**ASTHMASENSE: Detection of Asthma Using Artificial Intelligence**

**A PROJECT REPORT**

***Submitted to***

******

**ASSAM DON BOSCO UNIVERSITY**

***by***

**RAJ ROY**

**DC2022BCA0029**

**AMAN SAH**

**DC2022BCA0025**

**BISANT CHOWDHURY**

**DC2022BCA0035**

***in partial fulfilment for completion of Major Project***

***of***

**SIXTH SEMESTER**

**OF**

**BACHELOR OF COMPUTER APPLICATIONS**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**SCHOOL OF TECHNOLOGY**

**ASSAM DON BOSCO UNIVERSITY**

**AZARA, GUWAHATI 781017,**

**ASSAM, INDIA.**

**BATCH (2022-2025)**

**ASTHMASENSE: Detection of Asthma Using Artificial Intelligence**

**A PROJECT REPORT**

***Submitted to***

******

**ASSAM DON BOSCO UNIVERSITY**

***by***

**RAJ ROY**

**DC2022BCA0029**

**AMAN SAH**

**DC2022BCA0025**

**BISANT CHOWDHURY**

**DC2022BCA0035**

***in partial fulfilment for completion of Major Project***

***of***

**SIXTH SEMESTER**

**OF**

**BACHELOR OF COMPUTER APPLICATIONS**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**SCHOOL OF TECHNOLOGY**

**ASSAM DON BOSCO UNIVERSITY**

**AZARA, GUWAHATI 781017,**

**ASSAM, INDIA.**

**BATCH (2022-2025)**

**CERTIFICATE**

This is to certify that the Project Report entitled “**ASTHMASENSE: Detection of Asthma using Artificial Intelligence**” submitted by **RAJ ROY (DC2022BCA0029), AMAN SAH (DC2022BCA0025)** and **BISANT CHOWDHURY (DC2022BCA0035)** to the Assam Don Bosco University, Guwahati, Assam, in partial fulfilment of the requirement for Major project of 6th semester of Bachelor of Computer Applications. It is a bonafide record of the project work carried out by themunder my supervision during the semester January 2025 to June 2025.

(Signature of the Guide)

**Dr. Kishore Medhi**

**Assistant Professor (stage I), Dept. of Computer Applications**

**School of Technology**

**Assam Don Bosco University**

Date: .......................................................

**CERTIFICATE**

This is to certify that the Project Report entitled “**ASTHMASENSE: Detection of Asthma using Artificial Intelligence**” submitted by **RAJ ROY (DC2022BCA0029), AMAN SAH (DC2022BCA0025)** and **BISANT CHOWDHURY (DC2022BCA0035)** to the Assam Don Bosco University, Guwahati, Assam, in partial fulfilment of the requirement for the Major project of 6th semester of Bachelor of Computer Applications. It is a bonafide record of the project work carried out by them during the semester January 2025 to June 2025.

**Dr. Gypsy Nandi Prof. Manoranjan Kalita**

Head of the Department Director, School of Technology

Date: ........................................ Date:........................................

**EXAMINATION CERTIFICATE**

This is to certify that **RAJ ROY (DC2022BCA0029), AMAN SAH (DC2022BCA0025)** and **BISANT CHOWDHURY (DC2022BCA0035)** of theDepartment of Computer Applications have carried out the Project Work in a manner satisfactory to warrant its acceptance and also defended it successfully.

I wish them all the success in theirfuture endeavours.

Examiners:

1. External Examiner:
2. Internal Examiner:

**DECLARATION**

We hereby declare that the project work entitled “**ASTHMASENSE: Detection of Asthma using Artificial Intelligence**” submitted to the Assam Don Bosco University, Guwahati, Assam, in partial fulfilment of the requirement for Major project of 6th semester of Bachelor of Computer Applications. It is an original work done by us under the guidance of

**Dr. Kishore Medhi** **(**Assistant Professor (stage I), *Dept. of Computer Applications, School of Technology, Assam Don Bosco University)* and has not been submitted for the award of any degree.

(Signature of the student)

**RAJ ROY**

**DC2022BCA0029**

**Department of Computer Applications, School of Technology**

**Assam Don Bosco University**

(Signature of the student)

**AMAN SAH**

**DC2022BCA0025**

**Department of Computer Applications, School of Technology**

**Assam Don Bosco University**

(Signature of the student)

**BISANT CHOWHURY**

**DC2022BCA0035**

**Department of Computer Applications, School of Technology**

**Assam Don Bosco University**

**ACKNOWLEDGEMENT**

We want to thank our project Guide, **Dr. Kishore Medhi**, from the bottom of our hearts. He gave us priceless advice always backed us up, and shared helpful thoughts as we worked on this project. He know-how, cheering us on, and smart ideas helped us get past many roadblocks and finish our work on schedule.

We also say a big thank you to the **Head of Department, Dr. Gypsy Nandi.** She stood by us, pointed us in the right direction, and made sure we had what we needed to work. We're just as thankful to all the teachers and tech staff in the **Department of Computer Applications**. They lent a hand and shared what they knew, which helped when we were testing things out and building our project.

We want to thank **Assam Don Bosco University** for its help. We appreciate the Director and administrative staff for creating a place where we could work well and learn a lot.

We're grateful to our friends and peers. You helped us, cheered us on, and shared ideas with us as we worked. Your thoughts and teamwork played a big part in making this project work out.

We also thank our families. You were so patient with us, stood by us, and kept pushing us forward while we worked on this. When things got tough, your faith in us kept us going.

At last, we offer our sincere gratitude to the **Almighty**. His grace and blessings provided us with the power, insight, and resolve to finish this key step in our academic path.

**ABSTRACT**

AsthmaSense is a clever Android app that uses AI and machine learning to spot and forecast asthma risk by analyzing breathing sounds. This project aims to create an easy-to-use, affordable, and instant system to detect asthma through sound examination. It's useful in places where people can't get to advanced medical centers.

The system uses Convolutional Neural Networks (CNN) and audio signal processing methods like Mel Frequency Cepstral Coefficients (MFCCs) to pull out key features from .wav audio files of breathing noises. These features go through a trained machine learning model set up with a Flask API, which then tells you if the breathing pattern sounds asthmatic or normal. The app is easy to use, lets you upload audio files from your device or the cloud, and gives you quick results.

With a prediction accuracy of about 95.48%, AsthmaSense proves itself as a scalable prototype that has an influence on deep learning, mobile computing, and healthcare diagnostics. The project also uses top methods in model training quick communication through APIs, and Android UI/UX design showing how different fields can work together to apply tech in medicine.

This project points to AI's future role in keeping people healthy and sets the stage to add more advanced features. These could include analyzing live recordings, alerting users about air quality, and linking doctors in real time in later versions.

