

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI – 590018**



**“LUNG CANCER DETECTION USING
CT(COMPUTER TOMOGRAPHY) IMAGE
PROCESSING & MACHINE LEARNING”**

A report submitted in partial fulfilment of the requirements for the

Award Degree of,

**BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING**

Submitted By,

Name	USN
Mr. Aftab I Yaragatti	2HN22CS005
Miss. Bhagyashree S Poojari	2HN22CS013
Miss. Kavita K Dodagoudanavar	2HN22CS024
Miss. Laxmi A Bilur	2HN22CS026

Under The Guidance of,

Prof. M. G. Ganachari



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
S. J. P. N. Trust's
HIRASUGAR INSTITUTE OF TECHNOLOGY, NIDASOSHI – 591236**

Inculcating Values, Promoting Prosperity

Approved by AICTE, Recognized by Govt. of Karnataka, Affiliated to VTU
Belagavi

Accredited at 'A+' Grade by NAAC
Programmes Accredited by NBA: CSE, ECE
Academic Year 2025 - 26

S. J. P. N. Trust's

HIRASUGAR INSTITUTE OF TECHNOLOGY, NIDASOSHI – 591236

Inculcating Values, Promoting Prosperity

Approved by AICTE, Recognized by Govt. of Karnataka, Affiliated to VTU
Belagavi

Accredited at 'A+' Grade by NAAC

Programmes Accredited by NBA: CSE, ECE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Academic Year 2025 - 26



CERTIFICATE

This is to certify that the project entitled, **“Lung Cancer Detection Using CT(Computer Tomography) Image Processing & Machine Learning”** is a bonafide work carried out by **Mr. Aftab I Yaragatti (2HN22CS005), Miss. Bhagyashree S Poojari (2HN22CS013), Miss. Kavita K Dodagoudanavar (2HN22CS024), Miss. Laxmi A Bilur (2HN22CS026)** the students of **Hirasugar Institute of Technology, Nidasoshi** in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year **2025 – 26**, is certified that all correction/ suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Prof. M. G. Ganachari
Guide

Dr. S. V. Manjaragi
H. O. D.

Dr. S. C. Kamate
Principal

Name of the Examiners

Signature with Date

1. _____

2. _____

ACKNOWLEDGEMENT

First and foremost, we express our gratitude to the Almighty for providing us with the strength, wisdom, and perseverance to complete this project successfully.

We extend our sincere gratitude to our Principal **Dr. S. C. Kamate**, Hirasugar Institute of Technology, for his unwavering support and encouragement throughout the duration of this project. We would like to convey our heartfelt gratitude to **Dr. S. V. Manjaragi**, Head of the Department of Computer Science and Engineering, for his guidance and unique ideas, which have greatly contributed to the success of our project.

Our sincere appreciation goes to our respected guide, **Prof. M. G. Ganachari**, for his mentorship, expertise, and continuous support, which have been instrumental in shaping our project and fostering our academic growth. We would like to acknowledge and thank **Prof. M.G.Ganachari**, the project coordinator, for his assistance, coordination, and organizational efforts that have facilitated the smooth progress of our project. We would also like to express our gratitude to all the professors and staff members of the Computer Science and Engineering department for their support, encouragement, and invaluable contributions at every stage of our project journey.

Last but not the least; we are very much indebted to our parents, classmates and friends who directly or indirectly helped us in completing this project. Their collective efforts have been indispensable in the successful completion of our project "**Lung Cancer Detection Using CT(Computer Tomography) Image Processing & Machine Learning.**"

Grateful for your unwavering support and guidance, We proudly present this project report, a testament to our collective dedication and perseverance. With sincere appreciation, we extend our heartfelt thanks, always hopeful for continued success.

Warm regards,

Mr. Aftab I Yaragatti

Miss. Bhagyashree S Poojari

Miss. Kavita K Dodagoudanavar

Miss. Laxmi A Bilur

ABSTRACT

Lung cancer remains the leading cause of cancer-related mortality worldwide, necessitating the development of advanced diagnostic tools for early detection. This project presents an innovative approach to lung cancer detection by integrating computed tomography (CT) imaging with machine learning (ML) techniques. The methodology encompasses several key stages: Preprocessing of CT images to enhance quality and reduce noise, segmentation to isolate regions of interest, feature extraction to identify relevant patterns, and classification to differentiate between benign and malignant lesions. A convolutional neural network (CNN)-based model is employed to automate the analysis process, trained on a diverse dataset of annotated CT images. The model's performance is evaluated using metrics such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC). Preliminary results indicate promising outcomes, with the model achieving high accuracy.

INDEX

Acknowledgement	iii
Abstract	iv
List of Figures	v
List of Tables	vi

Chapter	Page
1 Introduction	
1.1 Purpose	03
1.2 Document convention	04
1.3 Project scope	04
2 Literature Survey	
2.1 Existing system	07
2.2 Proposed system	08
2.3 Comparison	08
3 Requirement Specification	
3.1 Functional Requirements	12
3.2 Use case diagrams	13
3.3 Use case description	14
3.4 Non-Functional requirements	15
3.5 S/W requirements	16
3.6 H/W requirements	17
4 System analysis and design	
4.1 System overview	20
4.2 System architecture	22
4.3 Component design	23
4.4 Data Flow diagram	24
4.5 Sequence diagram	26
4.6 Data Design and Description	27
4.7 Entity Relationship Diagram (ERD)	28
5 System Implementation	
5.1 Language description	31
5.2 Software description	31
5.3 Pseudo code	32
6 Testing and Validation	
6.1 Testing Strategies	36
6.2 Model Testing	37
6.3 Integration Testing	39
6.4 System Testing	40
6.5 Acceptance Testing	41
7 Results and Screen shots	43
8 Conclusion and future work	49
References	51

LIST OF FIGURES

Fig. 3.2	Use case diagram	14
Fig. 4.1	General working of our system	22
Fig. 4.2	System Architecture	22
Fig. 4.4	Data flow diagram	24
Fig. 4.5	Sequence Diagram	26
Fig. 4.7	ERD for lung cancer detection	28
Fig. 7.1	User Dashboard	43
Fig. 7.2	Admin Dashboard	44
Fig. 7.3.1	Lung Cancer Dashboard	44
Fig. 7.3.2	Recent Scans	45
Fig. 7.3.3	Patient list screen	45
Fig. 7.3.4 (a)	Select Patient	46
Fig. 7.3.4 (b)	Upload CT Scan Image	46
Fig. 7.3.4 (c)	Select Model for Prediction	47
Fig. 7.3.4 (d)	Lung Cancer Detection Result	47

LIST OF TABLES

Table 1.2	Document Conversions	04
Table 2.3	Existing System Vs Proposed System	09
Table 6.2	Module Testing	37

**DEDICATED TO PARENTS
FAMILY FRIENDS AND TEACHERS**

CHAPTER 1

CHAPTER 1

INTRODUCTION

Lung cancer is one of the most serious and life-threatening diseases affecting millions of people worldwide. It is the leading cause of cancer-related deaths globally, accounting for a significant number of fatalities each year. The high mortality rate associated with lung cancer is primarily due to late-stage diagnosis, as the disease often remains asymptomatic during its early phases. When symptoms such as persistent cough, chest pain, or shortness of breath become noticeable, the cancer has typically progressed to an advanced stage, making treatment more complex and less effective. Medical imaging plays a crucial role in the diagnosis and monitoring of lung cancer. Among various imaging modalities, Computed Tomography (CT) scans are widely used due to their ability to provide high-resolution, cross-sectional images of lung tissues, enabling the detection of small nodules and abnormalities that may not be visible in standard X-ray images. Despite their effectiveness, the interpretation of CT scans relies heavily on the expertise of radiologists. Manual analysis is not only time-consuming and labor-intensive but also subject to human error and variability, especially when handling large volumes of imaging data. With the rapid advancement of Artificial Intelligence (AI) and Machine Learning (ML), there has been a growing interest in applying these technologies to the field of medical diagnosis. Machine learning algorithms have the ability to learn from historical medical data, recognize complex patterns, and make accurate predictions. In the context of lung cancer detection, ML-based systems can analyze CT scan images, extract relevant features, identify suspicious lung nodules, and classify them as benign or malignant with high accuracy.

1.1 Purpose

The primary objective of this project is to design and develop an advanced, automated lung cancer detection system that utilizes Computed Tomography (CT) imaging in conjunction with Machine Learning (ML) techniques to enable early, accurate, and reliable diagnosis of lung cancer. Lung cancer is one of the most life-threatening diseases worldwide and continues to be the leading cause of cancer-related deaths, primarily due to late-stage diagnosis. In many cases, symptoms appear only after the disease has progressed to advanced stages, significantly reducing survival rates and limiting effective treatment options. Early detection plays a critical role in improving patient outcomes, as identifying lung cancer at an initial stage greatly increases the chances of successful treatment, reduces mortality rates, and enhances overall quality of life. However, conventional diagnostic approaches rely heavily on manual examination of CT scans by radiologists, which can be time-consuming, subjective, and prone to inter-observer variability. Additionally, the increasing volume of medical imaging data places a substantial burden on healthcare professionals, potentially leading to delayed diagnoses. To address these

challenges, this project proposes an intelligent, automated system capable of analyzing lung CT images with minimal human intervention. The system employs machine learning and image processing techniques to extract meaningful features from CT scans, detect abnormal lung nodules, and accurately classify them as benign or malignant. By learning from large datasets of labeled medical images, the ML model can identify complex patterns and subtle variations that may be difficult to detect through traditional visual inspection.

1.2 Document conventions

The following table 1.2 conventions are used through the document and should be understood accordingly. The table below shown a list of all the conventions used on the document given below.

Table 1.2 : Document Conversions

Abbreviation	Full Form
AI	Artificial Intelligence
ML	Machine Learning
CV	Computer Vision
CNN	Convolutional Neural Network
UI	User Interface
GUI	Graphical User Interface
API	Application Programming Interface
DFD	Data Flow Diagram
SRS	Software Requirement Specification
IDE	Integrated Development Environment
DB	Database
HCI	Human–Computer Interaction
PDF	Portable Document Format
FPS	Frames Per Second
CPU	Central Processing Unit
GPU	Graphics Processing Unit

1.3 Project Scope

This project aims to develop an automated system for the early detection of lung cancer by integrating Computed Tomography (CT) imaging with Machine Learning (ML) algorithms. The system is designed to analyze CT scan images through a structured pipeline that includes image preprocessing, lung region segmentation, feature extraction, and classification of suspicious regions. By automating these processes, the proposed solution seeks to assist radiologists in identifying potential malignancies at an early stage, thereby reducing diagnostic time, minimizing human subjectivity, and improving accuracy and consistency in lung cancer diagnosis. The system is intended to function as a computer-aided diagnostic

tool that supports medical professionals rather than replacing them.

The scope of the project includes the compilation and utilization of a diverse CT image dataset comprising both benign and malignant cases for model training and evaluation. A comprehensive image preprocessing pipeline will be developed to enhance image quality, normalize intensity values, and reduce noise across the dataset. The system will incorporate a segmentation module to accurately isolate lung regions and detect potential nodules within CT images. Relevant features such as shape, texture, and intensity will be extracted to characterize the segmented regions. A suitable machine learning model, such as a Convolutional Neural Network (CNN), will be designed, trained, and optimized to classify lung nodules as benign or malignant. The performance of the system will be evaluated using standard metrics including accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC). Detailed technical and user documentation will also be produced, outlining system design, methodologies, algorithms, and experimental results. The project scope explicitly excludes several components to ensure feasibility within academic and resource limitations. It does not involve conducting real-world clinical trials or patient studies, nor does it cover obtaining medical or regulatory approvals. The design or manufacturing of specialized CT imaging hardware is outside the scope, as the project focuses solely on software-based solutions. Additionally, the system will not be integrated with existing hospital information systems, electronic health records, or clinical workflows, and will be evaluated only in a simulated or experimental environment.

