DEEP NEURAL NETWORKS FOR RECOGNIZE AND DIAGNOSE COUGHS

***Dissertation submitted to***

***Shri Ramdeobaba College of Engineering & Management, Nagpur in partial fulfillment of the requirement for the award of a degree of***

**Bachelor of Technology (B.Tech)**

In

**COMPUTER SCIENCE AND ENGINEERING**

*By*

### Aastha Singh (B-01) Dhanshree Dharpure (B-03) Girija Chachada (B-04) Pranav Darak (B-55)

*Of*

**VI Semester**

*Guide*

**Dr. Swati Hira**



## Department of Computer Science and Engineering

**Shri Ramdeobaba College of Engineering & Management, Nagpur 440 013** (An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur) **April 2024**

DEEP NEURAL NETWORKS FOR RECOGNIZE AND DIAGNOSE COUGHS

***Dissertation submitted to***

***Shri Ramdeobaba College of Engineering & Management, Nagpur in partial fulfillment of the requirement for the award of a degree of***

**Bachelor of Technology (B.Tech)**

In

**COMPUTER SCIENCE AND ENGINEERING**

*By*

### Aastha Singh (B-01) Dhanshree Dharpure (B-03) Girija Chachada (B-04) Pranav Darak (B-55)

*Of*

**VI Semester**

*Guide*

**Dr. Swati Hira**



## Department of Computer Science and Engineering

**Shri Ramdeobaba College of Engineering & Management, Nagpur 440 013** (An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur) **April 2024**

ii

**SHRI RAMDEOBABA COLLEGE OF ENGINEERING MANAGEMENT, NAGPUR**

(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur)

**Department of Computer Science and Engineering**

**CERTIFICATE**

This is to certify that the Thesis on **“Deep Neural Networks for Recognise and Diagnose Coughs”** is a Bonafide work of

1. Aastha Singh
2. Dhanashree Dharpure
3. Girija Chachada
4. Pranav Darak

, submitted to the Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur in partial fulfillment of the award of a Degree of Bachelor of Technology (B.Tech), in Computer Science and Engineering. It has been carried out at the Department of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur during the academic year 2023-2024.

Date:

Place: Nagpur

Dr. Swati Hira Dr. R. Hablani

Project Guide H.O.D

Department of Computer Science Department of Computer Science and Engineering and Engineering

Dr. R. S. Pande Principal

**DECLARATION**

We hereby declare that the thesis titled “**Deep Neural Networks for Recognise and Diagnose Coughs**” submitted herein, has been carried out in the Department of Computer Science and Engineering of Shri Ramdeobaba College of Engineering and Management, Nagpur. The work is original and has not been submitted earlier as a whole or part for the award of any degree/diploma at this or any other institution / University.

Date:

Place: Nagpur

Aastha Singh B-01 Dhanshree Dharpure B-03 Girija Chachada B-04 Pranav Darak B-55

**APPROVAL SHEET**

This report entitled “**Deep Neural Networks for Recognise and Diagnose Coughs**” by

1. Aastha Singh
2. Dhanshree Dharpure
3. Girija Chachada
4. Pranav Darak

is approved for the degree of Bachelor of Technology (B.Tech).

Dr. Swati Hira

Project Guide External Examiner

Dr. R. Hablani H.O.D, CSE

Date:

Place: Nagpur

**ACKNOWLEDGEMENTS**

We would like to express our sincere gratitude to all those who have contributed to the completion of this report. We would like to extend a special appreciation to our project guide, Dr. Swati Hira, for their invaluable suggestions and encouragement throughout the project.

We are thankful to Shri Ramdeobaba College of Engineering and Management, Nagpur for providing us with this great opportunity to be working on this project.

We take this opportunity to acknowledge our profound indebtedness and extend our deep sense of gratitude to Dr. R. Hablani, Head of Department, for his support and guidance for our project.

Lastly, we would also like to express our appreciation for the guidance provided by our other supervisor and the panel members, particularly during our project presentation. Their constructive comments and advice have greatly enhanced our presentation skills and provided insights into improving our project.

# ABSTRACT

Respiratory infections pose a significant global health burden, with cough being a prominent symptom across various illnesses. Addressing this challenge, our project endeavors to revolutionize respiratory disease screening with an innovative automated system leveraging Convolutional Neural Networks (CNNs).

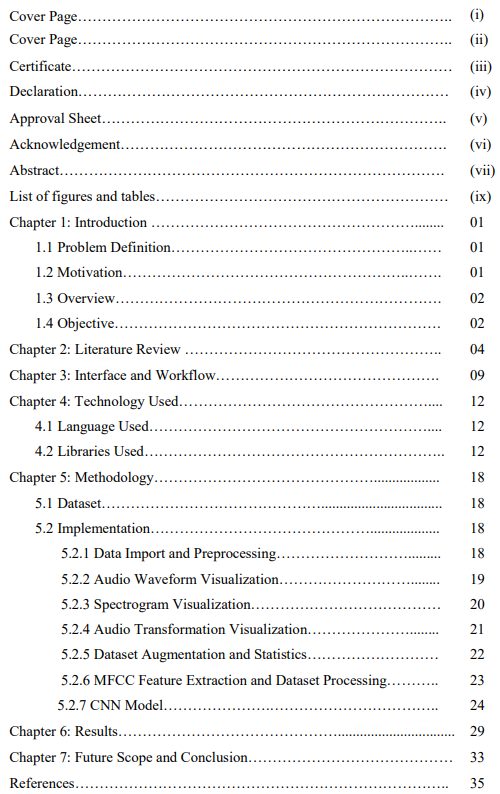
In today's fast-paced world, where health is paramount, early detection and diagnosis of respiratory illnesses are pivotal. Our project's focus is to develop an advanced tool capable of analyzing raw cough data to screen for four prevalent respiratory infections: bronchitis, asthma, pneumonia and pertussis.

Drawing inspiration from the success of CNNs in pattern recognition, our system extracts Mel- Frequency Cepstral Coefficients (MFCC) spectrograms from cough audio, enabling precise identification of respiratory symptoms. By meticulously training our model on a diverse dataset augmented with noise addition and time stretching, we ensure robustness and adaptability to real- world scenarios.

The architecture of our CNN model is meticulously crafted, comprising convolutional layers for spatial feature extraction, max-pooling layers for dimensionality reduction, and fully connected dense layers for classification. Incorporating dropout regularization mitigates overfitting, ensuring reliable performance.

Beyond mere detection, Our system marks a new time in respiratory healthcare by detecting problems early and creating treatment plans just for you. Through seamless integration into medical facilities and research institutions, our automated tool empowers healthcare professionals with timely insights, leading to improved patient outcomes and reduced healthcare costs. As we navigate through an era where respiratory health takes center stage, our project stands at the forefront, driving innovation and transforming healthcare delivery. Embracing the power of AI and CNNs, we pave the way for a future where respiratory diseases are no longer a silent threat but a conquerable challenge.

# TABLE OF CONTENTS

****

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **List of Figures** | **Pg. no.** |
| Figure 1: Interface for recognizing and diagnosis cough | 08 |
| Figure 2: Complete workflow of the model | 09 |
| Figure 3: Waveforms of diseases | 19 |
| Figure 4: Mel Spectrogram of Diseases | 20 |
| Figure 5: Waveforms after Transformation | 21 |
| Figure 6: Accuracy v/s Epoch and Loss v/s Epoch | 26 |
| Figure 7: Confusion matrix | 27 |
| Figure 8: Prediction Result for Pneumonia | 28 |
| Figure 9: Prediction Result for Bronchitis | 29 |
| Figure 10: Prediction Result for Pertussis | 30 |
| Figure 10: Prediction Result for Asthma | 31 |

# LIST OF TABLES

|  |  |
| --- | --- |
| **List of Tables** | **Pg. no.** |
| Table 1: Original Dataset | 17 |
| Table 2: Dataset after Augmentation | 22 |
| Table 3: Evaluation Metrics and Test Results | 27 |

**CHAPTER 1: INTRODUCTION**

## Problem Definition:

One of the notable modern medical concerns that impose an immense worldwide health burden are respiratory infections. Since cough is an essential symptom of many respiratory infections, an automated system to screen for respiratory diseases based on raw cough data would have a multitude of beneficial research and medical applications. Develop an automated recognition and diagnostic tool for screening respiratory infections that utilizes Convolutional Neural Networks (CNNs) to detect cough within environment audio and diagnose four potential illnesses (i.e., bronchitis, pneumonia, asthma, and pertussis) based on their unique cough audio features.

Each illness has unique cough audio signatures:

**Bronchitis**: Coughs in bronchitis patients tend to be frequent and productive, often accompanied by wheezing or rattling sounds due to inflammation of the bronchial tubes.

**Pneumonia**: Pneumonia, caused by bacteria or viruses, inflames the lungs, leading to fever, chills, chest pain, and coughing with mucus. Severe cases may cause wheezing or gasping, needing swift diagnosis and treatment to prevent complications.

**Asthma**: Asthma inflames and narrows airways, causing wheezing, shortness of breath, chest tightness, and coughing. Coughs are often dry and persistent, with wheezing. Severe cases show increased coughing, indicating worsening distress and highlighting early management importance.

**Pertussis (Whooping Cough):** Pertussis coughs typically have a characteristic "whooping" sound due to rapid inhalation following a coughing fit. The cough may be accompanied by vomiting and exhaustion.

## Motivation:

The prevalence of respiratory infections underscores the urgency to develop automated systems capable of accurately screening and diagnosing these illnesses. Manual symptom assessments are prone to variations and delays, hindering timely interventions and increasing

1

the risk of disease transmission. By leveraging advanced technologies such as Convolutional Neural Networks (CNNs) and deep learning, this project seeks to revolutionize respiratory disease detection by automating the analysis of raw cough audio data.

**Pneumonia**: In 2020, there were 1,81,160 deaths due to respiratory diseases like pneumonia in India, which was higher than the 1,52,311 deaths reported in 2019.

**Asthma**: A study by the Global Asthma Network in India revealed a prevalence of wheeze in the previous 12 months at 3.16% for children aged 6-7 years, 3.63% for children aged 13-14 years, and 3.30% for parents.

**Pertussis:** Research indicated an increased risk in patients with asthma, with 38% of pertussis cases having asthma before the index date compared to 26% in control subjects.

**Bronchitis**: Chronic bronchitis, less common but more severe, affected 9.3 million people in the United States in 2018. There is no specific statistic provided for bronchitis in India in the given data.

By developing a sophisticated CNN-based model trained on diverse cough sound features associated with different respiratory illnesses, this project aims to enhance accuracy, efficiency, and scalability in respiratory infection identification. The motivation behind this project lies in addressing the critical need for reliable, efficient tools that can streamline healthcare workflows, reduce errors, and improve patient outcomes in the context of respiratory disease management.

## Overview:

The project "Automated Recognition and Diagnosis of Respiratory Infections Using Convolutional Neural Networks (CNNs)" aims to develop an innovative technology solution for the automated screening and diagnosis of respiratory infections based on raw cough audio data. By harnessing the power of CNNs, the project endeavors to revolutionize the field of respiratory disease detection, offering a multitude of benefits for medical research and clinical practice. Through the integration of cutting-edge technologies and innovative approaches, the project seeks to improve respiratory disease detection and diagnosis, ultimately leading to better health outcomes for patients and communities.

