DISEASE OF LUNG INFECTION DETECTION

USING CNN MODEL -BAYESIAN OPTIMISATION

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Poojitha Gutha

December 2023

DISEASE OF LUNG INFECTION DETECTION

USING CNN MODEL-BAYESIAN OPTIMISATION

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Poojitha Gutha

December 2023

Approved by:

Dr. Jennifer Jin, Advisor, School of Computer Science

Dr. Khalil Dajani,Committee Member

Dr. Ronald Salloum, Committee Member

ABSTRACT

Auscultation is a fundamental aspect of the general examination used in disease diagnosis and identification, but its effective application often necessitates specialized training and expertise. This study aims to address this challenge by introducing an innovative respiratory categorization model designed to classify respiratory sounds into eight distinct categories: URTI, Healthy, Asthma, COPD, LRTI, Bronchiectasis, Pneumonia, and Bronchiolitis. For this categorization, the study employs a Convolutional Neural Network (CNN) model that has been optimized using Bayesian techniques. The dataset used in the study comprises 920 audio samples obtained from 126 patients, with sound durations ranging from 10 to 90 seconds. Impressively, the model demonstrates a noteworthy 83% validation accuracy and an impressive 86% training accuracy, highlighting its robust and effective performance. To enhance user interaction and facilitate result visualization, the research team has developed a user-friendly interface using Flask, HTML, and CSS. This interface provides healthcare professionals and other stakeholders with the means to access and interpret the results of the experimental analysis. Overall, this research marks a significant stride in making respiratory sound analysis more accessible and accurate, thus contributing to improved disease diagnosis and patient care.

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude and appreciation to the individuals and institutions that have contributed significantly to the completion of this project.

I would like to express my sincere gratitude to my respected advisor, Dr. Jennifer Jin, and my committee members, Dr. Ronald Salloum and Dr. Khalil Dajani. Their mentorship, expertise, and unwavering support have been indispensable throughout this project. Their valuable input and guidance significantly shaped the course and excellence of this endeavor.

I am indebted to the faculty and staff of California State University, San Bernardino**,** whose provision of essential resources and conducive academic environment played a pivotal role in the successful execution of this project. Their dedication to academic excellence has been a constant source of inspiration.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Weather changes can greatly worsen breathing problems and increase sickness and death in adults who already have ongoing lung issues like asthma, COPD, or other serious lung diseases. Chronic respiratory diseases (CRDs) impact the airways and lung structures. Common examples include chronic obstructive pulmonary disease (COPD), asthma, occupational lung ailments, and pulmonary hypertension. Aside from tobacco smoke, other risk factors encompass air pollution, exposure to occupational chemicals and dust, and frequent lower respiratory infections during childhood. Unfortunately, CRDs cannot be cured. Nevertheless, various treatments are available to ease breathing difficulties and enhance the quality of life for individuals affected by these conditions.Respiratory anomalies, such as asthma, COPD, pneumonia, bronchitis, and infections, disrupt the normal functioning of the respiratory system. These conditions vary in severity, symptoms, and treatments. Accurate diagnosis is crucial to provide customized care for each patient's needs.

Respiratory sounds are important for diagnosing and monitoring lung and airway health. These sounds can be divided into two main types: normal and abnormal. Normal breath sounds are soft and low-pitched. Abnormal breath sounds include wheezes, which are high-pitched whistling sounds, and crackles, which are discontinuous and crackling sounds. These sounds are non-invasive and valuable for medical evaluation.

Purpose

Respiratory sound classification aims to achieve important goals in healthcare and technology some of the goals are:

- Improving diagnostic precision through automated classification of respiratory sounds.
- Enhancing treatment personalization by offering deep insights, leading to better patient care.
- Enabling real-time patient monitoring through automation.

Motivation

Manual categorization of respiratory sounds is not only laborious but also susceptible to differing interpretations based on an individual's level of expertise. In recent times, progress in machine learning, specifically deep learning, has brought about a transformation in healthcare and medical diagnostics. The intricacies and subtleties involved in the analysis of respiratory sounds require a refined method for precise categorization and understanding.

CHAPTER TWO

LITERATURE SURVEY

Respiratory sounds provide important information about how our lungs work. Diagnosing lung issues often involves listening for abnormal sounds during auscultation. Listening to respiratory sounds helps diagnose respiratory conditions early.

To make this process less subjective and reduce the burden on doctors, many methods automate sound analysis. However, the success of these methods heavily depends on the quality and breadth of the respiratory sound database they use. In our research, we've created SPRSound, the first freely accessible pediatric respiratory sound database [1].

In 2019, almost 8 million people died due to respiratory illness. Auscultation, a common method for diagnosing respiratory issues. However, it cannot provide visual results. Moreover, a doctor's experience can influence the diagnosis, emphasizing the need for a quantitative analysis diagnostic system [2].

In recent years, automating the sorting of respiratory sounds has become a key research area. However, the success of modern deep learning methods in this field is hindered by the small and unevenly distributed datasets available. This is crucial for continuous monitoring of our respiratory system to prevent respiratory-related diseases[3]. In this study, we propose exploring the effectiveness of a new approach, MBTCNSE, which combines a multi-branch

Temporal Convolutional Network (TCN) with a Squeeze-and-Excitation Network (SEnet), to improve respiratory sound classification [4][5].

Analyzing respiratory sounds is crucial for understanding the lung's current state and can aid medical professionals in making precise diagnoses. The need for automated respiratory sound software is essential, as it can speed up diagnosis and ease the workload of physicians. In our study, we aim to differentiate between normal lung sounds, those with airway obstructions, and those with parenchymal issues using respiratory sound recordings from the RALE database [6][7].

Utilizing a wearable wireless acoustic sensor and a smartphone for respiratory sound classification can enhance asthma diagnosis and management. This innovation has the potential to improve long-term asthma care and reduce treatment costs. Moreover, it optimizes power consumption during signal acquisition, which is critical for a battery-powered system [8] [9].

Abnormal respiratory sounds, such as crackles, play a crucial role in diagnosing various respiratory diseases. Hence, understanding the characteristics of crackles is essential for developing a computerized diagnostic approach. In this study, we propose a methodology that integrates a random forest classifier and Empirical Mode Decomposition (EMD) to classify subjects into six respiratory conditions: healthy, bronchiectasis, bronchiolitis, COPD, pneumonia, and URTI [10][11].

CHAPTER THREE

SYSTEM REQUIREMENTS

Frameworks

Python

Version: Python 3.7 or higher is required for this project.

