12-2023

DISEASE OF LUNG INFECTION DETECTION USING CNN MODEL -BAYESIAN OPTIMIZATION

poojitha gutha

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 plays a role, in diagnosing and identifying diseases during examinations. However it requires training and expertise, for application. This study aims to tackle this challenge by introducing a model that categorizes respiratory sounds into eight groups: URTI, Healthy, Asthma, COPD, LRTI, Bronchiectasis, Pneumonia and Bronchiolitis. To achieve this categorization the study utilizes a Convolutional Neural Network (CNN) model that has been optimized using techniques. The dataset used in the study consists of 920 audio samples obtained from 126 patients with 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. It is of importance to monitor our respiratory system in order to prevent diseases related to respiration [3]. In this research we aim to investigate the efficacy of a method called MBTCNSE. This method combines a branch Temporal

Convolutional Network (TCN) with a Squeeze and Excitation Network (SEnet) in order to enhance the classification of respiratory sounds [4][5].

Understanding the condition of the lungs and being able to diagnose respiratory issues is extremely important, in the medical field. To assist healthcare professionals it is crucial to have automated software that can analyze sounds as it can speed up the diagnosis process and lighten the workload for doctors. In our research we are focusing on using recordings from the RALE database [6][7] to differentiate between normal lung sounds those with airway obstructions and those with parenchymal problems.

By utilizing a acoustic sensor and a smartphone for classifying respiratory sounds we can greatly enhance asthma diagnosis and management. This innovative approach has the potential to improve long term care for asthma patients while also reducing treatment costs. Additionally it optimizes power consumption during signal acquisition, which is especially important, for systems that rely on battery power [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 widely used in the fields of machine learning and data analysis due, to its robustness and versatility. Its extensive library support, user interface and active community make it an excellent choice for our project.

We rely on Python as the programming language to connect us with tools and frameworks like TensorFlow, Keras, scikit learn and Flask. It plays a role in developing respiratory sound categorization models, particularly Convolutional Neural Networks (CNNs). Additionally Python efficiently handles tasks such as data organization and segmenting respiratory sound data. Moreover we leverage Python for Bayesian optimization techniques to determine the settings for our machine learning model.

Tensorflow /Keras

Our project focuses on using CNN to classify sounds. We use TensorFlow to implement Convolutional Neural Networks (CNNs) which is well known for its power and flexibility, in machine learning. With TensorFlow we can efficiently create, train and improve these CNN models. It provides a range of built layers, activation functions and optimization algorithms that make designing and enhancing the CNN architecture much easier.

Librosa

In our project we're using Librosa, a Python library specifically designed for analyzing audio and music. Librosa offers tools for working with data, which is crucial, for our goal of categorizing respiratory sounds. This library enables us to load files extract features from the sound and create spectrograms that visually represent frequency content over time. These features play a role, in training our machine learning models, Convolutional Neural Networks (CNNs) implemented through TensorFlow. By utilizing the capabilities of Librosa we enhance our projects efficiency in processing sound data ultimately contributing to the identification of respiratory conditions.

Hyperopt

Hyperopt, a Python library that's source was developed with the purpose of optimizing hyperparameters, in machine learning models. It utilizes optimization, which's a highly efficient method for exploring the hyperparameter space by drawing insights, from past trials. This particular approach proves to be particularly effective when it comes to refining and enhancing machine learning models.

NumPy

NumPy, short, for Numerical Python is an reliable Python library that focuses on computation. It provides support for arrays matrices and various mathematical functions to carry out operations on these data structures. In our project we heavily depend on NumPy as it serves as a tool, in handling amounts of numerical data effortlessly especially during the analysis of respiratory sound data.

Matplotlib

Matplotlib, an used Python package is a tool, for creating diverse visual representations. It offers options for plots, charts, histograms and more making it a popular choice in fields when it comes to presenting data visually. One of the benefits of Matplotlib is its ability to generate plots that display the frequency, amplitude and spectrograms of sound data. These visualizations play a role in providing insights into the characteristics of sound and assisting us in analyzing data, for our project.

Pandas

In our project we heavily rely on Pandas, a Python library that allows us to manipulate and analyze data. Specifically we use Pandas to combine information from files that contain diagnosis data. These files play a role, in labeling each file. By utilizing Pandas we efficiently. Organize this data creating a dataset that adds valuable context, to our respiratory sound classification. With this data we can improve the accuracy and relevance of our machine learning models.