1.4 Summary

This project focuses on the design and implementation of an intelligent and automated system for the early detection of lung cancer by integrating computed tomography (CT) imaging with machine learning (ML) techniques. Lung cancer continues to be one of the leading causes of cancer-related mortality worldwide, largely due to late diagnosis and the limitations of traditional diagnostic approaches that rely on manual interpretation of medical images. The proposed system aims to overcome these challenges by providing a computer-aided diagnostic solution that can analyze CT scan images efficiently and accurately, thereby supporting medical professionals in early-stage cancer detection. The system follows a structured workflow that includes image preprocessing to enhance quality and reduce noise, segmentation of lung regions to isolate areas of interest, and extraction of meaningful features such as shape, texture, and intensity from detected nodules. Advanced machine learning models, particularly convolutional neural networks (CNNs), are trained using labeled CT image datasets to classify lung nodules as benign or malignant.

CHAPTER 2

CHAPTER 2

LITERATURE SURVEY

Early detection of lung cancer is critical due to its high mortality rate and the significant improvement in patient survival when diagnosed at an early stage. Traditional diagnosis relies on radiologists interpreting CT scans, which, although detailed, is time-consuming and prone to human error when handling large datasets [1]. This has led to increased research interest in automated and computer-aided diagnostic (CAD) systems for lung cancer detection. Early automated approaches used classical image processing techniques such as thresholding, edge detection, and morphological operations for lung segmentation and nodule detection [2]. However, these methods depended on handcrafted features and predefined rules, limiting robustness due to variations in nodule shape and intensity [3]. With the advancement of machine learning, feature-based classifiers such as SVM, k-NN, and Random Forests improved detection accuracy but still required manual feature extraction and expert knowledge [4]. Recent studies have adopted deep learning models, particularly CNNs and architectures like ResNet, which automatically learn discriminative features from CT images and achieve higher accuracy [5], [6]. Despite their effectiveness, many deep learning systems demand large datasets and high computational resources, highlighting the need for efficient and scalable AI-based lung cancer detection solutions.

2.1 Existing System

The existing lung cancer detection system primarily relies on manual analysis of medical imaging data, especially computed tomography (CT) scans, performed by experienced radiologists. In this approach, radiologists visually examine CT images to identify lung nodules and determine whether they are benign or malignant. While CT imaging provides high-resolution and detailed views of lung tissues, the diagnostic process is time-consuming, highly dependent on human expertise, and subject to inter-observer variability. The accuracy of diagnosis may vary based on the radiologist's experience, workload, and fatigue, particularly when large volumes of CT scans need to be analyzed. In some cases, small or early-stage nodules may be overlooked, leading to delayed diagnosis and reduced treatment effectiveness. Some existing systems incorporate traditional computer-aided diagnosis (CAD) tools that use basic image processing and classical machine learning techniques. These systems often rely on handcrafted features and rule-based methods for segmentation and classification. Although such approaches can assist in detection, they generally suffer from limited accuracy, poor scalability, and lack robustness when applied to diverse datasets. Moreover, these systems typically require significant manual intervention and do not fully exploit the potential of modern machine learning and deep learning techniques.

2.2 Proposed System

The proposed system introduces an automated and intelligent lung cancer detection framework that integrates CT scan image processing with advanced machine learning and deep learning algorithms. The system is designed to minimize human intervention by automating key stages such as image preprocessing, lung region segmentation, feature extraction, and classification of lung nodules. Advanced models, particularly Convolutional Neural Networks (CNNs), are employed to automatically learn discriminative features directly from CT images, enabling accurate differentiation between benign and malignant nodules.

Unlike the existing system, the proposed solution aims to provide higher diagnostic accuracy, consistency, and efficiency. By leveraging machine learning, the system can analyze large volumes of CT scans quickly and reliably, reducing the workload on radiologists and minimizing the chances of human error. The proposed system serves as a computer-aided diagnostic tool, offering decision support rather than replacing medical professionals. It also enables early detection of lung cancer, which is crucial for effective treatment planning and improved patient survival rates. Overall, the proposed system represents a scalable, reliable, and cost-effective solution that overcomes the limitations of traditional diagnostic methods and advances intelligent healthcare systems.

2.3 Comparison

Table 2.3 presents a comprehensive comparison between the proposed AI-based Eye Tracking System for paralyzed patients, traditional assistive communication methods, and existing AI-based eye-tracking solutions. The comparison is carried out based on key performance parameters, including usability, response time, adaptability, cost efficiency, user independence, and operational complexity. This analysis highlights the strengths and limitations of each approach and demonstrates how the proposed system offers improved efficiency, enhanced user autonomy, and reduced complexity compared to conventional and existing solutions.

Table 2.3 : Existing System vs Proposed System

Feature	Existing System	Proposed System
Segmentation Approach	Manual or traditional segmentation techniques	Automated lung and nodule segmentation using deep learning
Feature Extraction	Handcrafted and basic features	Advanced texture, shape, and intensity features learned automatically
Classification Algorithms	Rule-based or traditional ML models	Deep learning–based models (CNN, hybrid DL models)
Learning Capability	Limited, static learning	Adaptive learning from large datasets
Automation Level	Low, requires manual intervention	High, end-to-end automation
Accuracy	Moderate and inconsistent	High and consistent
Scalability	Limited scalability	Highly scalable for large datasets
Processing Time	Time-consuming	Faster and efficient processing
Handling Complex Patterns	Poor handling of complex variations	Excellent handling of complex and subtle patterns
Robustness to Noise	Sensitive to noise and image variations	Robust due to advanced preprocessing and learning
Integration	Standalone tools	Integrated end-to-end workflow
Interpretability	Moderate	High (with explainable AI techniques)
Clinical Adoption	Widely used traditional methods	Under evaluation and experimental
Human Dependency	High dependence on expert analysis	Reduced dependency, decision support system
Error Rate	Higher due to manual analysis	Lower due to automated and optimized models
Adaptability	Difficult to update and retrain	Easy to retrain with new data
Support for Early Detection	Limited	Strong support for early-stage detection

2.4 Summary

This chapter highlights the critical importance of early and accurate detection of lung cancer, given its high mortality rate and the significant impact timely diagnosis has on patient survival. It explains how traditional diagnostic approaches rely heavily on manual interpretation of CT scans by radiologists, which can be time-consuming and prone to human error, especially when handling large volumes of medical imaging data. To address these challenges, the chapter introduces an AI-based lung cancer detection system that utilizes CT image processing and machine learning techniques to automatically identify and classify lung nodules. The system employs advanced preprocessing, segmentation, and deep learning models to support accurate diagnosis through an efficient and user-friendly interface. The chapter concludes by emphasizing the project's purpose, scope, and significance in modern healthcare, demonstrating how artificial intelligence can enhance diagnostic accuracy, reduce workload on medical professionals, and improve early detection outcomes.

CHAPTER 3

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Functional Requirements

The Lung Cancer Detection System is designed to perform a set of essential operations that collectively aim to support early, accurate, and efficient diagnosis of lung cancer using CT scan images and machine learning techniques. The following functional requirements define the core capabilities and behavior of the proposed system.

3.1.1 Image Acquisition Module

The system shall accept and process lung CT scan images obtained from medical imaging devices or publicly available datasets. It shall support standard medical and image formats such as DICOM, PNG, and JPEG. The module shall ensure proper loading, validation, and storage of image data for further processing while maintaining data integrity and consistency.

3.1.2 Image Preprocessing Module

The system shall preprocess the acquired CT images to improve image quality and suitability for analysis. Preprocessing operations shall include noise reduction, intensity normalization, contrast enhancement, resizing, and artifact removal. This module shall ensure uniformity across images originating from different sources and imaging conditions, thereby enhancing the performance of subsequent segmentation and classification stages.

3.1.3 Lung Region Segmentation Module

The system shall automatically identify and segment lung regions and potential nodules from CT images using advanced image processing and deep learning-based segmentation models. The segmentation module shall accurately delineate regions of interest, enabling focused analysis while minimizing interference from non-lung structures.

3.1.4 Feature Extraction Module

The system shall extract relevant features from the segmented lung regions to characterize potential abnormalities. Extracted features shall include texture patterns, shape descriptors, size measurements, and intensity-based characteristics. These features shall serve as inputs to the machine learning classifiers for accurate diagnosis.

3.1.5 Classification Module

The system shall classify detected lung nodules as benign or malignant using trained machine learning or deep learning models such as Convolutional Neural Networks (CNNs). The classification module shall provide reliable predictions based on learned patterns from labeled datasets.

3.1.6 Model Training and Validation Module

The system shall support training of the classification model using labeled CT image datasets. It shall perform model validation using appropriate techniques and evaluation metrics such as accuracy, sensitivity, specificity, and area under the ROC curve (AUC) to assess diagnostic performance.

3.1.7 Prediction Module

The system shall generate real-time or near real-time predictions for newly uploaded CT scan images. It shall display diagnostic outcomes along with confidence scores to assist medical professionals in clinical decision-making.

3.1.8 Visualization Module

The system shall visually present the segmentation results, detected nodules, and classification outputs through an intuitive graphical interface. This module shall allow users to view original CT images alongside processed results for better interpretability.

3.1.9 Reporting Module

The system shall generate comprehensive diagnostic reports summarizing analysis results, including patient image details, classification outcomes, and performance metrics. The reports shall be exportable in standard formats such as PDF and CSV.

3.1.10 User Management and Security Module

The system shall implement role-based access control to manage different user roles such as administrators, radiologists, and researchers. It shall ensure secure access to data and system functionalities, maintaining confidentiality and compliance with data protection requirements.

3.2 Use Case Diagram

The Use Case diagram is a diagrammatic representation that illustrates how a system functions by showing the interaction between various entities (actors) and the actions they perform. It provides a clear overview of the working of the system and the processes involved in performing specific tasks. In the case of the Lung Cancer Detection System, the following Fig. 3.2 shows the use case diagram for detecting lung cancer using CT image processing and machine learning. This figure elaborates the interaction of the user (radiologist or researcher) with the system, including actions such as registering, uploading CT images, selecting machine learning models (CNN, ResNet, Naive Bayes), processing images, analyzing features, classifying nodules, viewing results, and generating diagnostic reports.

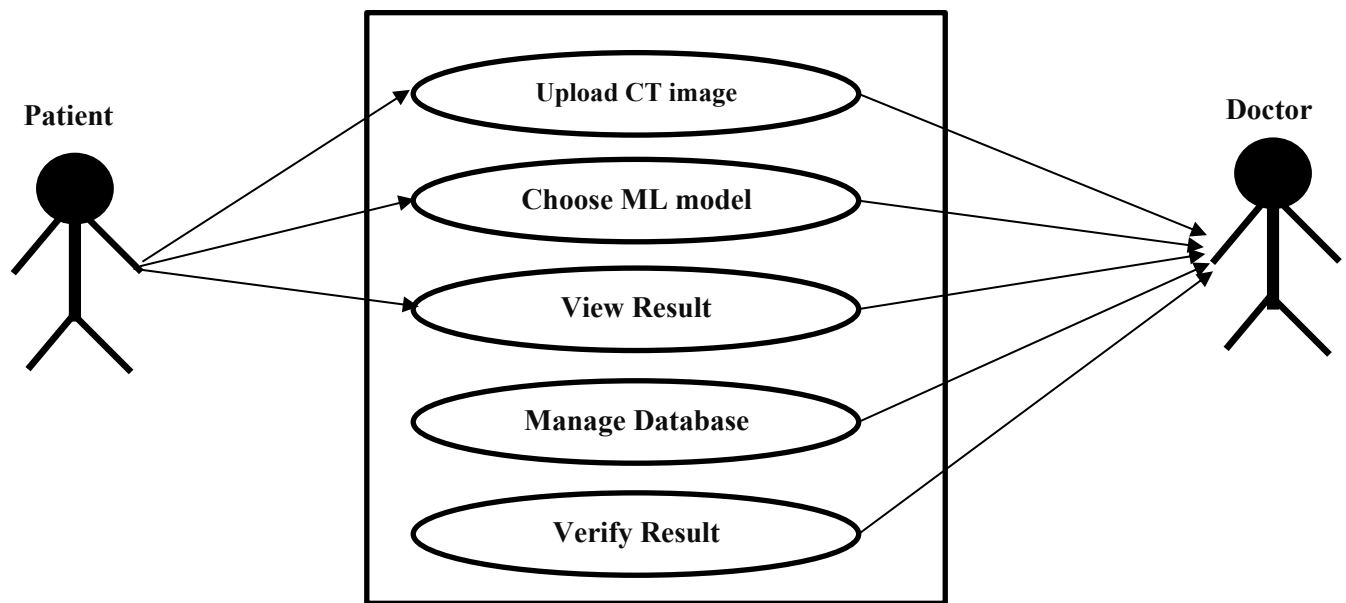


Fig 3.2 Use Case Diagram

3.3 Use Case Description

The Lung Cancer Detection System is designed to assist radiologists and researchers in detecting and classifying lung nodules using CT scan images and machine learning models. The system automates image preprocessing, segmentation, feature extraction, classification, and report generation, providing accurate diagnostic support. The following are detailed descriptions of each use case shown in the use case diagram:

3.3.1 Use Case 1: Upload CT Image

Actors: Patient / User, System

Pre-conditions: User is logged into the system. CT scan images are available in standard formats (DICOM, PNG, JPEG).