Description: Python is a powerful and frequently used programming language in machine learning and data data analysis. Its extensive library and framework support, ease of usage, and strong community support make it a great choice for our project.

Python serves as the primary interface, connecting us to critical tools and frameworks such as TensorFlow, Keras, scikit-learn, and Flask. It aids in the development of machine learning models, specifically Convolutional Neural Networks (CNNs), for accurately categorizing respiratory sounds. Python also handles data processing tasks like data organization and segmenting respiratory sound data. Furthermore, Python is useful for Bayesian optimization, which determines the best settings for our machine-learning model.

Tensorflow /Keras

Our project centers on the classification of respiratory sounds utilizing CNN. The implementation of Convolutional Neural Networks (CNNs) is accomplished through TensorFlow. TensorFlow, known for its strength and adaptability in machine learning, empowers us to effectively build, train, and refine these CNN models. It offers an extensive array of ready-made layers, activation functions, and optimization algorithms, streamlining the CNN architecture's design and enhancement process.

Librosa

In our project, we are utilizing Librosa, a powerful Python library specifically designed for audio and music analysis. Librosa provides essential tools for working with audio data, which is fundamental to our goal of classifying respiratory sounds. This library allows us to load audio files, extract critical features from the audio, and generate spectrograms, providing a clear visual representation of frequency content over time. These features are crucial for training our machine learning models, especially Convolutional Neural

Networks (CNNs) implemented using TensorFlow. By leveraging the capabilities of Librosa, we enhance our project's ability to efficiently process respiratory sound data, ultimately contributing to the accurate classification of respiratory conditions.

Hyperopt

Hyperopt, an open-source Python library, is created to optimize hyperparameters in machine learning models. It uses Bayesian optimization, a method that efficiently navigates the hyperparameter space by learning from previous attempts. This makes it especially effective for fine-tuning machine learning models.

NumPy

NumPy, an abbreviation for Numerical Python, stands as a fundamental and robust Python library dedicated to numerical computing. It offers assistance for arrays, matrices, and a variety of mathematical functions to efficiently perform operations on these data structures.In our project, we heavily rely on NumPy. NumPy acts as a cornerstone, aiding us in managing large volumes of numerical data with ease, particularly during the analysis of respiratory sound data.

Matplotlib

Matplotlib, a basic Python package, is a useful tool for constructing a wide variety of visualizations. It has an abundance of plots, charts, histograms, and other features, making it a popular choice for data visualization in a variety of areas. Matplotlib helps us create plots showing frequency, amplitude, and spectrograms of respiratory sound data. These visuals are crucial, offering insights into sound characteristics and aiding in the analysis of audio data for our project.

Pandas

In our project, we make extensive use of Pandas, a powerful Python library for data manipulation and analysis. Specifically, we utilize Pandas to merge data from separate files containing demographic and diagnosis information. These files are crucial for generating labels for each audio file. By leveraging Pandas, we efficiently merge and organize this data, creating a comprehensive dataset that enriches the context for our respiratory sound classification. This structured data is then utilized to enhance the accuracy and relevance of our machine learning models.

Data

Two research teams based in Portugal and Greece collaborated to establish the Respiratory Sound Database, featuring 920 annotated recordings. The durations of these recordings vary from 10 seconds to 90 seconds. These recordings were obtained from a diverse pool of 126 unique patients. The database encompasses a total of 5.5 hours of recordings, encompassing 6898 respiratory cycles. Among these cycles, 1864 contain crackles, 886 contain wheezes, and 506 exhibit both crackles and wheezes.There are two types of data files: demographic information and annotation text.

Algorithms

CNN

A Convolutional Neural Network (CNN) is a powerful tool in deep learning, specifically designed to analyze and interpret visual and spatial information. CNNs are built to automatically learn patterns and features from data. They have convolutional, pooling, and fully connected layers, as well as activation functions, all working together.

CNNs are the cornerstone of our classification model, leveraging their effectiveness in processing audio data. These networks are specifically tailored to extract intricate features from the audio spectrograms, enabling precise classification of respiratory sounds.

In our project, we train and update CNNs to improve the accuracy and strength of our audio classification system. These CNNs are meticulously designed to recognise critical patterns, assisting in the differentiation of various respiratory diseases based on sound characteristics. By emphasizing CNNs' significance in audio categorization, we hope to improve respiratory sound analysis and healthcare diagnosis.

Bayesian optimisation

Bayesian optimization efficiently examines complex hyperparameter space. It uses a probabilistic model to forecast the best collection of hyperparameters for testing.In our project, we've used Bayesian optimization to fine-tune the hyperparameters of our Convolutional Neural Network (CNN). We create an objective function and Bayesian optimization suggests which hyperparameters to test next by considering the model's predictions. We repeat this process until we discover the optimal hyperparameters.

Hardware Requirements

CPU: Intel Core i7 or equivalent, quad-core or higher

RAM: 16 GB or higher

Storage: Minimum 256 GB SSD for operating system, software, and datasets

Software Requirements

Operating System: Windows 10 (64-bit) or Ubuntu 18.04 (or higher)

Programming Languages: Python 3.7 or higher

Development Environment: PyCharm or Visual Studio Code

Libraries/Frameworks:

- TensorFlow 2.0 or higher
- Keras 2.2 or higher
- scikit-learn

- NumPy
- pandas
- matplotlib
- Librosa for audio analysis

Additional Tools:

- Bayesian optimization libraries hyperopt for hyperparameter tuning
- Flask for building the user interface (UI)

CHAPTER FOUR

DESIGN

Flow of events

UML diagrams offer a clear visual representation of the components and interactions in complex machine learning systems. They are especially valuable when dealing with many elements and processes. UML diagrams assist in designing machine learning projects, helping plan and organize the system. They enable effective structuring of machine learning models, data preprocessing, and deployment components.

1.The user logs into the “Respiratory classification system” .

2. UI presents the options to select the input files

3. User selects the specific audio file recording

4.The “Respiratory classification system” retrieves the audio file and processes

5. The CNN model predicts the class based on model training

A Data Flow Diagram (DFD) is a visual way to show how data moves within a system or process. It helps depict how data is processed, where it comes from (data sources), where it goes (data destinations), and where it's stored. DFDs are widely used in software engineering and systems analysis to understand and communicate how data is managed in an organization or software.