Data

Two research teams, from Portugal and Greece collaborated to create the Respiratory Sound Database. This database consists of 920 annotated recordings with varying durations ranging from 10 seconds to 90 seconds. The recordings were collected from a group of 126 patients. In total the database contains 5.5 hours of recordings encompassing 6898 cycles. Out of these cycles crackles were present in 1864 instances wheezes in 886 instances and both crackles and wheezes in 506 instances. The data files, in this database include information and annotation text.

Algorithms

CNN

A Convolutional Neural Network (CNN) is a tool, in the field of learning specifically created to analyze and understand visual and spatial information. CNNs are designed to learn patterns and features from data. They consist of pooling and connected layers as well as activation functions, which all work together.

CNNs play a role in our classification model by processing audio data. These networks are specifically tailored to extract features from spectrograms allowing for accurate classification of respiratory sounds.

In our project we. Update CNNs to enhance the accuracy and strength of our classification system. These CNNs are carefully designed to identify patterns that aid, in differentiating 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

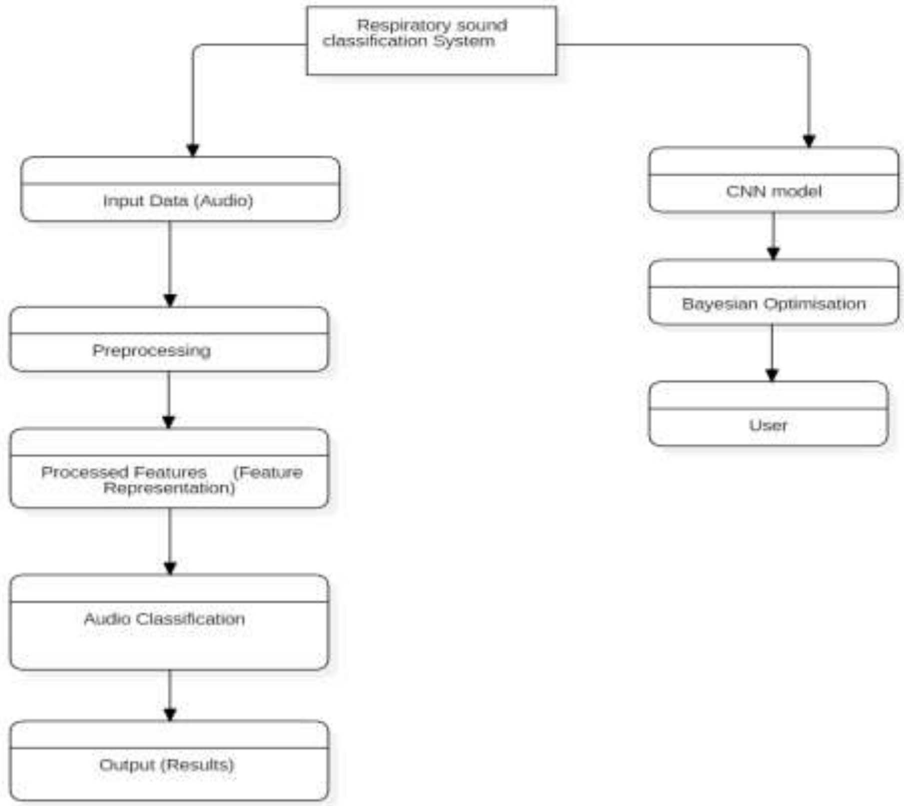

Figure1.Data Flow Diagram

Figure1 Data Flow Diagram (DFD) shows the graphical representation that how data flows within a system, showing the processes, data sources, data destinations, and data transformations. In the context of the "Respiratory Classification System”, The "Respiratory Classification System" operates through a structured workflow with five key processes. First, users log in, providing authentication data to achieve a successful login status. Once logged in, they're presented with options to select input files. The user's login status determines the available choices for selecting input files, typically audio recordings.

