**ACKNOWLEDGEMENT**

We would like to express our deep sense of gratitude to our esteemed institute **Gayatri Vidya Parishad College of Engineering (Autonomous),** which has provided us an opportunity to fulfil our cherished desire.

We express our sincere thanks to our Principal **Dr. A.B. KOTESWARA RAO, Gayatri Vidya Parishad College of Engineering (Autonomous)** for his encouragement to us during this project, giving us a chance to explore and learn new technologies in the form of main project.

We express our deep sense of Gratitude to **Dr. R.V.V. Murali Krishna, Associate Professor and Head of the Department of Information Technology, Gayatri Vidya Parishad College of Engineering (Autonomous)** for giving us an opportunity to do the project in college.

We express our profound gratitude and our deep indebtedness to our guide **Mr. D. Arun Kumar, Assistant Professor, Department of Information Technology,** whose valuable suggestions, guidance, and comprehensive assistance helped us a lot in realizing my present project.

We also thank our coordinator, **Mr. B. Srinu, Assistant Professor, Department of Information Technology,** for the kind suggestions and guidance for the successful completion of our project work.

Finally, we would also like to thank all the members of the teaching and non-teaching staff of Information Technology Department for all their support in completion of our project.

**MEDAVARAPU ADITYA (Regd No:20131A1232)**

**RAHUL TAMANAM (Regd No:20131A1254)**

**MOHAN RAO BODASINGI (Regd No:20131A1246)**

**RAMADALAI KARTHIK (Regd No:20131A1212)**

**ABSTRACT**

Respiratory diseases are the world’s third leading cause of mortality. Early detection is critical in dealing with respiratory diseases, as it improves the effectiveness of intervention, including treatment and reducing the spread. The main aim of this application is to propose a novel lightweight inception network to classify a wide spectrum of respiratory diseases using lung sound signals. The proposed framework consists of three stages: 1) preprocessing; 2) mel spectrogram extraction and conversion into a three-channel image; and 3) classification of the mel spectrogram images into different pathological classes using the proposed lightweight inception network.

To analyze respiratory sounds on a computer, we developed a cost-effective and easy-to-use Algorithm that can be used with any device. Utilizing the proposed architecture, we have achieved a high classification accuracy of 96.6%, 99.6%, and 94.0% for seven class classification, six-class classification, and healthy versus asthma classification. To the best of our knowledge, this is the first work on seven-class respiratory disease classification using lung sounds. Whereas our proposed network outperforms all the existing published works for six-class and binary classifications. The suggested framework makes use of deep-learning methods and offers a standardized evaluation with strong categorization capabilities. In order to distinguish between a wide range of respiratory diseases, our study is a pioneering one that focuses exclusively on lung sounds. The proposed framework can be translated into real-time clinical application, which will facilitate the prospect of automated respiratory health screening using lung sounds.

**INDEX**

**1. INTRODUCTION 5**

1.1 OBJECTIVE6

1.2 ABOUT THE PROJECT6

1.3 PURPOSE6

1.4 SCOPE7

**2. SRS DOCUMENT** 8

2.1 FUNCTIONAL REQUIREMENTS8

2.2 NON-FUNCTIONAL REQUIREMENTS8

2.3 MINIMUM HARDWARE REQUIREMENTS9

2.4 MINIMUM SOFTWARE REQUIREMENTS9

**3. ANALYSIS 12**

3.1 EXISTING SYSTEM12

3.2 PROPOSED SYSTEM12

3.3 FEASIBILITY STUDY13

3.3.1 TECHNICAL FEASIBILITY13

3.3.2 ECONOMIC FEASIBILITY13

3.3.3 OPERATIONAL FEASIBILITY14

3.3.4 SCHEDULE FEASIBILITY14

3.4 COST BENEFIT ANALYSIS14

**4.SOFTWARE DESCRIPTION 16**

4.1 FACE RECOGNITION MODEL16

4.2 DJANGO17

4.3 MVT ARCHITECTURE18

4.4 BOOTSTRAP19

**5.PROJECT DESCRIPTION 21**

5.1 PROJECT DESCRIPTION21

5.2 PROJECT OVERVIEW22

5.3 MODEL DESCRIPTION23

5.3.1 CNN24

5.3.2.1 DEEP METRIC LEARNING24

5.3.2.2 DEEP RESIDUAL NETWORKS24

**6.SYSTEM DESIGN 26**

6.1 SYSTEM ARCHITECTURE26

6.2 INTRODUCTION TO UML27

6.3 FLOWCHARTS30

6.4 BUILDING BLOCKS OF UML32

6.5 UML DIAGRAMS UML33

**7.DEVELOPMENT 34**

7.1 SAMPLE CODE34

7.2 CONCUSION56

7.3FUTURE SCOPE57

7.4 BIBLIOGRAPHY58

# **1. INTRODUCTION**

Respiratory diseases are leading causes of death and disability in the world. Chronic respiratory diseases pose a major public health problem and about 65 million people suffer from chronic obstructive pulmonary disease and with an estimated 3.91 million deaths in 2017 which accounts for 7% of all deaths worldwide and its third leading cause of death. Between 1990 and 2017, the number of deaths due to chronic respiratory diseases increased by 18%, from 3.32 million in 1990 to 3.91 million in 2017. About 334 million people suffer from asthma, the most common chronic disease of childhood affecting 14% of all children globally.