## Objective (Proposed Plan of Work):

The proposed plan of work for this project includes the following key objectives:

* + 1. Develop a comprehensive dataset of raw cough audio data representing a diverse range of respiratory infections.
    2. Preprocess the raw cough audio data to extract relevant features and prepare it for input into the CNN-based model.
    3. Design and implement a CNN architecture optimized for the classification of respiratory infections based on cough sound features.
    4. Train the CNN model using the prepared dataset to accurately classify different respiratory infections.
    5. Evaluate the performance of the trained CNN model using appropriate metrics and validate its effectiveness in real-world scenarios.
    6. Iterate on the model design and training process to further improve accuracy, efficiency, and scalability.
    7. Document and disseminate the findings of the project through research publications and presentations, contributing to the advancement of respiratory medicine and healthcare technology.

# CHAPTER 2: LITERATURE REVIEW

We studied multiple papers to understand the underlying workings of convolutional neural networks. Some referred papers are mentioned below with a brief discussion of their studies:

1. **Automatic diagnosis of COVID-19 disease using a deep convolutional neural network with multi-feature channels from respiratory sound data: Cough, voice, and breath.**

The Department of Computer Applications at NIT Tiruchirappalli, India, has spearheaded a groundbreaking project aimed at enhancing COVID-19 diagnosis through the analysis of respiratory sounds. Leveraging Artificial Intelligence (AI) and deep learning techniques, the project implements a multi-channeled Deep Convolutional Neural Network (DCNN) architecture. This innovative approach encompasses the preprocessing of a diverse dataset, feature extraction using methods like Gamma-tone Frequency Cepstral Coefficients (GFCC) and Improved Multi-frequency Cepstral Coefficients (IMFCC), and the application of data augmentation techniques. Through rigorous training and optimization, the DCNN model achieves an outstanding 95.45% accuracy in diagnosing COVID-19 from respiratory sounds, surpassing conventional diagnostic methods.

Respiratory sound classification has emerged as a critical area of focus for medical researchers and clinical scientists, particularly in the context of COVID-19 diagnosis. The project's contribution lies in its development of a robust and accurate diagnostic tool that utilizes AI-based models to identify COVID-19 from human- generated sounds such as voice, cough, and breath. By integrating multiple feature channels and employing advanced deep learning techniques, including data denoising and augmentation, the proposed approach significantly enhances the performance and accuracy of COVID-19 diagnosis. This achievement not only underscores the potential of AI in healthcare but also offers a tangible solution to the pressing need for rapid and reliable COVID-19 detection, thereby contributing to global efforts in combating the pandemic.

1. **Cough Sound Detection and Diagnosis Using Artificial Intelligence Techniques.**

Respiratory diseases pose a significant global health challenge, particularly in regions with limited access to healthcare resources. Cough, a common symptom across various respiratory ailments, serves as a valuable indicator for disease detection. The Department of Computer Science at the College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Saudi Arabia, has embarked on a pioneering study to harness Artificial Intelligence (AI) and Machine Learning (ML) techniques for efficient respiratory disease diagnosis. By prioritizing accessible mobile and recorder implementations, the methodology aims to deliver affordable and scalable solutions for global healthcare. Through the analysis of cough sounds, AI algorithms offer a promising avenue for early disease detection, potentially mitigating the impact of respiratory infections on public health and economic well-being.

The study employs a combination of AI and ML models to detect and diagnose respiratory diseases based on cough sounds. A Convolutional Neural Network (CNN) model achieves impressive sensitivity and specificity rates of 98.5% and 94.2%, respectively, in COVID-19 detection, demonstrating high accuracy. Additionally, an SVM-based model reports an AUC of 0.79 with 68% sensitivity and 82% specificity, showcasing effective diagnostic capabilities across various respiratory conditions. By analyzing cough sounds, the AI algorithms reliably identify diseases such as pneumonia, pulmonary edema, asthma, tuberculosis (TB), pertussis, and COVID-19. This approach not only aids in early detection but also ensures timely intervention, thus mitigating the impact of respiratory infections on public health and economic well-being.

## Classification of lung sounds using convolutional neural networks.

With the integration of computer systems capable of handling vast datasets, the medical field is witnessing a surge in non-invasive diagnostic methods. Aiming to contribute to this trend, the Faculty of Engineering at Yıldırım Beyazıt University, Ankara, Turkey, embarked on a

study to develop a non-invasive classification method for respiratory sounds. Leveraging a cost-effective electronic stethoscope and audio recording software, the study recorded an extensive dataset comprising 17,930 lung sounds from 1630 subjects. The methodology employed two machine learning approaches: mel frequency cepstral coefficient (MFCC) features with support vector machine (SVM) and spectrogram images with convolutional neural network (CNN). This approach aimed to benchmark the efficacy of CNN against the widely accepted SVM algorithm, particularly in diverse respiratory sound classifications.

The study prepared four datasets for each classification task, ranging from healthy versus pathological to singular respiratory sound type classification. Comparative analysis between CNN and SVM algorithms revealed comparable accuracy rates across various classifications. For instance, in the healthy versus pathological classification task, both CNN and SVM achieved an accuracy of 86%. Similarly, in tasks such as rale, rhonchus, and normal sound classification, CNN and SVM attained accuracies of 76% and 75%, respectively. Notably, spectrogram image classification with CNN demonstrated effectiveness comparable to SVM, indicating accurate pre-diagnosis of respiratory audio. These results underscore the potential of CNN and SVM algorithms in accurately classifying and pre-diagnosing respiratory audio, leveraging extensive datasets for enhanced diagnostic capabilities.

## Cough Detection System Using Machine Learning Technique.

Coughing, while often a natural reflex to clear irritants from the respiratory tract, can indicate underlying health issues if persistent or severe. Recognizing the significance of timely detection, this research presents a robust cough detection system leveraging machine learning techniques. Developed by the Innovative Electromobility (ITEM) Research Lab at the College of Engineering, UiTM, the system comprises five modules: audio sampling, feature extraction, model training, cough detection, and real-time analysis, facilitated by a Modular and Open System (MOST). Through rigorous testing, we have achieved an impressive accuracy rate of 97.5% in distinguishing coughs from other ambient sounds, underscoring the reliability and precision of our innovative approach. This innovative system promises to transform healthcare diagnostics, providing an efficient way to detect respiratory illnesses like COVID-19. Beyond cough detection, its potential spans various respiratory conditions, signaling a significant advancement in proactive healthcare.

While showcasing promising results, ongoing data refinement is imperative for enhanced performance and broader applicability. Future endeavors aim to augment the system for detecting COVID-19 infections, underscoring its potential in advancing healthcare diagnostics.

Coughing serves as a vital bodily response to clear air passages, yet prolonged or intense coughing can signify underlying health concerns warranting medical attention. Addressing the critical need for timely detection, this study introduces a novel cough detection system empowered by machine learning. Developed by the Innovative Electromobility (ITEM) Research Lab at the College of Engineering, UiTM, the system integrates five distinct modules: audio sampling, feature extraction, model training, cough detection, and real-time analysis. Leveraging an Artificial Neural Network (ANN) model trained on a comprehensive cough sound dataset, the system effectively analyzes audio signals to identify instances of coughing. Real-time testing, facilitated through an embedded controller within the Modular and Open System (MOST), validates the system's precision, achieving an impressive 97.5% accuracy in discerning coughs from other sounds. This innovative approach holds promise for revolutionizing healthcare diagnostics, with potential applications extending to the detection of respiratory illnesses such as COVID-19, marking a significant advancement in proactive healthcare management.

## A classification framework for identifying bronchitis and pneumonia in children based on a small-scale cough sounds dataset.

Bronchitis and pneumonia represent prevalent respiratory diseases in children, with pneumonia notably posing a significant threat to pediatric mortality globally. This study introduces a novel Classification Framework based on Cough Sounds (CFCS) aimed at accurately identifying bronchitis and pneumonia in pediatric patients. Leveraging a dataset comprising cough sounds from 173 outpatients at the West China Second University Hospital, Sichuan University, Chengdu, China, the framework employs an aggregation operation to extract disease features from cough chunks. In the classification stage, a Support Vector Machine (SVM) is initially utilized due to the dataset's limited scale, augmented

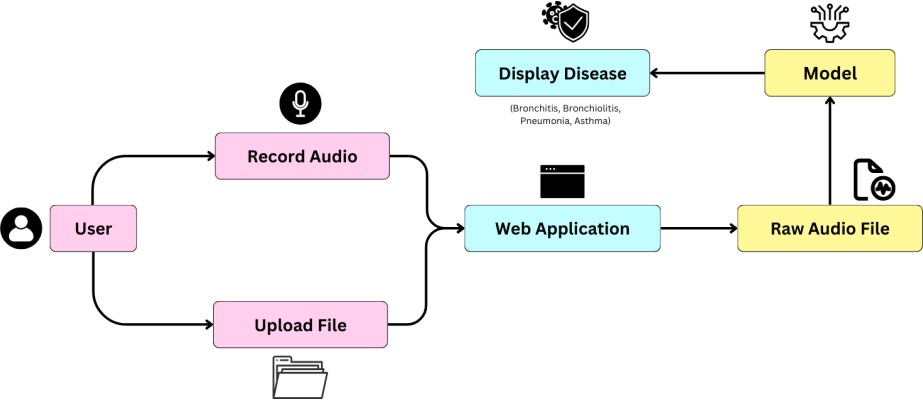
through data augmentation techniques. Additionally, a Long Short-Term Memory Network (LSTM) is employed for classification after data augmentation. Experimental results showcase SVM achieving the highest classification accuracy of 86.04%, with a standard deviation of 4.7%. The precision and recall rates for bronchitis and pneumonia range from 87.5% to 93.75% and 88.24% to 93.33%, respectively. Furthermore, the AUC values for SVM and LSTM classification models on augmented datasets reach

0.92 and 0.93, respectively, affirming the effectiveness of CFCS in accurately classifying pediatric respiratory diseases.

Bronchitis and pneumonia present significant challenges in pediatric healthcare, with pneumonia being a leading cause of mortality among children globally. Addressing the critical need for accurate diagnosis, this paper proposes a novel Classification Framework based on Cough Sounds (CFCS) to effectively classify bronchitis and pneumonia in pediatric patients. By analyzing cough sounds from a dataset sourced from the West China Second University Hospital, Sichuan University, Chengdu, China, CFCS employs an aggregation operation to extract disease features, effectively mitigating the influence of irrelevant cough segments. Through a staged classification process, initially utilizing a Support Vector Machine (SVM) and later integrating a Long Short-Term Memory Network (LSTM) after data augmentation, CFCS demonstrates robust classification capabilities. Experimental results underscore CFCS's efficacy, with SVM achieving a peak classification accuracy of 86.04%, and LSTM further enhancing classification performance post-augmentation. These findings highlight CFCS's potential as a reliable tool for pediatric bronchitis and pneumonia detection, paving the way for improved healthcare management in pediatric respiratory diseases. Future work aims to refine CFCS and explore its application in diverse healthcare scenarios.