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

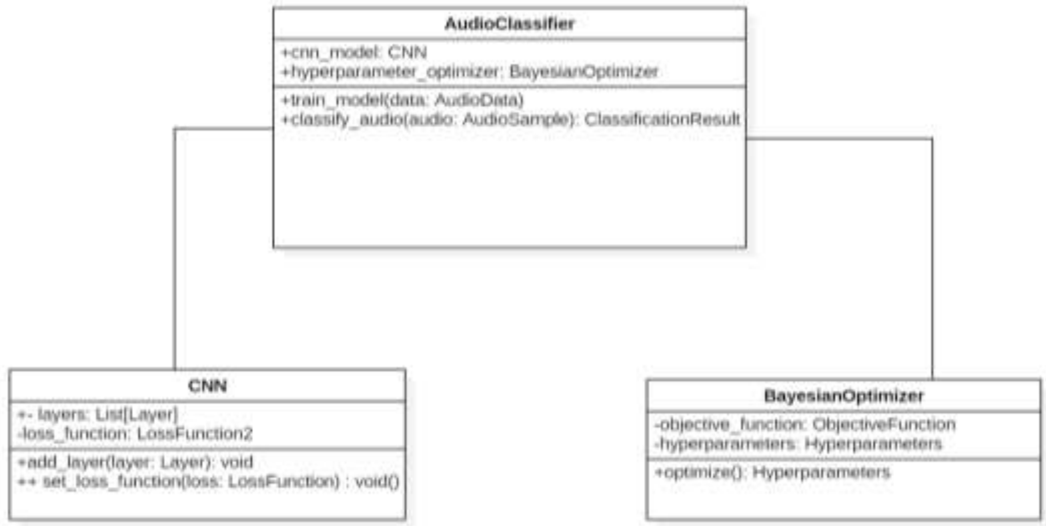

Figure2.Class Diagram

Figure2 Class diagram describe 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: "Audio Classifier," "CNN," and "Bayesian Optimization." Let's create a basic class diagram for these classes. The "Audio Classifier" class serves as the central component for audio classification, encompassing attributes related to hyper parameter tuning, CNN model training, and the actual classification process. The "Bayesian Optimization" class, on the other hand, is responsible for the crucial task of hyper parameter optimization. In the class diagram, connections between these classes, represented by arrows, signify associations and dependencies.

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 class known as "CNN" serves to represent the Convolutional Neural Network and its associated characteristics. These include the collection of layers and the loss function utilized in 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.

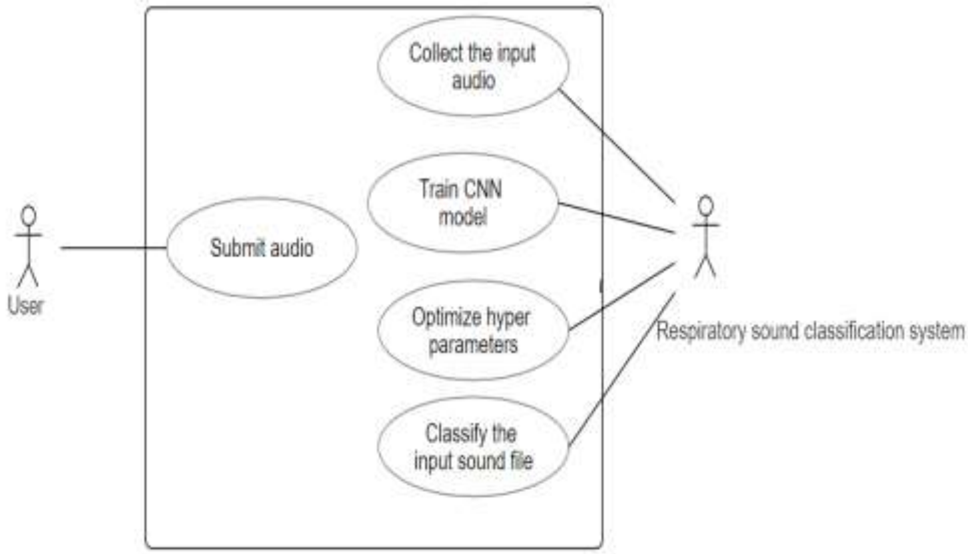

Figure3. Use Case Diagram

Figure3 visually represents the Use Case diagram it 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 main person involved in the system is someone who represents individuals interacting with the "Respiratory Classification System" to submit recordings and receive diagnosis results.

- Classification System: It is, like a person who operates the automated system for handling audio recordings. It also trains machine learning models optimizes hyperparameters and classifies input data into diagnosis types.