So, treatment of lung diseases, which are the most common cause of death in the world, is of great importance in the medical ﬁeld. For these reasons, a lot of research is going on for early diagnosis and intervention in respiratory diseases. Considering these statistics into account, to study individually and diagnose every patient by health specialists who are already overbooked, mistakes can happen. This is why ﬁnding new ways to help doctors to save time is a priority. Hence, automatic and reliable tools can help in diagnosing more people and it can also help specialists to make less mistakes due to the work overload.

Sounds heard over the chest wall are useful tools for diagnosing pulmonary diseases. Modern lung sound analysis, which began in the last four decades, is focused on digital sound processing and graphic representation of the signals. Audio analysis aids in timely diagnosis of respiratory ailments more effortlessly in the early stages of a respiratory dysfunction. Apart from respiratory check-ups, every cardiac assessment also includes an audio auscultation in which the medical specialist listens to sounds from the patient body with different tools like stethoscope or sonography. This shows how important sound analysis is for heart and lungs disease detection. This application is aimed to classify the lung sounds in an accurate manner and predict the disease by extending the deep learning techniques. These methods also give more relevant information from respiratory sounds extracted and contribute to reducing the time for diagnosis, consequently increasing treatment efﬁciency.

* 1. **OBJECTIVE**

The objective of this initiative is to leverage modern technologies and build a reliable diagnostic tool for the prediction of pulmonary diseases. Through proactive registration, easy audio file uploading, and the prediction of the disease, the initiative aims to allocate the users needs in an efficient manner. The main advantage of using a computer-based lung sound analysis algorithm rather than a spirometry test or through lung auscultation is that this method is non-invasive, less time consuming and cost effective than other methods. Furthermore, implementing the deep learning techniques has proven to be a huge contribution towards accurate audio analysis.

* 1. **ABOUT THE PROJECT**

The project outlined in this introduction is a comprehensive initiative aimed at addressing the pervasive issue of lung diseases prevalent in India. Utilizing the ICBHI 2017 challenge database, a benchmark data repository of lung auscultation sounds used in this study for accurate respiratory disease detection we were able to develop a deep learning model that is trained in a sophisticated manner. We were able to classify 7 types of diseases including a healthy lung sound in order to achieve a higher rate of accurate prediction.

* 1. **PURPOSE**

Pulmonary diseases, ranging from asthma to chronic obstructive pulmonary disease (COPD) and beyond, pose significant risks to human health and well-being. The core purpose of this system is a lifeline, designed to save lives by swiftly identifying potential issues. By enabling early intervention, it offers patients better outcomes and an improved quality of life. It also represents a collaborative effort between engineers and healthcare professionals united in the common goal of improving patient care. Its development reflects a commitment to innovation and a dedication to serving the greater good.

Moreover, this system was developed with cost efficiency in mind. By detecting diseases early, it can prevent costly complications, making healthcare more affordable and accessible for all. In essence, it stands as a testament to our capacity to create positive change, ensuring that every breath is cherished and safeguarded.

* 1. **SCOPE:**

The scope of the pulmonary disease detection system extends across various critical domains, particularly in utilizing audio-based lung sounds analysis:

**1) Early Detection and Diagnosis:** The foremost aim of the system is the early identification of pulmonary diseases through the analysis of lung sounds captured via audio recordings. By discerning subtle abnormalities in these sounds, such as crackles, wheezes, or decreased breath sounds, the system can detect conditions like pneumonia, bronchitis, or asthma in their nascent stages.

**2) Disease Monitoring and Progression Tracking:** Beyond initial diagnosis, the system facilitates continuous monitoring and tracking of pulmonary diseases using audio-based analysis. It enables healthcare providers to monitor changes in lung sounds over time, allowing for the assessment of disease progression and the refinement of treatment plans accordingly.

**3) Risk Stratification and Prognostication:** Through the analysis of lung sound patterns and characteristics, the system aids in stratifying patients based on their risk of developing or experiencing complications related to pulmonary diseases. This risk assessment enables targeted interventions and closer monitoring of high-risk individuals to prevent adverse outcomes.

**4) Personalized Treatment Planning:** Leveraging its ability to analyse nuanced features within lung sounds, the system supports personalized treatment planning for patients with pulmonary diseases. By integrating audio data with clinical information and patient-specific factors, it facilitates the development of tailored treatment strategies aimed at optimizing outcomes and enhancing patient satisfaction.

**5) Healthcare Efficiency:** By automating the analysis of lung sounds and expediting the diagnostic process, the system contributes to resource optimization and healthcare efficiency. It reduces the need for unnecessary tests and procedures, minimizes healthcare costs, and eases the burden on healthcare systems by averting avoidable hospitalizations and emergency room visits.

## **2.SRS REQUIREMENTS**

A software requirements specification (SRS) is a document that captures a complete description of how the system is expected to perform. It is usually signed off at the end of the requirements engineering phase.