# CHAPTER 3: INTERFACE AND WORKFLOW

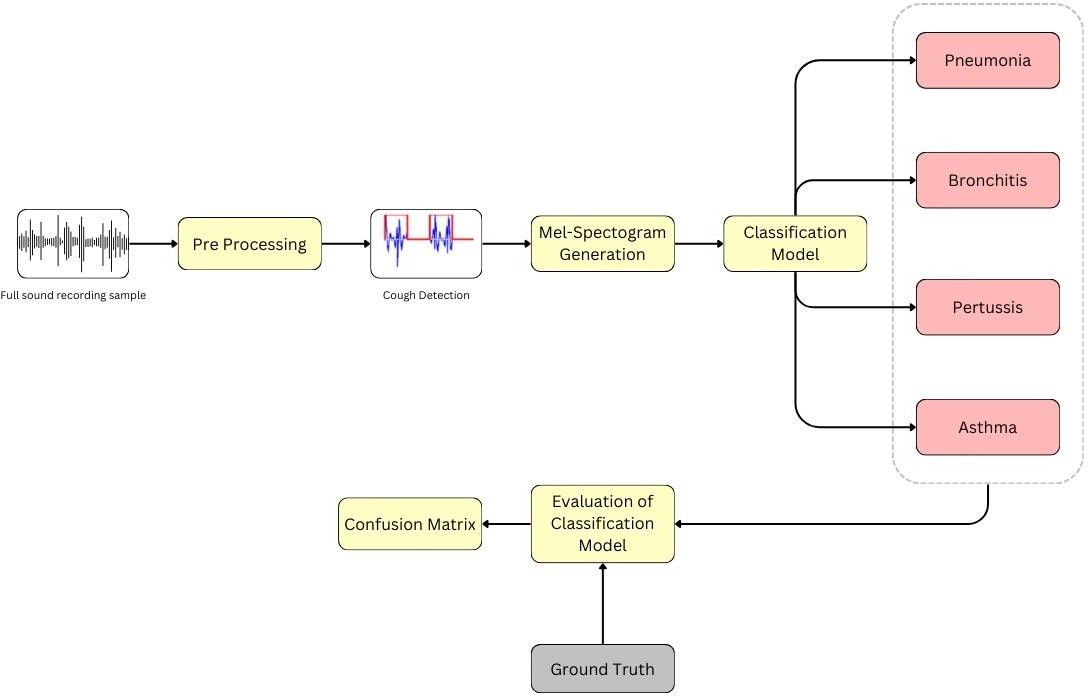
## Interface



**Figure 1: Interface for recognizing and diagnosis cough**

Using our web application for cough recognition and diagnosis is simple and intuitive. Users can begin by navigating to the application's interface, where they are presented with options to either record a cough sound in real time or upload a pre-recorded audio file. Upon selecting their preferred method, users can follow the on-screen instructions to complete the submission process. Once the cough sound data is received by the application, it is swiftly processed and analyzed by our deep neural network model. Within moments, users receive feedback in the form of a displayed diagnosis, detailing any potential respiratory conditions associated with the submitted cough sound. Additionally, users may have the option to view further information or recommendations based on the diagnosis provided. Our user-friendly interface and streamlined process ensure that individuals can easily access valuable insights into their respiratory health with just a few clicks or taps.

## Workflow



**Figure 2: Complete workflow of the model**

Our project unfolds through a meticulously orchestrated workflow, carefully designed to navigate the complexities of respiratory infection screening via cough analysis. Each stage in the process is imbued with purpose, aiming to unravel the mysteries hidden within raw audio data and unveil the respiratory conditions lurking beneath.

* **Full Sound Recording Sample**: The commencement of our project initiates with the procurement of unprocessed audio recordings, resonating with a variety of cough cadences. These recordings, obtained from individuals representing a range of respiratory conditions such as bronchiolitis, bronchitis, pertussis, and normal respiratory functions, serve as the fundamental elements of our dataset, encapsulating the broad spectrum of respiratory distress manifestations.
* **Pre-Processing**: During the initial pre-processing phase, our approach involves careful refinement similar to that of a sculptor refining their sculpture. The goal is to remove imperfections and reveal the essence of our creation. In this stage, we refine raw audio data by removing unnecessary noise, adjusting amplitude, and isolating cough sounds from environmental variables.
* **Mel-Spectrogram Generation**: In our continued exploration of audio analysis, we employ the Mel-Spectrogram, a powerful tool that captures the essence of sound by mapping its frequency and time components. This intricate technique provides a detailed representation of audio signals, allowing us to extract valuable insights for our model's interpretation. By leveraging the Mel-Spectrogram, we gain a deeper understanding of the underlying structure and characteristics of the audio data, facilitating more accurate analysis and decision-making processes.
* **Feature Extraction and Model Training**: Guided by the architectural marvel of our CNN model, we embark on the extraction of intricate features from the Mel- Spectrograms, each a testament to the spectral richness of cough sounds. With a total of 97,414,148 trainable parameters, our model stands poised to learn the subtle nuances of respiratory conditions. Through 15 epochs of training, with a batch size of 32 and the Adam optimizer at our helm, we traverse the landscape of audio data, sculpting our model into a beacon of diagnostic prowess.
* **Cough Detection and Evaluation**: Armed with our trained model, we embark on a voyage of discovery—a quest to detect the telltale signs of respiratory distress hidden within the cough sounds. As our model navigates this sonic terrain, we evaluate its performance with keen scrutiny, measuring its efficacy against the backdrop of test loss and accuracy. Model 1, with a test loss of 0.7592 and a test accuracy of 0.8714, showcases the fruits of our labor, while Model 2, with a test loss of 0.3260 and a test accuracy of 0.8571, stands as a testament to our dedication to excellence.
* **Ground Truth Comparison**: During validation, we compare our model's predictions to the confirmed truths of respiratory diagnosis. This helps us make sure our model stays accurate and reliable.

# CHAPTER 4: TECHNOLOGY USED

## Programming Languages Used:

**Python**

Python is a versatile and widely used programming language that has become popular for developing neural networks and other machine learning models. Its simplicity, readability, and extensive ecosystem of libraries and frameworks make it an ideal choice for implementing neural networks. Neural networks are computational models inspired by the human brain's structure and functioning. They consist of interconnected layers of artificial neurons, also known as nodes or units. These networks are designed to learn and extract patterns from input data by adjusting the weights and biases of the connections between neurons. Python's libraries and frameworks, such as TensorFlow, PyTorch, and Keras, provide high-level abstractions and APIs that handle much of the underlying theory and implementation details. These libraries enable efficient neural network development and training, allowing practitioners to focus on model design and experimentation. Understanding the theoretical concepts underlying neural networks empowers Python users to make informed decisions and effectively utilize these libraries to build powerful machine learning models.

## Libraries Used:

1. **TensorFlow**

TensorFlow is an open-source software library widely used for machine learning and deep learning tasks. Developed by the Google Brain team, TensorFlow provides a flexible and efficient framework for building and deploying various types of artificial intelligence models. The core component of TensorFlow is its computational graph, which represents the flow of operations and data dependencies in a machine learning model. The graph consists of nodes that represent mathematical operations and edges that denote the data arrays, called tensors, flowing between these operations. This graph-based approach enables 10 TensorFlow to efficiently distribute computations across multiple CPUs or GPUs, allowing for high-performance training and inference on large-scale datasets.

Key features of TensorFlow include:

* 1. **Versatility:** TensorFlow supports a broad range of machine learning tasks, including classification, regression, clustering, and neural network-based tasks such as image recognition, natural language processing, and speech recognition.
  2. **Flexibility:** TensorFlow provides a flexible programming interface that allows users to define and train custom models. It supports both high-level APIs, such as Keras, which simplifies the model creation process, as well as lower-level APIs that offer finer control over the model architecture and training process.
  3. **Scalability:** TensorFlow is designed to scale from running on a single machine to distributed computing environments. It can efficiently distribute computations across multiple devices or machines, making it suitable for training large neural networks on vast amounts of data.
  4. **TensorBoard:** TensorFlow includes a visualization tool called TensorBoard, which helps users analyze and monitor the training process. It allows for the inspection of model architectures, the visualization of training curves, and the exploration of intermediate representations in the network.
  5. **Deployment:** TensorFlow supports various deployment scenarios, including integration with production environments, mobile devices, and embedded systems. It provides tools and libraries for converting trained models into formats that can be deployed in different environments.
  6. **Community and Ecosystem:** TensorFlow benefits from a large and active community of developers, researchers, and enthusiasts. This vibrant ecosystem contributes to the development of new features, provides support through forums and documentation, and offers numerous pre-trained models and resources for accelerating development.

Overall, TensorFlow is a powerful and widely adopted library for building and deploying machine learning models. It combines flexibility, scalability, and a rich ecosystem, 11 making it suitable for both research and production use cases in the field of artificial intelligence.

## Keras

Keras is an open-source neural network library written in Python. It is designed to provide a user- friendly interface for building, training, and deploying deep learning models. Keras allows developers to create complex neural network architectures with minimal code, making it suitable for both beginners and experienced machine learning practitioners.

Key features of Keras include:

* 1. **TensorBoard:** In our project, we utilized Keras' Sequential API to construct the convolutional neural network (CNN) model. The Sequential API enables us to create and configure layers in a sequential manner, starting from the input layer and progressing through hidden layers to the output layer. Each layer is added to the model using Keras' layer functions, such as Conv2D, MaxPooling2D, Flatten, and Dense.
  2. **Layer Creation and Configuration:** The Conv2D layer is used to create convolutional layers that perform operations such as feature extraction. We configured parameters such as the number of filters, kernel size, and activation function to customize the behavior of these layers. Additionally, MaxPooling2D layers were employed to downsample the spatial dimensions of the input data, reducing computational complexity. The Flatten layer was used to flatten the input data into a one-dimensional array, while Dense layers created fully connected layers with configurable activation functions.
  3. **Model Compilation:** After defining the model architecture, we compiled it using the compile() method. During compilation, we specified the loss function, optimizer, and evaluation metrics. The loss function measures the model's performance on the training data, while the optimizer updates the model's weights based on the computed loss. Evaluation metrics, such as accuracy, were chosen to assess the model's performance during training and testing.
  4. **Model Training:** The fit() method was used to train the compiled model on the training data. This method iteratively updates the model's weights using backpropagation to minimize the loss function. We provided parameters such as batch size, number of epochs, and validation data to customize the training process. Additionally, callbacks such as ReduceLROnPlateau and EarlyStopping were employed to enhance training efficiency and prevent overfitting.

Overall, Keras provided a convenient and efficient framework for implementing the CNN model in our project. Its Sequential API allowed us to easily define the model architecture and configure layers, while its compile() and fit() methods facilitated model compilation and training. By

leveraging Keras' capabilities, we were able to build a robust CNN model for our classification task with minimal effort.

## Librosa

Librosa is a Python library designed for audio and music signal-processing tasks. It provides tools for analyzing, manipulating, and extracting features from audio data, making it invaluable for tasks such as audio classification, speech recognition, and music information retrieval. Librosa offers a wide range of functionalities for loading audio files, visualizing waveforms, and computing various audio features, including spectrograms, mel-frequency cepstral coefficients (MFCCs), and chromatograms.

Key features of Librosa include:

* 1. **Audio File Loading:** In our project, we utilized Librosa to load audio files from the dataset. The `librosa.load()` function was used to load audio files in the waveform format, allowing us to obtain the raw audio data and the sample rate (in Hz) of each audio file. This step was crucial for preprocessing and feature extraction tasks, as it provided access to the underlying audio signals.
  2. **Feature Extraction:** Librosa offers a rich set of functions for extracting informative features from audio signals. In our project, we focused on extracting Mel-frequency cepstral coefficients (MFCCs), a widely used feature representation for audio classification tasks. The `librosa.feature.mfcc()` function was employed to compute MFCCs from the raw audio data, providing a compact and discriminative representation of the audio content. These MFCC features were then used as input to our convolutional neural network (CNN) model for disease classification.
  3. **Preprocessing:** In addition to feature extraction, Librosa facilitated preprocessing steps to enhance the quality of the audio data. For instance, we applied data augmentation techniques such as adding random noise and shifting the audio signals to create augmented versions of the original audio data. Librosa provided functions for adding noise (`np.random.randn()`) and shifting (`np.roll()`) the audio signals, allowing us to generate diverse training examples for the CNN model.
  4. **Visualization:** Librosa includes visualization tools that enable users to visualize audio waveforms, spectrograms, and other relevant representations. While not extensively utilized in our project, these visualization capabilities can be valuable for inspecting and understanding the characteristics of audio data. For instance, the

`librosa.display.waveplot()` function can be used to plot audio waveforms, providing insights into the temporal structure of the audio signals.