**LIST OF TABLES**

**Table Title Page**

3.1.1 Hardware Requirements 7

3.1.2 Software Requirements 8

3.2.3.1 COCOMO Model Estimation 11

5.2 Respiratory Dataset 19

5.3 Model Architecture 20

6.2 Test Case 24

6.3 Evaluation Metrics 25

6.6 Comparative Experiments 28

**LIST OF FIGURES**

**Figure Title Page**

3.2.4.1 Work Breakdown Structure 12

3.2.4.2 Gantt Chart 13

4.1 System Architecture Diagram 14

4.2 Activity Diagram 15

4.3 Block Diagram 16

4.4 Flowchart 17

5.7 User Interface 23

6.4.1 Training vs Validation Accuracy 26

6.4.2 Confusion Matrix 26

6.4.3 AUC-ROC 27

6.4.4 Classification Report 27

**ABBREVIATIONS**

AI Artificial Intelligence

ML Machine Learning

CNN Convolutional Neural Network

MFCC Mel Frequency Cepstral Coefficient

API Application Programming Interface

AQI Air Quality Index

UI/UX User Interface / User Experience

IoT Internet of Things

IDE Integrated Development Environment

**CONTENTS**

**Title Page**

**CERTIFICATE** ii

**CERTIFICATE** iii

**EXAMINATION CERTIFICATES** vi

**DECLERATION** v

**ACKNOWLEDGEMENTS** vi

**ABSTRACT** vii

**LIST OF TABLES** viii

**LIST OF FIGURES** xi

**ABBREVIATIONS** x

**CHAPTER 1 INTRODUCTION**

1.1 Aim of the Project 1

1.2 Objective of the Project 1

1.3 Motivation 2

1.4 Problem Statement 2

**CHAPTER 2 BACKGROUND STUDY**

2.1 Medical Background: Understanding Asthma 3

2.2 Existing System 4

2.3 Proposed System 4

2.4 Literature Review 5

2.5 Methodology, Research Data and Dataset 6

**CHAPTER 3 SYSTEM REQUIREMENTS AND FEASIBILITY STUDY**

3.1 System Requirements 7

3.1.1 Hardware Requirements 7

3.1.2 Software Requirements 8

3.2 Feasibility Study 9

3.2.1 Technical Feasibility 9

3.2.2 Operational Feasibility 9

3.2.3 Economic Feasibility 10

3.2.4 Schedule Feasibility 11

**CHAPTER 4 SYSTEM DESIGN**

4.1 System Architecture Overview 14

4.2 Activity Diagram 15

4.3 Block Diagram 16

4.4 Flowchart 17

4.5 Component Description 18

**CHAPTER 5 IMPLEMENTATION**

5.1 System Overview 19

5.2 Dataset and Pre-processing 19

5.3 Model Architecture 20

5.4 Model Implementation 21

5.5 BackendIntegration 22

5.6 Android Application Implementation 22

5.7 AsthmaSense User Interface Screenshots 23

**CHAPTER 6 TESTING AND REPORTS**

6.1 Testing Strategy 24

6.2 Test Case 24

6.3 Evaluation Metrics 25

6.4 Graphical Results 26

6.5 Model Performance 28

6.6 Comparative Experiments 28

6.7 Discussion 28

6.8 Tools Used For Testing 28

**CHAPTER 7 CONCLUSIONS and FUTURE WORK**

7.1 Conclusion 29

7.2 Future Work 30

**REFERENCES** ................................................................................................................31

**CHAPTER 1**

**INTRODUCTION**

* 1. **Aim of the Project**

The aim of this project is to create a smartphone app called AsthmaSense that uses AI to spot possible asthma signs by analysing breathing sounds. By combining ML with mobile tech, this project aims to offer an easy-to-use, gentle, and affordable way to catch asthma. This app will let people those in far-off or underserved places, check for asthma symptoms with just their phone cutting down the need for pricey medical tools or trips to the hospital.

* 1. **Objectives of the Project**

**1.2.1 AI to Detect Asthma**

The goal is to create and train a machine learning system that listens to respiratory sounds like wheezing, coughing, or typical breathing. This system will aim to predict asthma with strong accuracy.

**1.2.2 App for Mobile Users**

The plan involves creating an app for Android users. This app will let people upload their breathing sounds and get quick predictions through a Flask-powered backend connected to the trained model.

**1.2.3 Relying on Public Data**

The project will make use of available respiratory sound data from sites like Kaggle. These public datasets will allow effective training of the model without needing to gather custom data.

**1.2.4 Simple and User-Friendly Design**

Efforts will go towards building an interface that is easy to navigate and understand. To design a user interface and experience that feels natural and works so people of any age can access and use the tool without much technical know-how.

* 1. **Motivation**

Asthma is a long-term breathing problem that impacts more than 300 million people around the world. A lot of people do not know they have it or find out when the symptoms get really bad. In poorer areas, not having enough lung doctors and proper testing machines makes it take even longer to figure out and treat the issue. The growing air pollution and changing weather patterns are causing more asthma cases in kids and older adults.

The motivation behind this project lies in bridging the healthcare accessibility gap using Artificial Intelligence. With most people owning smartphones, we saw an opportunity to leverage this technology for early asthma detection using AI models that analyze respiratory audio patterns. A mobile-based application allows flexibility, ease of use, and instant feedback — thereby empowering individuals to monitor their respiratory health proactively.

* 1. **Problem Statement**

Doctors diagnose asthma using costly medical tests like spirometry, chest x-rays, or lab checks. These tests need patients to visit a hospital and depend on skilled professionals to analyze the results. Many people ignore or identify early signs like coughing or wheezing. Right now, no popular mobile tool exists to give fast, AI-powered asthma predictions based on breathing sounds recorded by users.

Thus, there is a need for a low-cost, portable, and intelligent system that can help in early asthma detection by analyzing sound data through deep learning techniques. This project aims to solve this problem by offering an application that combines ML models with mobile accessibility to predict asthma risk based on audio recordings.

**CHAPTER 2**

**BACKGROUND STUDY**

**2.1 Medical Background: Understanding Asthma**

Asthma is a condition in which your airways narrow and swell and produce extra mucus. It can happen to anyone at any age, but the condition generally begins in childhood. Asthma According to World Health Organization (WHO), asthma complicates the lives of over 300 million people globally and is one of the most common chronic diseases among all ages, especially in urban and polluted areas.

Asthma attacks happen when the airways become inflamed, constricted and produce extra mucus, leading to difficulty breathing. This response may be provoked by allergens (pollen, dust, pets, etc.), respiratory infection, exercise, cold; or pollution.

**2.1.1 Common Symptoms of Asthma:**

* Shortness of breath
* The sound of whistling or wheezing while breathing
* Tightness in the chest
* Cough that goes on and on (especially at night and early morning)

**2.1.2 Diagnosis:**

Traditional diagnosis methods include spirometry (lung function tests), peak flow measurements, chest X-rays, and clinical evaluation. However, these often require trained personnel and hospital infrastructure, making early detection difficult in rural or remote areas.

**2.1.3 Treatment & Care:**

* Asthma is not curable but it can be control. Treatment includes:
* Inhaled corticosteroids (preventer) These are steroid medications that you inhale and are used to prevent symptoms.
* Bronchodilators (to give relief at the time of an attack)
* Lifestyle modifications: trigger avoidance, air purification, hydration and monitoring period.

**2.1.4 Challenges in Diagnosis:**

* Early presentation is often subtle and symptoms can here easily be confused with common cold syndromes
* Many Do Not Seek Medical Help Until the Disease Is More Advanced
* Scarcity of accessible healthcare in deprived regions
* Not having or disposing of tools for monitoring the patient in real time

**2.2 Existing System**

At present, diagnosis and treatment of asthma rely on hospital-based measures of physical examination and medical apparatus. These can include spirometry tests to measure the amount and speed of air exhaled, peak flow meters, chest X-rays and sometimes C.T. scans.

There are also some smartphone apps that can be used to manually record medication usage or symptoms but they don't dynamically detect asthma based on AI or respiratory sound analysis.

**2.2.1 Drawbacks of Existing System**

* Requires physical presence and trained healthcare personnel
* Expensive and not portable
* Lacks real-time and continuous monitoring
* Often results in delayed diagnosis
* No AI-based sound classification available to users

**2.3 Proposed System**

The proposed system, AsthmaSense, addresses these limitations by providing an AI-powered mobile application that can analyze a user’s respiratory sounds and predict whether they may be experiencing asthma symptoms. The core of the system is a machine learning model (CNN) trained on respiratory sound datasets (normal and asthma wheeze).