A Data Flow Diagram (DFD) *Figure1* is a graphical representation that illustrates how data flows within a system, showing the processes, data sources, data destinations, and data transformations. In the context of the "Respiratory Classification System," I'll explain the theoretical DFD for the steps you've described:

1. User logs into the “Respiratory Classification System” (Process 1):

In the DFD, you would represent this as a process labeled "User Login." This process takes user authentication data as input and produces a successful login status as output.

- Input: User authentication data
- Output: Successful login status

2. UI presents the options to select the input files (Process 2):

- This can be represented as a process called "Display Input Options" or similar. It takes the user's logged-in status as input and presents the available options for selecting input files (e.g., audio recordings).

- Input: User login status
- Output: Options for selecting input files

3. User selects the specific audio file recording (Process 3):

- This is a user action and doesn't involve a traditional process in a DFD. You can represent it with an arrow indicating data flow from the user to the selected audio file.

4. The “Respiratory Classification System” retrieves the audio file and processes (Process 4):

- This process can be labeled "Data Retrieval and Processing." It takes the selected audio file as input and performs various actions, including data processing.

- Input: Selected audio file

- Output: Processed data

5. The CNN model predicts the class based on model training (Process 5):

- This process represents the core of the classification system. It's labeled "CNN Model Prediction" and takes the processed data as input to make predictions based on the trained Convolutional Neural Network (CNN) model.

- Input: Processed data

- Output: Predicted class

In a DFD, these processes, data flows, and data stores (not mentioned in your description but may be relevant) are connected with arrows to show the flow of data between them. External entities, such as the user and the CNN model, are also depicted in the diagram. The DFD helps to provide a clear and visual representation of how data and processes interact within the "Respiratory Classification System."

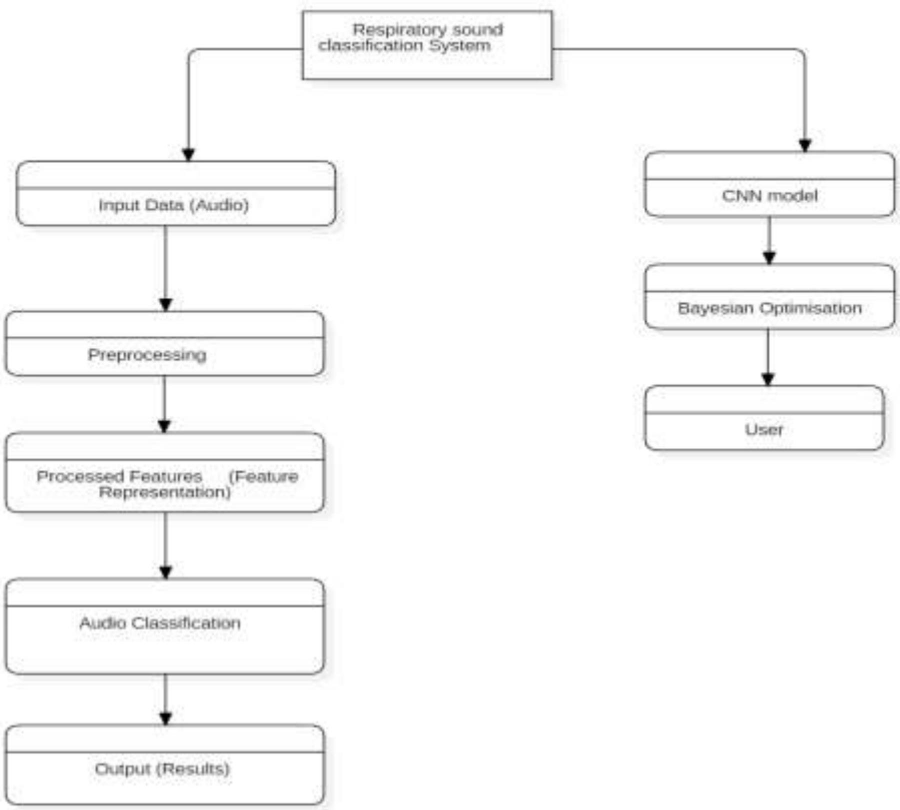

Figure1.Data Flow diagram

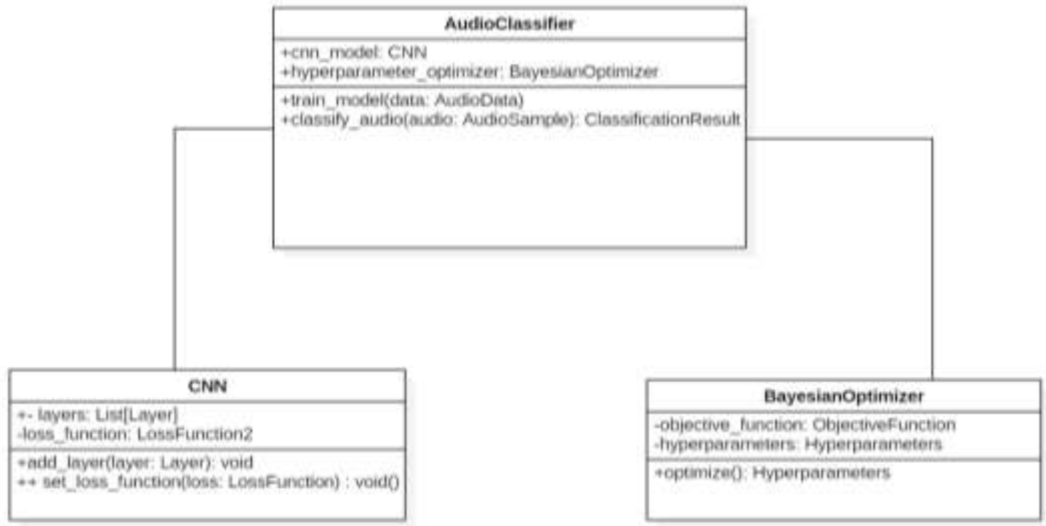

Figure2.Class diagram

A class diagram *Figure2* is a visual representation in UML that illustrates classes, attributes, and relationships in a software system. In the "Respiratory Classification System" context, we have three main classes: "AudioClassifier," "CNN," and "BayesianOptimization." Let's create a basic class diagram for these classes.

The connections between these classes are depicted using arrows, signifying associations or dependencies. For instance, you can establish associations between the "AudioClassifier" and "CNN" classes to indicate that the audio classifier relies on the CNN model for classification. These associations also represent interactions and relationships between the classes

- The "AudioClassifier" class represents the high-level functionality for audio classification. It has attributes related to hyperparameter tuning, CNN model training, and the classification process.