Librosa offers crucial tools for audio processing and feature extraction. With its intuitive interface, comprehensive feature set, and robust visualization capabilities, we successfully preprocessed audio data, extracted informative features, and prepared input data for our CNN model. Leveraging Librosa's functionalities enabled us to efficiently handle audio data, leading to promising results in disease classification.

## Scikit-learn

Scikit-learn, often abbreviated as sklearn, stands as a cornerstone in the realm of machine learning libraries, offering a robust and user-friendly framework for a myriad of tasks. Developed as an open-source project, scikit-learn boasts an extensive array of functionalities tailored towards facilitating machine learning endeavors, from data preprocessing to model evaluation. At its core lies a plethora of algorithms, ranging from classical supervised and unsupervised learning methods to cutting-edge techniques in ensemble learning and dimensionality reduction.

Key features of scikit-learn include:

* 1. **Comprehensive Set of Algorithms:** Scikit-learn encompasses a diverse collection of algorithms for classification, regression, clustering, dimensionality reduction, and more. From traditional models like linear regression and decision trees to advanced methods such as support vector machines and random forests, scikit-learn provides a versatile toolbox to tackle various machine learning tasks.
  2. **Consistent API:** With a unified and intuitive interface, scikit-learn ensures a seamless user experience across different algorithms. Its consistent API design facilitates rapid prototyping and experimentation, enabling users to easily switch between models and compare their performance.
  3. **Data Preprocessing Tools:** Scikit-learn offers a wide range of tools for data preprocessing, including feature scaling, imputation of missing values, encoding categorical variables, and feature selection. These utilities streamline the data preparation process, allowing users to focus on building and refining their models.
  4. **Model Evaluation and Selection:** Scikit-learn provides robust mechanisms for model evaluation and selection, with built-in functions for cross-validation, hyperparameter tuning, and performance metrics computation. These tools enable users to assess the generalization performance of their models and fine-tune them for optimal results.
  5. **Integration with Scientific Computing Ecosystem:** Scikit-learn seamlessly integrates with other popular libraries in the Python scientific computing ecosystem, such as NumPy, SciPy, and matplotlib. This interoperability enhances the library's capabilities and facilitates seamless data manipulation, visualization, and analysis workflows.

In essence, scikit-learn epitomizes simplicity, versatility, and reliability, making it a go-to choice for machine learning enthusiasts, researchers, and practitioners alike. Its rich assortment of algorithms, coupled with its user-friendly interface and extensive documentation, positions it as a cornerstone in the field of machine learning and data science.

# CHAPTER 5: METHODOLOGY

## Dataset

The dataset utilized for this analysis comprises a diverse collection of audio files annotated with corresponding disease labels. Each entry in the dataset encapsulates a unique disease category along with the count of its associated audio files (in .wav format). This compilation of audio recordings forms the basis of our exploration into the relationship between sound patterns and disease manifestation.

Below is a summary of the diseases included in the dataset along with the respective counts of their associated .wav files:

## Table 1: Original Dataset

|  |  |
| --- | --- |
| **Disease** | **Count of Audio Files** |
| Bronchitis | 92 |
| Pertussis | 9 |
| Pneumonia | 80 |
| Asthma | 110 |

* 1. **Implementation**

### Data Import and Preprocessing

To prepare the dataset for analysis, we developed a custom Python script to import and preprocess the audio files related to various respiratory diseases. This script enables us to efficiently manage the dataset and perform necessary preprocessing steps prior to analysis. The process begins by traversing through the dataset directory, which is organized into subdirectories, each representing a different respiratory disease. Within each disease category, the script iterates through the corresponding audio files, leveraging the librosa library for loading and processing the audio data.

Key steps in the preprocessing pipeline include:

* + - 1. Loading Audio Files: Using the librosa.load function, the script reads each audio file, ensuring compatibility with various sampling rates and formats.
      2. Normalization: To maintain consistent volume levels across all audio samples, we normalize the audio data. This step is essential for mitigating variations in recording conditions and ensuring uniformity in subsequent analyses.
      3. Label Assignment: Concurrently, the script assigns appropriate labels to each audio sample based on the disease category it represents. This labeling process is crucial for supervised learning tasks and facilitates accurate classification and analysis.

By encapsulating these preprocessing steps within a structured Python script, we streamline the data preparation phase, ensuring consistency and efficiency in our subsequent analyses. The processed dataset, comprising normalized audio data and corresponding disease labels, forms the foundation for our exploratory and predictive analyses of respiratory diseases.

### Audio Waveform Visualization:

Understanding the temporal characteristics of audio signals is crucial in the analysis of respiratory diseases using audio data.

It provides insights into the underlying patterns and structures within the audio data.

A custom waveform visualization technique was developed to graphically represent the amplitude variations of individual audio files over time.

This technique enables the visualization of key characteristics of the audio signals. Process Overview:

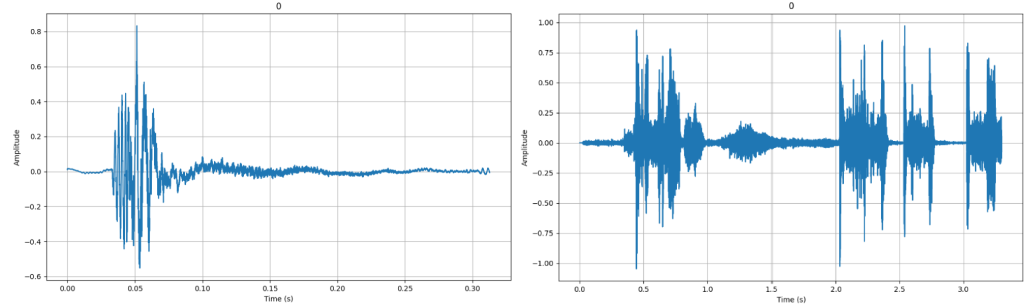
* + - 1. Audio File Loading: The librosa library is utilized for loading audio files, a powerful tool for audio processing.
      2. Duration Extraction and Time Axis Creation: The duration of each audio file is extracted, and a time axis spanning the duration of the audio signal is created.
      3. Amplitude vs. Time Plotting: The amplitude of the audio signal is plotted against time, constructing a visual representation of the audio waveform.

Visualization Details:

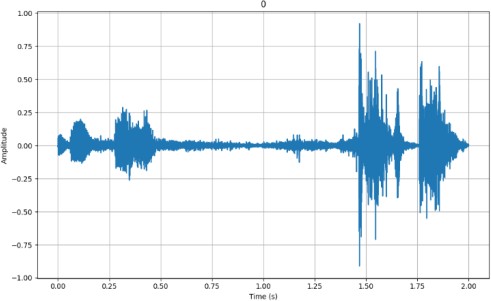
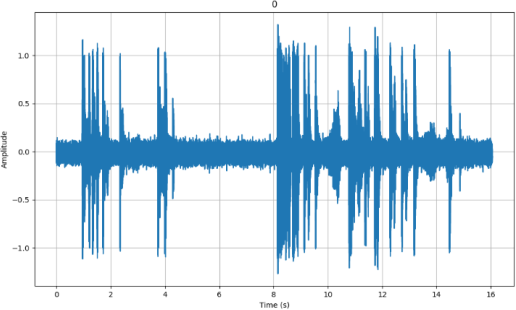
1. Plot Representation: Each plot in the visualization corresponds to a single audio file. The x-axis represents time (in seconds), while the y-axis represents the amplitude of the audio signal.
2. Insights from Waveform Plot: The waveform plot captures fluctuations in amplitude over time, revealing characteristics such as intensity, frequency, and duration of sound.

Key Functions Used:

1. librosa.load: Utilized for loading audio files.
2. librosa.get\_duration: Used to extract the duration of each audio file.
3. matplotlib.pyplot.plot: Function employed to plot the amplitude of the audio signal against time, generating waveform plots.



Asthma Bronchitis



Pertussis Pneumonia

**Figure 3: Waveforms of diseases**

## Spectrogram Visualization

Understanding the frequency content of audio signals is essential in analyzing respiratory diseases using audio data. To facilitate this, we developed a custom spectrogram visualization technique. This method enables us to visually represent the frequency distribution of audio signals over time, providing valuable insights into the spectral characteristics of the audio data.

Process Overview:

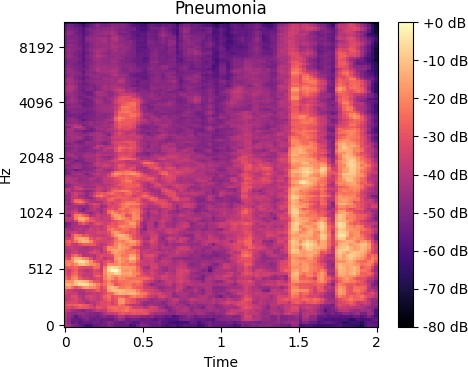
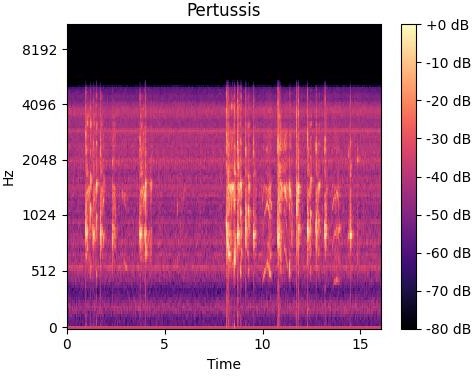
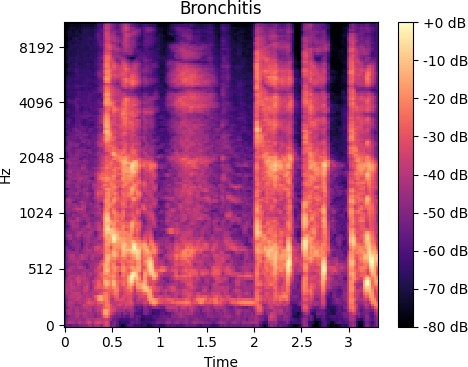
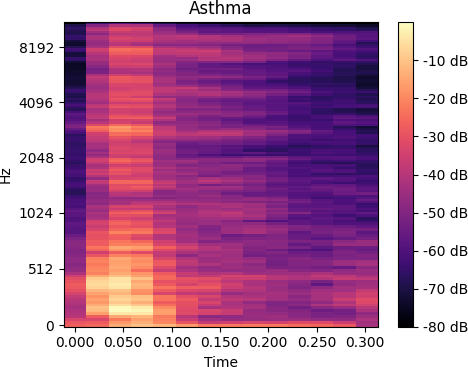
* + - 1. Spectrogram Computation: The spectrogram is computed using the librosa.feature.melspectrogram function, which calculates the Mel spectrogram of the audio signal.
      2. Plotting the Spectrogram: The spectrogram is visualized using

**librosa.display.specshow**, with options for adjusting parameters such as the color

map and axis labels. We also apply power scaling and convert to decibels using librosa.power\_to\_db to enhance the visualization of the spectrogram.