The Android app allows users to upload an audio file (or eventually record one), sends it to the backend Flask server where the model analyzes it, and provides instant prediction results, along with care suggestions.

**2.3.1 Advantages of Proposed System**

* Smartphones offer non-invasive, affordable, and easy access
* Breathing sound analysis enables quick predictions
* Open datasets allow free, expandable, and scalable use
* Users can operate it at home without doctors watching

**2.4 Literature Review**

Several research papers in AI biomedical signal processing, and mobile health applications have an influence on and provide support for the development of AsthmaSense.

Here's a summary of key studies we reviewed:

* + 1. **"An IoT-Based Asthma Intensity Prediction Using Classification Models"**
* Authors: A. Wagh et al.
* The authors created an IoT-enabled system that uses classification algorithms such as SVM and Random Forest to detect asthma severity.
* They applied supervised learning algorithms to patient datasets.
* Their work emphasized the significance of sensor data and real-time alerts.
  + 1. **"Cough Sound Detection and Diagnosis Using AI Techniques: Challenges and Opportunities"**
* Authors: I. B. Ashraf et al.
* Looked at different AI models to detect breathing problems using sound.
* Talked about CNNs and how to get MFCC features from audio.
* Found that using sound to diagnose shows promise for helping with breathing issues.
  + 1. **"Identification of Related Technologies Associated with Asthmatic Wheeze Detection Systems – A Review"**
* Authors: D. Saluja et al.
* Explored how sound waves and pictures of sound can spot wheezing in people with asthma.
* Highlighted the need to create systems that work , don't cost much, and are easy to carry around.
  + 1. **"Predicting Asthma Attacks Through AI-Powered Thermal Imaging Analysis of Breathing Patterns"**
* Authors: R. H. Khan et al.
* Applied AI and heat pictures to check for unusual breathing patterns.
* Promoted blending audio with additional information such as temperature readings and thermal imaging.

**2.5 Methodology, Research Data and Dataset**

**2.5.1 Machine Learning Algorithms:**

* Convolutional Neural Networks (CNN): CNN technology was applied to extracting the features of the audio MFCC and classifying the features.
* Dense layers: Assisted in better generalization, and preventing overfitting.

**2.5.2 Dataset Used (Kaggle Respiratory Sound):**

* Contains labelled recordings of breathing sounds (normal and wheezing)
* Format: wav audios
* Model training and assessment

**2.5.3 Preprocessing Pipeline:**

* Copy audio to MFCC (mel-Frequency Cepstral Coefficients)
* Median filter audio normalization cleaning
* Learn CNN on features
* Validate by confusion matrix and AUC-ROC curve.

**CHAPTER 3**

**SYSTEM REQUIREMENTS & FEASIBILITY STUDY**

**3.1 System Requirements**

The successful development of the AsthmaSense application requires both hardware and software resources. These specifications are based on the needs of machine learning model training, backend integration, and Android application development.

**3.1.1 Hardware Requirements**

The hardware resources listed below are needed to develop, train models, and run tests. These specs might change depending on how much the system needs to grow and work with bigger datasets later on.

Developer's Perspective:

* Processor: Intel Core i5 or better (64-bit architecture)
* RAM: At least 8 GB (16 GB is better)
* Storage: 250 GB SSD (to process data and load models faster)
* GPU: NVIDIA GPU (to speed up model training)
* Smartphone (to Test the App): Android 10 or newer

Table 3.1.1 : Hardware Requirements

|  |  |
| --- | --- |
| **Component** | **Specification** |
| Processor | Intel i5 or above |
| RAM | Minimum 8 GB |
| Storage | 250 GB SSD or more |
| GPU (Optional) | NVIDIA CUDA-enabled GPU |
| Smartphone | Android 10+, Min. 4 GB RAM |

**3.1.2 Software Requirements**

The software stack consists of machine learning frameworks, libraries, Android development tools, and middleware for backend integration.

**Required Tools:**

* Operating System: Windows 11
* Language: Python 3.10, Java
* Android IDE: Android Studio (Meerkat version)
* ML Libraries: TensorFlow, Keras, Librosa, NumPy, Matplotlib
* Backend: Flask (Python)
* API Testing: Postman
* Others: VS Code, Google Colab

Table 3.1.2: Software Requirements

| **Component** | **Software / Tools** |
| --- | --- |
| OS | Windows 11 |
| Programming Language | Python 3.10, Java (Android) |
| IDE | Android Studio, Visual Studio Code |
| Libraries | TensorFlow, Keras, Librosa, NumPy |
| Backend Framework | Flask |
| ML Platform | Google Colab |
| Testing Tool | Postman |

**3.2 Feasibility Study**

A feasibility study examines how practical a proposed project is. It checks if the project can work in terms of cost, technology, time, and usability.

**3.2.1 Technical Feasibility**

Technical feasibility checks if the current development tools, platforms, and programming environments are right to implement the project.

* The system uses Python and its AI libraries (TensorFlow, Librosa) which have good documentation and are open-source.
* Android Studio helps create a smooth UI with Java so most smartphones can run the app.
* Flask, as a small backend, makes it easy to connect Android and the machine learning model.

This means looking at it from a technical angle, we have all the resources at hand. They're well-backed and can be put into action within the time we've got.

**3.2.2 Operational Feasibility**

Operational feasibility shows how well the system fixes issues and helps users in their everyday tasks.

* The app has a simple design that even people without tech skills can use.
* You don't need medical or tech know-how; just upload a sound file for quick results.
* Because the model runs on a basic backend (Flask), you can host it on local or cloud servers without much trouble.
* Proper documentation must be maintained to ensure easy debugging and future expansion.
* The model should be optimized for low-latency inference, making it suitable for real-time applications.

This system cuts down the need for pricey tools and allows more people to access healthcare support — making it work well in practice.

**3.2.3 Economic Feasibility**

Economic feasibility figures out if we can build and launch the project with a sensible budget using tools that don't cost anything.

* We're using open-source tools that are free (Python, TensorFlow, Flask, Android Studio, Google Colab).
* We don't need any hardware sensors or paid APIs right now.
* We just need to put in time and effort, which makes this solution good value for money.

**3.2.3.1 COCOMO Model (Basic)**

The COCOMO (Constructive Cost Model) helps us guess how much work and time we'll need to develop software.

Assumptions:

* Mode: Organic (simple project skilled team)
* Size: 2.8 KLOC (Estimated 2800 lines of code)
* Team Members: 3
* Time Period: 5 months

COCOMO Formulas:

* Effort (E) = a × (KLOC)^b
* Time (T) = c × (Effort)^d
* Constants for Organic Mode: a = 2.4 b = 1.05, c = 2.5, d = 0.38

Calculation:

* Effort (E): 2.4 × (2.8)^1.05 ≈ 7.07 Person-Months
* Time (T): 2.5 × (7.07)^0.38 ≈ 5.25 Months

Table 3.2.3.1: COCOMO Model Summary

| **Parameter** | **Value** |
| --- | --- |
| Mode | Organic |
| a | 2.4 |
| b | 1.05 |
| c | 2.5 |
| d | 0.38 |
| Size (KLOC) | 2.8 |
| Effort | ≈ 7.07 Person-Months |
| Time | ≈ 5.25 Months |
| Team Members | 3 |

The team has 3 members. This balances the workload. The project can finish within 5 months with tasks running at the same time.

**3.2.4 Schedule Feasibility**

Schedule feasibility evaluates whether the project can be completed within the desired timeline of 5 to 5.5 months.