Post-conditions: CT images are successfully uploaded, validated, and stored in the system.

Description: The user uploads lung CT scan images through the system interface. The system verifies the image format and quality, performs basic preprocessing, and stores the images securely for further analysis.

3.3.2 Use Case 2: Choose Machine Learning Model

Actors: User (Radiologist/Researcher), System

Pre-conditions: CT images are successfully uploaded. Multiple ML/DL models (CNN, ResNet, SVM, etc.) are available in the system.

Post-conditions: Selected machine learning model is loaded and ready for image analysis.

Description: The user selects a preferred machine learning or deep learning model for lung cancer detection. The system initializes the selected model and prepares it to process the uploaded CT images.

3.3.3 Use Case 3: View Result

Actors: User (Patient/Doctor), System

Pre-conditions: Image processing and classification are completed.

Post-conditions: Classification results are displayed to the user.

Description: The system displays the detection results, indicating whether the lung scan is benign or malignant, along with confidence scores and visual highlights of detected nodules to assist diagnosis.

3.3.4 Use Case 4: Manage Database

Actors: Admin / Doctor, System

Pre-conditions: User has authorized access rights. Database system is operational.

Post-conditions: Patient data, CT images, and diagnostic results are securely stored or updated.

Description: The system manages storage of patient details, CT images, classification results, and metadata. Authorized users can update, retrieve, or delete records while ensuring data security and integrity.

3.3.5 Use Case 5: Verify Result

Actors: Doctor / Radiologist, System

Pre-conditions: Classification results are generated by the ML model.

Post-conditions: Results are verified and approved or updated by the doctor.

Description: The doctor reviews the AI-generated diagnosis, validates the classification results, and may add expert remarks or corrections. The verified result is then finalized for medical reference and reporting.

3.4 Non-Functional Requirements

Non-functional requirements specify the quality attributes and performance standards that the system must satisfy to be effective, reliable, and user-friendly.

3.4.1 Performance: The system shall provide real-time or near real-time predictions for uploaded CT images. Processing time per image should be minimized (<5 seconds for inference using pre-trained models on GPU).

3.4.2 Accuracy: The classification system shall achieve high accuracy, sensitivity, and specificity for nodule detection and malignancy classification, targeting >90% performance on validation datasets.

3.4.3 Scalability: The system shall be able to handle large datasets and accommodate future expansion, such as integration with hospital PACS systems or additional ML models.

3.4.4 Reliability: The system shall ensure robust and continuous operation, with minimal downtime and effective error handling in case of image corruption or system failure.

3.4.5 Security: The system shall implement user authentication, role-based access control, and secure storage of patient data, complying with data privacy standards.

3.4.6 Usability: The system shall provide a user-friendly graphical interface, supporting intuitive navigation for radiologists and researchers, with clear visualization of results.

3.4.7 Maintainability: The system shall be modular and well-documented, allowing easy updates to ML models, preprocessing algorithms, or UI components.

3.4.8 Portability: The system shall be deployable on multiple platforms, including Windows and Linux environments.

3.5 Software Requirements

- **Operating System:** The system should run on Windows 10/11 or Linux (e.g., Ubuntu 20.04 or higher). The OS must be compatible with required machine learning libraries, GPU drivers, and software dependencies.
- **Programming Language:** Python 3.x is the primary language for implementing image processing, feature extraction, machine learning models, and system logic due to its extensive support for scientific computing and AI frameworks.
- **Integrated Development Environment (IDE):** PyCharm, VS Code, or Jupyter Notebook should be used for coding, debugging, and testing. IDEs provide efficient development tools and help manage large projects effectively.
- **Machine Learning Libraries:** TensorFlow, Keras, PyTorch, and scikit-learn are required for implementing CNN, ResNet, and Naive Bayes models. These libraries also provide utilities for training, validation, and evaluation of ML models.
- **Image Processing Libraries:** OpenCV, PIL (Python Imaging Library), and NumPy are required to handle CT image preprocessing, segmentation, feature extraction, and numerical operations.
- **Visualization Libraries:** Matplotlib, Seaborn, and Plotly are used to display images, segmentation outputs, feature analysis, and classification results in an intuitive and informative manner.
- **Database:** SQLite or MySQL is required to store user data, uploaded images, analysis results, and system logs in an organized and secure manner.
- **Reporting Tools:** ReportLab or CSV export libraries are used to generate diagnostic reports in standard formats like PDF or CSV for documentation and record keeping.
- **Version Control System:** Git, along with platforms like GitHub or GitLab, is recommended to manage code changes, collaborate efficiently, and maintain project version history.

- **Optional Tools:** Docker can be used to containerize the development environment for easy deployment. NVIDIA CUDA Toolkit and cuDNN are optional but highly recommended for GPU acceleration to improve ML model training.

3.6 Hardware Requirements

- **Processor (CPU):** The system requires a high-performance CPU to handle image preprocessing, segmentation, and machine learning computations. A minimum Intel Core i5 or equivalent is acceptable, but an Intel Core i7 or higher is recommended for smoother and faster processing of high-resolution CT images.
- **Graphics Processing Unit (GPU):** A GPU is necessary for accelerating deep learning model training and inference. A minimum NVIDIA GTX 1050 can handle basic processing, while a recommended NVIDIA RTX 2060/3060 or higher ensures faster training and execution of CNN, ResNet, and other models.
- **RAM (Memory):** The system should have at least 8 GB of RAM to support image processing and model computations. However, 16 GB or more is recommended to efficiently handle large datasets and parallel processing without performance bottlenecks.
- **Storage:** Adequate storage is essential to store CT scan datasets, trained models, and system logs. A minimum 500 GB HDD is sufficient, but a 1 TB SSD is recommended for faster read/write operations and improved system performance.
- **Display:** A high-resolution display is required for clear visualization of CT images, segmentation results, and classification outputs. While a 13-inch monitor is sufficient, a 15–17 inch high-resolution display is recommended for better clarity and accuracy.
- **Input Devices:** Standard input devices such as a keyboard and mouse are necessary for interacting with the system. A graphic tablet is optional and can assist in annotating or reviewing images.
- **Network:** Internet connectivity is required for downloading datasets, libraries, and software updates. High-speed internet is recommended to reduce waiting times and improve overall workflow efficiency.

3.7 Summary

This chapter presents the key functional and non-functional requirements necessary for designing and implementing the AI-based Lung Cancer Detection System. The functional requirements include CT image acquisition, image preprocessing, lung and nodule segmentation, feature extraction, machine learning–based classification, result visualization, and report generation. The chapter also outlines important non-functional requirements such as system accuracy, reliability, scalability, security, performance, and usability, which are critical for deployment in medical environments. Use case diagrams are used to illustrate the interactions between patients, doctors, and the system, highlighting processes such as image upload, model selection, result verification, and report generation. The chapter concludes by detailing the complete hardware and software requirements, including Python, OpenCV, TensorFlow/Keras, machine learning frameworks, database systems, and computational hardware specifications, ensuring accurate, efficient, and reliable lung cancer detection in real-world clinical settings.

CHAPTER 4

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 System Overview

The proposed Lung Cancer Detection System is designed to automate the detection of lung cancer from CT images by combining advanced image processing techniques with powerful machine learning algorithms. The system operates through a structured workflow that ensures precise and early diagnosis. The first stage involves image acquisition, where CT scan images are uploaded into the system. Next, the preprocessing stage enhances image quality by performing noise reduction, contrast adjustment, and normalization to prepare the images for analysis. The Lung Cancer Detection System is an intelligent, automated framework designed to assist medical professionals in the early detection and diagnosis of lung cancer using CT scan images. The system aims to reduce diagnostic time, minimize human error, and improve accuracy by leveraging advanced image processing and machine learning techniques. It provides a reliable computer-aided diagnosis (CAD) tool that supports radiologists and clinicians in identifying malignant lung nodules at an early stage.

The system utilizes CT scan images as input and applies preprocessing, segmentation, and machine learning-based analysis to detect abnormalities in lung tissues. By employing deep learning and traditional ML models, the system automatically extracts critical features and classifies lung nodules as benign or malignant. This automated approach helps in handling large volumes of imaging data efficiently while ensuring consistent diagnostic results. A user-friendly Graphical User Interface (GUI) allows doctors, radiologists, or researchers to upload CT images, select appropriate machine learning models, and view detailed analysis results. The interface presents classification outcomes along with confidence scores and visual segmentation outputs, aiding clinical decision-making.

The key highlights of the system are listed below:

4.1.1 User Interface

The primary interface of the lung cancer detection system is a web-based application developed using modern frontend technologies such as HTML, CSS, JavaScript, and React (or Flask-based UI). The interface enables users to register/login, upload CT scan images, select machine learning models (CNN, ResNet, Naive Bayes), and visualize prediction results. The UI is designed to be intuitive and user-friendly, ensuring ease of use for medical professionals with minimal technical training.

4.1.2 Image Acquisition & Preprocessing Module

This module handles the acquisition of CT scan images uploaded by the user in standard formats such as DICOM, PNG, and JPEG. The preprocessing stage applies image enhancement techniques including noise reduction, resizing, normalization, and contrast enhancement. These steps improve image quality and ensure uniform input for further processing, thereby increasing the accuracy of lung nodule detection.

4.1.3 Segmentation Module

The segmentation module identifies and isolates lung regions and potential nodules from CT images. Advanced image processing and deep learning–based segmentation techniques are employed to accurately delineate regions of interest. This step ensures that the classification models focus only on relevant lung areas, improving diagnostic precision.

4.1.4 Feature Extraction & Classification Module

Once segmentation is completed, the system extracts important features such as texture, shape, and intensity from the lung nodules. These features are then passed to selected machine learning models, including CNN, ResNet, and Naive Bayes, to classify nodules as benign or malignant. The system provides confidence scores along with classification results to support reliable medical interpretation.

4.1.5 Result Visualization & Report Generation

After classification, the system displays results in both graphical and textual formats. Segmented images, prediction labels, and confidence levels are shown on the interface. Additionally, the system can generate diagnostic reports in PDF or CSV format, enabling doctors to store, share, or include them in patient medical records.

4.1.6 Database Management & External Integration

All uploaded images, prediction results, user details, and processing metadata are securely stored in a database such as MySQL or SQLite. The system maintains a complete diagnostic history for each user. Fig. 4.1 illustrates the overall workflow of the lung cancer detection system, starting from image upload and preprocessing to model selection, classification, result visualization, and report storage. The system can be extended to integrate with hospital databases or electronic health record (EHR) systems for real-time clinical deployment.