* + - 1. Visualization Details: Frequency vs. Time Representation: The x-axis represents time (in seconds), while the y-axis represents frequency (in Mel scale). The color intensity indicates the energy level of each frequency component at a specific time.

The spectrogram visualization technique serves as a valuable tool in our analysis pipeline, enabling us to qualitatively assess the frequency distribution of audio signals.

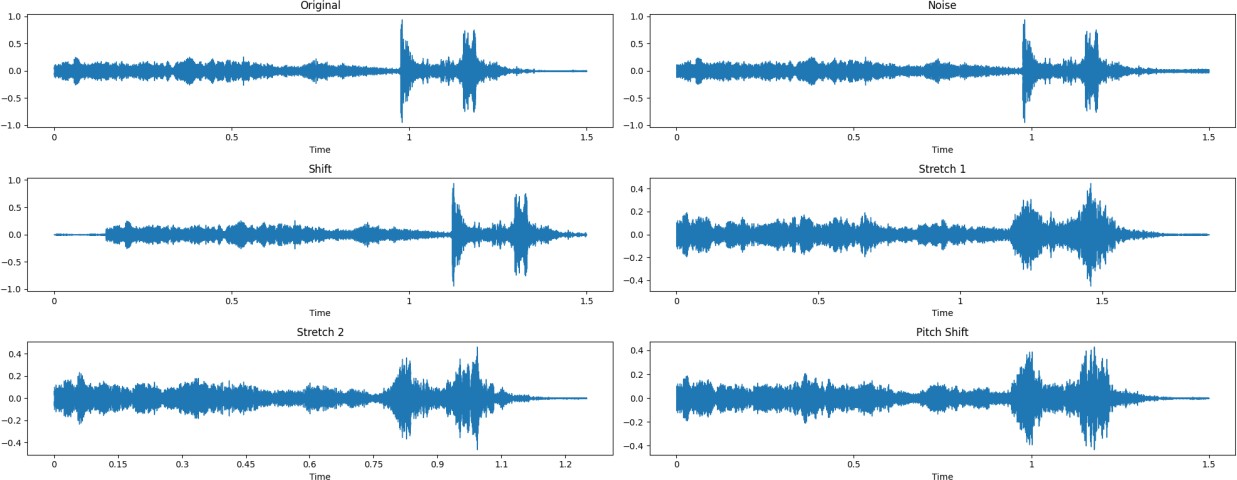
By visually examining the spectrogram patterns, we gain insights into the spectral characteristics of respiratory diseases, complementing our understanding derived from waveform analysis.

**Figure 4: Mel Spectrogram of Diseases**

## Audio Transformation Visualization

In the analysis of respiratory diseases using audio data, it's imperative to explore various transformations that can augment our dataset and improve the robustness of our models. We have implemented several audio transformation techniques to generate augmented audio samples, enhancing the diversity and variability of our dataset.

Transformation Techniques:

* + - 1. Adding Noise: Random noise is introduced to the audio signal using the add\_noise function, simulating environmental noise or recording artifacts.
      2. Shifting:The shift function shifts the audio signal by a specified number of samples, mimicking temporal displacements in the audio recording.
      3. Time Stretching: Time stretching is applied to alter the temporal duration of the audio signal. The stretch1 and stretch2 functions perform time stretching with different stretch rates, allowing manipulation of the speed of the audio playback.
      4. Pitch Shifting:The pitch\_shift function adjusts the pitch of the audio signal by a specified number of semitones, simulating variations in vocal characteristics or instrument tuning.

**Figure 5: Waveforms after Transformation**

## Dataset Augmentation and Statistics

During the analysis pipeline for respiratory disease classification using audio data, the importance of dataset augmentation is recognized to enhance the diversity and robustness of the dataset. Below, the steps involved in augmenting the dataset are outlined, along with detailed statistics comparing the original and augmented datasets for each respiratory disease category.

Dataset Augmentation Process:

* + - 1. File Duplication: Initially, the original audio files from each disease category are duplicated to create a base for augmentation.
      2. Audio Transformation:Various audio transformation techniques are applied to the duplicated files to generate augmented samples. These techniques include adding noise, shifting the audio, and stretching the time duration.
      3. Destination Directory Setup:The augmented audio samples are saved in designated destination directories corresponding to each respiratory disease category.

Detailed Augmentation Techniques:

1. Adding Noise: A small amount of random noise is added to each audio sample to simulate environmental noise or recording artifacts.
2. Shifting: The audio signal is shifted by a fixed number of samples, creating temporal displacements in the audio recording.
3. Time Stretching: The duration of the audio signal is manipulated using time stretching techniques with varying stretch rates, altering the speed of the audio playback.

**Augmented Dataset Counts**: Counts of the augmented audio files generated for each disease category are presented after applying the augmentation techniques.

## Table 2: Dataset after Augmentation

|  |  |
| --- | --- |
| **Disease** | **Count of Audio Files** |
| Bronchitis | 460 |
| Pertussis | 45 |
| Pneumonia | 400 |
| Asthma | 550 |

* + 1. **MFCC Feature Extraction and Dataset Processing**

In the respiratory disease classification pipeline, MFCC (Mel-frequency cepstral coefficients) features extracted from audio files are leveraged to train machine learning models. Below, the process of extracting MFCC features, processing the dataset, and loading the preprocessed data for model training is outlined.

MFCC Feature Extraction:

* + - 1. Function to Extract MFCC Features: A function named extract\_mfcc is defined to extract MFCC features from audio files using the librosa library. This function takes an audio file path as input and returns the MFCC matrix.

Dataset Processing:

1. Processing Augmented Data: The augmented audio data from the duplicated directories for each respiratory disease category is processed. MFCC features are

extracted from the augmented audio files, and labels are assigned based on the disease category.

1. Padding and Splitting: The MFCC sequences are padded to ensure uniform length, and then the dataset is split into train, validation, and test sets. The split sizes can be adjusted using parameters such as test\_size and val\_size.
2. Saving Processed Data: The processed datasets are saved as .npy files in the specified output directory for future use.

Loading Preprocessed Data:

1. Loading Train, Validation, and Test Sets: The preprocessed train, validation, and test sets are loaded from the saved .npy files.
2. Shapes of Loaded Arrays: The shapes of the loaded arrays are verified to ensure consistency and compatibility with model input requirements.
3. Input Shape and Number of Classes: The input shape of the data and the number of unique classes for classification are determined.

Preparing Data for Model Training:

a. Reshaping Input Data: Input data is reshaped to include an additional dimension to accommodate the channel axis for convolutional neural networks.

By processing the audio data into MFCC features and organizing it into train, validation, and test sets, we prepare a structured dataset suitable for training and evaluating machine learning models for respiratory disease classification.

## Convolutional Neural Network (CNN) Architecture

In the respiratory disease classification pipeline, a Convolutional Neural Network (CNN) architecture was utilised to process MFCC features extracted from audio data. The CNN model is designed to effectively learn and classify patterns indicative of different respiratory diseases.

Model Architecture:

* + - 1. The CNN model consists of multiple convolutional layers followed by max-pooling layers to extract and capture hierarchical features from the input MFCC data.
      2. The architecture comprises two sets of convolutional layers, each followed by max- pooling layers, allowing the model to learn increasingly abstract representations of the input data.
      3. The flattened output from the convolutional layers is passed through fully connected dense layers to perform classification based on the learned features.
      4. Dropout layers are incorporated to prevent overfitting by randomly dropping a fraction of connections during training.
      5. Batch normalization layers are added to normalize the activations of the previous layer, which accelerates the training process and improves generalization.

Model Compilation and Training:

1. The model is compiled with the Adam optimizer, utilizing a lower learning rate to facilitate stable training and convergence.
2. The loss function is specified as sparse categorical cross-entropy, suitable for multi-class classification tasks.
3. The model is trained on the preprocessed dataset consisting of MFCC features extracted from augmented audio data.

Functions used in model :

1. **create\_cnn\_model\_v2**: This function creates a Convolutional Neural Network (CNN) model with an updated architecture. It defines the layers of the model including convolutional layers, max-pooling layers, dense layers, and dropout layers. The architecture is designed to effectively process MFCC features extracted from audio data and perform multi-class classification of respiratory diseases.
2. **extract\_mfcc**: This function extracts Mel-frequency cepstral coefficients (MFCC) features from audio files using the librosa library. It takes an audio file path as input and returns the MFCC matrix representing the features of the audio signal.
3. **process\_data**: This function processes the augmented audio data from duplicated directories for each respiratory disease category. It extracts MFCC features from the augmented audio files and assigns labels based on the disease category. The MFCC sequences are padded to ensure uniform length, and the dataset is split into train, validation, and test sets.
4. **load\_data**: This function loads the preprocessed train, validation, and test sets from the saved .npy files. It reads the MFCC features and labels from the files and returns them as numpy arrays.
5. **count\_files**: This function counts the number of audio files in a directory with a specific extension (.wav). It is used to determine the number of original and augmented audio files for each respiratory disease category.

Training Phase

In the training phase, the CNN model was trained using the preprocessed dataset consisting of MFCC features extracted from audio files. The training process involved the following key parameters and configurations:

1. Training Data: The input features (MFCC data) for training the model were provided as X\_train, while the corresponding labels were provided as y\_train.
2. Batch Size: During training, a batch size of 32 was utilized. The batch size determines the number of samples processed before the model's parameters are updated. A larger batch size was chosen to expedite the training process.
3. Epochs: The model was trained for a total of 15 epochs. An epoch refers to one complete pass through the entire training dataset. Increasing the number of epochs allows the model to learn from the data for a longer duration.
4. Validation Data: Validation data (X\_val and y\_val) were used to evaluate the model's performance at the end of each epoch. This validation data helps monitor the model's generalization and prevent overfitting.
5. Early Stopping: To prevent overfitting and ensure optimal performance, the training process employed the Early Stopping callback mechanism. This callback monitors the validation loss and halts training if the loss does not improve for a specified number of epochs (patience). Additionally, the best weights achieved during training are restored (restore\_best\_weights=True) to retain the model's best performance.

Evaluation on Validation Data:

The model.evaluate function is used to evaluate the model's performance on the validation data (X\_val and y\_val).The validation loss and accuracy are calculated and stored in the variables val\_loss and val\_accuracy, respectively.These metrics are then printed to the console.

Evaluation on Test Data:

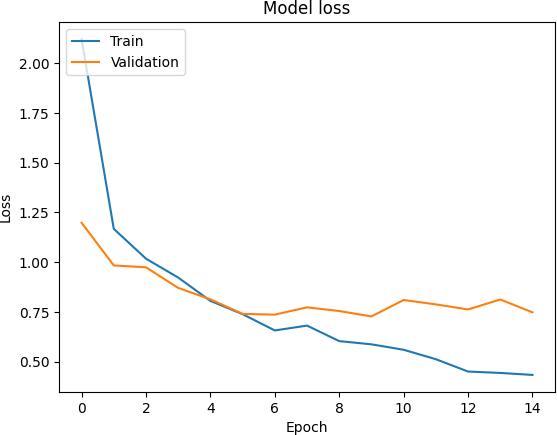
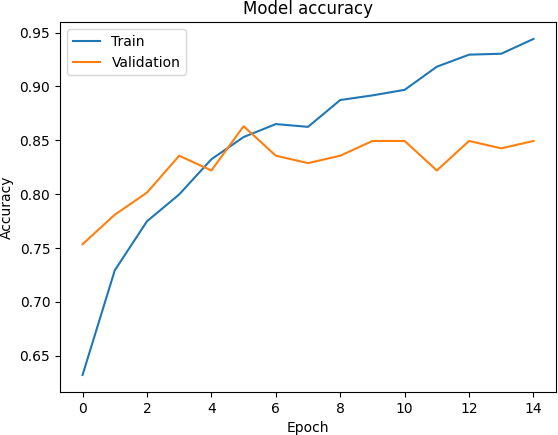
Similarly, the model.evaluate function is used to evaluate the model's performance on the test data (X\_test and y\_test).The test loss and accuracy are calculated and stored in the variables loss and accuracy, respectively.These metrics are then printed to the console.