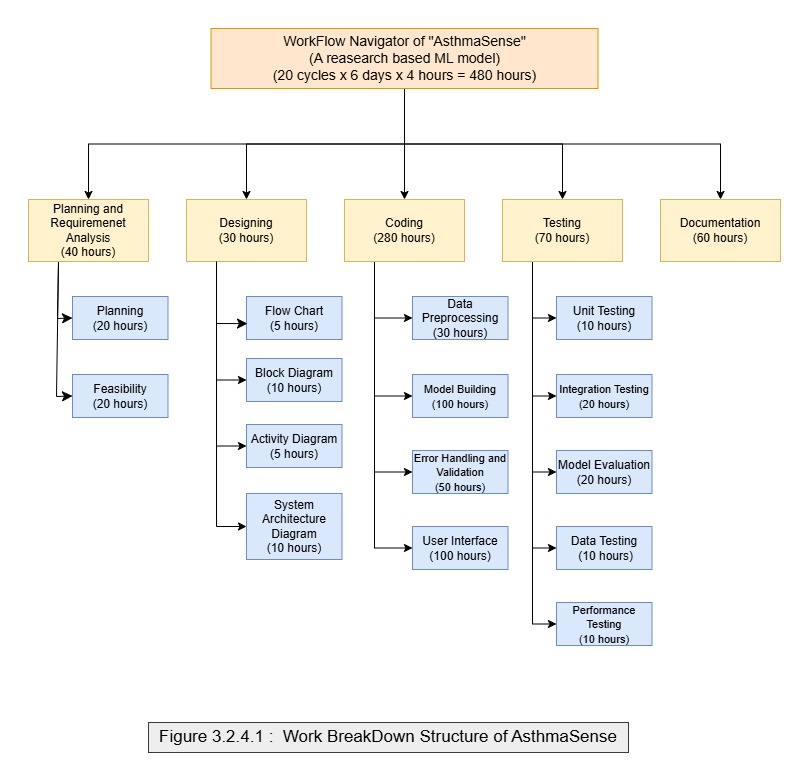
Project Duration: AsthmaSense project completion is planned within a 5 to 5.5 month of time frame. Therefore, effective management practice, regular monitoring of progress, and timeless decision-making are key so that all tasks are completed in due course.

The project was broken into phases including planning, dataset analysis, model training, app development, testing, and documentation.

* Machine learning model and backend were developed in parallel with frontend app UI.
* Using Google Colab accelerated training speed.
* The Android app and API were tested iteratively to ensure robustness

**3.2.4.1 Work Breakdown Structure (WBS):**

WBS means breaking down a project into manageable chunks or estimation of effort in hours. With this detailed breakup, the manager will be better equipped to plan the tasks and resources. Benefits: WBS provides an outline of the project. The team stays focused on their immediate tasks with respect to overall objectives. It also helps find out which areas will become bottlenecks or would possibly need deployed extra resources.



**3.2.4.2 Gantt Chart**

A Gantt chart will be adopted to visualize the project timeline, indicating when each of the tasks is to be done. This will be important in monitoring progress and ensuring that everything runs according to the set schedule. Pros: Through a Gantt chart, the project manager can easily track the deadlines and dependencies, thereby allocating resources properly and adjusting the schedule as need be.

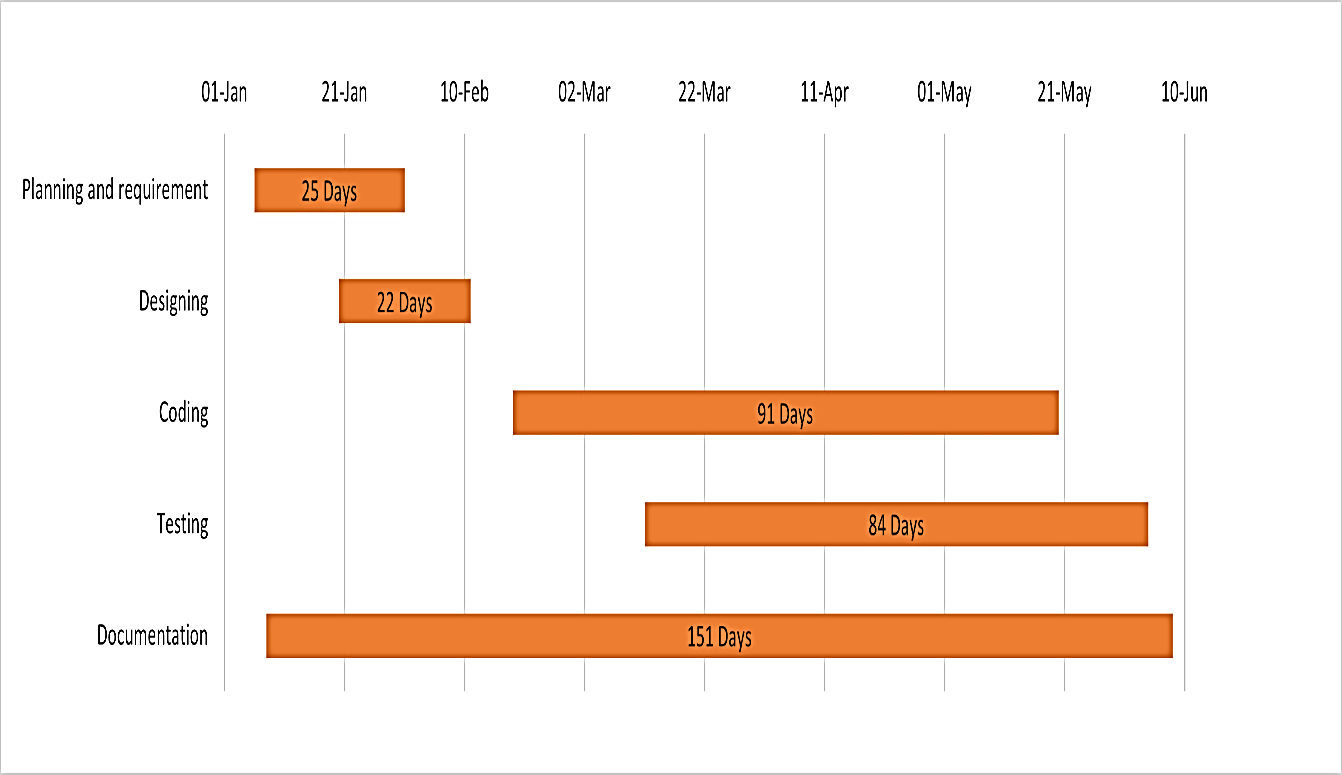


Figure 3.2.4.2 : Gantt Chart of AsthmaSense

**CHAPTER 4**

**SYSTEM DESIGN**

The system design phase focuses on creating the structural and functional layout of the project components. It ensures smooth integration of modules such as the machine learning backend, the Android application frontend, and the communication between them. In this chapter, we illustrate how different parts of the system work together to fulfil the objectives of AsthmaSense.

**4.1 System Architecture Overview**

The system is composed of three main components:

* Android Frontend App: Lets users pick and send an audio file (breathing sound).
* Flask Backend (Python): Gets the audio input from the app, works on it, and passes it to the trained ML model.
* Machine Learning Model: Figures out if the audio comes from someone with asthma or not by looking at audio features (MFCCs).