**2.1 FUNCTIONAL REQUIREMENTS**

A Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behaviour, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements in Software Engineering are also called Functional Specification. Predict message received by the user is spam or not with the algorithm.

**2.1.1: Data Acquisition:** The system must be capable of recording and colleting lung sounds from patients and it should support multiple recording devices including electronic stethoscopes and microphones.

**2.1.2: Preprocessing:** It implements preprocessing techniques to remove noise and artifacts from recorded lung sounds. Extract relevant features, such as spectral characteristics, duration, and amplitude.

**2.1.3: Classification Algorithms:** Integrate Machine-Learning algorithms (e.g. SVM, neural networks) for respiratory disease classification along with training the system on a diverse dataset containing labelled samples of various respiratory conditions.

**2.1.4: Disease Identification:** Implement algorithms to identify and classify respiratory diseases, including but not limited to COPD, pneumonia, and bronchitis. Provide the capability to handle multiple disease classifications within a single sample.

**2.1.5: Reporting:** Generate a detailed report on disease classifications, including confidence levels and any additional insights. Allow users to export reports in standard formats.

**2.2 NON-FUNCTIONAL REQUIREMENTS**

Non-Functional Requirement (NFR) specifies the quality attribute of a software system.

They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system.

> Accuracy-Achieve a high level of accuracy in disease classification to ensure reliable results.

> Real-Time Processing– Support real-time processing for immediate feedback during patient examinations and also minimize processing delays to enhance the system’s responsiveness.

> Scalability – The system can handle concurrent users and increasing volumes of patient data.

> Security and Privacy – Implements robust security measures to protect patient data.

Non-Functional requirements in Software Engineering allows you to impose constraints or restrictions on the design of the system across the various agile backlogs.

**2.3 MINIMUM HARDWARE REQUIREMENTS**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements. The Hardware Interfaces Required are:

1. Ram: Minimum 4GB or higher
2. GPU: 4GB dedicated.
3. SSD: 128GB
4. Processor: Intel i5 10th Gen or Ryzen 5 with Octa core

**2.4 MINIMUM SOFTWARE REQUIREMENTS**

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view.

1. **Operating System**: Windows, macOS, Linux

Windows is a graphical operating system developed by Microsoft. It allows users to view and store files, run the software, play games, watch videos, and provides a way to connect to the internet. It was released for both home computing and professional works.

MacOS is the computer operating system (OS) for Apple desktops and laptops. It is a proprietary graphical OS that powers every Mac. OSes interact with a computer's hardware, allocating the resources necessary to complete tasks given to it, for example, running an application. OSes allocate resources including memory, processing power and file storage.

Linux is an operating system. In fact, one of the most popular platforms on the planet, Android, is powered by the Linux operating system. An operating system is software that manages all of the hardware resources associated with your desktop or laptop. To put it simply, the operating system manages the communication between your software and your hardware. Without the operating system (OS), the software wouldn’t function

1. **Programming Languages:** Python
2. **Libraries and Frameworks:** Flask

**Development Tools:** Jupyter Notebook, Visual Studio

## **Python Libraries To be installed:**

The following libraries with the specific versions type must be installed for this project to function. These can be installed using command “pip install library name==version"

Numpy: NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

TensorFlow: **TensorFlow** is a popular framework of **machine learning** and **deep learning.** It is **a free** and **open-source** library which is developed by **Google Brain Team**. It is entirely based on Python programming language and use for numerical computation and data flow, which makes machine learning faster and easier. TensorFlow can train and run the deep neural networks for image recognition.

Librosa: Librosa is a versatile and powerful library for handling audio files in Python. It provides a comprehensive set of tools and functionalities for audio data preprocessing, feature extraction, visualization, analysis, and advanced techniques.

Seaborn: Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.

# **3. ANALYSIS**

## **EXISTING SYSTEM**

Despite efforts to utilize MFCC features with Inception Network classifiers for audio scene classification, challenges arise in effectively combining these features and integrating information from different channels with varying data formats. Additionally, adapting Inception Network architecture designed for image analysis to speech recognition may lead to suboptimal performance due to neglecting speech-specific properties. The complexities of integrating MFCC features within Inception Network architectures result in cumbersome and inefficient processes, hindering seamless classification of audio scenes. Furthermore, the inherent differences between image and audio data present obstacles in achieving optimal performance, leading to decreased accuracy and reliability in classification tasks. Despite attempts to overcome these limitations, the efficacy of Inception Network for speech recognition remains questionable, with shortcomings in capturing nuanced speech characteristics and variability. Overall, while some success has been observed, the adaptation of Inception Network for audio-related tasks is marred by inherent challenges and limitations, highlighting the need for further refinement and exploration of alternative approaches.

