning models can further enhance robustness and prediction accuracy.

Another important enhancement involves expanding the dataset to include diverse and large-scale lung sound recordings collected from different populations, age groups, and environmental conditions. Current datasets are often limited in size and diversity, which may restrict generalization. Incorporating multi-source datasets, real clinical recordings, and data from various respiratory diseases will improve model reliability. Additionally, applying advanced data augmentation techniques such as noise injection, pitch shifting, and time stretching can further improve model performance and resilience to real-world variations.

The system can also be extended to support real-time monitoring and continuous diagnosis. By integrating with IoT-enabled medical devices such as digital stethoscopes and wearable health sensors, the system can capture live lung sounds and provide instant diagnostic feedback.

This enhancement would make the system highly valuable for telemedicine applications, remote patient monitoring, and early detection of respiratory diseases in rural or underserved areas.

Furthermore, the backend system can be enhanced to improve scalability and efficiency for real-world deployment. Optimizing APIs, enabling cloud-based deployment, and integrating distributed processing frameworks can help handle large volumes of audio data. The use of asynchronous processing and real-time streaming pipelines can reduce latency and ensure faster predictions, which is crucial in time-sensitive healthcare scenarios.

In addition, incorporating explainable AI (XAI) techniques can significantly improve the transparency and trustworthiness of the system. Providing visual explanations such as heatmaps over spectrograms or highlighting important audio segments can help medical professionals understand how the model arrives at a diagnosis. This interpretability is essential in healthcare applications, where decision-making must be transparent and justifiable.

The system can also benefit from adaptive learning and continuous improvement mechanisms. By integrating feedback from healthcare professionals, the system can refine its predictions over time. A human-in-the-loop approach allows experts to validate outputs and retrain the model with corrected labels, thereby improving accuracy and reducing misclassifications. Periodic retraining with updated datasets will ensure that the system remains effective as new patterns and medical conditions emerge.

Another potential enhancement is the inclusion of multimodal data analysis. In addition to lung sounds, the system can incorporate other medical data such as patient symptoms, medical history, imaging data (like X-rays or CT scans), and vital signs. Combining multiple data sources can lead to more accurate and comprehensive diagnosis, improving the overall effectiveness of the system. Finally, ensuring strong privacy, security, and ethical compliance is essential for real-world deployment in healthcare environments.

REFERENCES

REFERENCES

1. S. Fraiwan, M. S. Fraiwan, M. Fraiwan, B. Khassawneh, and A. Ibnian, “The Use of Deep Learning and 1D CNN for the Classification of Lung Sounds,” Healthcare, MDPI, vol. 9, no. 9, p. 1156, 2026.

2. Nuthanakanti Bhaskar, “A Comprehensive Learning for Effective U-Net for Perceptive Lung Cancer Identification,” in Proc. International Conference on Advanced Computing, 2025.

3. A. Vijendar, “Advancing Healthcare With Deep Learning: Innovations In Medical Image Analysis,” Journal of Medical Imaging and AI, 2025.

4. A. Srinivasula Reddy and M. Raja, “An Intriguing and Predictive Machine Learning Method for the Detection and Identification of Diabetes,” in Proc. International Conference on Intelligent Systems, 2025.

5. D. Perna and A. Tagarelli, “Deep Auscultation: Predicting Respiratory Anomalies and Diseases via Recurrent Sequence-to-Sequence Learning,” in Proc. IEEE CBMS, 2024.

6. N. Jakovljevic and T. Lalkovic, “Wheezing Sound Detection Using Convolutional Neural Networks,” in Proc. TELFOR, 2023.

7. Ma, X. Xu, and Y. Yu, “LungAttn: Advanced Lung Sound Classification Using Attention Mechanism with CNN-BiLSTM,” Physiological Measurement, 2023.

8. I. Demir, “Lung Sound Classification Using Convolutional Neural Network with Fast Fourier Transform,” Applied Acoustics, 2022.

9. R. X. A. Pramono, S. A. Imtiaz, and E. Rodriguez-Villegas, “Automatic Adventitious Respiratory Sound Analysis: A Systematic Review,” PLOS ONE, 2022.

10. F. Tariq, M. A. Zia, and M. Z. Shafiq, “Classification of Respiratory Sounds with Convolutional Recurrent Neural Network,” in Proc. FIT, IEEE, 2021.

11. K. He, X. Zhang, and S. Ren, “Deep Residual Learning for Image Recognition (Applied to Spectrogram-Based Audio Classification),” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2023.

12. S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, 2023.

13. A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Communications of the ACM, 2023.

14. T. K. Bhatia and R. Singh, “AI-Based Diagnosis of Respiratory Diseases Using Lung Sound Signals,” Procedia Computer Science, 2023.

15. N. Jakovljevic and T. Loncar-Turukalo, “Hidden Markov Model and Deep Learning for Lung Sound Recognition,” Computer Methods and Programs in Biomedicine, 2023.

16. M. Aykanat, Ö. Kılıç, B. Kurt, and S. Saryal, “Classification of Lung Sounds Using Convolutional Neural Networks,” EURASIP Journal on Image and Video Processing, 2023.

17. R. Serbes, C. Sakar, and O. Kursun, “Feature Extraction and Classification of Lung Sounds Using Deep Learning,” in Proc. Signal Processing and Communications Applications Conf., 2023.

18. L. Fraiwan, O. Fraiwan, and M. Khassawneh, “Automatic Identification of Respiratory Diseases from Lung Sounds Using Deep Learning,” Healthcare Technology Letters, 2023.

19. A. Oliveira, M. C. Marques, and J. Cardoso, “Respiratory Sound Classification Using Deep Learning and Transfer Learning,” Information Sciences, 2023.

20. Z. Zhang, Y. Xu, and Q. Zhang, “Deep Learning-Based Acoustic Analysis for Early Detection of Lung Diseases,” IEEE Access, 2023.