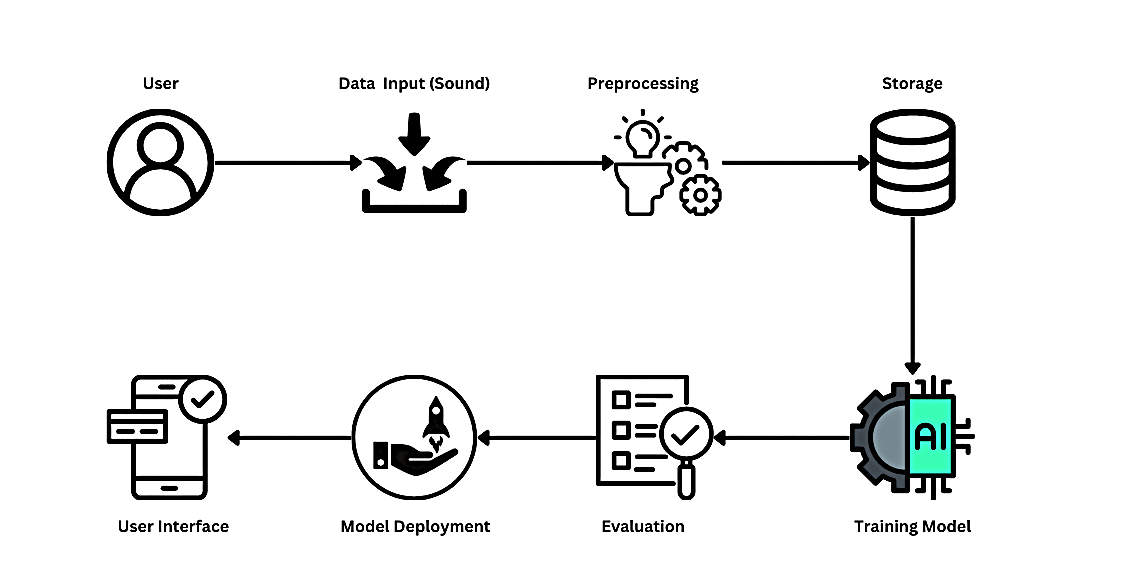


Figure 4.1: System ArchitectureDiagram of AsthmaSense

**4.2 Activity Diagram**

This diagram shows how users interact with the app step by step.

* Users open the app and choose an audio file from their device.
* When they click "Upload & Predict," the app sends the audio to the backend.
* The backend then sends back a prediction, which shows up on the app's screen.

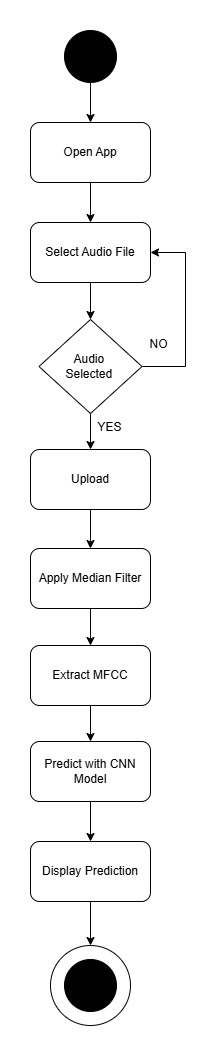


Figure 4.2 : Activity Diagram of AsthmaSense

**4.3 Block Diagram**

The block diagram simplifies the logical structure of the entire system:

* Input Layer: Captures audio data from the user.
* Preprocessing: Extracts MFCC features and removes noise.
* Trained Model: Predicts output using CNN architecture.
* Output Layer: Sends prediction (Asthma / Normal) back to the mobile application.

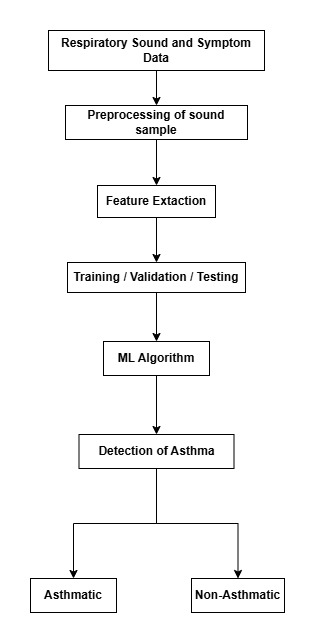


Figure 4.3: Block Diagram

**4.4 Flowchart**

This flowchart visualizes the step-by-step decision-making logic used in the system. It represents the flow from user interaction to model inference and result display.

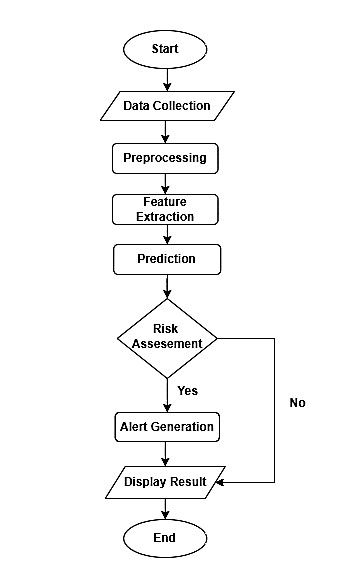


Figure 4.4: Flowchart of AsthmaSense

**4.5 Component Descriptions**

**4.5.1. User Interface (Mobile App)**

* Developed in Android Studio using Java.
* Provides a clean layout with options to choose and upload audio files.
* Displays prediction output received from the backend.

**4.5.2. Backend Server (Flask)**

* Handles file uploads and sends them for processing.
* Connects with the ML model stored locally and returns prediction JSON.

**4.5.3. ML Model (CNN Architecture)**

* Processes MFCC-extracted features from audio data.
* Trained on respiratory datasets using TensorFlow and Keras.
* Model saved as .h5 file and loaded during prediction.

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 System Overview**

The proposed system classifies respiratory sounds into *asthma* or *normal* categories.

The pipeline consists of:

* Data acquisition – .wav files stored in Google Drive.
* Noise reduction – Median filter (kernel = 5).
* Feature engineering – Extraction of 13‑dimensional MFCC vectors.
* Data augmentation – Random pitch‑shift or time‑stretch to balance classes.
* CNN classifier – Trained on MFCC features to perform the binary classification.

Technologies: Google Colab, Python 3.10, TensorFlow 2.x, librosa 0.10, SciPy 1.11, scikit‑learn 1.3, seaborn 0.13.

**5.2 Dataset & Pre‑processing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Datasets** | **Asthma** | **Normal** | **Samples** |
| Dataset 1 [5] | 383 | 133 | 516 |
| Dataset 2 [6] | 99 | 105 | 204 |
| **Total** | **482** | **238** | **720** |

Table 5.2: Respiratory Dataset

**5.2.1 Median filtering**

def median\_filter(audio, kernel\_size=5):  
 return medfilt(audio, kernel\_size)

Removes impulsive noise while preserving transient features of breath sounds.

**Formula:**

For a 1D signal *x[n]*, median filtering with a window size *k* replaces each value by the median of its neighboring values:

*y[n] = median(x[n –r],...,x[n],...,x[n + r])*

Where:

* *y[n]* is the filtered output,
* *r = (k-1)/2* is the radius of the window (for odd *k*),
* Median helps remove impulsive or salt-and-pepper noise.

**Python Package Used:**

from scipy.signal import medfilt

**5.2.2 MFCC**

**Extraction:**

librosa.feature.mfcc(y, sr, n\_mfcc=13, n\_fft=1024) – mean of each coefficient gives a fixedlength 13D vector.