2. Use Cases(Processes):

- Submit Audio: In this scenario the "User" provides recordings of sounds, for analysis. The system then. Processes the 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: In this scenario the "Classification System" undergoes training, by refining its Convolutional Neural Network (CNN) model. By utilizing a dataset the system enhances its capability to effectively classify conditions, with accuracy.

- Optimize Hyperparameters: The "Classification System" optimizes the settings of its machine learning models like the CNN to improve their accuracy, in identifying conditions.

- Classify Input into Type of Diagnosis: This particular scenario represents the purpose of the system. It includes analyzing the data provided as input using trained models and categorizing it into types of respiratory diagnoses, such, as Chronic Obstructive Pulmonary Disease (COPD) pneumonia and so on.

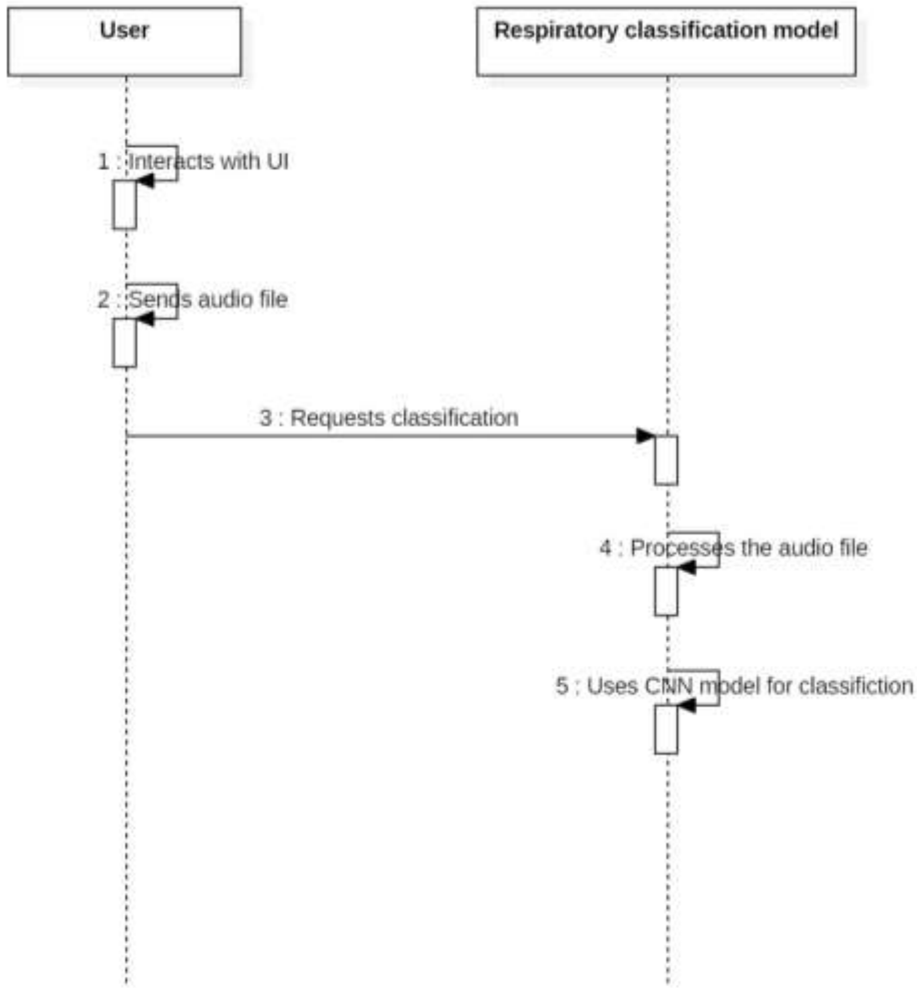

Figure.4.Sequence Diagram

Figure.4 characterizes the Sequence diagram, which illustrates the interactions and chronological order of events, among objects or components, within a system. In this scenario we have two sequences; one representing the "User" and the other representing the "Respiratory Classification Model." Here is a description of how these two sequences interact in the sequence diagram:

Sequence Diagram Description:

1. User Sequence:

- Interacts with UI: The user starts the interaction by engaging with the User Interface (UI). This involves actions, like accessing the system choosing options and making requests.
- Requesting Audio File: Following their interaction, with the UI the user asks the system to process a file. The request entails uploading a recording of sounds for purposes.