- The "CNN" class represents the Convolutional Neural Network and its related attributes, such as the list of layers and the loss function used for training the model.

- The "BayesianOptimization" class is responsible for hyperparameter optimization. It includes attributes like the objective function, hyperparameters to be optimized, and the optimization process.

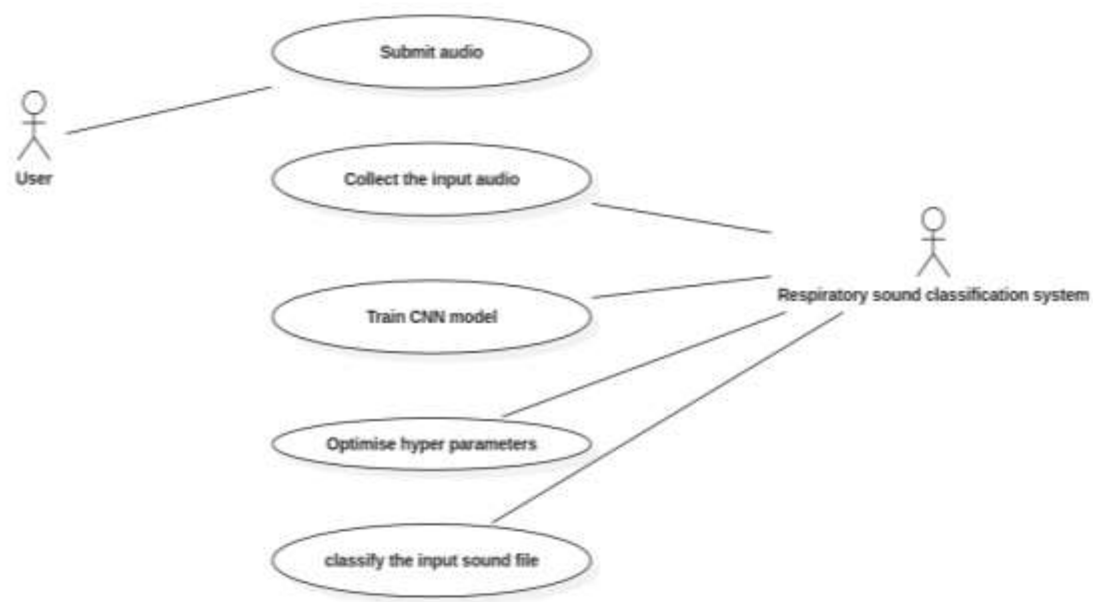

Figure3. Use Case diagram

A use case diagram *Figure3* visually represents the interactions between actors (in this case, "User" and "Classification System") and the various processes (use cases) they engage in. Here's a description of the use case diagram for the "Respiratory Classification System":

In this use case diagram, "User" interacts with the system by submitting audio recordings, and the "Classification System" performs a series of processes to analyze and classify the respiratory data accurately. The diagram provides a high-level overview of how these actors and processes interact within the "Respiratory Classification System."

Use Case Diagram Description:

1. Actors:

- User: The primary actor in the system, representing individuals who interact with the "Respiratory Classification System" to submit audio recordings and receive diagnosis results.

- Classification System: This actor embodies the automated system responsible for processing audio recordings, training machine learning models, optimizing hyperparameters, and classifying input data into different respiratory diagnosis types.

2. Use Cases(Processes):

- Submit Audio: This use case represents the action where the "User" submits audio recordings of respiratory sounds for analysis. The system receives and processes the audio data.

- Collect Input Audio: The process by which the "Classification System" collects and stores the user-provided audio data for further analysis and classification.

- Train CNN Model: This use case involves the "Classification System" training its Convolutional Neural Network (CNN) model. The system utilizes a dataset to improve the model's ability to classify respiratory conditions accurately.

- Optimize Hyperparameters: The "Classification System" fine-tunes the hyperparameters of its machine learning models, such as the CNN, to enhance their performance in diagnosing respiratory conditions.

- Classify Input into Type of Diagnosis: This use case signifies the core functionality of the system. It involves processing the input audio data using the trained models and classifying it into specific types of respiratory diagnoses, such as Chronic Obstructive Pulmonary Disease (COPD), pneumonia, etc.

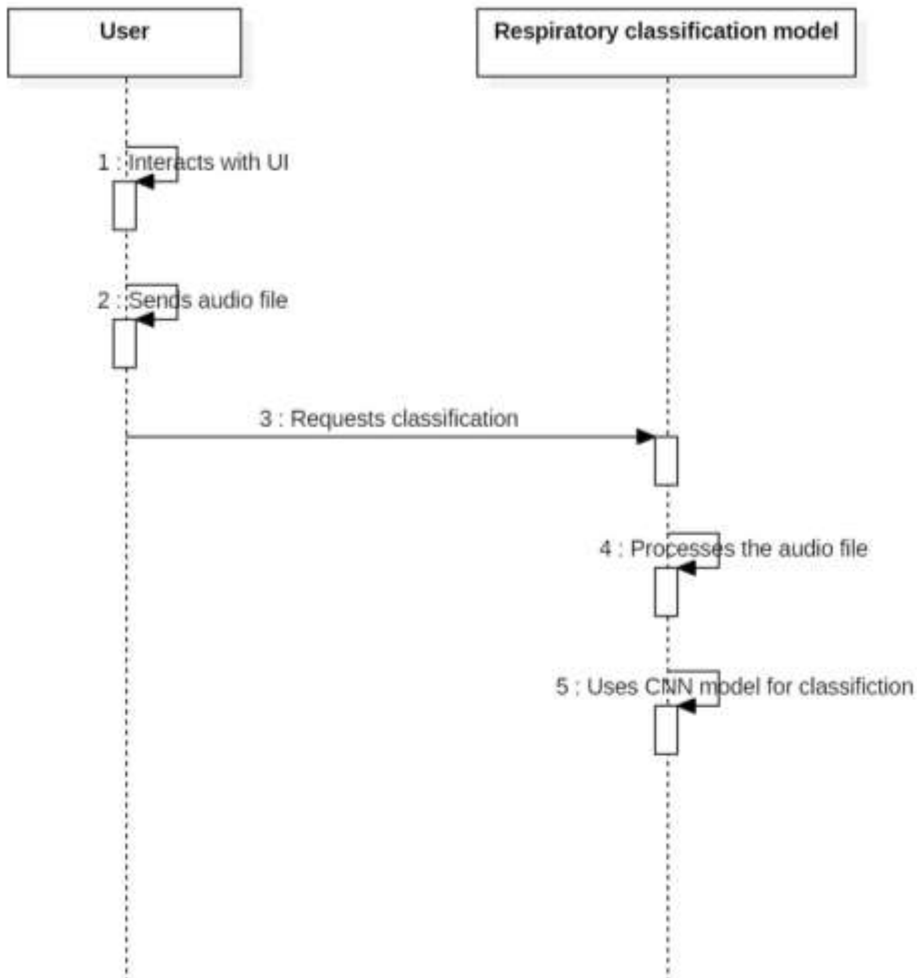

Figure.4.Sequence diagram

A sequence diagram *Figure.4* visually depicts the interactions and order of events between different objects or components in a system. In this case, we have two sequences: one representing the "User" and the other representing the