Visualization of Training Metrics:

Matplotlib is used to create two subplots: one for plotting the training and validation accuracy values and another for plotting the training and validation loss

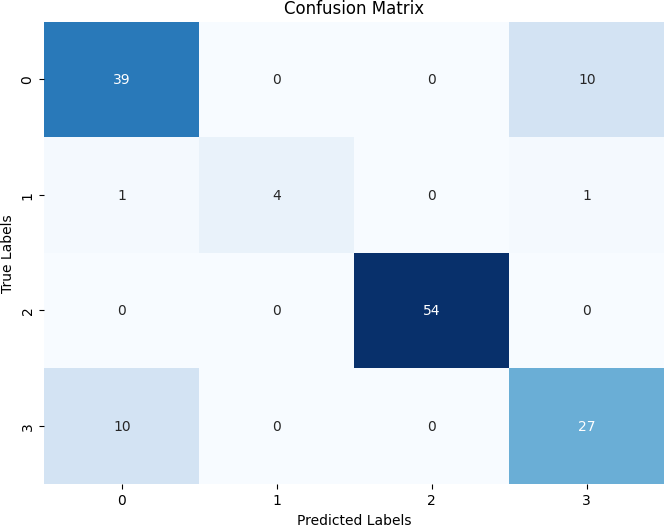
values. The training accuracy and loss values over epochs are accessed from the history object returned by the model.fit function during training. The validation accuracy and loss values over epochs are also accessed from the history object.

Both plots are displayed to visualize the model's training progress and performance.



**Figure 6: Accuracy v/s Epoch and Loss v/s Epoch**

Model Inference:

1. A random index is selected from the test data (X\_test) to perform inference on a random sample.
2. The input data (input\_data) and its corresponding label (input\_label) are retrieved based on the selected random index.
3. The input data is reshaped to match the model's input shape and passed through the model to obtain predictions.
4. The predicted label (predicted\_label) is obtained by finding the class index with the highest probability in the model's prediction.
5. Both the actual and predicted labels are printed to the console for comparison. Confusion Matrix:
6. Predicted probabilities for each class are obtained for the validation data (X\_val) using the trained model.
7. The predicted labels (y\_pred) are extracted by selecting the class index with the highest probability for each sample.
8. The confusion matrix (cm) is calculated based on the true labels (y\_val) and predicted labels (y\_pred).
9. The confusion matrix is visualized using a heatmap to provide an overview of the model's performance in terms of class-wise predictions.

**Figure 7: Confusion matrix**

Evaluation Metrics and Test Results

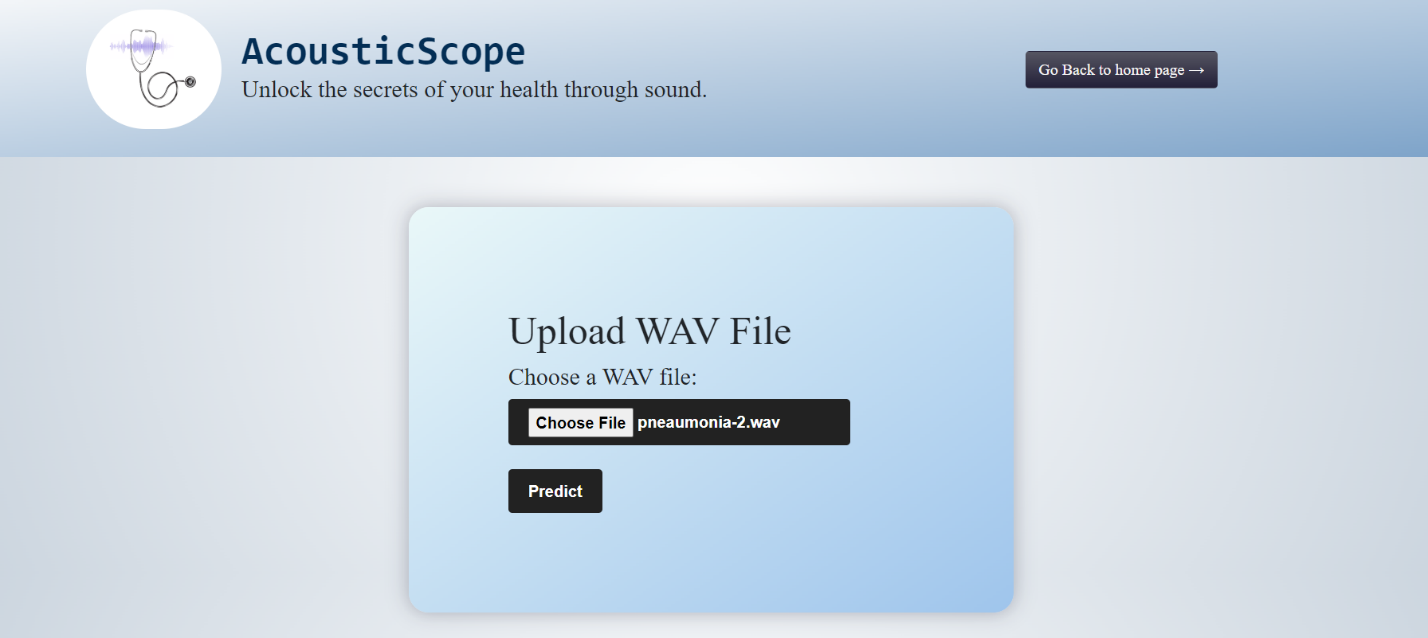
This section presents the evaluation metrics and test results obtained from the trained model. Precision, recall, and F1 score are calculated to assess the model's performance, alongside the overall test loss and accuracy. Additionally, a comparison between predicted labels and true labels on the test data is provided to further evaluate the model's effectiveness.

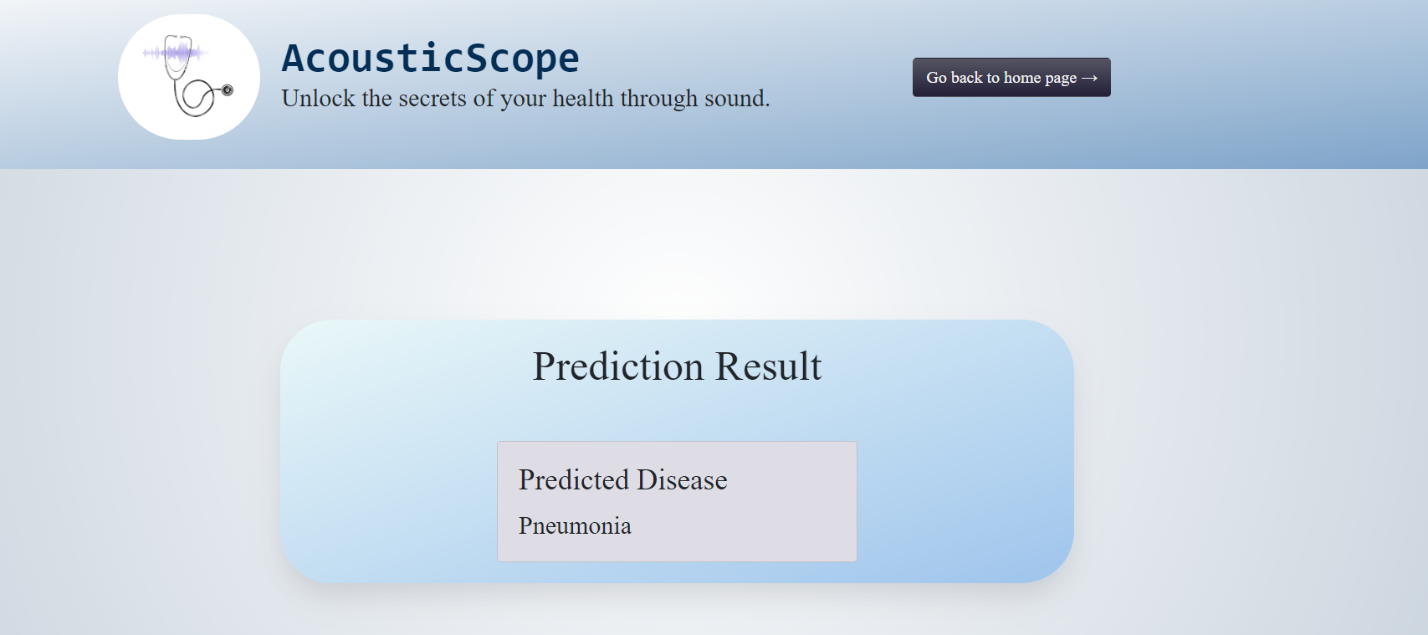
**Table 3: Evaluation Metrics and Test Results**

|  |  |
| --- | --- |
| **Metric** | **Score** |
| Precision | 0.85280 |
| Recall | 0.84931 |
| F1 Score | 0.84963 |
| Test Loss | 0.64490 |
| Test Accuracy(model) | 0.84246 |
| Test Accuracy | 0.84246 |