2. Respiratory Classification Model Sequence:

- Processes the Audio File: When the audio file request is received the Respiratory Classification Model kicks, into action. It starts by processing the file performing tasks like data preprocessing and feature extraction to prepare the data for analysis.
- Uses CNN Model for Classification: The processed audio data is then classified using a Convolutional Neural Network (CNN) model within the Respiratory Classification Model. This CNN model is responsible, for identifying and categorizing conditions based on the audio input.

3. Interactions and Messages:

The sequence diagram uses arrows and vertical lines to show how the "User" and the "Respiratory Classification Model" interact and exchange messages. It presents the actions, in a manner illustrating the order in which they take place.

This diagram clearly demonstrates how the users actions prompt the Respiratory Classification Model to analyze data and employ the CNN model for classification.

Essentially this diagram offers an overview of the interactions and operations between these two sequences, within the framework of the "Respiratory Classification System."

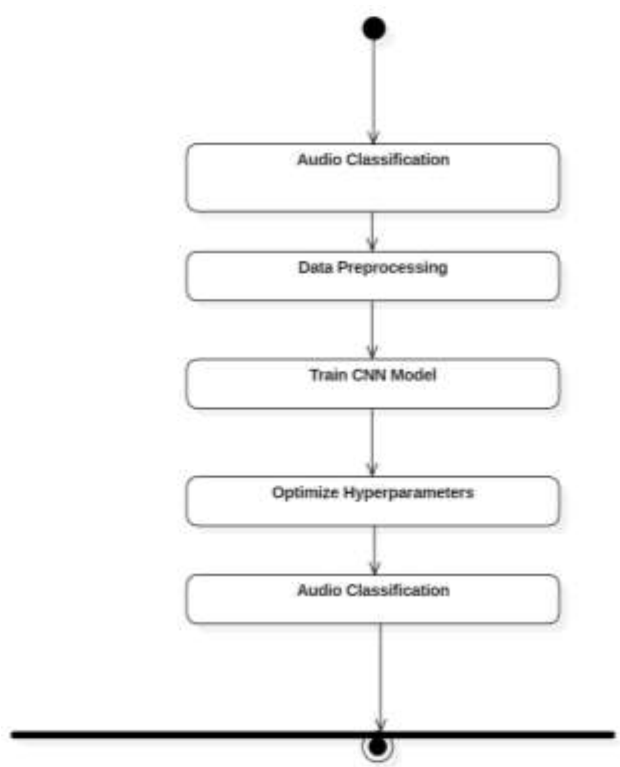

Figure 5.Activity Diagram

Activity diagrams as stated in Figure 5 depict the flow of activities or processes within a system. In the case of the "Respiratory Sound Classification System," an activity diagram can be created to illustrate the steps encompassed in the classification process. Below is a description of the activity diagram pertaining to this system:

Activity Diagram Description:

1. Data Preprocessing Activity:

The initial crucial phase involves preparing the data, for analysis. This includes tasks like cleaning and formatting the audio data extracting features and getting it ready, for examination.

2. Train CNN Model Activity:

After the data has been prepared the system moves on to training a Convolutional Neural Network (CNN) model. During this stage the preprocessed data is utilized to teach the model how to identify patterns and characteristics in recordings.

3. Optimize Hyperparameters Activity:

Once the initial CNN model has been trained the system proceeds with hyperparameter optimization. This crucial step involves tuning the model by making adjustments to hyperparameters such, as learning rates batch sizes and epochs. The goal is to enhance the models performance and achieve results.

4. Audio Classification Activity:

In the stage we perform classification. In this stage we utilize a trained and optimized CNN model to categorize the input audio data into respiratory condition categories. By doing we can provide a diagnosis based on the audio.

5. End Activity:

The activity diagram ends with a step, which indicates that the classification process has been completed. The activity diagram provides an overview of the following stages:

1. Begin the classification process.

2. Carry out data preprocessing to get audio data ready.

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.