"Respiratory Classification Model." Here's a description of the sequence diagram for the interactions between these two sequences:

Sequence Diagram Description:

1. User Sequence:

- Interacts with UI: The user initiates the interaction by interacting with the User Interface (UI). This includes actions such as accessing the system, selecting options, and submitting requests.
- Sends Audio File Request: After interacting with the UI, the user sends a request to the system for processing an audio file. This request includes uploading an audio recording of respiratory sounds for diagnosis.

2. Respiratory Classification Model Sequence:

- Processes the Audio File: Upon receiving the audio file request, the Respiratory Classification Model sequence comes into action. It processes the audio file by performing tasks like data preprocessing and feature extraction to make the audio data suitable for analysis.
- Uses CNN Model for Classification: The Respiratory Classification Model then utilizes its Convolutional Neural Network (CNN) model for the classification of the processed audio data. The CNN model is responsible for identifying and categorizing respiratory conditions based on the audio input.

3. Interactions and Messages:

The sequence diagram includes arrows and vertical lines to represent the flow of interactions and messages between the "User" and the "Respiratory Classification Model."

Messages and actions are depicted sequentially, showing the order in which they occur.

In this sequence diagram, it's evident how the user's actions trigger the Respiratory Classification Model to process audio data and utilize the CNN model for classification. The diagram provides a chronological view of the interactions and activities between these two sequences in the context of the "Respiratory Classification System."

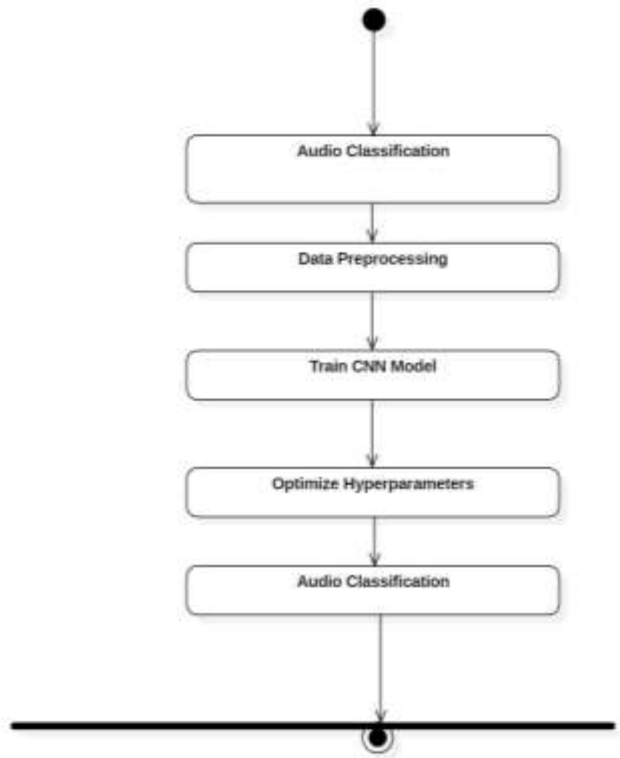

Figure 5.Activity diagram

An activity diagram *Figure 5* is a visual representation that illustrates the flow of activities or processes within a system. In the context of the "Respiratory Sound Classification System," you can create an activity diagram to show the steps involved in the classification process. Here's a description of the activity diagram for this system:

Activity Diagram Description:

1. Data Preprocessing Activity:

The first major step is data preprocessing. This activity encompasses tasks such as cleaning and formatting the audio data, extracting relevant features, and preparing it for further analysis.

2. Train CNN Model Activity:

Once the data is preprocessed, the system proceeds to train a Convolutional Neural Network (CNN) model. This step involves using the preprocessed data to train the model so that it can recognize patterns and features in respiratory audio recordings.

3. Optimize Hyperparameters Activity:

After training the initial CNN model, the system engages in hyperparameter optimization. This process fine-tunes the model by adjusting hyperparameters like learning rates, batch sizes, and epochs to improve its performance.

5. Audio Classification Activity:

The final step is audio classification. Here, the trained and optimized CNN model is applied to classify the input audio data into specific respiratory condition categories. This step provides the diagnosis based on the audio recording.

6. End Activity:

- The activity diagram concludes with an end activity, signifying the completion of the classification process.

The activity diagram outlines the following steps:

1. Start the classification process.

2. Perform data preprocessing to prepare audio data.

3. Train the CNN model.

4. Optimize hyperparameters for the model.

5. Apply the model for audio classification.

6. Conclude the classification process.

This activity diagram provides a clear visual representation of the sequence of activities and their flow in the "Respiratory Sound Classification System." It helps to understand the logical order of processes and their dependencies.

CHAPTER FIVE

IMPLEMENTATION

Data preparation

In our audio dataset, each file name includes both the sound data and the device name. Additionally, we have numerous accompanying text files containing valuable demographic and diagnosis details. To extract insights from this information, we employed the pandas library in Python.

We carefully analyzed the demographic and diagnosis data present in these text files. Leveraging the unique patient ID column as a reference point, we merged these dataframes. This merging process allowed us to generate labels that were then associated with the respective sound data. Utilizing pandas facilitated a structured and organized approach to efficiently manage and process this essential information, enriching our dataset for further analysis.

Data analysis

Data analysis is a crucial part of the respiratory classification system, as it provides insights into the quality and characteristics of the dataset used to train and test the machine learning models. In the *Figure 6* data analysis phase, a significant and noteworthy observation is the presence of class imbalance within the target variables, which represent different respiratory diseases. This imbalance affects the distribution of the dataset with respect to these diseases, and it has important implications for model training and prediction accuracy. The "COPD" class is the dominant class in the dataset. In contrast, the "Asthma" class exhibits a lower representation in the dataset. Similarly, the "LRTI" class is observed to be less represented in the dataset.The class imbalance in the dataset has several implications for the machine learning models and the system's predictive performance.