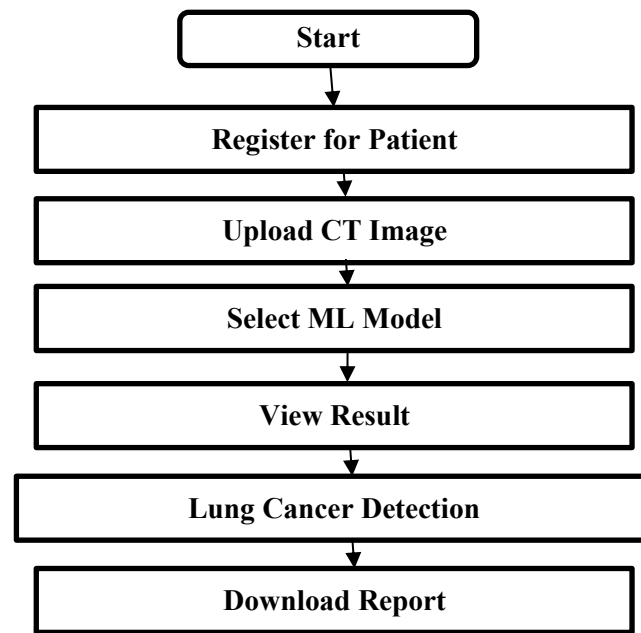


Fig 4.1 General working of our System

4.2 System Architecture

The system architecture is designed to efficiently manage CT image processing and deliver accurate diagnostic predictions. At the front end, a user-friendly interface developed using React allows users, such as radiologists or researchers, to upload CT images, select machine learning models, and view results in an intuitive format.

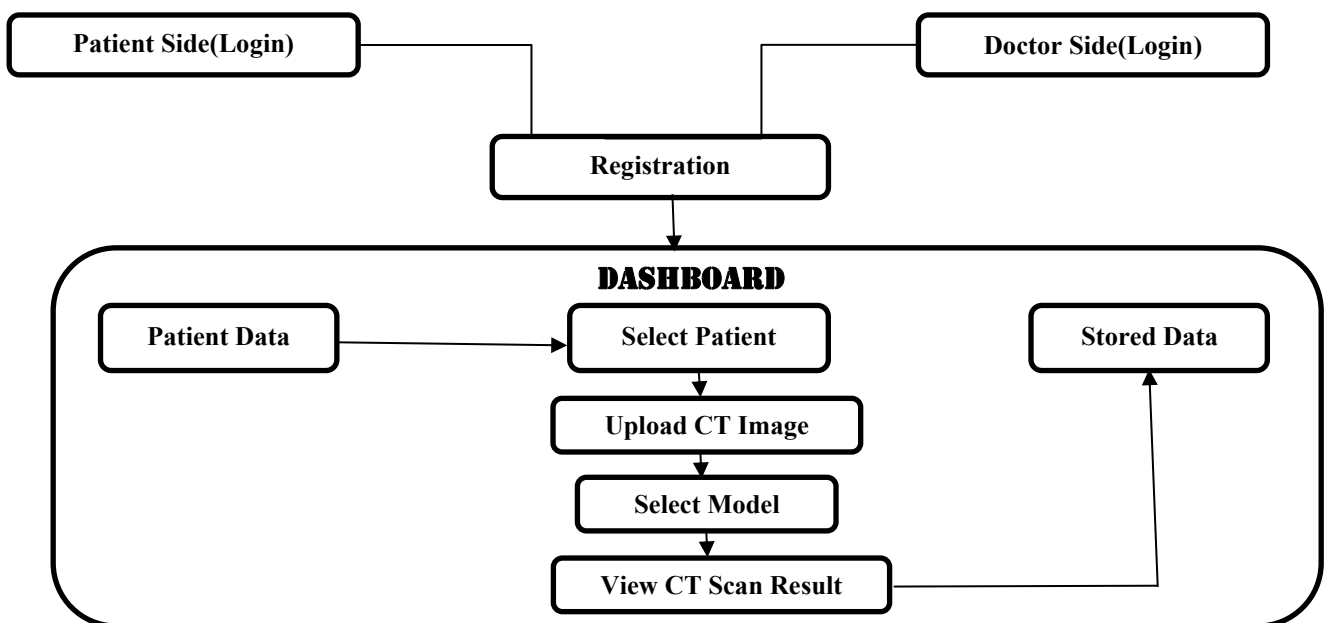


Fig 4.2 System Architecture

The backend server, built with Flask, handles core operations including image uploads, preprocessing, and communication with the machine learning model. The machine learning component consists of a convolutional neural network (CNN) implemented in TensorFlow or PyTorch, trained to detect and classify lung nodules based on extracted features. A SQL database stores user information, image metadata, and system results, ensuring organized and secure data management. Additionally, a security layer is implemented, incorporating authentication and authorization mechanisms to protect sensitive medical data and ensure that only authorized users can access the system. Together, these components create an integrated, robust architecture that delivers efficient, accurate, and secure lung cancer detection from CT images.

4.3 Component Design

The Lung Cancer Detection System is composed of several interrelated components, each responsible for a specific functionality to ensure accurate, efficient, and secure detection of lung cancer from CT images. The system architecture is modular, allowing easy maintenance, scalability, and updates of individual components without affecting the overall workflow.

4.3.1 Frontend Interface: The frontend is a user-friendly interface developed using React. It enables users, such as radiologists or researchers, to register, log in, upload CT images, select machine learning models (CNN, ResNet, or Naive Bayes), and view classification results. The interface provides clear visualizations of segmented images, feature extraction outputs, and final diagnostic results, ensuring intuitive interaction.

4.3.2 Backend Server: The backend, built using Flask, handles all server-side operations, including receiving uploaded images, performing preprocessing and segmentation, interacting with the machine learning models for classification, and sending results to the frontend. It also manages requests from multiple users concurrently, ensuring smooth and reliable operation.

4.3.3 Image Preprocessing and Segmentation Module: This component is responsible for enhancing image quality by performing noise reduction, normalization, and contrast adjustments. Segmentation algorithms accurately delineate lung regions and nodules, isolating areas of interest for further analysis. This module ensures that only relevant regions are analyzed by the ML models, improving classification accuracy.

4.3.4 Machine Learning Module: The ML module implements deep learning models such as CNN and ResNet, as well as classical models like Naive Bayes. This component is responsible for feature extraction, training, validation, and classification. It takes segmented images as input, automatically learns hierarchical features, and predicts whether nodules are benign or malignant.

4.3.5 Database Component: A PostgreSQL database stores user information, CT image metadata, model outputs, and diagnostic reports. This component ensures data integrity, secure storage, and easy retrieval for reporting, auditing, and further analysis.

4.3.6 Security Layer: The security component implements authentication, authorization, and data encryption mechanisms. It ensures that only authorized users can access sensitive medical data and maintains compliance with data privacy standards.

4.3.7 Reporting Module: This component generates detailed diagnostic reports in standard formats (PDF or CSV), including images, segmentation results, extracted features, classification outcomes, and probability scores. These reports aid medical professionals in clinical decision-making and record-keeping.

4.4 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) represents the flow of information within the Lung Cancer Detection System, illustrating how data moves between users, system components, and databases. It provides a clear visual understanding of how CT scan images are processed, analyzed, and used to generate diagnostic results.

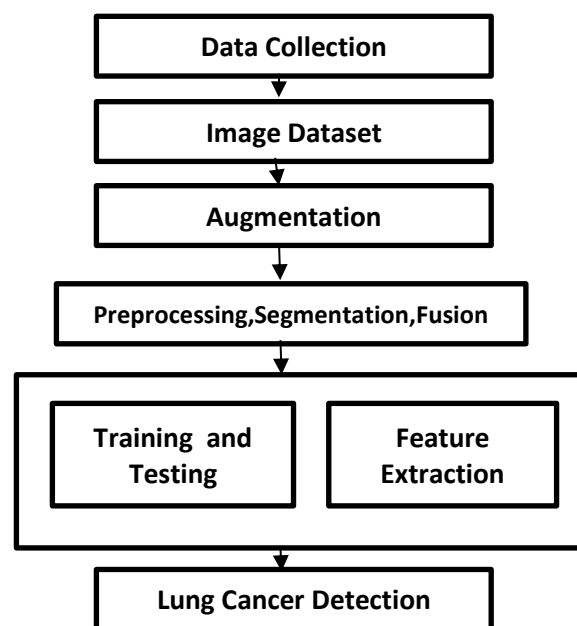


Fig 4.4 Data Flow Diagram

4.4.1 Data Collection : The first step of the system involves collecting lung CT scan images from reliable and diverse sources such as hospitals, online medical repositories like Kaggle, and publicly available datasets such as LIDC-IDRI. These datasets contain CT images from different patients, scanners, and clinical conditions. The inclusion of both tumor and non-tumor cases ensures data diversity and reduces bias. Collecting data from multiple sources helps the model learn real-world variations in lung anatomy and tumor appearance, which is essential for building a robust and accurate detection.

4.4.2 Image Dataset : After collection, the CT scan images are organized into a structured dataset. Each image is carefully labeled as either “tumor” or “non-tumor” based on medical annotations or dataset documentation. The dataset is usually divided into training, validation, and testing subsets to ensure proper evaluation of the model. Maintaining a well-organized dataset allows smooth data handling, reduces errors during training, and ensures reproducibility of results.

4.4.3 Augmentation : Data augmentation techniques are applied to increase the effective size of the dataset without collecting new images. Operations such as rotation, horizontal and vertical flipping, scaling, zooming, and brightness adjustment are performed on the existing CT images. These transformations simulate real-world variations in scan orientation and imaging conditions. Augmentation helps reduce overfitting, allowing the deep learning model to generalize better and perform accurately on unseen CT scans.

4.4.4 Pre-processing, Segmentation, and Fusion: Pre-processing improves the quality of CT images before they are fed into the model. This includes noise reduction to remove scanning artifacts, resizing images to a uniform dimension, normalization to scale pixel values, and contrast enhancement to highlight lung tissues. These steps ensure consistent input data and improve learning efficiency.

4.4.5 Segmentation: Segmentation focuses on identifying and isolating the region of interest (ROI), such as lung areas or suspected tumor regions. By separating the tumor-relevant portions from the background, the model concentrates on medically important areas, improving detection accuracy and reducing computational complexity.

4.4.6 Fusion: Fusion combines multiple pre-processed or segmented outputs, or integrates information from different imaging views or modalities. This process enhances feature representation by combining complementary information, leading to better discrimination between tumor and non-tumor regions.

4.4.7 Deep Learning Model: A deep learning model, typically a Convolutional Neural Network (CNN), is used to automatically extract meaningful features from the segmented CT images. These features include edges, textures, shapes, and complex tumor patterns that are difficult to define manually. Training and Testing: During training, the model learns from labeled images by adjusting its parameters to minimize classification errors. The trained model is then tested on unseen images to evaluate its performance using metrics such as accuracy, sensitivity, and specificity.

4.4.8 Classification: Once features are extracted, the classification layer of the model assigns each CT image to one of two categories: Tumor or Non-Tumor. This decision is made based on learned patterns and probability scores generated by the deep learning model. Accurate classification is crucial for assisting doctors in identifying abnormal lung conditions at an early stage.

4.4.9 Output : The final output of the system is presented in a user-friendly format, such as a diagnostic label or visual indication of tumor presence. This output can be accessed by doctors, radiologists, or patients for clinical evaluation. The system aims to support medical professionals in early diagnosis, reduce human error, and improve treatment planning for lung cancer patients.