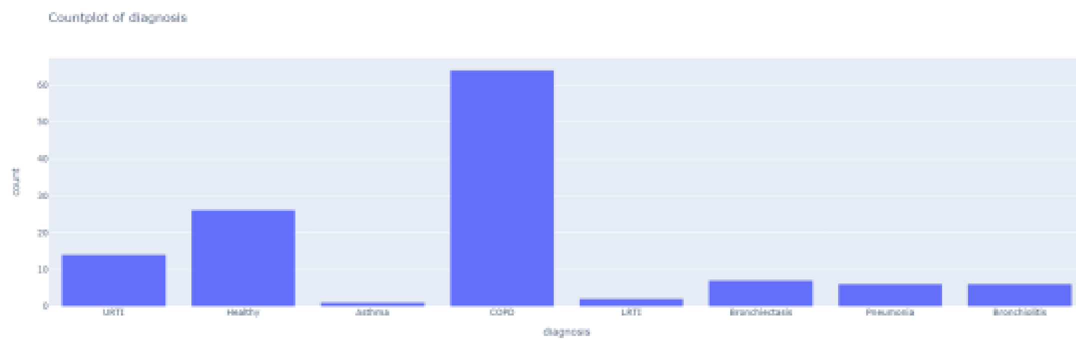

Figure 6. Diagnosis Distribution

Figure 6 represents the Distribution of diagnosis 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.

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.

Figure 7 is classify the base model architecture 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 over fitting. 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%.

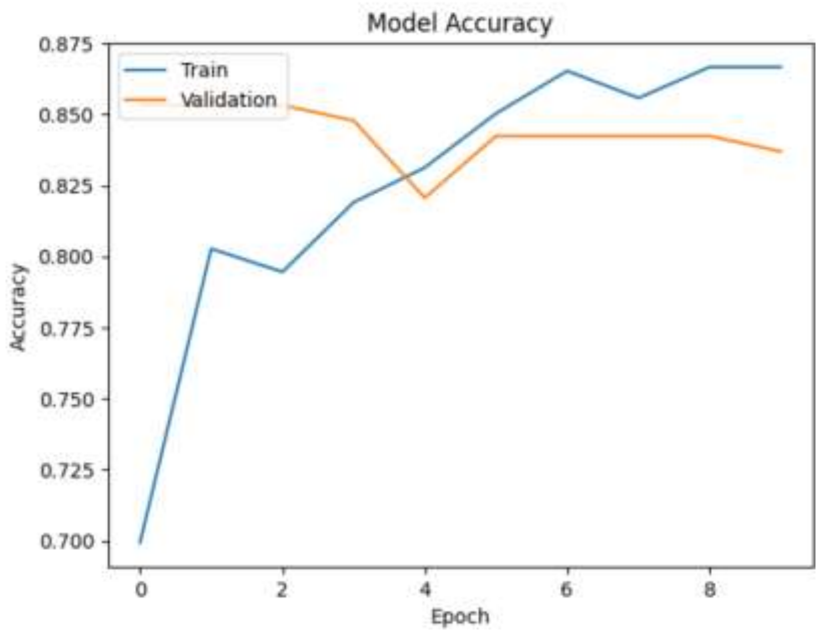

Figure 8.Model Training and Validation Accuracy

The Figure 8 represents the model accuracy curve which is a graphical representation of how well a machine learning model is correctly classifying or predicting data during training and validation. It's expressed as a line graph that shows the accuracy of the model on the y-axis and the number of training epochs (iterations) on the x-axis.

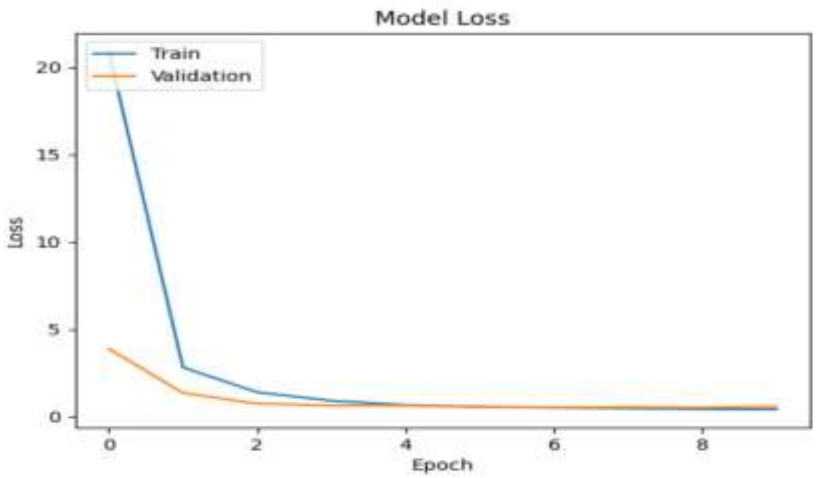

Figure 9. Model Loss

Figure 9 represents the model loss curves for the classification model. The loss curve reveals how effectively the model is converging toward minimizing errors. It should ideally show a steady decrease over time. The loss converged after 8 epochs for training and validation.