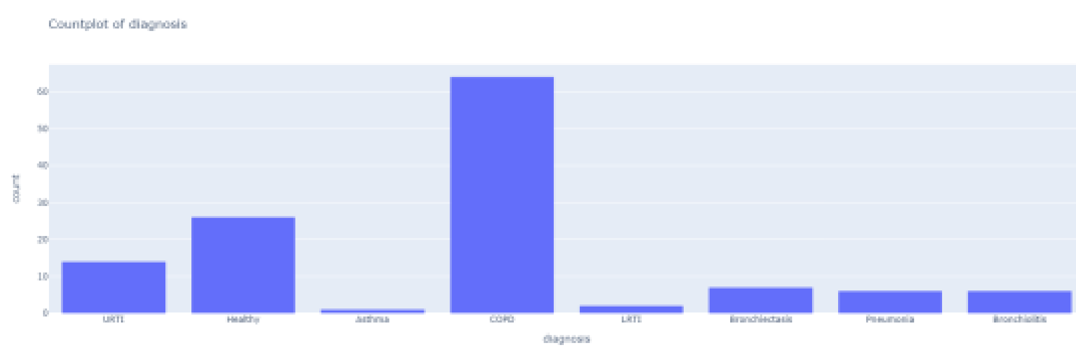

Figure 6.Distribution of diagnosis

Base model

The Convolutional Neural Network (CNN) architecture we implemented for the respiratory sound classification is structured to effectively extract features and classify audio data.

In Figure 7 the first layer, Conv1D, employs 64 filters with a kernel size of 3, aiming to capture local patterns in the audio data. Following this, max pooling is applied to reduce spatial dimensions, focusing on the most relevant features. This process is repeated with another Conv1D layer of 128 filters and kernel size 3, further enhancing feature extraction. Subsequently, max pooling is again employed to consolidate significant information. The resulting features are then flattened to create a one-dimensional vector, enabling effective input to subsequent dense layers. The dense layers help in learning high-level abstractions by employing 256 neurons. A dropout layer is incorporated to prevent overfitting. The final dense layer with 8 neurons represents the output classes corresponding to respiratory conditions. The Convolutional Neural Network (CNN) architecture described above represents the base model for our respiratory sound classification project

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 11, 64)            19264

 max_pooling1d (MaxPooling1D  (None, 5, 64)            0
 )

 conv1d_1 (Conv1D)           (None, 3, 128)            24704

 max_pooling1d_1 (MaxPooling  (None, 1, 128)           0
 1D)