## **PROPOSED SYSTEM**

In the developed system, one of the major objectives is to provide an automated algorithmic approach that can categorize lung sounds in a variety of diseased states. Another objective of this present research work is to propose a lightweight deep learning architecture that can classify lung sounds accurately while keeping parameter size and computational complexity less. The majority of these respiratory diseases have almost similar kind of symptoms; therefore, it becomes difficult for the doctor to predict the actual disease just by hearing the lung sound only and requires additional tests, such as spirometry test. Classification of seven respiratory diseases for the first time, utilizing three publicly available lung sound databases: ICBHI 2017 challenge database and chest wall lung sound database. Engaging all the databases also ensures the robustness of the classification mechanism, as the DLM is trained with a wide variety of lung sounds. Computing the ablation study and classification report containing the statistics of layers, parameters, accuracy, precision, recall, F1 score, and so on, in order to have a thorough performance/classification accuracy analysis of the proposed CNN model.

1. **FEASIBILITY STUDY**

It is an assessment of the practicality of a proposed plan or method. This helps to find the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects for success. A feasibility study is used:

1. To determine whether the objectives stated in the assignment brief are reasonably attainable within the limitation and financial constraints period.

2. To define major problem areas, so that the system analyst can plan the strategy for the field investigation.

3. To find areas where potential exists for making saving in money, time or effort.

4. To determine the approximate time required for the full investigation and cost. There are different types of a feasibility study, which are explained below

1. Technical feasibility

2. Economic feasibility

3. Operational feasibility

4. Schedule feasibility

### **3.3.1 TECHNICAL FEASIBILITY**

It measures the availability of technical resources (hardware components or technical equipment). It also studies the availability of the technical manpower for the project. If the work performances of the technical manpower are not experienced, the entire system will be certainly insufficient.

**3.3.2 ECONOMIC FEASIBILITY**

Economic feasibility measures whether finances (investments) are available for the proposed solution, i.e. it looks at the financial aspects (cost/ effectiveness) of the project. This is often called a cost benefit analysis.

**3.3.3 OPERATIONAL FEASIBILITY**

It is a measure of how well the solution of problems, or a specific alternative solution will work in the organization. It is also a measure of how people feel about the system. If the system is not easy to operate than the operational process would be difficult. The operator of the system should be given proper training. The system should be made such that the user can interface the system without any problem.

**3.3.4 SCHEDULE FEASIBILITY**

If a Schedule feasibility deadline (time limit) is established, it is called schedule feasibility, the deadline of the project is studied under the scheduled feasibility. The scheduled feasibility also depends upon available manpower and economic condition as well.

**3.4 COST BENEFIT ANALYSIS**

It can be explained as a procedure for estimating all costs involved and possible profits to be derived from a business opportunity or proposal. It takes into account both quantitative and qualitative factors for analysis of the value for money for a particular project or investment opportunity. Benefits to costs ratio and other indicators are used to conduct such analyses.

The objective is to ascertain the soundness of any investment opportunity and provide a basis for making comparisons with other such proposals. All positives and negatives of the project are first quantified in monetary terms and then adjusted for their mentioned correct estimates for conduct of cost benefit analysis.

The cost estimation done for Project also depend upon the baseline metrics collected from existing system and these were used in conjunction with estimation variables to develop co and effort projections.

The baseline metrics collected from the existing system may include:

1. Data Volume: The amount of data stored in the existing system, including information integrated from 3 databases that contain the lung sounds with different parameters, helps in determining storage requirements and database design for the new platform.
2. System Performance: Metrics related to system performance, such as response times, processing speeds, and uptime, provide insights into the system's efficiency and reliability. Understanding performance issues in the existing system helps in designing a more scalable and responsive solution.
3. User Feedback: Feedback from users of the existing system regarding usability, functionality, and satisfaction levels helps in identifying pain points and areas for improvement in the new platform. Incorporating user feedback into the project requirements ensures that the new system meets user needs and expectations.
4. Error Rates: Metrics related to error rates, data inconsistencies, and data integrity issues in the existing system highlight areas that require attention during development. Addressing these issues in the new platform improves data accuracy and reliability.
5. Process Efficiency: Metrics related to process efficiency, such as the time taken to enter and retrieve data, the number of manual steps involved in data management, and the frequency of errors, help in identifying opportunities for automation and streamlining processes in the new platform.

# **4. SOFTWARE DESCRIPTION**

**4.1 Convolution Neural Network MODEL:**

Different CNN architectures use the TFR image of the lung sound signal as input, have been used in recent years to classify respiratory diseases from pathological lung sounds. The growing popularity of lightweight CNN architectures in the deep learning research field is due to their quick processing speeds and small size without sacrificing accuracy. In this project, to counteract the high computational cost which results in limited resources we developed a novel lightweight DLM that can diagnose seven distinct respiratory disorders by extracting prominent features from TFR images obtained from the lung sound signal.

**Preprocessing:** In the pre-processing stage of respiratory disease classification using lung sounds, four sub-modules are employed. Firstly, all signals are resampled to 4kHz to accommodate variations in sampling frequency across different databases. Following resampling, the signals undergo temporal snippet generation, where they are divided into 5-second windows with 50% overlap between adjacent frames. Subsequently, baseline removal using Discrete Fourier Transform (DFT) filtering is applied to eliminate the baseline wandering component (0-1 Hz) from each temporal snippet. The normalization of the lung sound snippet is performed between the range [-1,1].