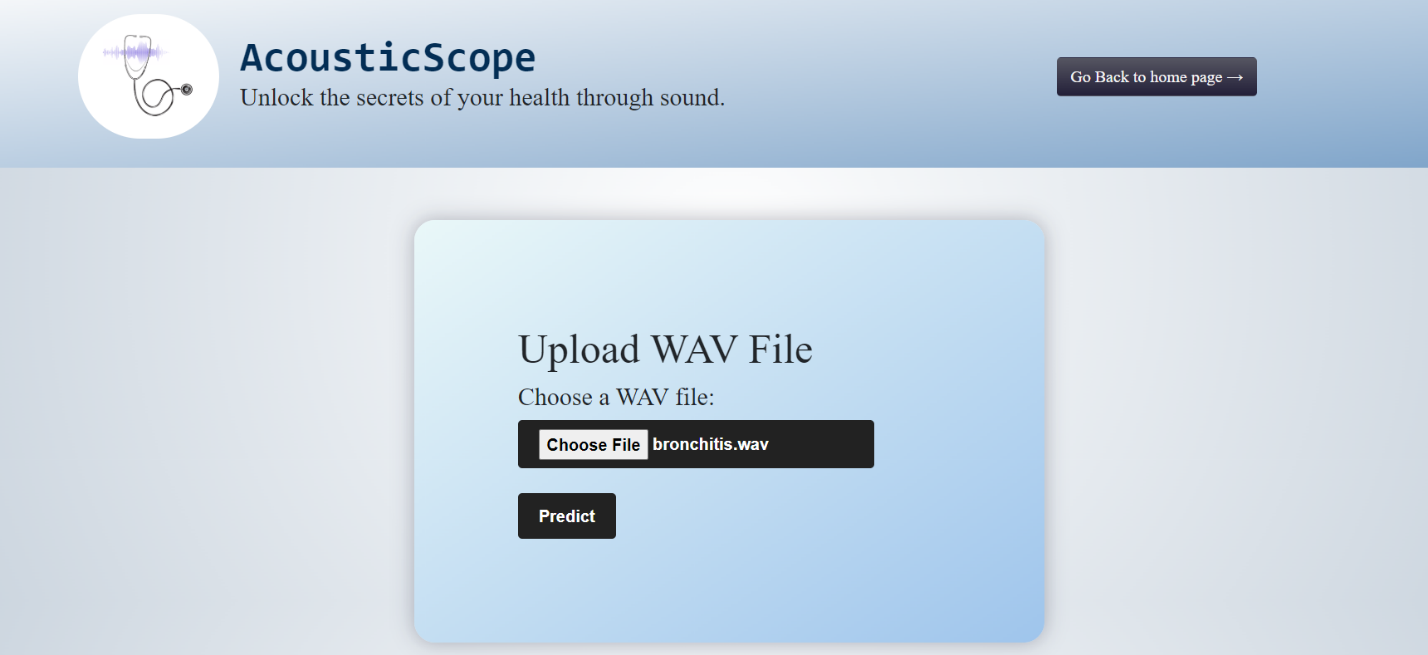
# CHAPTER 6: RESULTS

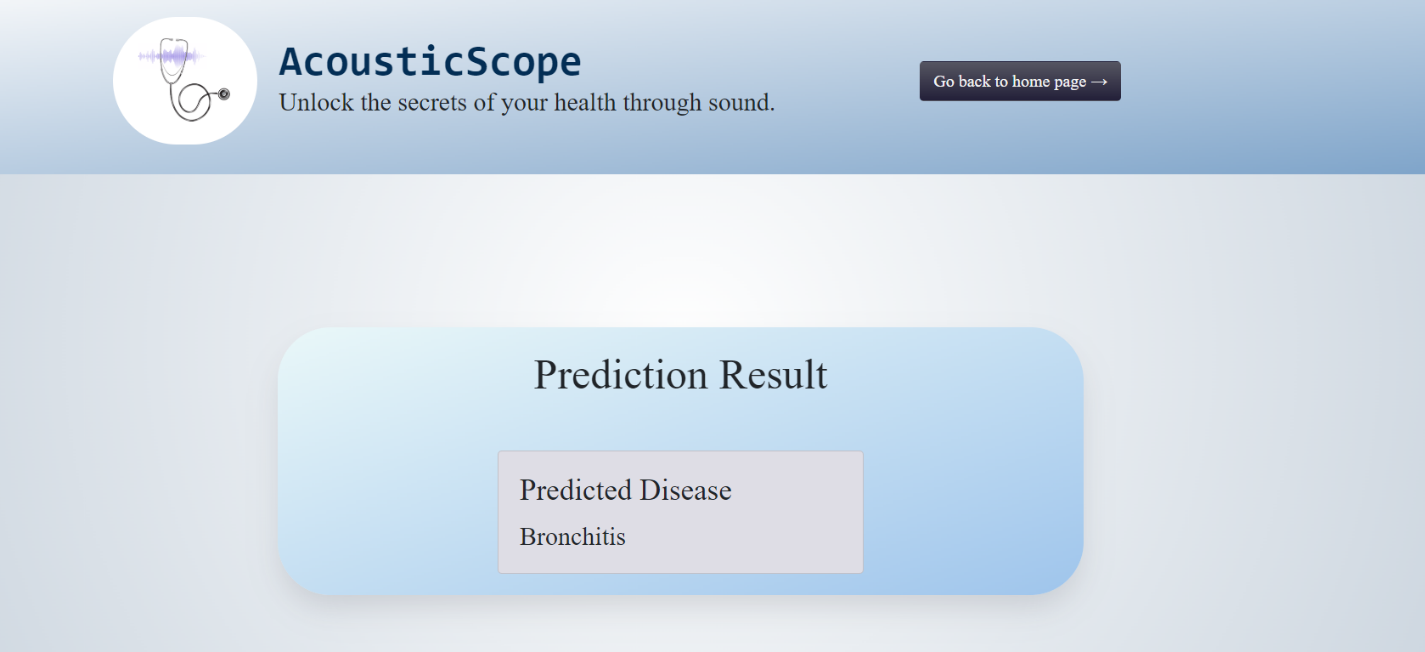
The outcomes of the conducted experiments and analyses are presented. Through rigorous testing and evaluation, the performance and efficacy of the proposed methodologies and models are unveiled, addressing the outlined research objectives. The findings shed light on the effectiveness and potential implications of the approaches within the studied domain.



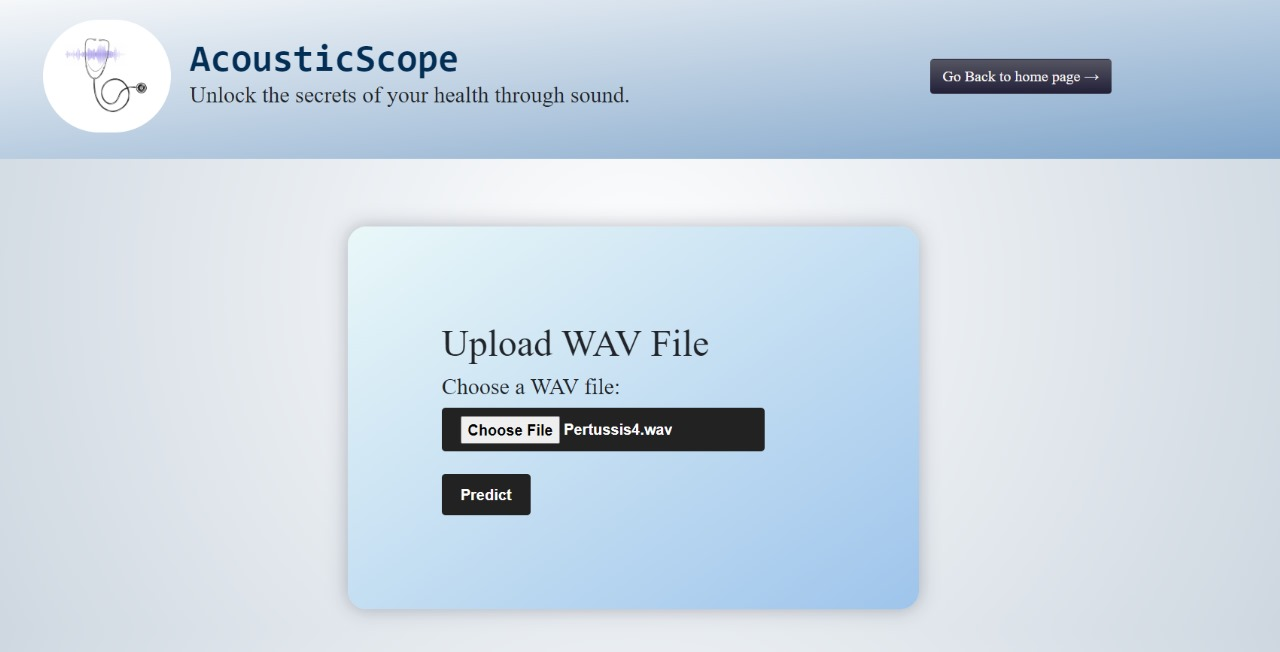


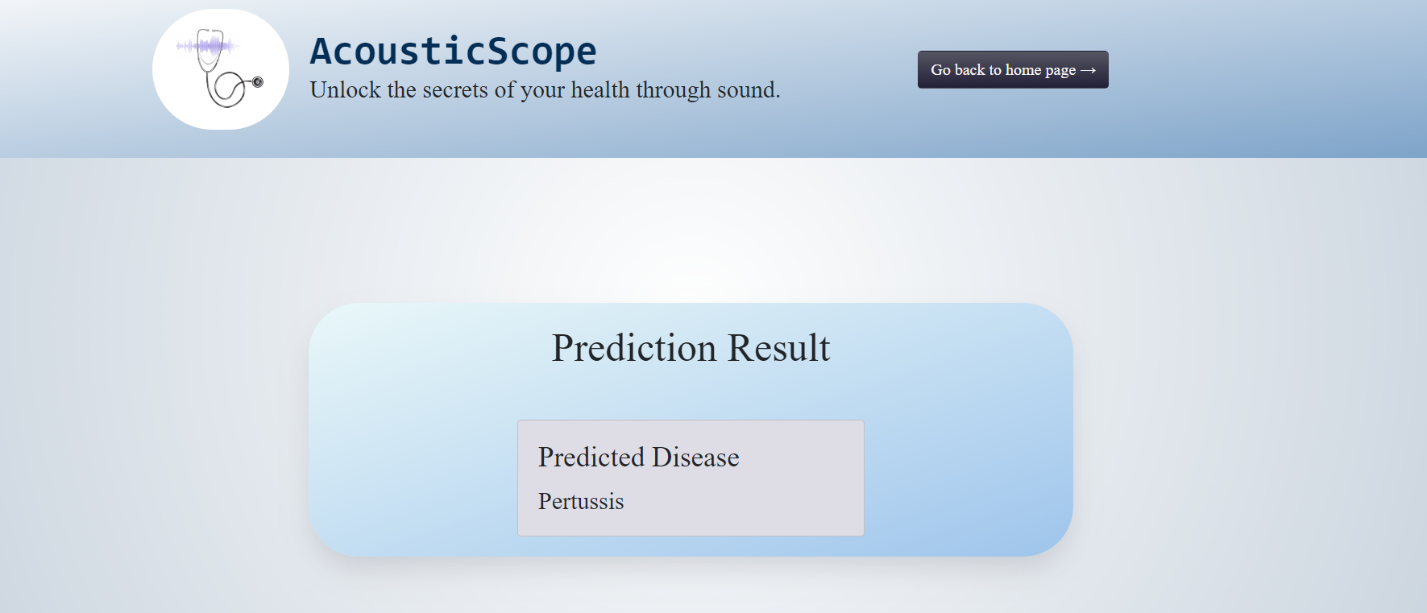
**Figure 8: Prediction Result for Pneumonia**





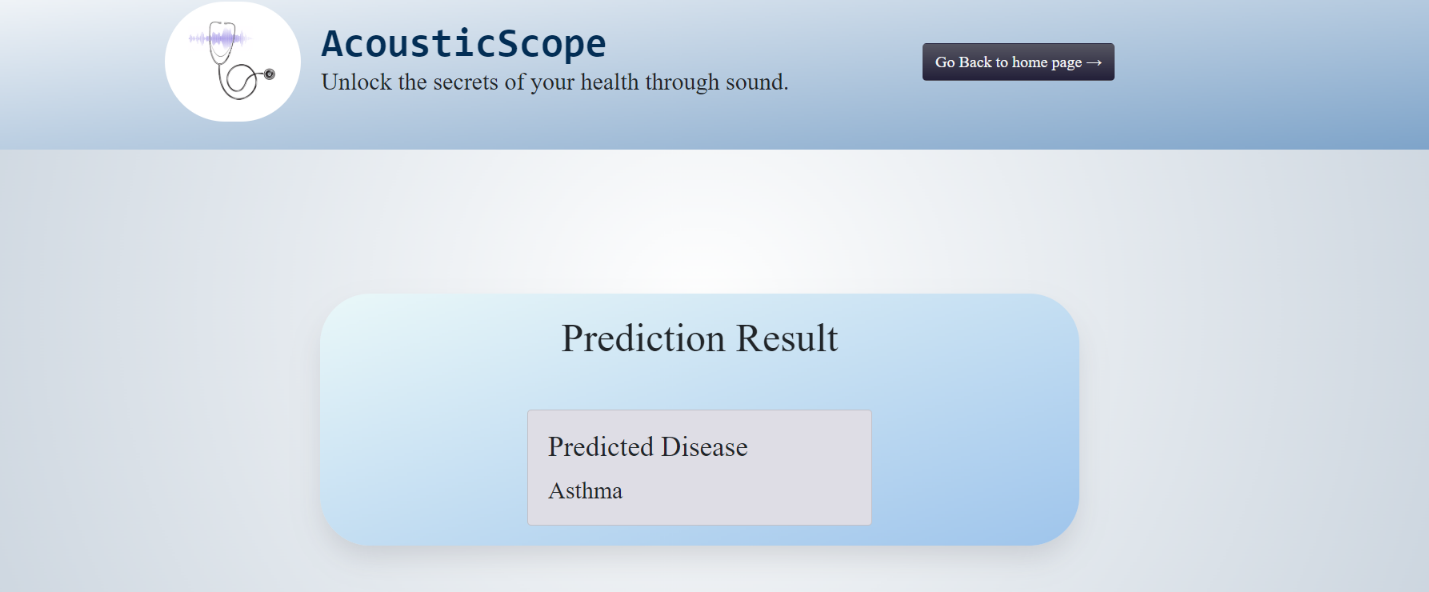
**Figure 9: Prediction Result for Bronchitis**





**Figure 10: Prediction Result for Pertussis**





**Figure 11: Prediction Result for Asthma**

In conclusion, the results obtained from our experiments provide valuable insights into the performance and effectiveness of the proposed methodologies and models. These findings not only validate the feasibility of our approaches but also highlight their potential significance and applicability in addressing the research objectives within the studied domain.