 flatten (Flatten)           (None, 128)               0

 dense (Dense)               (None, 256)               33024

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 8)                 2056

=================================================================
Total params: 79,048
Trainable params: 79,048
Non-trainable params: 0
```

Figure 7.Base model architecture

Bayesian Optimisation

In our project, we initially developed a base model and achieved a validation accuracy of 83%. To enhance this accuracy, we decided to utilize Bayesian optimization, a technique available through the Hyperopt framework. We focused the optimization on key model parameters: convolutional units, dense units, and dropout rate. The process was set to run a maximum of 30 evaluations, aiming to fine-tune these parameters and improve the overall model performance. The improved accuracy after optimization is 85%.The Figure 8 and Figure 9 show the model accuracy and loss curves for the classification model.

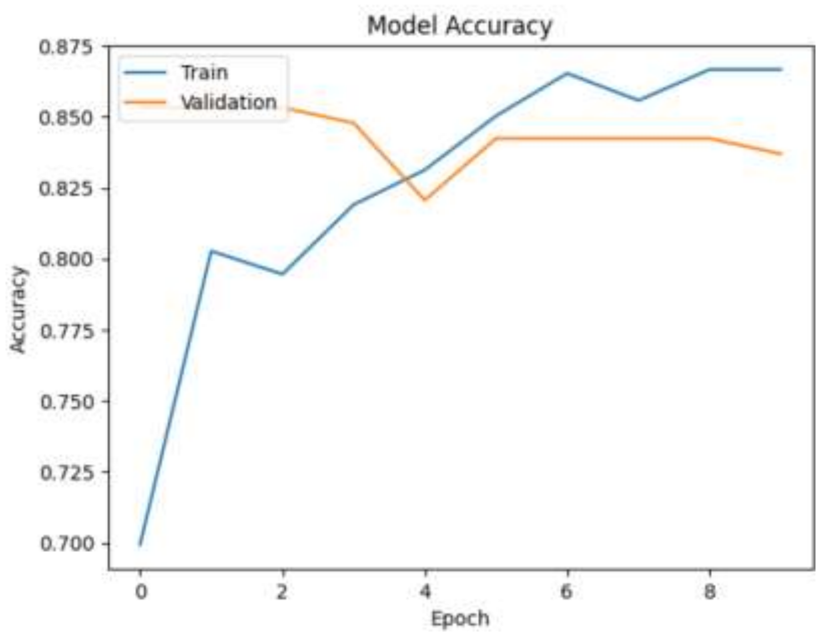

Figure 8.Model training and validation accuracy

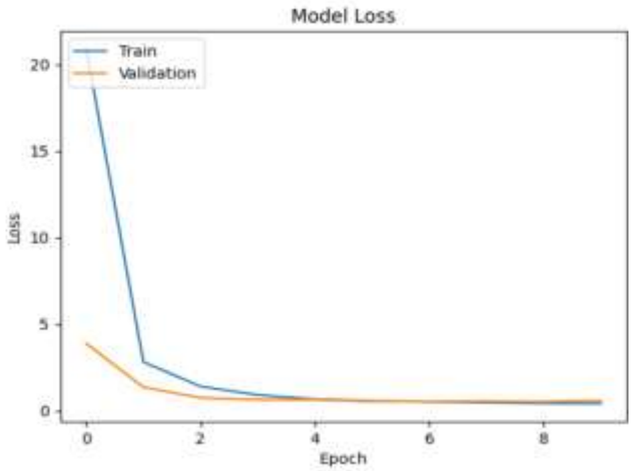

Figure 9. Model loss

A confusion matrix is a table that is used to evaluate the performance of a classification model. It presents a comprehensive summary of the model's predictions compared to the actual ground truth. Figure 10 shows that the highest true positive rate is for COPD.

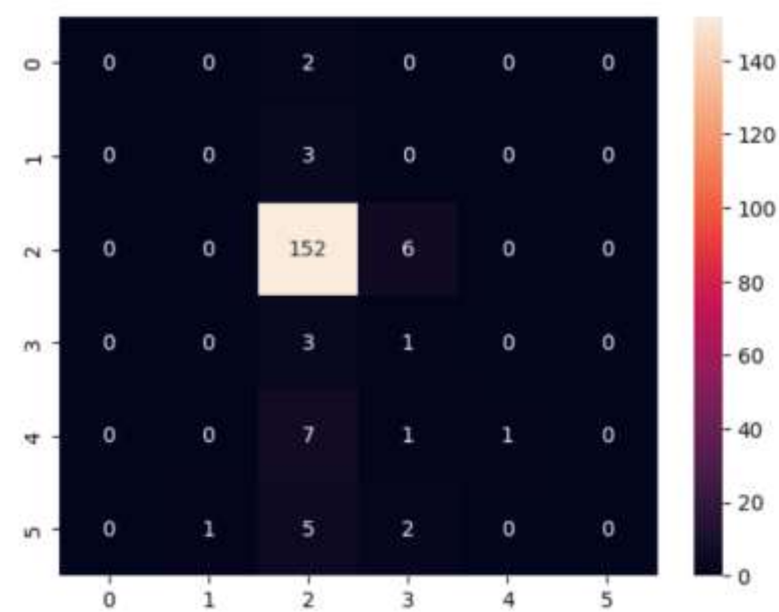

Figure 10.Confusion matrix

Integration with UI

In our project, we used Flask, a Python web framework, to construct the UI. Flask acts as a vital connection between the UI that users interact with and the backend where the trained machine learning model operates. Users can upload respiratory sounds, trigger the classification process, and visualize the results.

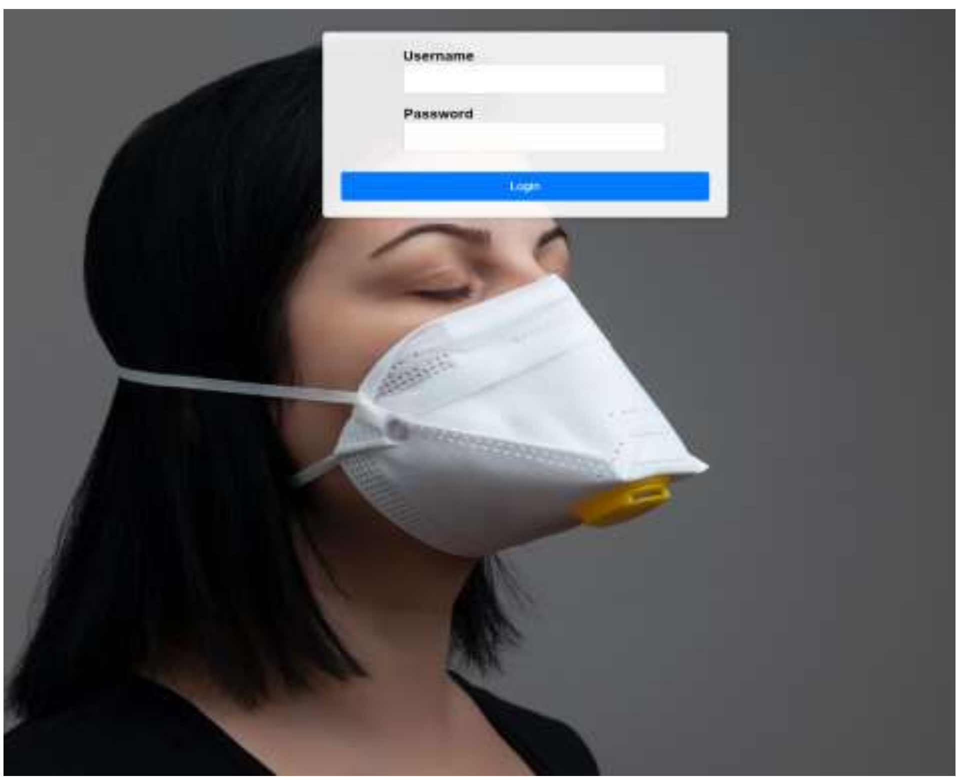

Figure 11 .Login page

The system features a secure login page *Figure 11* where users are required to authenticate their identity. Default credentials are verified to ensure authorized access.Upon successful login, users are directed to the home page

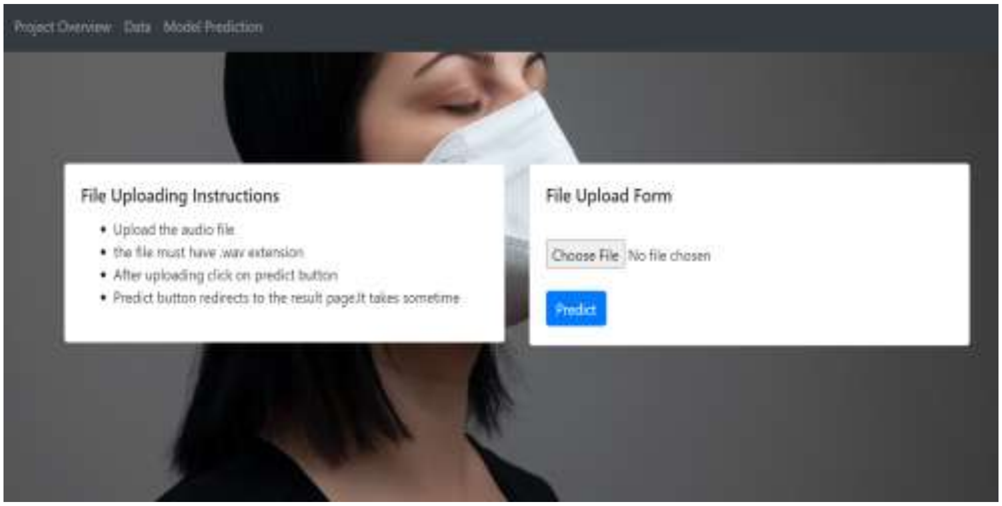

Figure 12.Home page

Figure 12, which serves as the central hub for system functionality. Project Overview tab provides an overview of the system's purpose, goals, and its potential impact on respiratory disease classification. Users can access data insights, including visualizations and statistical information about the dataset used for training and testing the classification models.

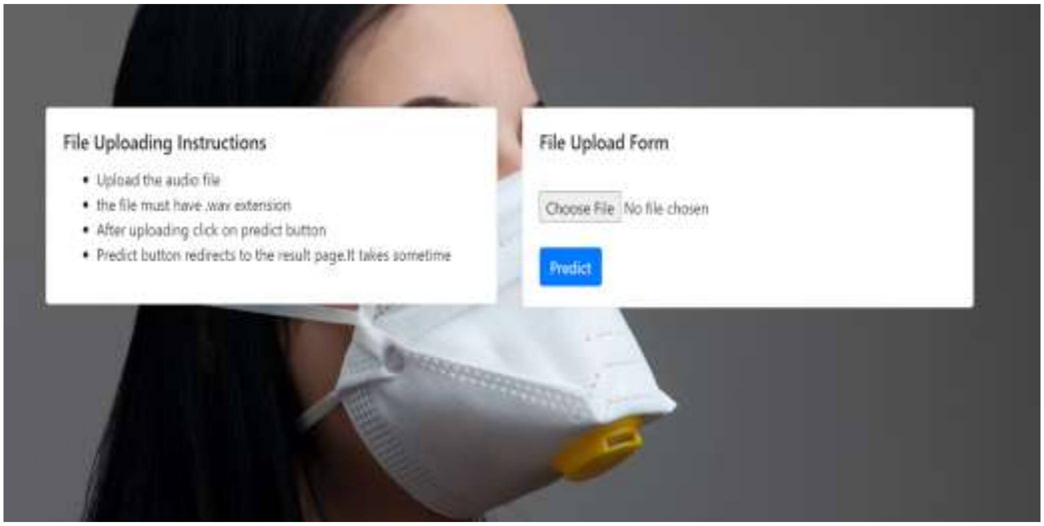

Figure 13.Diagnosis prediction page

Figure 13 tab, users can upload an audio file containing respiratory sounds. The system then processes the audio data through pre-trained machine learning models to predict the presence of specific respiratory diseases. The results are displayed to the user, offering valuable insights into the potential diagnosis.

REFERENCES

[1] Z. Zhao et al., "Automatic Respiratory Sound Classification Via Multi-Branch Temporal Convolutional Network," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 9102-9106, doi: 10.1109/ICASSP43922.2022.9746182.

[2] Q. Zhang et al., "SPRSound: Open-Source SJTU Paediatric Respiratory Sound Database," in IEEE Transactions on Biomedical Circuits and Systems, vol. 16, no. 5, pp. 867-881, Oct. 2022, doi: 10.1109/TBCAS.2022.3204910.