**Feature Extraction:** The core of the proposed system is based on the machine learning tasks such as MFCCs, Chroma STFT, and Mel Spectrogram which are commonly used in audio signal processing such as speech recognition and sound event detection. Each feature type captures different aspects of the audio signal, providing rich information for analysis and modelling.

**Training:** First, the whole data are spitted into 80%, 10%, and 10% for training, validation, and testing sets, respectively. During training, the model adjusts its parameters to minimize the loss function. Engaging all the databases also ensures the robustness of the classification mechanism, as the DLM is trained with a wide variety of lung sounds.

## **4.2 FLASK**

Flask is a lightweight and flexible web framework written in Python. It is designed to make it easy to build web applications quickly and with minimal code. Flask provides the tools and libraries necessary for creating web applications, handling routing, and managing HTTP requests and responses.

**Routing:** Flask offers developers the flexibility to define route parameters, allowing for dynamic URL patterns and enabling the passing of data within the URL. This enables more versatile routing capabilities, facilitating the creation of RESTful APIs and more complex web applications.

**HTTP Request Handling:** HTTP request handling in Flask extends beyond basic request methods, offering support for handling file uploads, handling form data, and accessing request headers. This comprehensive request handling functionality simplifies the development of web applications that interact with client-side data and resources.

**Template Rendering:** Flask's template rendering capabilities are not limited to HTML content. Developers can render various types of content, including JSON, XML, and even plain text responses. This versatility enables Flask applications to serve a wide range of content types, catering to diverse client requirements and preferences.

**Extensibility:** Flask integrates seamlessly with various third-party extensions and libraries, providing developers with access to a rich ecosystem of tools and resources. This extensibility allows developers to enhance their applications with additional functionality, such as authentication, database integration, and caching, while maintaining the simplicity and elegance of Flask's core framework.

**Error Handling:** Flask provides built-in error handling mechanisms, allowing developers to define custom error pages and handle exceptions gracefully. This enhances the user experience by presenting informative error messages and guiding users through unexpected scenarios.

**4.3 BOOTSTRAP**

Bootstrap is a free and open-source front-end framework for developing websites and web applications. It's one of the most popular HTML, CSS, and JavaScript frameworks for designing responsive and mobile-first projects on the web. Bootstrap makes front-end web development faster and easier by providing a set of features and components that you can use as the foundation for your website design.

1. Responsive Design: Bootstrap's grid system and ready-made components automatically adjust to fit the size of the screen, ensuring your site looks great on all devices, from phones to tablets to desktops.
2. Pre-styled Components: It comes with a wide range of pre-styled components such as buttons, navigation bars, modals, carousels, and much more, which you can use to build your UIs.
3. Customizable: Bootstrap can be customized to fit the specific needs of your project. You can select which components to include, set your own sizes, colours, and more.
4. Cross-browser Compatibility: It ensures that your site looks consistent across all modern web browsers by handling all the CSS compatibility issues with different browsers.

Bootstrap can be included in your project in several ways, such as:

Downloading the Bootstrap source files and including them in your project.

Linking to a Content Delivery Network (CDN) to include Bootstrap without hosting the files yourself.

Using package managers like npm or yarn to install Bootstrap in your project.

# **5.PROJECT DESCRIPTION**

**5.1 PROJECT DESCRIPTION**

This project is a web application that enables users to upload lung sound recordings and receive automated predictions regarding the presence of respiratory diseases. The system utilizes advanced machine learning techniques, including Convolutional Neural Networks (CNNs), for disease classification, and Flask, a lightweight web framework, for creating the user interface. Lung sound recordings are sourced from 3 main datasets (ICBHI 2017 Challenge Database, Chest Wall Lung Sound Database, RespiratoryDatabase@TR) which are integrated together for enough entries to train the data accordingly. These recordings are pre-processed to remove noise, normalize audio levels, and segment them into individual samples corresponding to specific respiratory events.

Various feature extraction techniques are employed to capture relevant characteristics from the lung sound data. This includes Mel-Frequency Cepstral Coefficients (MFCC), Chroma Short-Time Fourier Transform (Chroma STFT), and Mel Spectrogram (MSPEC). These features provide a compact representation of the audio data, which is essential for training the CNN model. A CNN architecture is designed and trained using the extracted features. The CNN model consists of convolutional layers for feature extraction, followed by pooling layers to reduce dimensionality and extract the most salient features. Batch normalization is applied to stabilize and accelerate the training process, while dropout regularization is used to prevent overfitting.

The trained CNN model is evaluated using standard metrics such as accuracy, precision, recall, and F1-score. A comprehensive classification report is generated to assess the model's performance across different respiratory disease classes. Cross-validation techniques may be employed to ensure robustness and generalizability.

Once the CNN model achieves satisfactory performance, it is serialized and saved using the pickle library. The serialized model is integrated into a Flask web application, where it can be accessed and utilized by users. The model file (model.h5) is hosted on the web server for real-time inference.