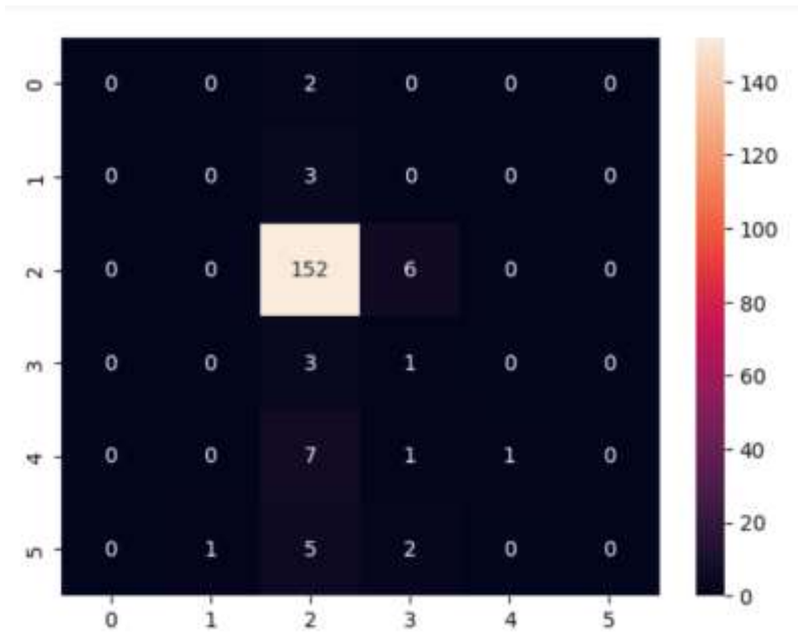

Figure 10.Confusion Matrix

Figure 10 represents the confusion matrix. It 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. It shows that the highest true positive rate is for COPD.

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.

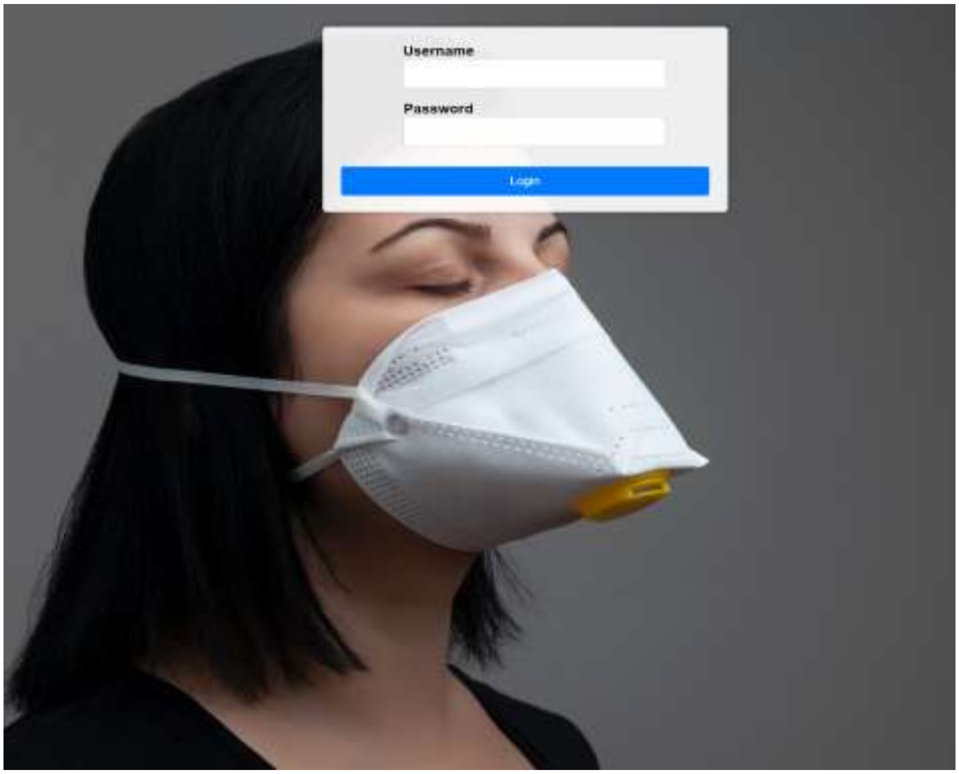

Figure 11 .Login Page

Figure 11 illustration the system features a secure login page 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.