**Python Package Used:**

import librosa

mfcc = librosa.feature.mfcc(y=signal, sr=sample\_rate, n\_mfcc=13)

Where:

* y: audio signal
* sr: sample rate
* n\_mfcc: number of MFCC features to extract

**5.2.3 Augmentation**

choice = random.choice(['stretch','pitch','none'])

* Time‑stretch : 0.8 – 1.2×
* Pitch‑shift : –2 … +2 semitones

### **5.3 Model Architecture**

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer** | **Output Shape** | **Params** | **Activation** |
| **Input** | (None, 13, 1) | 0 | — |
| Conv1D (64, 3) | (None, 11, 64) | 256 | ReLU |
| MaxPooling1D (2) | (None, 5, 64) | 0 | — |
| Conv1D (128, 3) | (None, 3, 128) | 24 704 | ReLU |
| MaxPooling1D (2) | (None, 1, 128) | 0 | — |
| Flatten | (None, 128) | 0 | — |
| Dense (128) | (None, 128) | 16 512 | ReLU |
| Dense (2) | (None, 2) | 258 | Softmax |
| **Total params** | **41 730** |  |  |

Table 5.3: Model Architecture

* Loss : categorical\_crossentropy
* Optimizer : Adam(lr = 0.001)
* Epochs : 50
* Batch size : 32
* Validation split : 0.20

**5.4 Model Implementation**

The machine learning model developed in this project utilizes a Convolutional Neural Network (CNN) for classifying respiratory audio signals into two categories — asthmatic and normal. The following algorithm was used to implement the model:

**5.4.1 Algorithm: CNN-Based Asthma Detection**

Data Collection: Use audio datasets from Kaggle (e.g., respiratory sound and asthma detection datasets).

Data Preprocessing: Use median filtering to clean up the audio signals. Then, get MFCC features from each sample.

Feature Extraction: Turn each signal into a matrix of Mel-frequency cepstral coefficients (MFCCs). To do this, use the Librosa library.

Model Design:

* Input Layer: MFCC features (e.g., 13×1)
* Conv1D and MaxPooling Layers
* Flatten Layer
* Dense Layer with Dropout
* Output Layer with Softmax

Model Compilation: Use categorical cross-entropy as the loss function and Adam optimizer.

Model Training: Train the model on preprocessed data for 50 epochs using 80:20 train-test split.

Model Saving: Save the trained model as asthma\_cnn\_model.h5.

**5.5 Backend Integration**

A lightweight Flask API was created to connect the trained model with the mobile application.

Steps:

1. Accept a .wav audio file uploaded from the app.
2. Preprocess the file using MFCC extraction.
3. Load the saved model (.h5) and predict the class.
4. Return the result (asthma / normal) in JSON format.

**5.6 Android Application Implementation**

The mobile app is developed using Java in Android Studio (Meerkat). The app allows users to:

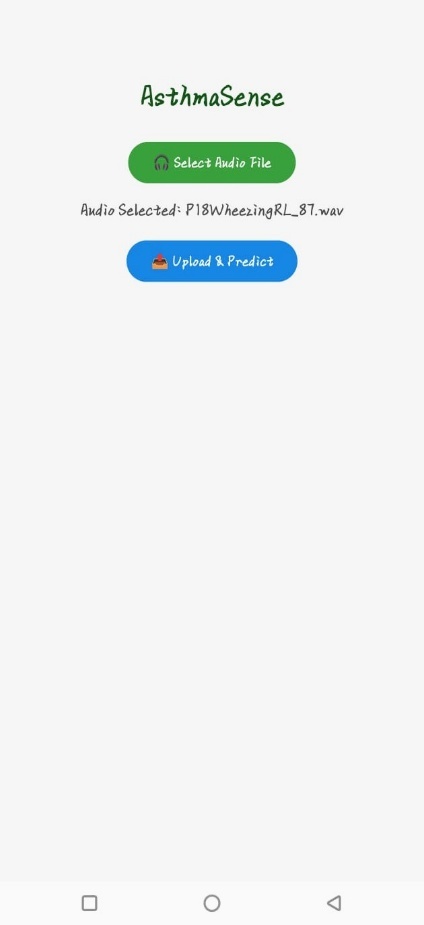
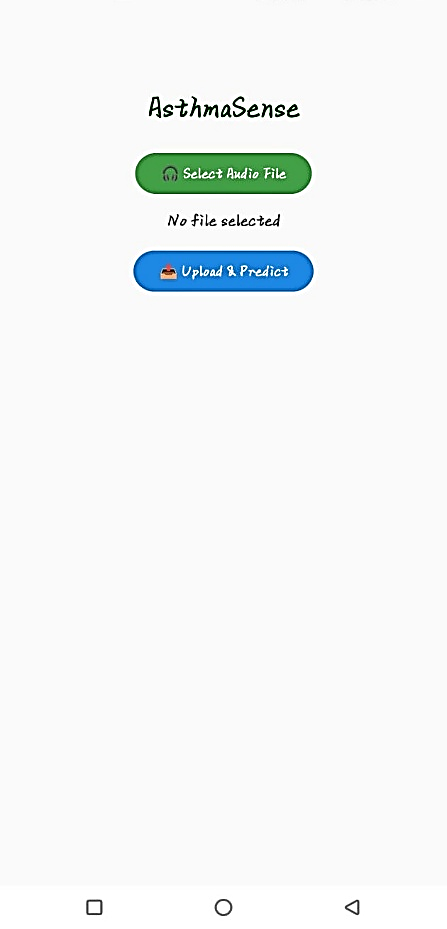
* Select a breathing sound .wav file (Stethoscope Audio Data) .
* Upload the file to the Flask server.
* View prediction results on the same screen.

App Workflow:

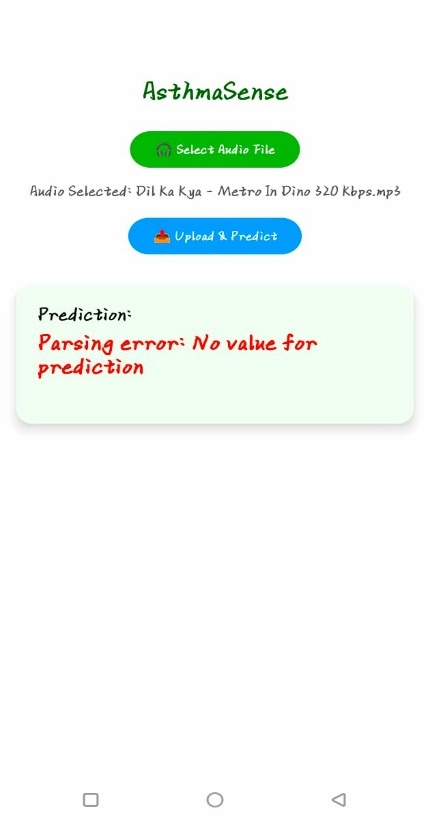
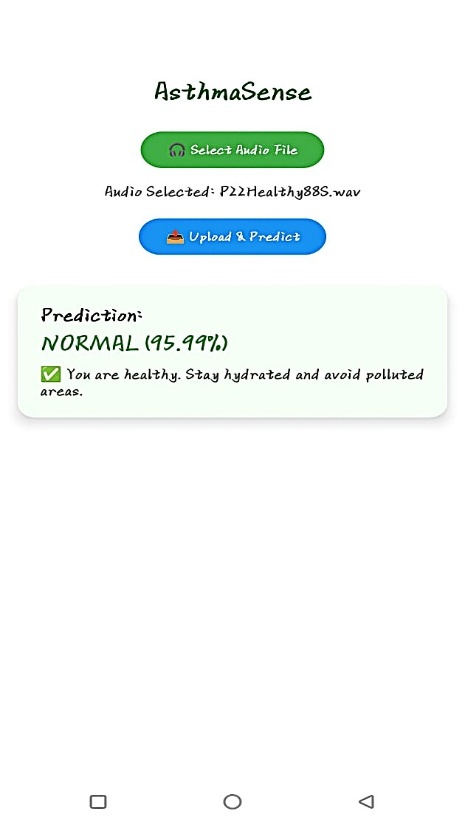
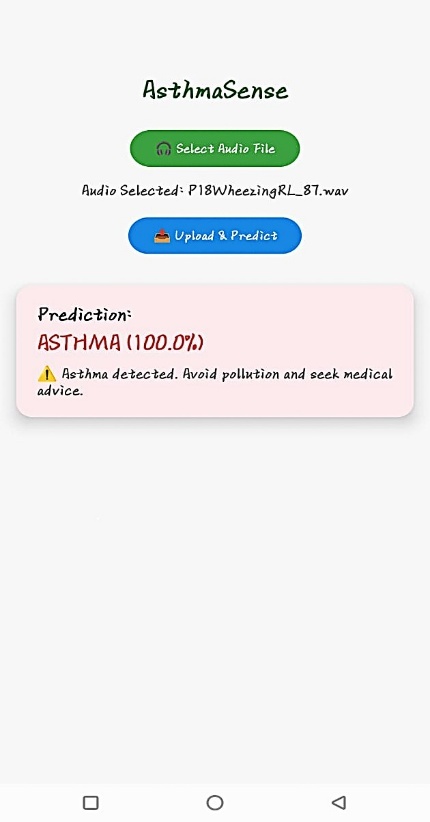
1. Homepage with two buttons: Select Audio File and Upload & Predict.
2. File chooser opens Google Drive/local storage.
3. Selected file name is displayed.
4. On clicking Upload, result is received and displayed as:

Prediction: Asthma Detected (Confidence: 87.5%)

**5.7 AsthmaSense User Interface Screenshots (Android App)**

****

UI 5.7.1: Home Page UI 5.7.2: Storage UI 5.7.3: Audio Selected



UI 5.7.4: Asthma Detected UI 5.7.5: Normal Detected UI 5.7.6: Invalid Audio Input

**CHAPTER 6**

**TESTING AND REPORTS**

**6.1 Testing Strategy**

The system is tested using both automated methods (model accuracy) and manual testing (via app and API). The goal is to verify:

* Upload functionality
* API communication
* Model inference reliability
* UI responsiveness

### **6.2 Test Cases Table**

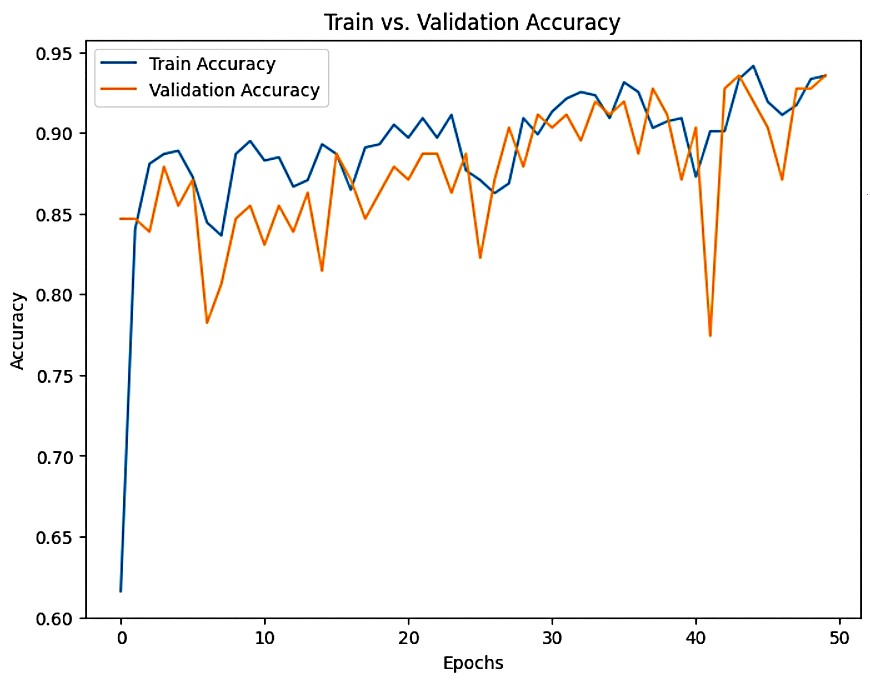
| **Test Case** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| --- | --- | --- | --- | --- |
| Upload valid .wav file | asthma.wav | "Asthma Detected" | Asthma Detected | Pass |
| Upload normal audio | normal.wav | "Normal" | Normal | Pass |
| Upload non-breathing sound (.mp3) | song.mp3 | "Parsing Error or Invalid" | Parsing Error | Pass |
| No file selected | — | "Please select file" | Please select file | Pass |
| Network error | — | Upload failed | Upload failed | Pass |

**6.3 Evaluation Metrics Table**

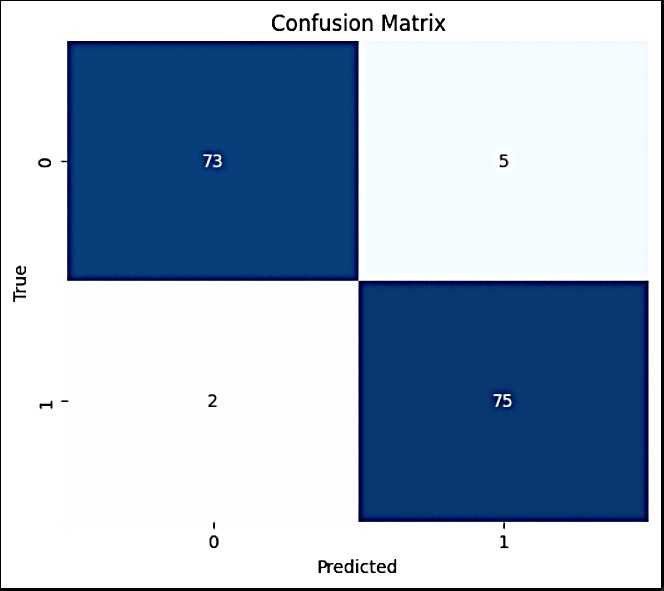
|  |  |  |
| --- | --- | --- |
| **Metric** | **Value** | **Comment** |
| **Training accuracy** | 92.84% | Final training accuracy after 50 epoch |
| **Validation accuracy** | 93.55 % | Shows low overfitting |
| **Test accuracy** | 95.48 % | Confirms good generalization |
| **Precision / Recall / F1** | Asthma → P: 0.97, R: 0.94, F1: 0.95  Normal → P: 0.94, R: 0.97, F1: 0.96 | Balanced classes |
| **AUC (Asthma)** | *0.98* | High separability |
| **AUC (Normal)** | *0.98* | High separability |
| **NSR before filter** | 0.0404 | Noise‑to‑signal ratio |
| **NSR after filter** | 0.0000 | Median filter reduces impulsive noise |