## **5.2 PROJECT OVERVIEW**

A user-friendly web interface is developed using Flask, HTML, and CSS. The web application allows users to upload lung sound recordings in WAV format and submit them for analysis. Additionally, informative content and instructions are provided to guide users through the process.

Upon uploading a lung sound recording, the web application processes the audio file using the trained CNN model to predict the likelihood of various respiratory diseases. The results are displayed to the user in an easily interpretable format, along with confidence scores for each prediction.

The user is presented with an HTML form to upload an audio file and provide patient information. When the form is submitted, the audio file is saved to the upload folder, and its URL is passed to the machine learning model for classification. The machine learning model generates classification results based on the audio file and returns them. The classification results, along with the patient information and the URL of the uploaded audio file, are rendered on the web page.

The classification results include both numerical predictions and a visualization of the sound wave and MFCC (Mel-frequency cepstral coefficients) features extracted from the audio file.

**5.3 MODEL DESCRIPTION**

A Convolutional Neural Network (CNN) is a type of Deep Learning neural network architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data. The Convolutional layer applies filters to the input image to extract features, the Pooling layer down samples the image to reduce computation, and the fully connected layer makes the final prediction.

**Feature Extraction Layers (Sub models):**

1) MFCC Model: This sub model processes the MFCC features. It consists of several convolutional layers with batch normalization and ReLU activation functions, followed by max-pooling layers for down sampling. The output is obtained by applying global max-pooling to the final convolutional layer.

2) Chroma Model: Similar to the MFCC model, this sub model processes chroma features using convolutional layers, batch normalization, ReLU activation, and max-pooling, followed by global max-pooling to obtain the output.

mSpec Model: This sub model processes mSpec features in a manner similar to the MFCC and chroma models.

3) Concatenation Layer: The outputs from the three sub models (MFCC, chroma, and mSpec) are concatenated along the channel axis using the concatenate layer.

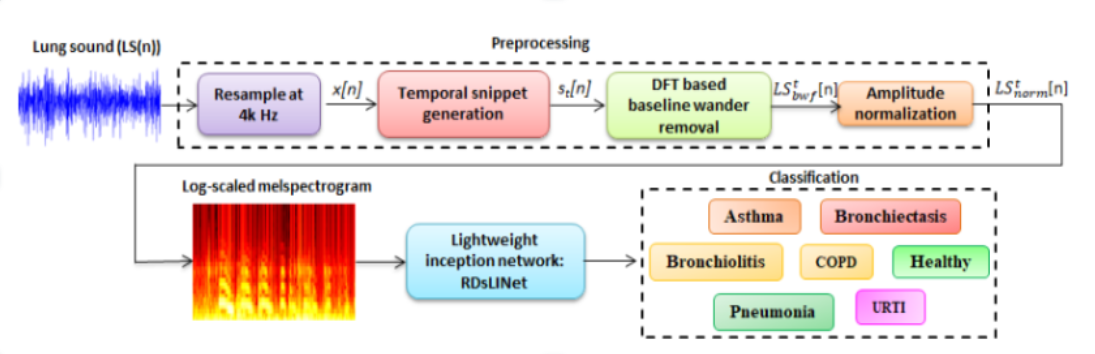
4) Hidden Layers: Dropout layers are added to mitigate overfitting by randomly dropping a fraction of input units during training. Dense layers with ReLU activation functions are used for feature transformation and nonlinear mapping. Multiple dense layers are stacked to capture complex patterns in the concatenated feature representation.

5) Output Layer: The final output layer consists of a dense layer with softmax activation, which produces probability distributions over the classes/categories. The number of units in this layer corresponds to the number of output classes (8 in this case).

6)Model Compilation: The CNN model (net) is instantiated using the Keras functional API, with the three input layers and the output layer specified.

# **6. SYSTEM DESIGN**

## **6.1 SYSTEM ARCHITECTURE**

System architecture refers to the high-level structure and design of a complex system, providing a blueprint for the arrangement and interaction of its components. It encompasses the organization of hardware elements, software components, communication interfaces, data storage, and the relationships between them to achieve specific functionalities or goals. System architecture is crucial in the development process as it helps guide the design, implementation, and maintenance of a system.

Overview of System Architecture

# **6.2 INTRODUCTION TO UML**

UML (Unified Modeling Language) is a standardized visual language used in software engineering for creating diagrams and models to represent software systems. UML is a flexible and powerful tool that enables software developers to design, document, and communicate their software systems effectively.

There are various types of UML diagrams that can be used to model different aspects of

software systems, including:

● Use case diagrams: A use case diagram represents the functional requirements of a system by showing the interactions between actors (users or external systems) and the system.

● Class diagrams: A class diagram represents the structure of a system by showing the classes, their attributes, and the relationships between the classes.

● Sequence diagrams: A sequence diagram represents the behavior of a system by showing the interactions between objects in chronological order.

● State machine diagrams: A state machine diagram represents the behavior of an object or a system by showing the possible states and transitions between the states.

● Activity diagrams: An activity diagram represents the flow of activities or processes in a system.