4.5 Sequence Diagram

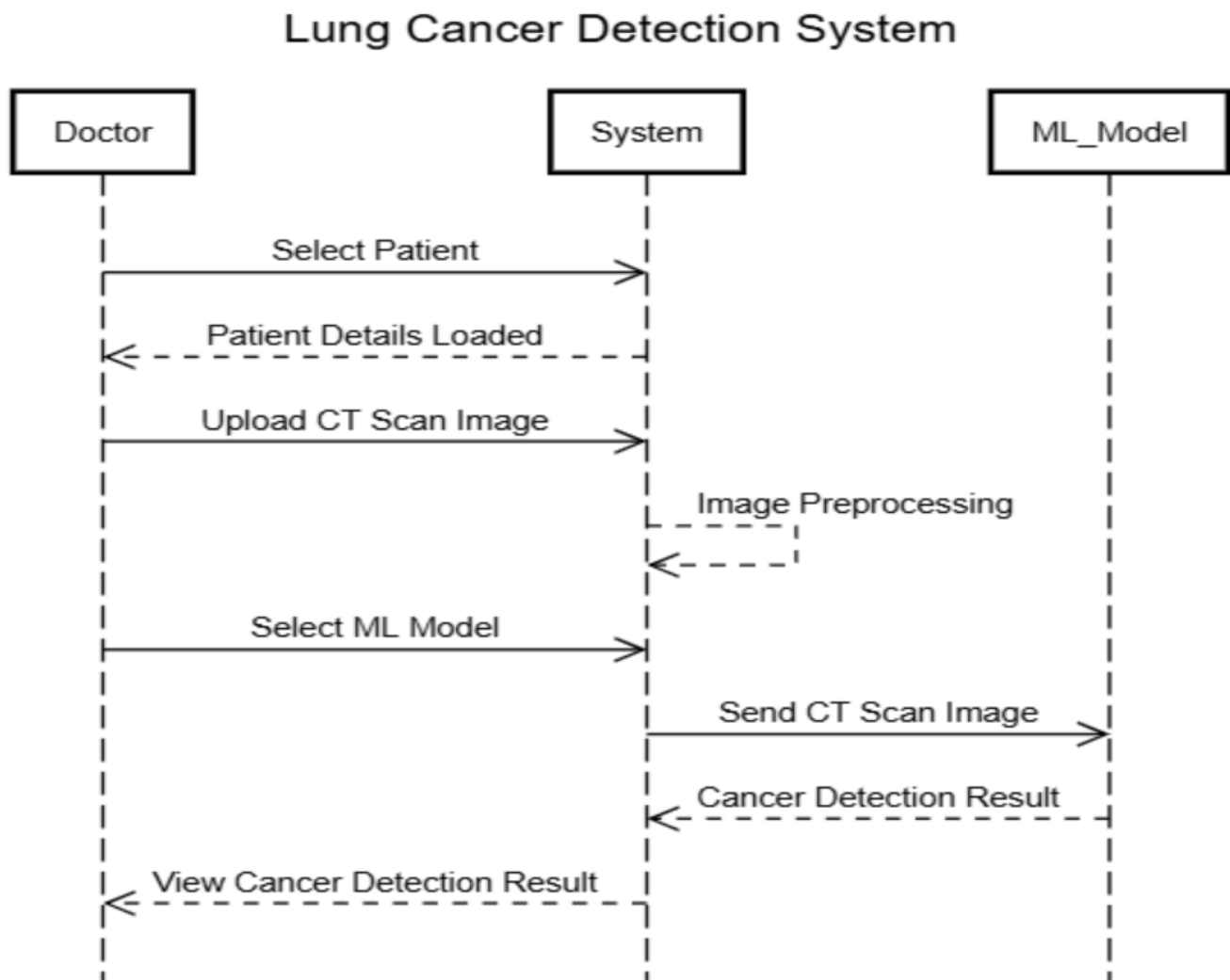


Fig 4.5 : Sequence Diagram

The Sequence Diagram illustrates the interactions between system components during the lung cancer detection workflow, providing a clear understanding of the step-by-step process. The sequence begins with the user uploading a CT scan image via the frontend interface. Once the image is uploaded, the backend server receives it and initiates the preprocessing stage to prepare the image for analysis. Preprocessing involves noise reduction, normalization, and contrast enhancement, ensuring that the input image is optimized for accurate feature extraction and classification.

After preprocessing, the image is sent to the machine learning model, such as a CNN, ResNet, or Naive Bayes, for classification of lung nodules into benign or malignant categories. The classification result is then returned to the backend, which forwards the information to the frontend for display to the user. Simultaneously, the result and associated metadata are stored in the database for future reference, record-keeping, and further analysis. This sequence ensures a smooth and efficient workflow from image upload to diagnostic output.

4.6 Data Design and Description

4.6.1 Patient (Input Source) : The patient acts as the primary data source by undergoing a CT scan of the lungs. These CT images serve as the fundamental input for the lung cancer detection system. The scans capture detailed cross-sectional views of lung tissues, enabling identification of abnormalities such as nodules or tumors. The system relies entirely on medical imaging data and does not require any manual interaction from the patient, ensuring a non-invasive and reliable diagnostic approach.

4.6.2 CT Image Acquisition : This module is responsible for acquiring lung CT images from medical imaging devices, hospital databases, or publicly available datasets such as Kaggle and LIDC-IDRI. The images are captured in high resolution to preserve anatomical details necessary for accurate diagnosis. Proper image acquisition ensures consistency in scan quality, slice thickness, and orientation, which is essential for effective processing and analysis.

4.6.3 Image Pre-processing : In this stage, the acquired CT images undergo pre-processing to improve image quality and standardization. Operations such as noise removal, resizing, normalization, and contrast enhancement are applied. Pre-processing reduces artifacts and variations caused by different scanning conditions, ensuring that all images are suitable for feature extraction and machine learning analysis.

4.6.4 Lung Segmentation: The lung segmentation module isolates the lung region from the background structures such as bones and surrounding tissues. Image processing techniques or deep learning-based segmentation models are used to extract regions of interest where tumors are likely to appear. This step focuses the analysis on relevant areas, reduces computational complexity, and improves detection accuracy.

4.6.5 Feature Extraction : Once segmentation is completed, significant features such as shape, texture, size, and intensity variations are extracted from the lung regions. These features help differentiate between normal lung tissue and cancerous nodules. In deep learning-based approaches, convolutional neural networks (CNNs) automatically learn hierarchical features directly from the segmented CT images.

4.6.6 Classification Using Machine Learning: The extracted features are passed to a machine learning or deep learning classifier that categorizes each CT image as Tumor or Non-Tumor. The model is trained using labeled datasets and tested on unseen images to evaluate performance. This classification stage forms the core decision-making component of the system.

4.6.7 Output Generation: The classification result is generated as the system output, indicating whether lung cancer is detected or not. The output may include probability scores, labels, or visual markers highlighting suspicious regions. This information assists doctors and radiologists in clinical decision-making and early diagnosis.

4.6.8 Doctor/Radiologist Side (User Interface): The doctor or radiologist acts as the system user and accesses the diagnostic results through a graphical user interface. The interface allows medical professionals to view CT images, classification results, and highlighted tumor regions. This ensures easy interpretation and supports faster medical evaluation.

4.6.9 Case Review and Validation : Medical professionals can review detected cases and validate the system's predictions. This step allows confirmation of results, reduces false positives, and builds trust in the system. Expert feedback can also be used to improve model performance over time.

4.6.10 Report Generation: Based on the classification results, the system generates a diagnostic report summarizing findings such as tumor presence, location, and confidence level. These reports can be stored digitally and shared for further medical consultation or treatment planning.

4.7 Entity Relationship Diagram(ERD)

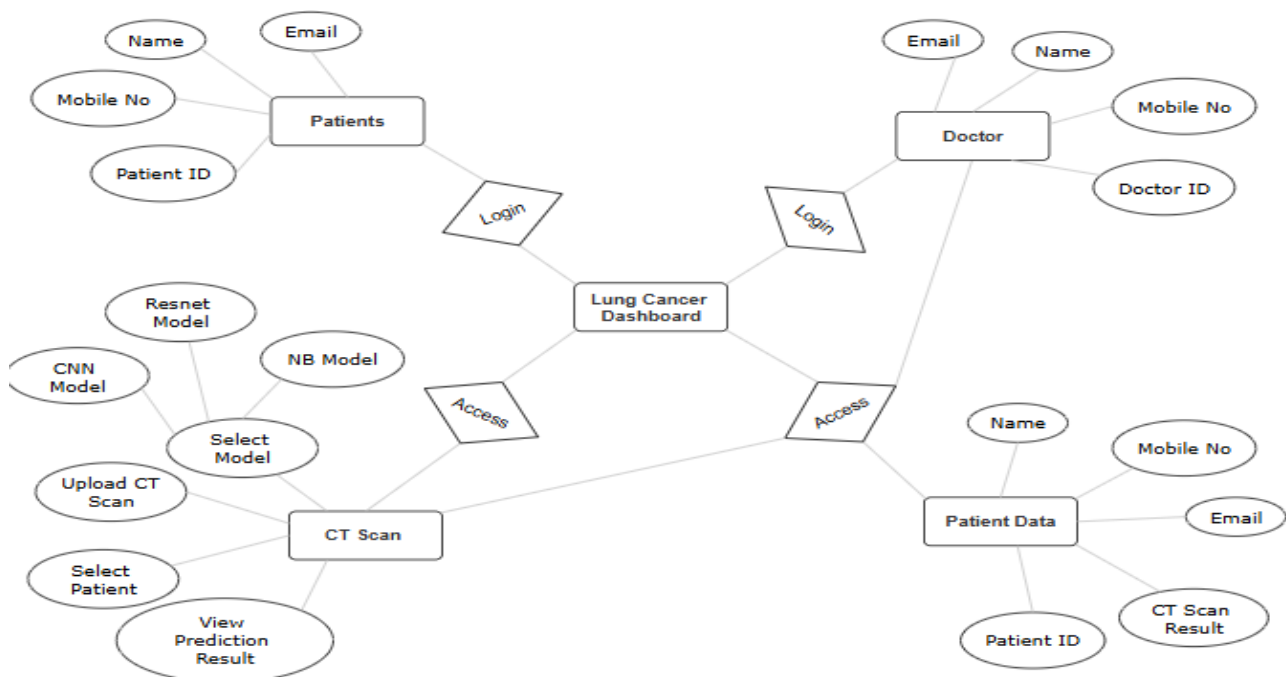


Fig 4.7 : ERD for Lung Cancer Detection

The Entity Relationship Diagram (ERD) represents the logical data structure of the Lung Cancer Detection System using CT Image Processing and Machine Learning. It illustrates the primary entities involved in the system, the key attributes associated with each entity, and the relationships that connect them. The ERD provides a clear understanding of how medical imaging data, patient information, and diagnostic results are organized and managed within the system database.

4.7.1 ERD Description

PATIENTS Entity stores details related to each patient, including patient ID, name, age, gender, and medical history. Each patient can undergo multiple CT scans over time for diagnostic or follow-up purposes. This entity serves as the primary reference for linking medical imaging data with individual patient records.

CT_SCANS Entity contains information related to lung CT images such as scan ID, patient ID (FK), scan date, image path, scan type, and resolution. Each CT scan is associated with a single patient but a patient may have multiple scan records. These scans form the primary input data for image processing and machine learning analysis.

DIAGNOSIS_RESULTS Entity records the outcome of lung cancer detection for each CT scan. It includes result ID, scan ID (FK), detected class (Tumor / Non-Tumor), confidence score, and diagnosis timestamp. This entity captures the final classification produced by the machine learning model and maintains the diagnostic history for each patient.

CHAPTER 5

CHAPTER 5

SYSTEM IMPLEMENTATIONS

5.1 Language Description

Python is a high-level, interpreted, and dynamically typed programming language widely used in the fields of machine learning, deep learning, and medical image processing. It provides a simple and readable syntax, which reduces development time and makes implementation easier compared to traditional programming languages. Python is extensively supported by powerful libraries such as NumPy, Pandas, OpenCV, Scikit-learn, TensorFlow, and Keras, making it an ideal choice for building AI-based medical diagnostic systems. In this project, Python is used for implementing the Lung Cancer Detection System using CT Image Processing and Machine Learning due to its seamless integration with deep learning frameworks and image processing tools. Libraries such as OpenCV are used for CT image preprocessing and segmentation, while TensorFlow/Keras are employed for designing, training, and testing convolutional neural network (CNN) models. Python also supports the use of DICOM processing libraries such as *pydicom*, enabling direct handling of medical CT scan formats.

Python supports both object-oriented and functional programming paradigms, offering flexibility in designing modular and scalable systems. Its ability to efficiently handle large datasets and perform complex numerical computations makes it suitable for processing high-resolution CT images. The availability of GPU acceleration through CUDA-enabled TensorFlow further improves training speed and model performance. Additionally, Python allows easy integration with data visualization libraries such as Matplotlib and Seaborn, which are used to analyze training performance and evaluate metrics like accuracy, sensitivity, and specificity. The trained machine learning models can also be deployed using Python-based web frameworks or integrated into clinical decision support systems. Overall, Python provides a reliable, efficient, and scalable development environment for implementing accurate and high-performance lung cancer detection solutions.

5.2 Software Description

There are many free and commercial IDEs and tools available for developing machine learning and image processing applications. Jupyter Notebook and PyCharm are widely used development environments for Python-based AI systems. Jupyter Notebook provides an interactive environment for experimentation, data visualization, and model training, while PyCharm offers advanced debugging, code completion, refactoring support, and seamless integration with version control systems. These IDEs make it easier to develop, test, and optimize machine learning models efficiently.