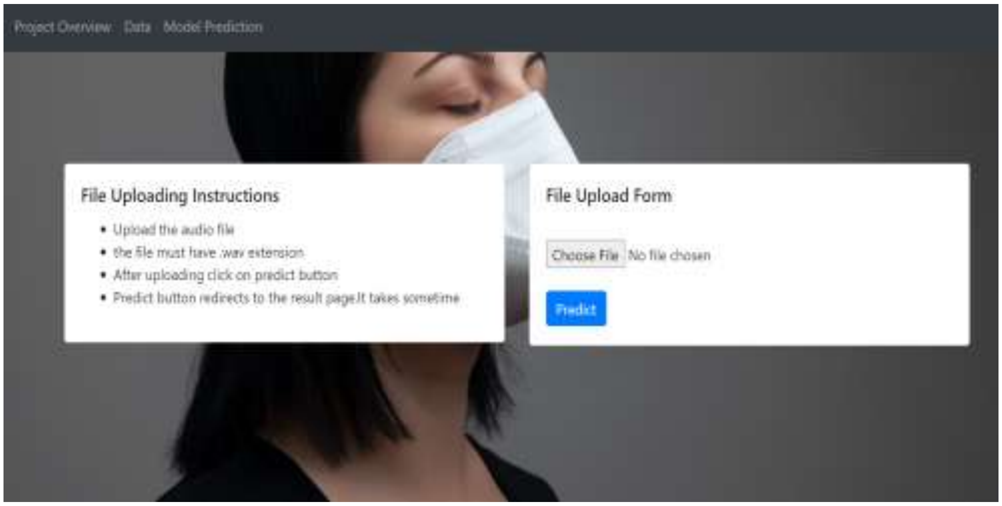

Figure 12.Home Page

Figure 12 shows 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.

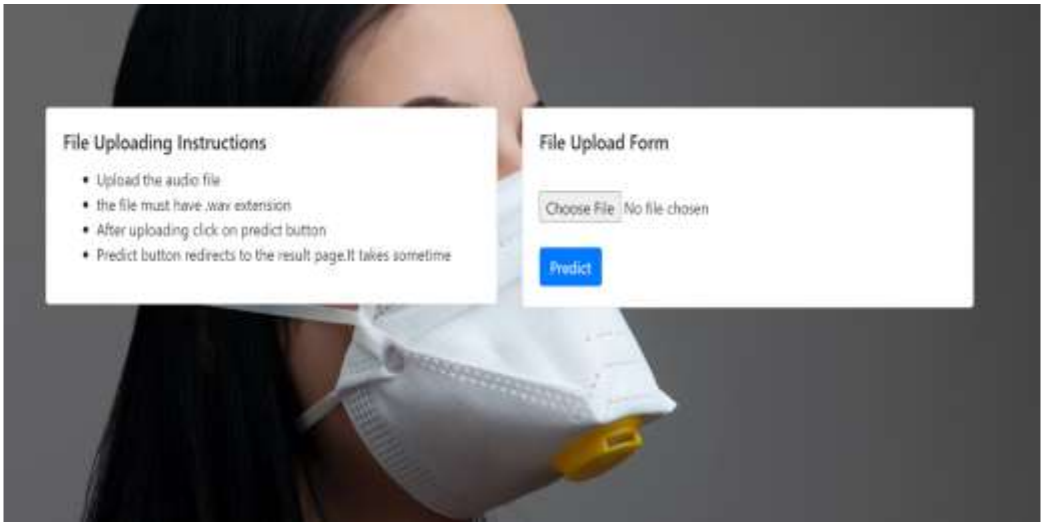

Figure 13.Diagnosis Prediction Page

Figure 13 represents the 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.