To create UML diagrams, many software tools are available that support UML modelings, such as Microsoft Visio, StarUML, and Visual Paradigm. UML diagrams can also be created by hand, but using software tools makes the process faster and more efficient. UML is a valuable tool for software development as it provides a common language and visual representation for software systems, which can help to reduce misunderstandings and miscommunications among team members. It can also help in the design and development process by providing a clear and organized view of the system being developed.

* **USE CASE DIAGRAM**

A diagram of a user flow

Description automatically generated

Use Case Diagram of Web UI

* **A diagram of a computer

  Description automatically generatedClass Diagram**.
* **Activity Diagram**

**A diagram of a diagram

Description automatically generated**

## **6.3 FLOWCHARTS**

The DFD is also known as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system. It maps out the flow of information for any process or system, how data is processed in terms of inputs and outputs. It uses defined symbols like rectangles, circles and arrows to show data inputs, outputs, storage points and the routes between each destination. They can be used to analyses an existing system or model of a new one. A DFD can often visually “say” things that would be hard to explain in words and they work for both technical and non- technical. There are four components in DFD:

* + - * 1. External Entity
        2. Process
        3. Data Flow
        4. Data Store

**External Entity:**

It is an outside system that sends or receives data, communicating with the system. They are the sources and destinations of information entering and leaving the system. They might be an outside organization or person, a computer system or a business system. They are known as terminators, sources and sinks or actors. They can be used to analyses an existing system or model of a new one. These are sources and destinations of the system’s input and output.

**Data Store:**

A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. They can be used to analyses an existing system or model of a new one. These are sources and destinations of the system’s input and output.

**Circle:**

A circle (bubble) shows a process that transforms data inputs into data outputs. It is just like a function that changes the data, producing an output. It might perform computations for sort data based on logic or direct the dataflow based on business rules.

**Data Flow:**

A curved line shows the flow of data into or out of a process or data store. A dataflow represents a package of information flowing between two objects in the data-flow diagram, Data flows are used to model the flow of information into the system, out of the system and between the elements within the system.

A flowchart of a computer program

Description automatically generated

Data Flow Diagram

**6.4 BUILDING BLOCKS OF UML**

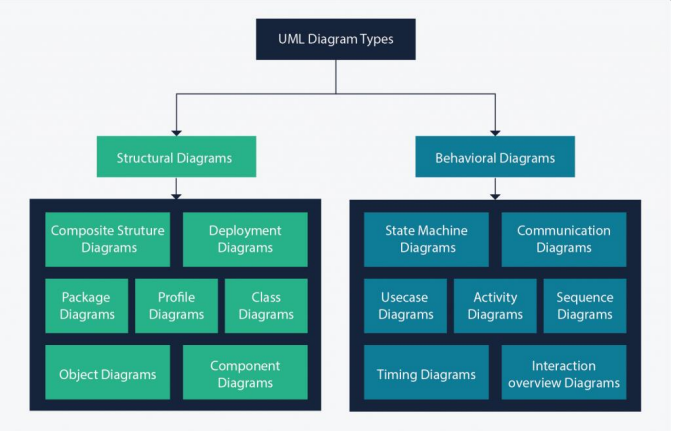
UML is linked with object-oriented design and analysis. UML makes use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as

1. Structural Diagrams – Capture static aspects or structures of a system. Structural Diagrams include Component Diagrams, Object Diagrams, Class Diagrams, and Deployment Diagrams.

2. Behaviour Diagrams – Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams, and Interaction Diagrams.

## **THE IMAGE BELOW SHOWS THE HIERARCHY OF DIAGRAMS**

## **ACCORDING TO UML**

**OBJECT ORIENTED CONCEPTS USED IN UML**

**Class** – A class defines the blue print i.e. structure and functions of an object.

**Objects** – Objects help us to decompose large systems and help us to modularize our system. Modularity helps to divide our system into understandable components so that we can build our system piece by piece. An object is the fundamental unit (building block) of a system which is used to depict an entity.