[3] Q. Zhang et al., "Grand Challenge on Respiratory Sound Classification for SPRSound Dataset," 2022 IEEE Biomedical Circuits and Systems Conference (BioCAS), Taipei, Taiwan, 2022, pp. 213-217, doi: 10.1109/BioCAS54905.2022.9948551.

[4] J. Li, X. Wang, X. Wang, S. Qiao and Y. Zhou, "Improving The ResNet-based Respiratory Sound Classification Systems With Focal Loss," 2022 IEEE Biomedical Circuits and Systems Conference (BioCAS), Taipei, Taiwan, 2022, pp. 223-227, doi: 10.1109/BioCAS54905.2022.9948543.

[5] K. Minami, H. Lu, H. Kim, S. Mabu, Y. Hirano and S. Kido, "Automatic Classification of Large-Scale Respiratory Sound Dataset Based on Convolutional Neural Network," 2019 19th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea (South), 2019, pp. 804-807, doi: 10.23919/ICCAS47443.2019.8971689.

[6] M. A. Awal et al., "An Early Detection of Asthma Using BOMLA Detector," in IEEE Access, vol. 9, pp. 58403-58420, 2021, doi: 10.1109/ACCESS.2021.3073086.

[7] S. Tewari, U. Agrawal, S. Verma, S. Kumar and S. Jeevaraj, "Ensemble Model for COVID-19 detection from chest X-ray Scans using Image Segmentation, Fuzzy Color and Stacking Approaches," 2020 IEEE 4th Conference on Information & Communication Technology (CICT), Chennai, India, 2020, pp. 1-6, doi: 10.1109/CICT51604.2020.9312076.

[8] R. Du et al., "Identification of COPD From Multi-View Snapshots of 3D Lung Airway Tree via Deep CNN," in IEEE Access, vol. 8, pp. 38907-38919, 2020, doi: 10.1109/ACCESS.2020.2974617.

[9] R. Du et al., "Identification of COPD From Multi-View Snapshots of 3D Lung Airway Tree via Deep CNN," in IEEE Access, vol. 8, pp. 38907-38919, 2020, doi: 10.1109/ACCESS.2020.2974617.

[10] Z. Farahany, J. Wu, K. M. Sajjadul Islam and P. Madiraju, "Oversampling techniques for predicting COVID-19 patient length of stay," 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 5253-5262, doi: 10.1109/BigData55660.2022.10020253.

[11] M. Moradi, M. Kordestani, A. Chaibakhsh and M. Saif, "Enhanced COVID-19 Detection by chest x-ray images using transfer learning-based extracted deep features and information fusion," 2023 International Conference on Control,

Automation and Diagnosis (ICCAD), Rome, Italy, 2023, pp. 1-8, doi: 10.1109/ICCAD57653.2023.10152327.

[12] S. Kulkarni, H. Watanabe and F. Homma, "Self-Supervised Audio Encoder with Contrastive Pretraining for Respiratory Anomaly Detection," 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSPW59220.2023.10193030.

[13] L. Pham, H. Phan, R. Palaniappan, A. Mertins and I. McLoughlin, "CNN-MoE Based Framework for Classification of Respiratory Anomalies and Lung Disease Detection," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 8, pp. 2938-2947, Aug. 2021, doi: 10.1109/JBHI.2021.3064237.

[14] L. Pham, I. McLoughlin, H. Phan, M. Tran, T. Nguyen and R. Palaniappan, "Robust Deep Learning Framework For Predicting Respiratory Anomalies and Diseases," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 2020, pp. 164-167, doi: 10.1109/EMBC44109.2020.9175704.

[15] L. Pham, H. Phan, A. Schindler, R. King, A. Mertins and I. McLoughlin, "Inception-Based Network and Multi-Spectrogram Ensemble Applied To Predict Respiratory Anomalies and Lung Diseases," 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Mexico, 2021, pp. 253-256, doi: 10.1109/EMBC46164.2021.9629857.