5.2.1 OpenCV: OpenCV (Open Source Computer Vision Library) is an open-source library extensively used for image processing and computer vision applications. In this project, OpenCV is utilized for processing lung CT images, including tasks such as noise reduction, image enhancement, segmentation, and feature extraction. Its optimized algorithms and cross-platform support enable efficient handling of high-resolution medical images, making it suitable for lung cancer detection applications.

5.2.2 TensorFlow / Keras: TensorFlow and Keras are powerful deep learning frameworks used for designing, training, and deploying convolutional neural networks (CNNs). In this project, these frameworks are employed to build and train the lung cancer classification model. They provide flexible APIs, GPU acceleration, and pre-trained model support, which help achieve high accuracy in tumor detection and classification.

5.2.3 Scikit-learn: Scikit-learn is a Python-based machine learning library that provides efficient tools for data preprocessing, feature extraction, classification, and performance evaluation. It is used for implementing traditional machine learning algorithms, model validation, and evaluation metrics such as accuracy, precision, recall, and F1-score.

5.2.4 Pydicom: Pydicom is a specialized Python library used for handling DICOM (Digital Imaging and Communications in Medicine) files. In this project, it enables the reading, processing, and conversion of medical CT scan data into formats suitable for analysis and machine learning, ensuring compatibility with clinical imaging standards.

5.2.5 Python Standard Library: The Python Standard Library offers a wide range of built-in modules for file handling, data processing, mathematical computations, and system-level operations. It supports efficient management of datasets, model files, and result storage, contributing to the overall robustness and scalability of the lung cancer detection system.

5.3 Pseudo Code

Pseudocode is a high-level representation of the algorithm used for lung cancer detection using CT scan images and machine learning models.

Main Logic (CT Image Processing and Cancer Detection)

BEGIN

Import required libraries

(NumPy, Pandas, OpenCV, TensorFlow/Keras, Scikit-learn, Matplotlib)

Load CT scan image dataset from dataset folder

FOR each CT image in dataset DO

 Resize image to fixed size (224×224)

 Convert image to grayscale or RGB

Normalize pixel values to range [0,1]

END FOR

Split dataset into:

 Training set (80%)

 Testing set (20%)

Display model selection menu

 Option 1: CNN Model

 Option 2: Naive Bayes Model

 Option 3: ResNet Model

 Option 4: Run All Models

IF Option 1 selected (CNN Model) THEN

 Initialize CNN architecture

 Add convolution, activation, and pooling layers

 Add fully connected dense layers

 Compile model using Adam optimizer

 Train CNN using training data

 Test CNN using testing data

 Display CNN accuracy and results

ENDIF

IF Option 2 selected (Naive Bayes Model) THEN

 Extract image features (pixel / HOG features)

 Flatten features into 1D vectors

 Initialize Naive Bayes classifier

 Train classifier using training data

 Predict results on testing data

 Display accuracy, confusion matrix, and report

ENDIF

IF Option 3 selected (ResNet Model) THEN

 Load pre-trained ResNet model

- Replace final layers with custom classifier
- Freeze initial layers for transfer learning
- Compile and train ResNet model
- Test model on testing data

Display ResNet accuracy and confusion matrix
ENDIF

IF Option 4 selected (Run All Models) THEN

- Train and test CNN model
- Train and test Naive Bayes model
- Train and test ResNet model
- Compare accuracy of all models
- Select best-performing model
- Save best model

ENDIF

Load new CT scan image uploaded by doctor

Preprocess image (resize and normalize)

Predict result using selected/best model

IF prediction indicates tumor THEN

- Display "Cancer Detected"

ELSE

- Display "No Cancer Detected"

ENDIF

Generate diagnostic report

- (Patient details, scan result, model used, date)

END

CHAPTER 6

CHAPTER 6

TESTING AND VALIDATION

The testing phase plays a crucial role in ensuring that the AI-Based Lung Cancer Detection System using CT Scan Images functions accurately, reliably, and efficiently. Various testing levels were employed throughout the Software Development Lifecycle (SDLC) to validate individual components, integrated modules, and the overall system performance. The primary objective of testing is to ensure correct diagnosis, system stability, and reliable performance in assisting medical professionals during lung cancer detection.

The four primary levels of software testing conducted are:

Unit Testing – Verifies the functionality of individual modules such as CT image preprocessing, feature extraction, model inference, and database operations.

Integration Testing – Ensures seamless interaction between modules including image upload, model selection, result generation, and data storage.

System Testing – Evaluates the complete lung cancer detection system under simulated real-world clinical scenarios.

Acceptance Testing – Confirms that the system meets end-user requirements and aligns with expectations of doctors and healthcare staff.

Testing was carried out using both local systems and cloud-based environments to validate performance across different devices, operating systems, and dataset variations.

6.1 Testing Strategies

A testing strategy defines the overall approach and techniques adopted to verify system correctness, performance, and robustness. The following testing strategies were applied:

6.1.1 Top-Down Testing: Higher-level modules such as doctor dashboard, patient selection, and result visualization were tested first, followed by lower-level modules including CT image preprocessing and model execution.

6.1.2 Bottom-Up Testing: Core components such as image normalization, feature extraction, and CNN-based classification models were tested independently before integrating them into the complete diagnostic workflow.

6.1.3 Thread Testing: Used to validate parallel operations such as CT image processing, real-time model inference, and database synchronization occurring simultaneously.

6.1.4 Stress Testing: Conducted to evaluate system behavior under heavy loads, including multiple CT image uploads, repeated model execution, and large patient datasets, ensuring system stability.

6.2 Module Testing

Each major module in the AI-based system was individually tested to confirm its functionality and performance. Testing was done using Android Studio Logcat, on-screen outputs, and Firebase real-time updates to verify expected results. Following table 6.2 info about module test

Table 6.2 : Module Testing

Test Case	Description	Expected Output	Actual Output	Status
User Authentication	Login using patient/doctor credentials	User authenticated successfully	Login successful	Pass
Registration Module	Register new patient and doctor details	User data stored in database	Data stored correctly	Pass
Patient Data Entry	Enter and save patient details	Patient data saved accurately	Data saved successfully	Pass
Select Patient	Select patient from database	Patient profile loaded	Correct patient selected	Pass
CT Image Upload	Upload lung CT scan image	Image uploaded successfully	Image uploaded correctly	Pass
Image Preprocessing	Apply resizing and normalization	Preprocessed image generated	Processed correctly	Pass
Model Selection	Select ML/DL model for prediction	Model loaded successfully	Model selected correctly	Pass
Lung Cancer Detection	Analyze CT scan using selected model	Cancer detection result generated	Result generated accurately	Pass
Result Visualization	Display prediction result	Clear result shown (Cancer/No Cancer)	Displayed correctly	Pass
Probability Score Display	Show confidence level of prediction	Probability score displayed	Score shown correctly	Pass
Data Storage	Store CT images and results	Data stored in database	Stored successfully	Pass
Report Generation	Generate diagnostic report	Report created successfully	Report generated	Pass
Doctor Dashboard	View patient history and reports	Dashboard loads correctly	Works as expected	Pass

System Performance	Process multiple CT images	Smooth performance	Stable performance	Pass
Database Synchronization	Sync patient and result data	Real-time synchronization	Works correctly	Pass
Error Handling	Upload invalid image	Error message displayed	Error handled properly	Pass
Compatibility Testing	Run system on different devices	Works on all tested systems	Compatible	Pass
Security Check	Restrict unauthorized access	Access denied	Security enforced	Pass

Model testing is a crucial phase in the lung cancer detection system to evaluate the performance, accuracy, and reliability of the trained machine learning models. After completing the training phase, the models were tested using unseen CT scan images to assess their ability to correctly classify lung cancer cases. The objective of model testing is to ensure that the system generalizes well to new data and produces consistent and clinically meaningful results.

6.2.1 CNN Model Testing

The Convolutional Neural Network (CNN) model was tested using the testing dataset, which comprised 20% of the total CT scan images. The model's performance was evaluated based on accuracy, loss, precision, recall, and confusion matrix. During testing, the CNN demonstrated strong feature extraction capability and achieved high classification accuracy. The loss value remained low, indicating good generalization and minimal overfitting. The CNN model successfully identified cancerous and non-cancerous lung images with reliable accuracy.

6.2.2 Naive Bayes Model Testing

The Naive Bayes model was tested by using feature vectors extracted from preprocessed CT scan images. Testing focused on evaluating prediction accuracy and class-wise performance. Although the Naive Bayes classifier showed faster execution and lower computational cost, its accuracy was comparatively lower than deep learning models. However, it performed reasonably well for binary classification and served as a baseline model for comparison.

6.2.3 ResNet Model Testing

The ResNet model was tested using transfer learning with a pre-trained ResNet architecture. The testing phase evaluated accuracy, confusion matrix, and loss metrics. The ResNet model achieved superior performance due to its deep architecture and residual connections, which helped in capturing complex lung tissue patterns. The results showed higher accuracy and better robustness compared to CNN and Naive Bayes models, making it the best-performing model among the three.

6.2.4 Comparative Model Evaluation

All three models—CNN, Naive Bayes, and ResNet—were tested on the same testing dataset to ensure a fair comparison. Performance metrics such as accuracy, precision, recall, F1-score, and inference time were analyzed. The comparison revealed that the ResNet model achieved the highest accuracy, followed by the CNN model, while the Naive Bayes model showed the lowest accuracy but fastest execution time. These results highlight the trade-off between accuracy and computational efficiency.

6.3 Integration Testing

Integration testing verifies the interaction between individual modules of the Lung Cancer Detection System after unit testing is completed. The primary objective of integration testing is to ensure that data flows correctly between modules and that combined components function together as expected. Since the system involves multiple processing stages—image acquisition, preprocessing, feature extraction, model inference, and result visualization—proper integration is critical to maintain diagnostic accuracy and system reliability.

The first level of integration testing focused on the interaction between the Image Loading Module and the Image Preprocessing Module. CT scan images in different formats were loaded and passed to the preprocessing stage. The test verified that images retained their integrity during resizing, normalization, and enhancement operations. The system successfully handled valid images and gracefully rejected unsupported or corrupted files.

The second integration test examined the connection between the Preprocessing Module and the Feature Extraction / Model Training Module. Preprocessed images were converted into tensors and fed into CNN, Naive Bayes (feature-based), and ResNet models. This test ensured correct input dimensions, pixel value scaling, and batch processing. The models accepted the data without dimensional mismatches or runtime errors, confirming smooth integration.

Further testing was carried out between the Model Training Module and the Prediction Module. Trained models were loaded from saved checkpoints and used to classify unseen CT scan images. The predictions generated were consistent with expected outputs, indicating that trained weights and model parameters were correctly passed to the inference stage.

6.4 System Testing

System testing evaluates the complete Lung Cancer Detection System as a whole under conditions that closely resemble real-world usage. This testing phase ensures that the system meets functional, performance, and reliability requirements and is suitable for deployment in clinical or diagnostic environments. During system testing, the entire workflow—from CT scan image upload to final diagnosis report generation—was tested end-to-end. The system was evaluated using real and benchmark datasets containing both cancerous and non-cancerous lung CT scans. The tests verified that the system consistently produced accurate predictions without crashes or unexpected behavior.

Performance testing was conducted to assess model inference time and system responsiveness. The system processed CT scan images efficiently, producing predictions within acceptable time limits. This is crucial for clinical decision support, where timely diagnosis plays a significant role in patient outcomes. The system was also tested for stability by running continuous image classification tasks over extended periods. No memory leaks, model failures, or degradation in performance were observed, demonstrating robustness during long-term operation.

Accuracy and reliability testing involved evaluating prediction consistency across different machine learning models (CNN, Naive Bayes, and ResNet). The system maintained stable accuracy levels, with deep learning models achieving higher performance on complex CT images. Confusion matrices and classification reports confirmed reliable detection of lung cancer cases. Usability testing ensured that the user interface was intuitive and suitable for medical professionals. Doctors could easily upload CT scans, view diagnostic results, analyze confidence scores, and download reports. Error messages were clear, and guidance was provided for invalid inputs.