**Inheritance** – Inheritance is a mechanism by which child classes inherit the properties of their parent classes.

## **Abstraction** – Mechanism by which implementation details are hidden from the user.

## **Encapsulation** – Binding data together and protecting it from the outer world is referred to as encapsulation.

## **Polymorphism** – Mechanism by which functions or entities are able to exist in different forms.

**7. DEVELOPMENT**

**7.1 SAMPLE CODE**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screen shot of a computer

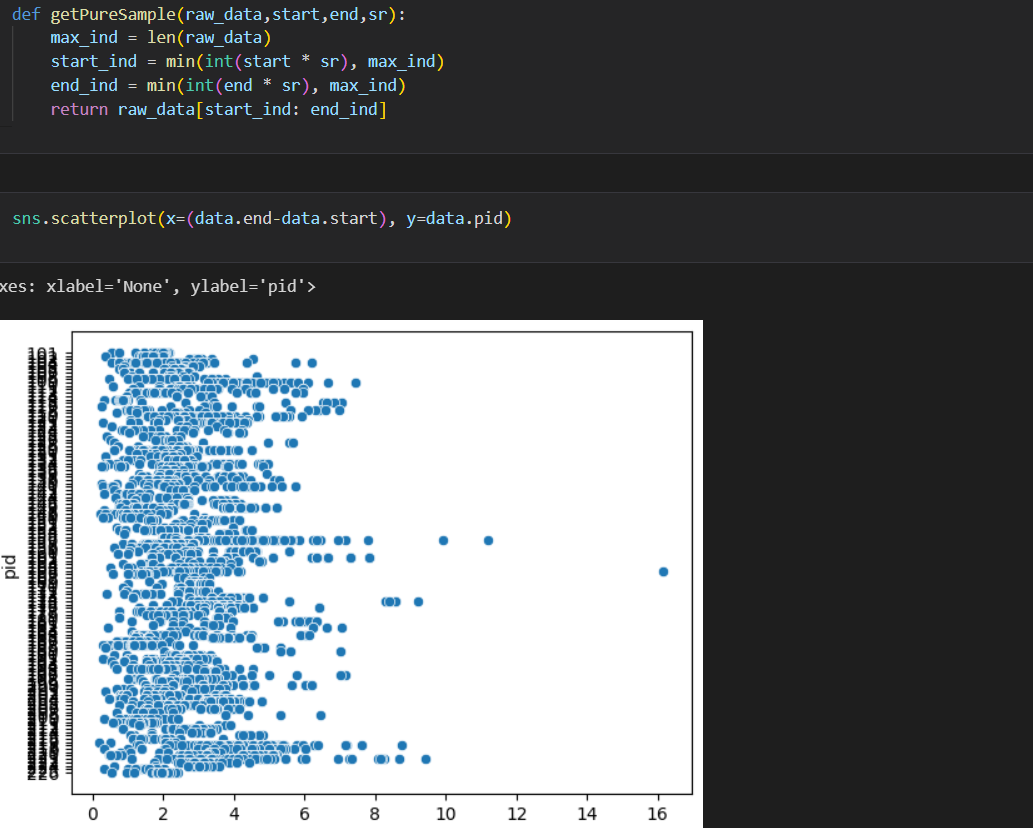
Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screen shot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a graph

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

A screen shot of a graph

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

A screen shot of a graph

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screen shot of a graph

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer screen

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedA black screen with white text

Description automatically generated

A screen shot of a graph

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

**7.2 RESULT**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A close-up of a pink mask

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# **7.3 CONCLUSION**

In conclusion, the development of this web application marks a significant advancement in the realm of respiratory disease diagnosis. By harnessing the power of advanced machine learning techniques, particularly Convolutional Neural Networks (CNNs), and integrating them with Flask, a lightweight web framework, we have created a user-friendly platform capable of automated disease prediction from lung sound recordings. The integration of data from multiple datasets ensures robust training of the CNN model, enhancing its accuracy and generalizability. Pre-processing techniques such as noise removal and feature extraction further refine the input data, enabling the model to capture relevant characteristics effectively.

The evaluation of the trained CNN model using standard metrics demonstrates its reliability and efficacy in disease classification. By serializing and integrating the model into the Flask web application, we have enabled real-time inference and streamlined accessibility for users. The user interface, designed with simplicity and intuitiveness in mind, empowers users to upload their lung sound recordings seamlessly and receive informative predictions regarding respiratory diseases. The inclusion of visualizations enhances the interpretability of results, facilitating informed decision-making.

Overall, this web application represents a valuable tool in the diagnosis and management of respiratory diseases, offering a blend of cutting-edge technology and user-centric design to improve healthcare outcomes and enhance patient care.

**7.4 FUTURE SCOPE**

Automated adventitious sounds detection or classiﬁcation provides a promising solu- tion to overcome the limitations of conventional auscultation. Furthermore, the subject area for future investigation will be:

* To use larger dataset and test further on robustness in presence of higher percent- ages of noise.
* Attempts will also be made towards isolation of breath sounds from the ambient noises and heartbeat sounds for better analysis.
* Other acoustic techniques will be applied for even better modelling of the lung sounds along with deep learning-based approaches.
* To have clinical use in pulmonary health screening and as a tool in differential diagnosis of pulmonary diseases.
* Finally, we will be trying to identify the nature and severity of infection from the breath sounds.

**7.5 BIBLIOGRAPHY**

* **L. Pham, H. Phan, R. Palaniappan, A. Mertins, and I. McLoughlin, “CNN-MoE based framework for classification of respiratory anomalies and lung disease detection,” IEEE J. Biomed. Health Information., vol. 25, no. 8, pp. 2938–2947, Aug. 2021.**
* **S. B. Shuvo, S. N. Ali, S. I. Swapnil, T. Hasan, and M. I. H. Bhuiyan, “A lightweight CNN model for detecting respiratory diseases from lung auscultation sounds using EMD-CWT-based hybrid scalogram,” IEEE J. Biomed. Health Informat., vol. 25, no. 7, pp. 2595–2603, Jul. 2021.**
* [**https://bhichallenge.med.auth.gr/ICBHI\_20**](https://bhichallenge.med.auth.gr/ICBHI_20)**﻿**[**17\_Challenge**](https://www.canva.com/design/DAF5T-EUxso/fDhDdyfHijTBB67DcAD_9Q/edit)