VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI – 590018

A Major Project User Manual on

"Lung Cancer Detection using CT(Computed Tomography) Image Processing & Machine Learning"

Submitted By

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

Recognized under 2(f) &12B of UGC Act, 1956

Accredited at "A+" Grade by NAAC

Programmes Accredited by NBA: CSE & ECE

Academic Year 2025-2026

PROJECT EXECUTION STEPS

Lung Cancer Detection System Using CT Scan Images (Machine Learning & Deep Learning)

1. Introduction

This document provides a complete, step-by-step execution workflow for running the **Lung Cancer Detection System**. The system consists of:

- **Backend:** Python Flask API integrated with ML/DL models (CNN/ResNet/Naïve Bayes).
- **Frontend:** Modern web interface built using React.
- **Model Processing:** Accepts CT scan images and returns lung cancer prediction results.

This guide ensures smooth execution on any machine with minimal setup effort.

2. Opening the Project in Visual Studio Code

2.1 Launch VS Code

- Open **Visual Studio Code** from your system applications.

2.2 Load the Project Workspace

1. Click **File → Open Folder**.
2. Select the **root project directory** containing both **backend** and **frontend** folders.
3. Click **OK** to load the complete workspace.

VS Code will automatically detect project files, configurations, and environment settings.

3. Backend Setup and Execution (Python – Flask API)

The backend is responsible for handling image uploads, running ML/DL models, and returning predictions.

3.1 Open Integrated Terminal

- Press **Ctrl** + ` inside VS Code to open the terminal.

3.2 Navigate to the Backend Directory

cd backend

3.3 Install Required Python Dependencies

(Only required during the first setup or after adding new libraries)

pip install -r requirements.txt

This installs all necessary packages such as Flask, NumPy, OpenCV, TensorFlow/PyTorch, and other ML/DL libraries.

3.4 Run the Flask Backend Server

python app.py

Expected Console Output:

Once the server starts, you will see a message similar to:

* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

This indicates that the backend is successfully running and ready to accept requests.

3.5 Copy the Backend URL

Copy the displayed URL, for example:

http://127.0.0.1:5000

This URL must be correctly configured in the frontend for communication.

4. Frontend Setup and Configuration (React)

The frontend provides the graphical interface for uploading CT scan images and viewing prediction results.

4.1 Navigate Back to the Project Root Directory

cd ..

4.2 Enter the Frontend Directory

cd frontend

4.3 Configure Backend API Endpoint

The frontend communicates with the backend through the API URL stored in config.json.

Steps:

1. Open **config.json** located inside the frontend directory.
2. Update the backend API URL.

Example configuration:

```
{
  "API_BASE_URL": "http://127.0.0.1:5000"
}
```

Ensure the URL exactly matches the backend output.

5. Install Frontend Dependencies

Inside the **frontend** directory, run:

npm install

This installs all required Node.js modules including:

- React
- Axios (for API communication)
- TailwindCSS / Material UI (if used)
- Image-handling libraries
- Routing modules

This step may take a few minutes.

6. Run the Frontend Development Server

Start the React UI:

npm run dev

Expected Terminal Output:

A development URL such as:

VITE v4.0.0 ready in 524 ms

➜ Local: http://localhost:5173/

This indicates the frontend is active and connected.

7. Accessing the Lung Cancer Detection System

Steps:

1. Copy the frontend URL displayed in the terminal.

Example:

http://localhost:5173/

2. Open your web browser (Chrome or Edge recommended).
3. Paste the URL and press Enter.
4. The **Lung Cancer Detection System UI** will load successfully.

8. Using the System (Functional Flow)

8.1 Upload CT Scan Image

- Click the **Upload / Choose File** button.
- Select a lung CT scan image (PNG/JPG/JPEG).

8.2 Submit for Prediction

- Click **Predict / Analyze**.
- The frontend sends the image to the backend via the API.

8.3 Prediction & Output

The backend model processes the image using:

- CNN
- ResNet
- Naïve Bayes
- Transfer Learning models

The result is returned and displayed on the screen as:

- Cancer Detected / Not Detected
- Probability/Confidence Score
- Model Insights (if implemented)