6.5 Acceptance Testing

Acceptance testing is performed to verify that the Lung Cancer Detection System satisfies the requirements and expectations of end users, particularly medical professionals such as doctors and radiologists. This testing ensures that the system delivers accurate results, is easy to use, and functions reliably in a real clinical environment. Acceptance testing acts as the final validation phase before system deployment.

The primary user requirements verified during acceptance testing include the ability to upload CT scan images, perform automatic lung cancer detection, view classification results clearly, and generate diagnostic reports. The system was tested to ensure that it correctly identifies cancerous and non-cancerous CT scans and presents results with appropriate confidence scores. Medical workflow simulation was carried out by testing the system with sample CT scans and patient details. Doctors were able to upload images, select or use the best-performing trained model, and receive prediction results without manual intervention. The generated reports included essential information such as patient name, age, scan result, model used, and date, meeting clinical documentation needs.

Usability testing confirmed that the user interface is intuitive and easy to navigate. Even users with minimal technical knowledge were able to operate the system without difficulty. Clear messages were displayed for successful predictions as well as for invalid or unsupported image formats, improving user experience and reducing operational errors. Reliability testing during acceptance testing ensured that the system produces consistent results for repeated inputs and performs stably during continuous usage. The system worked effectively on standard hardware configurations, making it suitable for deployment in hospitals and diagnostic centers without requiring high-end infrastructure.

CHAPTER 7

CHAPTER 7

RESULTS AND SCREENSHOTS

The Lung Cancer Detection System was successfully developed and tested to evaluate its accuracy, reliability, and diagnostic effectiveness. The system demonstrated consistent performance in analyzing lung CT scan images and generating accurate cancer detection results using machine learning and deep learning techniques. Across multiple test cases, the model efficiently processed CT images and produced reliable predictions indicating the presence or absence of lung cancer. The system showed stable performance with minimal processing delay, ensuring timely diagnosis support for medical professionals. The preprocessing and model inference pipeline operated smoothly, even with variations in CT image quality and resolution. The user interface proved to be intuitive and easy to navigate, allowing doctors to upload images, select models, and view results seamlessly. Overall, the system effectively supports early lung cancer detection and assists clinicians in making informed diagnostic decisions.

7.1 User Dashboard Interface

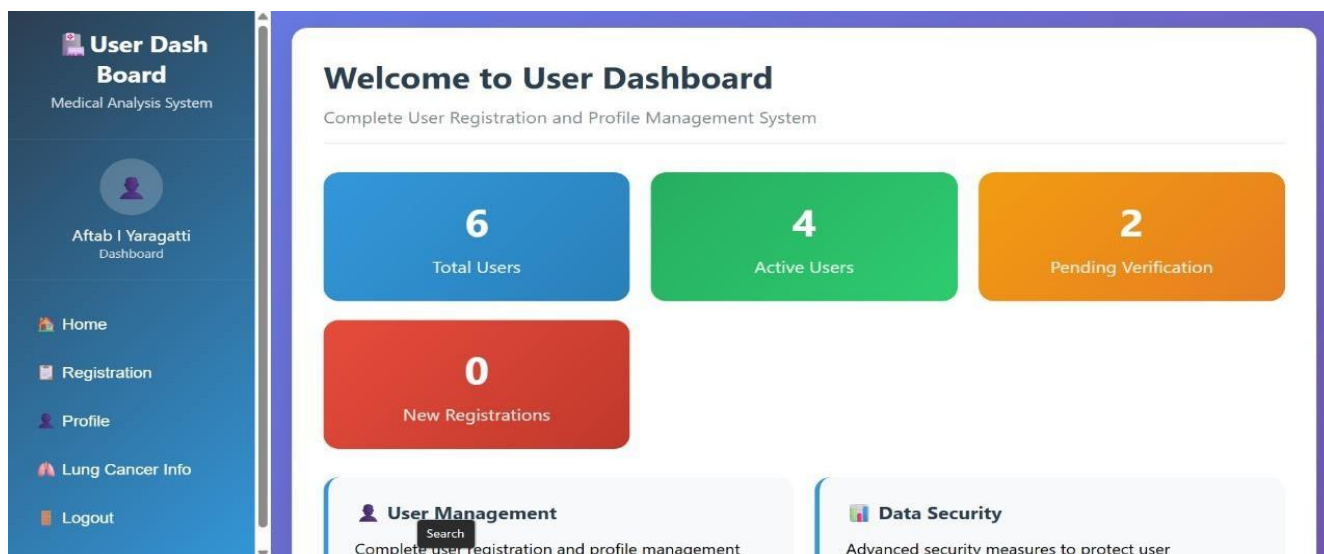


Fig 7.1 : User Dashboard

Fig 7.1 shows the User Dashboard of the Lung Cancer Detection System. This screen is displayed after successful user login. It provides an overview of total users, active users, pending verifications, and new registrations. The dashboard acts as the central interface for managing user profiles and accessing lung cancer-related information.

7.2 Admin Dashboard Interface

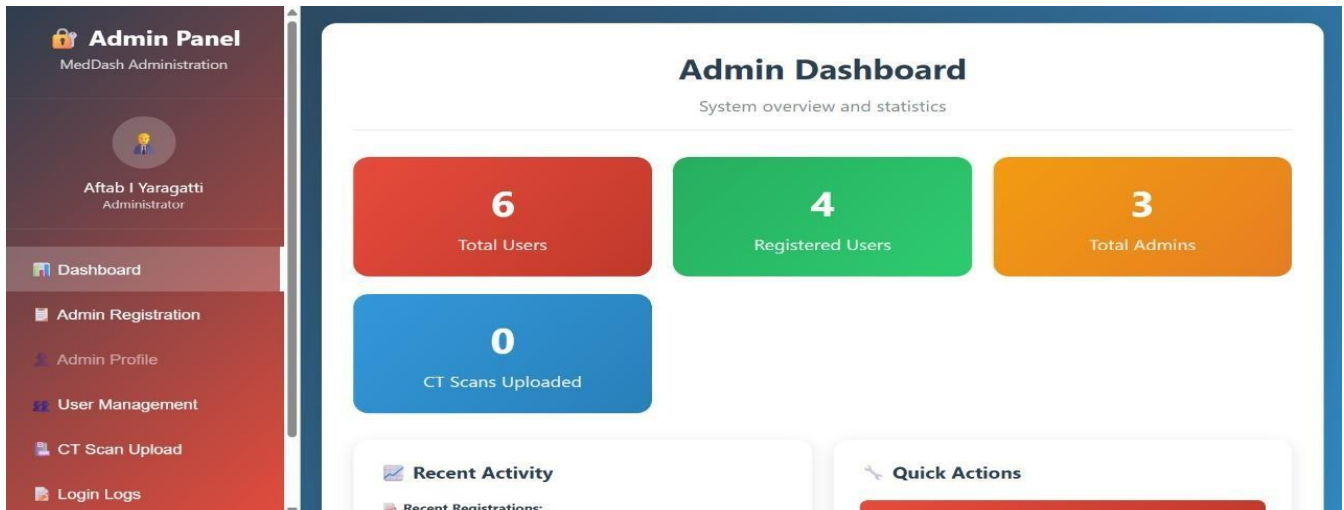


Fig 7.2 : Admin Dashboard

Fig 7.2 illustrates the Admin Dashboard, which provides a complete overview of system statistics. It displays total users, registered users, total administrators, and the number of CT scans uploaded. This interface allows the administrator to monitor system activity, manage users, and control CT scan uploads securely.

7.3 Lung Cancer Dashboard Interface

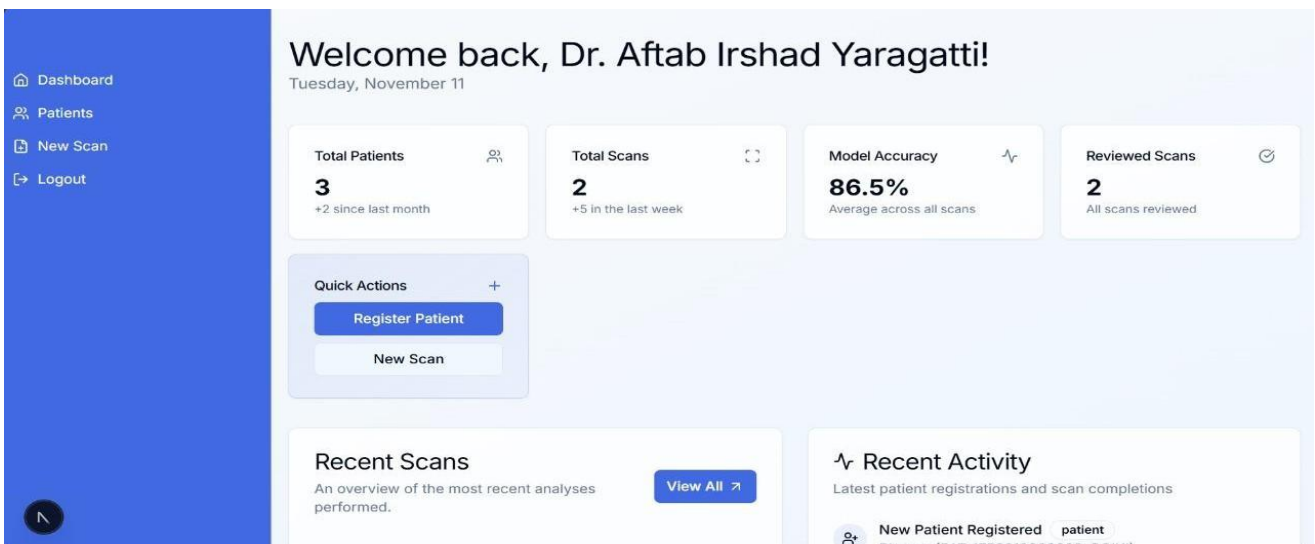


Fig 7.3.1 : Lung Cancer Dashboard

Fig 7.3.1 represents the Lung Cancer Dashboard, which appears after doctor authentication. It displays key metrics such as total patients, total CT scans analyzed, model accuracy, and reviewed scans. Quick action buttons allow the doctor to register a new patient or initiate a new CT scan analysis.

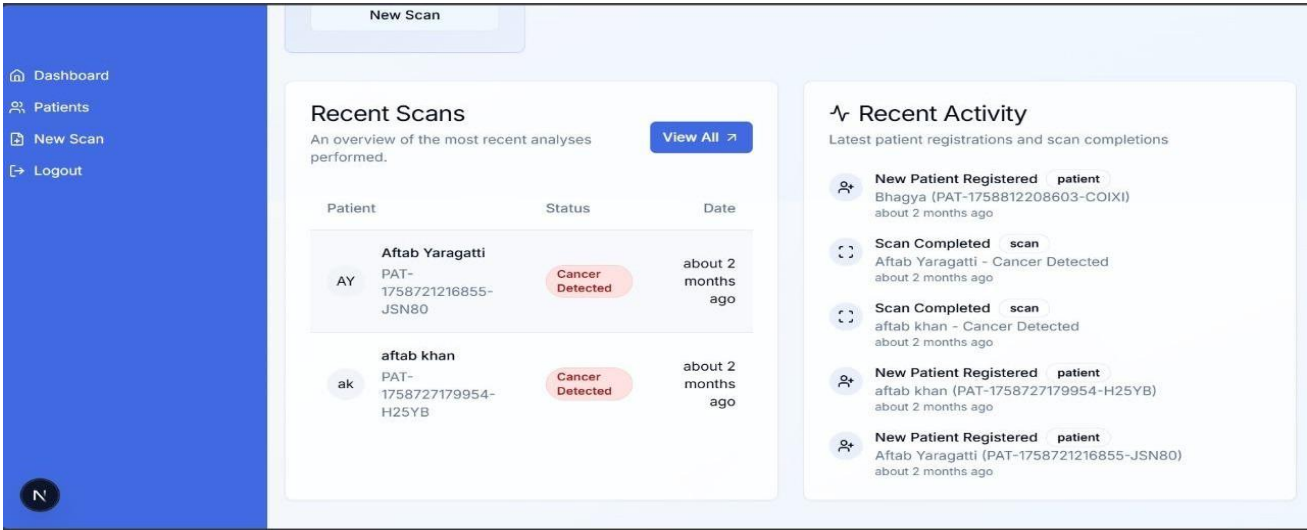


Fig 7.3.2 : Recent Scans

Fig 7.3.2 shows the Recent Scans section, where previously analyzed CT scans are listed along with patient details, scan status, and date. It also displays recent system activities such as patient registration and scan completion, enabling doctors to track diagnostic history efficiently.