**6.4 Graphical Results**

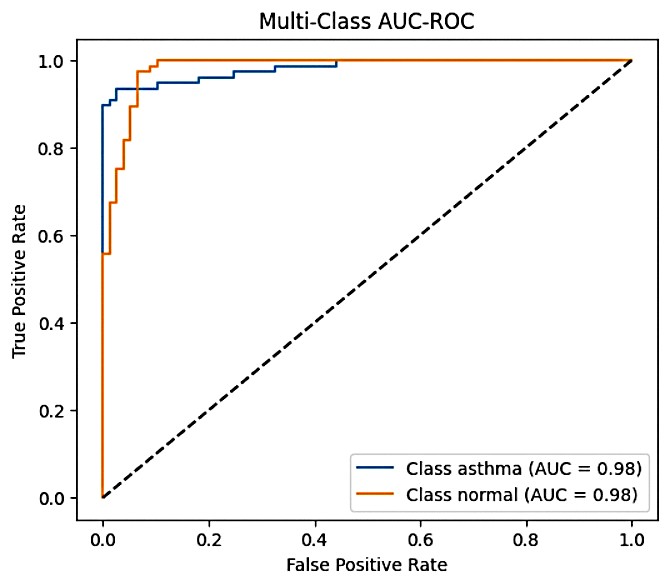
* + 1. **Training vs Validation Accuracy**

******

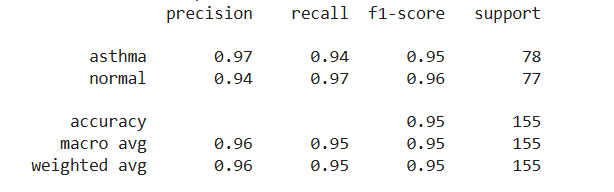
**6.4.2 Confusion Matrix for Testing Dataset**



**6.4.3 AUC-ROC Curves**



**6.4.4 Classification Report**

****

**6.5 Model Performance**

* Test Accuracy: 95.48%
* Validation Accuracy: 93.55%
* AUC-ROC Score: 0.98

**6.6 Comparative Experiments Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variant | Architecture | Changes / Notes | Training Acc | Val Acc | Testing ACC |
| Proposed CNN | 1D CNN | ReLU + Softmax | 92.84% | 93.55% | 95.48% |
| LSTM Model | CNN + LSTM | Added LSTM + BatchNorm | 82.91% | 75.29% | 80.49% |
| AlexNet 1D | Deep 1D CNN | AlexNet-style depth + BN + Dropout | 95.36% | 89.13% | 89.66% |
| 2D CNN | 2D CNN (MFCC reshaped to image) | Used Conv2D + MFCC (2D) input | 98.63% | 96.33% | 96.14% |

**6.7 Discussion**

* Median filtering significantly lowered NSR, improving feature quality.
* The modest gap between training and validation curves confirms minimal over‑fitting.
* Class imbalance was mitigated via augmentation (time‑stretch, pitch‑shift).
* Final model meets project objective of ≥ 90 % accuracy on unseen data.

**6.8 Tools Used for Testing**

* Postman: API testing
* Google Colab: Model evaluation
* Android Emulator + Real Device: UI testing
* TensorFlow: Model training logs and accuracy

**CHAPTER 7**

**CONCLUSION AND FUTURE WORK**

**7.1 Conclusion**

The project "AsthmaSense – AI-Based Asthma Prediction System" shows how machine learning and mobile tech can help in healthcare. It uses sound processing and special neural networks to spot asthma breathing patterns pretty well.

Users can record or pick their breathing sounds with the Android app and get quick predictions about asthma. The system hooks up a simple web service to connect on-device predictions with mobile ease of use giving people a straightforward way to check their health.

The model showed strong results in tests reaching an accuracy of 95.48%. This shows it can spot the difference between regular and asthmatic breathing noises. The project also highlights how strong open-source tools like TensorFlow, Librosa, and Android Studio can be. These tools helped build a cheap and easy-to-grow solution to manage breathing health.

What's more, working on this project taught us a lot. We learned about cleaning up data pulling out key features (MFCC), training models building APIs, and designing mobile apps that look good and work well. This made the project a big win across many fields.

**7.2 Future Work**

Though the present implementation of AsthmaSense has provided a solid foundation, there are several ways in which the system can be extended further:

**7.2.1 Increase Dataset and Enhance Model Accuracy**

The model used in the present implementation is based on a small dataset. By integrating larger and more heterogeneous audio datasets — such as wheeze, cough, and respiratory patterns in various demographics — model generalizability can be enhanced.

**7.2.2 Multi-Class Classification**

An additional feature in the future would be to teach the system to categorize input into three or more classes — for example, asthma, normal, and noise or irrelevant sound. This would go a long way toward eliminating false positives from music or background sound.

**7.2.3 Offline Prediction using TensorFlow Lite**

Adding TensorFlow Lite to the Android app would enable users to make predictions without an internet connection, which would make the app more accessible in rural and far-off locations.

**7.2.4 Real-Time Audio Recording**

Include a recording option within the app itself so that users are not required to upload pre-recorded audio files. The recorded audio can be sent directly to the backend or TensorFlow Lite model for prediction.

**7.2.5 Health Tips and Preventive Measures**

The app can suggest lifestyle tips, emergency contact details, or air pollution alerts based on the prediction to steer the user towards improved asthma control.

**7.2.6 AQI and Pollution Alert Integration**

Incorporate real-time Air Quality Index (AQI) information based on open APIs to inform users of high-risk surroundings and suggest preventive actions.

**7.2.7 Medical Community Integration**

The system can be made more advanced to maintain user history, build health profiles, and support secure communication with physicians or medical practitioners.

**REFERENCES**

[1] Uwanthika, Isuri & Madushanka, Pavithra & Deraniyagala, Dewni. (2023). Identification of related technologies associated with Asthmatic Wheeze Detection Systems: A Review.

[2] J. N. V. R. S. Kumar, B. N. Nishant, S. V. Suraj, I. S. S. Rao, S. R. Pappu and B. Lokesh, "An IoT-Based Asthma Intensity Prediction Using Classification Models," 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2024, pp. 404-410, doi: 10.1109/ICSCSS60660.2024.10625043. keywords: {Logistic regression;Machine learning algorithms;Medical services;Predictive models;Prediction algorithms;Sensor systems;Sensors;Asthma;Internet of Things (IoT);Sensors;Symptoms;Cloud Storage;Risk Prediction;Data Analysis;Machine Learning;Patient},

[3] K. S. Alqudaihi et al., "Cough Sound Detection and Diagnosis Using Artificial Intelligence Techniques: Challenges and Opportunities," in IEEE Access, vol. 9, pp. 102327-102344, 2021, doi: 10.1109/ACCESS.2021.3097559. keywords: {Pulmonary diseases;Artificial intelligence;COVID-19;Medical services;Tools;Lung;X-ray imaging;Artificial intelligence (AI);cough detection;2019 novel coronavirus disease (Covid-19);respiratory illness diagnosis;cough-based diagnosis},

[4] A. Hamza, Y. Himeur, A. Amira and A. Oulefki, "Predicting Asthma Attacks Through AI-Powered Thermal Imaging Analysis of Breathing Patterns," 2024 IEEE 10th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService), Shanghai, China, 2024, pp. 82-86, doi: 10.1109/BigDataService62917.2024.00018. keywords: {Measurement;Accuracy;Computational modeling;Imaging;Nose;Medical services;Predictive models;Data models;Thermal analysis;Asthma;Asthma attack prediction;Transfer learning;Thermal images;VGG16;Inhale;Exhale},

[5] Anik Chakraborty, Rittik Das Gupta, Md. Zesanul Kabir, Sourav Dhar, Development of an IoT-enabled cost-effective asthma patient monitoring system: Integrating health and indoor environment data with statistical analysis and data visualization, Internet of Things, 10.1016/j.iot.2023.100942, 24, (100942), (2023)

[6] Dataset 1: https://www.kaggle.com/datasets/mohammedtawfikmusaed/asthma- detection-dataset-version-2?resource=download

[7] Dataset 2: https://data.mendeley.com/datasets/jwyy9np4gv/3