# CHAPTER 7: FUTURE SCOPE AND CONCLUSION

As the landscape of healthcare continues to evolve, advancements in technology and innovative approaches play a pivotal role in shaping the future of respiratory disease management. From early detection to resource optimization, the integration of cutting-edge solutions holds immense promise in revolutionizing healthcare delivery and improving patient outcomes.

1. **Early Detection and Intervention:** The early detection of respiratory diseases, such as Pertussis, Bronchitis, and Bronchiolitis, stands as a cornerstone in ensuring effective treatment and management strategies. Moving forward, the development of more sophisticated diagnostic tools, including AI-driven algorithms and wearable sensors, promises to enhance early detection capabilities, enabling prompt intervention and personalized treatment plans tailored to individual patient needs.
2. **Remote Monitoring and Telemedicine:** In regions with limited access to healthcare facilities, the adoption of remote monitoring and telemedicine technologies emerges as a critical avenue for bridging gaps in healthcare delivery. The future lies in the integration of advanced telehealth platforms, offering real- time consultations, remote diagnostics, and continuous monitoring of respiratory health parameters. By leveraging telemedicine, healthcare providers can extend their reach to underserved populations, ensuring timely access to quality care and improving health outcomes.
3. **App-Enabled Cough Monitoring:** Innovations in healthcare are increasingly leveraging app technology to empower patients and enhance clinical decision- making. App-enabled cough monitoring solutions hold immense potential in providing real-time detection and analysis of cough sounds, facilitating immediate patient monitoring and intervention. By harnessing the power of machine learning algorithms and smartphone applications, healthcare providers can accurately track cough patterns, identify potential respiratory issues, and intervene proactively to prevent disease progression.
4. **Resource Optimization in Healthcare Settings:** The optimization of healthcare resources, including staff, equipment, and infrastructure, remains a pressing concern, particularly during periods of high demand or emergency situations. Looking ahead, the integration of predictive analytics, IoT devices, and AI-driven resource allocation algorithms offers promising avenues for optimizing resource utilization, improving operational efficiency, and enhancing patient care delivery in healthcare settings. By leveraging data-driven insights, healthcare organizations can anticipate demand, streamline workflows, and allocate resources more effectively, ensuring timely access to care and enhancing overall healthcare outcomes.
5. **Research and Epidemiological Studies:** The availability of large-scale datasets comprising cough sounds from confirmed respiratory cases presents unprecedented opportunities for advancing epidemiological research and deepening our understanding of disease dynamics. Future research endeavors will focus on leveraging big data analytics, machine learning, and epidemiological modeling techniques to analyze cough sound patterns, identify disease trends, and predict outbreaks. By harnessing the power of data-driven insights, researchers can inform public health policies, implement targeted intervention strategies, and ultimately mitigate the burden of respiratory diseases on a global scale.

In conclusion, the future of respiratory disease management holds immense promise, driven by technological innovations, data-driven insights, and collaborative efforts across the healthcare ecosystem. By embracing emerging technologies and adopting a proactive approach to healthcare delivery, we can pave the way for a future where respiratory diseases are detected early, managed effectively, and their impact on public health significantly reduced.

# REFERENCES

1. Kranthi Kumar Lella and Alphonse Pja, "Automatic diagnosis of COVID-19 disease using deep convolutional neural network with multi-feature channel from respiratory sound data: Cough, voice, and breath", Alexandria Engineering Journal, 2022
2. Nehad M. Abdel Rahman Ibrahim, Irfan Ullah Khan and Mohammed S. Alshahrani, "Cough Sound Detection and Diagnosis Using Artificial Intelligence Techniques: Challenges and Opportunities", PubMed Central, National Library of Medicine, 2021
3. Murat Aykanat,Ozkan Kilic,Bahar Kurt and Sevgi Saryal, "Classification of lung sounds using convolutional neural networks", EURASIP Journal on Image and Video Processing, 2017
4. Noor Fadzli Abdul Razak, Juliana Johari, Noor Azlina Mohd Salleh and Yupiter Harangan Prasada Manurung, "Cough Detection System Using Machine Learning Technique", European Chemical Bulletin, 2023
5. Siqi Liao, Chao Song, Xiaoqin Wang and Yanyun Wang, "A classification framework for identifying bronchitis and pneumonia in children based on a small-scale cough sounds dataset", PubMed Central, National Library of Medicine, 2022
6. Teghdeep Kapoor, Tanya Pandhi, Bharat Gupta, “Cough Audio Analysis for COVID‑19 Diagnosis”, March 2022.
7. Ravi Patel, Victor Miranda, and Ali Zaidi, "Real-Time cough and sneeze detection using convolutional and recurrent neural networks ", 2019
8. V Aravind, V Harish Kumar, S V Chandra Sekhar, V Sravani and V Vineela, "A Survey on the Identification of COVID-19 from Cough Audio Sounds Using Deep Learning", 2022
9. Madhurananda Pahar, Igor Miranda, Andreas Diacon and Thomas Niesler, "Deep Neural Network based Cough Detection using Bed-mounted Accelerometer Measurements", 2021
10. Sezer Ulukaya, Ahmet Alp Sarıca, Oğuzhan Erdem and Ali Karaali, "MSCCov19Net: multi-branch deep learning model for C detection from cough sounds", 2023
11. Gyeong-Tae Leea, Hyeonuk Nama, Seong-Hu Kima, Sang-Min Choia, Youngkey Kimb and Yong-Hwa Parka, "Deep learning based cough detection camera using enhanced features", 2022
12. B T Balamurali , ORCID,Hwan Ing Hee , ORCID,Saumitra Kapoor ORCID, Oon Hoe Teoh , Sung Shin Teng , Khai Pin Lee , Dorien Herremans ORCID and Jer Ming Chen, “ Deep Neural Network-Based Respiratory Pathology Classification Using Cough Sounds” , June 2021
13. Jiaming Liu, Mingyu You, Zheng Wang, Guo-Zheng Li, “ Cough Detection Using Deep Neural Networks”, 2014
14. Kawther S. Alqudaihi, Nida Aslam, Irfan Ullah Khan, Abdullah M. Almuhaideb,corresponding author Shikah J. Alsunaidi, Nehad M. Abdel Rahman Ibrahim, Fahd A. Alhaidari, Fatema S. Shaikh, Yasmine M. Alsenbel, Dima M. Alalharith, Hajar M. Alharthi, Wejdan M. Alghamdi, and Mohammed S. Alshahrani , "Cough Sound Detection and Diagnosis Using Artificial Intelligence Techniques: Challenges and Opportunities", 2021
15. Gabriela Augustinov, Patrick Fischer, Volker Gross, Ulrich Koehler, Keywan Sohrabi, and Seyed Amir Hossein Tabatabaei, "Automatic Detection and Classification of Cough Events Based on Deep Learning", 2020
16. Quan Zhou, Jianhua Shan, Wenlong Ding, Chengyin Wang, Shi Yuan, Fuchun Sun, Haiyuan Li and Bin Fang, "Cough Recognition Based on Mel-Spectrogram and Convolutional Neural Network", 2021
17. Muntasir Mamun, "Analyzing Cough Sounds for the Evidence of Covid-19 using Deep Learning Models", 2022
18. Harry Coppock, Alex Gaskell, Panagiotis Tzirakis, Alice Baird, Lyn Jones and Björn Schuller, "End-to-end convolutional neural network enables COVID-19 detection from breath and cough audio: a pilot study", 2021
19. Aneeqa Ijaz , Muhammad Nabeel , Usama Masood , Tahir Mahmood , Mydah Sajid Hashmi , Iryna Posokhova , Ali Rizwan and Ali Imran, "Towards using cough for respiratory disease diagnosis by leveraging Artificial Intelligence: A survey", 2021
20. Mingyu You, Weihao Wang, You Li, Jiaming Liu, Xianghuai Xu and Zhongmin Qiu, "Automatic cough detection from realistic audio recordings using C-BiLSTM with boundary regression", 2022
21. Madhurananda Pahar, Marisa Klopper, Robin Warren and Thomas Niesler, "COVID-19 cough classification using machine learning and global smartphone recordings", 2021
22. Seifeddine Messaoud, Soulef Bouaafia, Amna Maraoui and Mohsen Machhout, "COVID-19 Recognition Based on Patient’s Coughing and Breathing Patterns Analysis: Deep Learning Approach", 2021
23. Rumana Islam , Esam Abdel-Raheem , Mohammed Tarique , "A study of using cough sounds and deep neural networks for the early detection of Covid-19", 2022
24. Jordi Laguarta , Ferran Hueto, and Brian Subirana, "COVID-19 Artificial Intelligence Diagnosis Using Only Cough Recordings", 2020
25. Javier Andreu-Perez; Humberto Pérez-Espinosa; Eva Timonet; Mehrin Kiani; Manuel I. Girón-Pérez; Alma B. Benitez, " A Generic Deep Learning Based Cough Analysis System From Clinically Validated Samples for Point-of-Need Covid-19 Test and Severity Levels", 2021
26. Skander Hamdi, Mourad Oussalah, Abdelouahab Moussaoui & Mohamed Saidi ,"Attention-based hybrid CNN-LSTM and spectral data augmentation for COVID-19 diagnosis from cough sound", 2022
27. Vipin Bansal, Gaurav Pahwa and Nirmal Kannan, "Cough Classification for COVID-19 based on audio mfcc features using Convolutional Neural Networks", 2020
28. KC Santosh, Nicholas Rasmussen, Muntasir Mamun, and Sunil Aryal, "A systematic review on cough sound analysis for Covid-19 diagnosis and screening: is my cough sound COVID-19", 2022
29. Hwan Ing Hee, BT Balamurali, Arivazhagan Karunakaran, Dorien Herremans and Onn Hoe Teoh, "Development of Machine Learning for Asthmatic and Healthy Voluntary Cough Sounds: A Proof of Concept Study", 2019