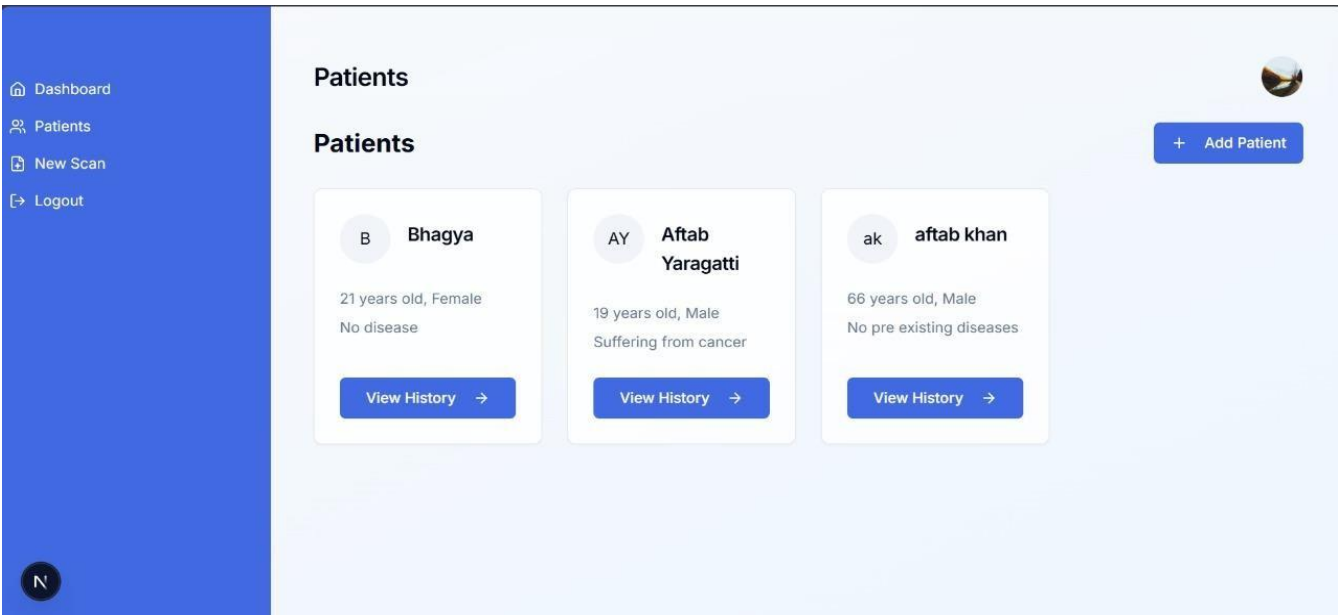


Fig 7.3.3 : Patient List Screen

Fig 7.3.3 illustrates the Patients Page, where all registered patients are displayed in card format. Each card shows patient name, age, gender, and medical status. Doctors can view patient history or add new patients from this interface.

The screenshot shows a web application interface for 'New Scan Analysis'. On the left is a blue sidebar with navigation links: 'Dashboard', 'Patients', 'New Scan', and 'Logout'. The main content area is titled 'New Scan Analysis' and features a user profile picture in the top right. Below the title, there's a section for 'Step 1: Select Patient' with the instruction: 'Choose the patient you want to perform the lung scan analysis for. You can add a new patient if none exists.' There are two tabs: 'Existing Patient' (active) and 'New Patient'. Under the 'Existing Patient' tab, it says 'Select a patient from your list' and shows a dropdown menu with the selected patient 'Bhagya (PAT-1758812208603-COIXI)'. A blue 'Continue' button is at the bottom of the form.

Fig 7.3.4 (a) : Select Patient

Fig 7.3.4 (a) shows Step 1: Select Patient for new CT scan analysis. The doctor selects an existing patient from the dropdown list or can add a new patient if required. This step ensures that each CT scan is accurately mapped to the correct patient record.

The screenshot shows the 'Step 2: Upload Scan for Bhagya' section of the application. It includes the same sidebar and navigation as the previous figure. The main content area has a title 'Step 2: Upload Scan for Bhagya' and an instruction: 'Upload a high-quality lung scan image (JPEG, PNG) for analysis. Ensure the scan is clear and well-lit.' Below this is a large dashed-border box containing a cloud upload icon and the text 'Drag & drop scan image here or click to browse'. At the bottom left of the main content area is a 'Back' button with a left arrow.

Fig 7.3.4 (b) : Upload CT Scan Image

Fig 7.3.4 (b) illustrates Step 2: Upload CT Scan Image, where the doctor uploads a lung CT scan image in supported formats such as JPEG or PNG. The drag-and-drop or browse option allows easy and secure image uploading for further analysis.

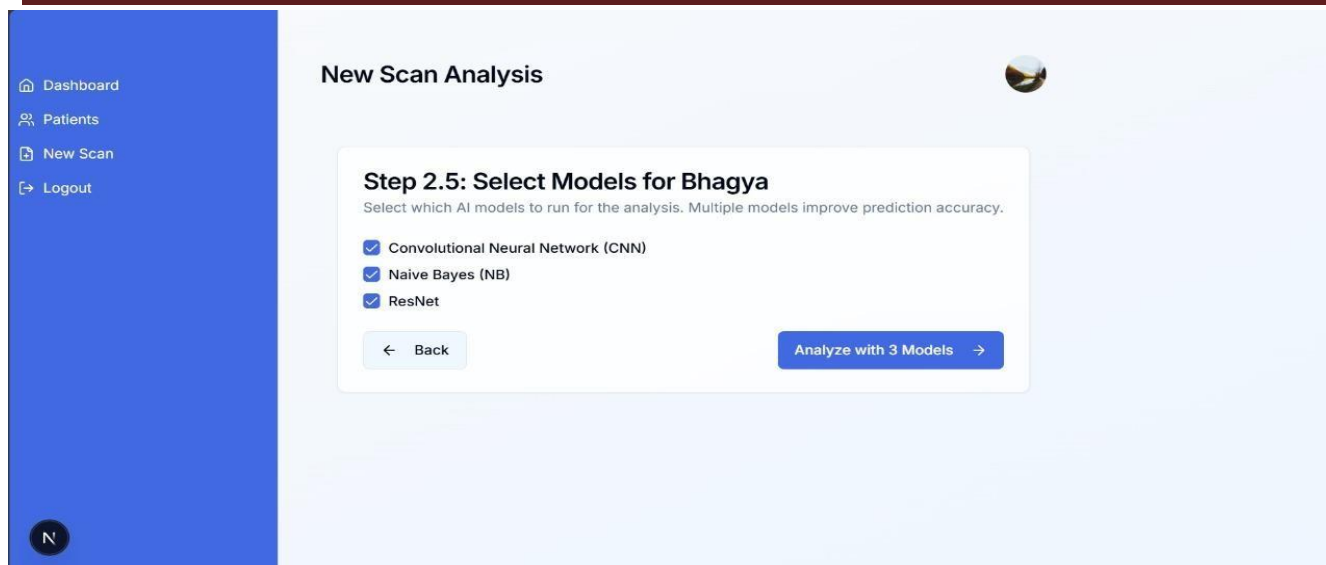


Fig 7.3.4 (c) : Select Model for Prediction

Fig 7.3.4 (c) represents Step 2.5: Model Selection, where the doctor selects one or more AI models such as CNN, Naive Bayes, and ResNet for lung cancer detection. Selecting multiple models helps improve prediction accuracy before initiating the analysis process.



Fig 7.3.4 (d) : Lung Cancer Detection Result

Fig 7.3.4 (d) displays the analysis execution stage, where the uploaded CT scan is processed using the selected machine learning and deep learning models. After analysis, the system generates results indicating whether lung cancer is detected, along with confidence scores.

CHAPTER 8

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

The AI-Based Lung Cancer Detection System using CT Scan Image Processing and Machine Learning successfully demonstrates how artificial intelligence can assist medical professionals in early and accurate diagnosis of lung cancer. By leveraging advanced image processing techniques and deep learning models, the system effectively analyzes lung CT scan images to detect cancerous patterns with high reliability.

The system provides an efficient platform for managing patient data, uploading CT scans, selecting diagnostic models, and generating clear prediction results. This reduces manual effort, minimizes diagnostic delays, and supports doctors in making informed clinical decisions. With features such as secure user authentication, historical data storage, and multi-model analysis, the system can be effectively deployed in hospitals, diagnostic centers, and clinical research environments. This project proves that AI-driven diagnostic tools can enhance healthcare efficiency, support early disease detection, and contribute to improved patient outcomes.

8.2 FUTURE WORK

To further enhance the Lung Cancer Detection System, several improvements can be introduced to increase diagnostic accuracy, scalability, and real-world clinical usability. Advanced deep learning architectures such as EfficientNet, Vision Transformers (ViT), or ensemble models can be integrated to improve cancer detection performance across diverse CT scan datasets. Incorporating multi-class classification can enable the system to identify cancer stages and types, supporting more detailed clinical analysis.

Integration with hospital information systems (HIS) and electronic health records (EHR) can allow automatic retrieval and storage of patient data, reducing manual entry and improving workflow efficiency. Real-time cloud deployment can enable remote access for doctors, support large-scale screening programs, and facilitate collaborative diagnosis. Adding explainable AI (XAI) techniques such as heatmaps and attention maps can improve transparency by highlighting affected lung regions, increasing clinician trust in AI predictions.

Future enhancements may also include longitudinal patient monitoring to track disease progression over time, mobile or web-based patient portals for viewing reports, and automated report generation in standardized medical formats. These improvements will make the system more robust, scalable, and suitable for widespread adoption in modern healthcare environments.

REFERENCES

1. S. Saxena, S. N. Prasad, A. M. Polnaya, and S. Agarwala, "Hybrid Deep Convolution Model for Lung Cancer Detection with Transfer Learning," arXiv preprint arXiv:2501.02785, Jan. 2025. [Online]. Available: <https://arxiv.org/abs/2501.02785>
2. A. Chaudhari, A. Singh, S. Gajbhiye, and P. Agrawal, "Lung Cancer Detection Using Deep Learning," arXiv preprint arXiv:2501.07197, Jan. 2025. [Online]. Available: <https://arxiv.org/abs/2501.07197>
3. S. A. Althubiti, A. J. S. Alshahrani, M. A. K. Alruwaili, A. M. Alsharif and A. A. Alqarni, "Ensemble learning framework with GLCM texture features for automated lung cancer detection from CT images," Scientific Programming, vol. 2022, Article ID 2733965, 2022. [Online]. Available: <https://www.hindawi.com/journals/sp/2022/2733965>
4. M. Mamun, M. I. Mahmud, M. Meherin and A. Abdelgawad, "LCDctCNN: Lung Cancer Diagnosis of CT scan Images Using CNN Based Model," arXiv preprint, Apr. 2023. [Online]. Available: <https://arxiv.org/abs/2304.04814>
5. Z. UrRehman et al., "Effective lung nodule detection using deep CNN with dual-path architecture," Scientific Reports, 2024. [Online]. Available: <https://www.nature.com/articles/s41598-024-51833-x>
6. C. Gao et al., "Deep learning in pulmonary nodule detection and segmentation: methods, datasets and challenges," European Radiology / PMC (overview), 2024. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/11632000/>
7. M. O. Oyediran, O. A. Afolabi and O. Adewale, "An optimized support vector machine for lung cancer detection using CT images," Scientific Reports / PMC, 2024. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11700792/>
8. L. Talukder, M. M. Islam, M. A. Uddin et al., "Machine learning-based lung and colon cancer detection using deep feature extraction and ensemble learning," arXiv preprint, Jun. 2022. [Online]. Available: <https://arxiv.org/abs/2206.01088>