

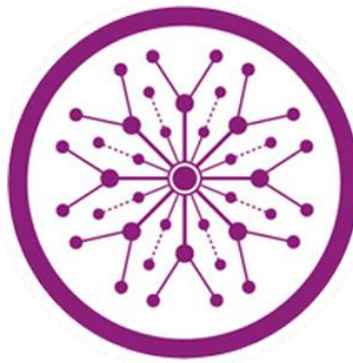
AI in Pneumonia Detection

Final Year Project

Session 2019-2023

A project submitted in partial fulfillment of the degree of

BS in information Technology



Department of Information Technology

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2023

Type (Nature of project)	<input type="checkbox"/> Development <input type="checkbox"/> Research <input type="checkbox"/> R&D			
Area of specialization				
FYP ID:	02			
Project Group Members				
Sr. #	Reg. #	Student Name	Email ID	Signature
(i)	BITM-F19-037	BITM-F19-037	bitm-f19-037@superior.edu.pk	
(ii)	BITM-F19-036	BITM-F19-036	bitm-f19-036@superior.edu.pk	
(iii)	BITM-F19-031	BITM-F19-031	bitm-f19-031@superior.edu.pk	

- The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I _____ S/D of _____, group leader of FYP under registration no at _____ Information Technology Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Name of Group Leader: _____ Signature: _____

Name of Supervisor: **Sir Javed Iqbal**

Designation: Lecturer

Signature: _____

HOD: Dr. Asad Ali Naqvi

Signature: _____

Project Report

AI in Pneumonia Detection

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Malaika	1.0	8 Sep	Template 1, Template 3	
Malaika	1.1 1.2 1.3	15 Sep	Prototype	
Allah Mafi Malaika	1.4 1.5	6 Oct	Template 5, Prototype	
Allah Mafi	1.6	2 Dec	Template 5	
	1.7 1.8 1.9			
	2.0			
	2.1			
	2.2			
	2.3			
	2.4			

APPROVAL

PROJECT SUPERVISOR

Comments: _____

Name: _____

Date: _____ Signature: _____

PROJECT MANAGER

Comments: _____

Date: _____ Signature: _____

HEAD OF THE DEPARTMENT

Comments: _____

Date: _____ Signature: _____

Dedication

"Dedicated to my parents, who have always believed in me and provided unwavering support throughout my academic journey. To my project supervisor **Mr. Javed Iqbal**, who has guided and mentored me with patience and expertise. To my core members, who has been a constant source of inspiration and encouragement. And to all the individuals who have played a part in my personal and professional growth, thank you for helping me reach this milestone."

ACKNOWLEDGEMENT

In the name of Allah, the Most Gracious and the Most Merciful, all praises to Allah for the strengths and His blessing for this task. I am dust on the path of Hazrat Muhammad (SAW) the chosen one. First, we would like to thank our parents for their endless love, prayers and encouragement throughout our degree program. Thanks to all our friends for their kindness and moral support during our study. Thanks for the friendship and memories. We would like to thank our family members for supporting, encouraging, appreciating and praying throughout our thesis duration.

Executive Summary

The Pneumonia Detector Web App in Python and Mobile App in Flutter is an application designed to assist medical professionals in diagnosing pneumonia in patients. The application utilizes a deep learning model to analyze chest x-rays and determine the likelihood of pneumonia.

The web app is built using the Flask web framework in Python and allows medical professionals to upload chest x-ray images for analysis. The mobile app, built in Flutter, provides a user-friendly interface for patients to submit their chest x-rays for analysis.

The deep learning model used in the application was trained on a large dataset of chest x-rays and is capable of achieving high accuracy in pneumonia detection. The model was developed using Tensor Flow, a popular machine learning library, and was fine-tuned using transfer learning techniques.

The Pneumonia Detector Web App and Mobile App have the potential to improve pneumonia diagnosis accuracy, reduce the time required for diagnosis, and enhance patient outcomes. The application provides a cost-effective and efficient solution to pneumonia diagnosis, especially in areas with limited access to medical professionals.

Overall, the Pneumonia Detector Web App and Mobile App have the potential to make a significant impact in the field of medical diagnostics, and have the potential to save lives by improving the accuracy of pneumonia diagnosis.

Table of Contents

Plagiarism Free Certificate	ii
Dedication	v
Executive Summary	vii
Table of Contents	viii
List of Figures	xi
List of Tables	xii
Introduction.....	2
1.1. Background	4
1.2. Motivations and Challenges	4
1.3. Goals and Objectives.....	6
1.4. Literature Review/Existing Solutions	6
1.5. Gap Analysis	7
1.6. Proposed Solution	7
1.7. Project Plan	7
1.7.1. Work Breakdown Structure	9
1.7.2. Roles & Responsibility Matrix	10
1.7.3. Gantt Chart	12
1.8. Report Outline	12
Chapter 2.....	15
Software Requirement Specifications.....	15
2.1. Introduction	16
2.1.1. Purpose	16
2.1.2. Document Conventions	16
2.1.1 Intended Audience and Reading Suggestions	16
2.1.3. Product Scope	17
2.2. Overall Description	17
2.2.1. Product Perspective	17
2.2.2. User Classes and Characteristics	18
2.2.1 User Classes.....	18
2.2.3. Operating Environment	18
2.2.4. Design and Implementation Constraints.....	19
2.2.5. Assumptions and Dependencies	20
Install Python Dependencies.....	20
2.3. External Interface Requirements.....	20
2.3.1. User Interfaces	20
User interact with default web browser.	20
2.3.2. Hardware Interfaces.....	21
2.3.3. Software Interfaces	21
2.4. System Features.....	22
2.2.2 System Features	22
2.2.2.1 Description and Priority	23
2.2.2.2 Stimulus/Response Sequences	24
2.2.2.3 Functional Requirements	25
2.1.4 Functional Requirements	25

2.5 Other Nonfunctional Requirements	26
2.1.1 Performance Requirements.....	26
2.1.2 Safety Requirements.....	26
2.1.3 Security Requirements.....	26
Chapter 3.....	28
Use Case Analysis.....	28
3.1. Use Case Model	29
3.2. Use Case Descriptions.....	36
Sign Up Use Case:	36
Chapter 4.....	40
System Design	40
4.1. Architecture Diagram.....	41
4.2. Domain Model.....	42
4.3. Methodology	46
4.4. flowchart Diagram.....	47
4.5. Sequence / Collaboration Diagram	48
4.6. Operation contracts	49
4.7. Activity Diagram.....	50
4.8. State Transition Diagram	51
4.9. Component Diagram	52
4.10. Deployment Diagram	53
Chapter 5.....	54
Implementation	54
5.1. Important Flow Control/Pseudo codes.....	60
5.2. Components, Libraries, Web Services and stubs	61
5.3. Deployment Environment	61
5.4. Tools and Techniques.....	62
5.5. Best Practices / Coding Standards.....	62
Naming Conventions and Standards	63
Good Programming Practices	63
Comments	63
Exception Handling	63
5.6. Version Control	63
Chapter 6.....	65
Testing and Evaluation	65
6.1. Use Case Testing.....	66
6.2. Equivalence partitioning	67
6.3. Boundary value analysis.....	67
6.4. Data flow testing	67
6.5. Unit testing.....	68
6.6. Integration testing.....	68
6.7. Performance testing.....	68
6.8. Stress Testing	68
Chapter 7.....	69
Summary, Conclusion and Future Enhancements	69
7.1. Project Summary	70

7.2. Achievements and Improvements	71
7.3. Critical Review.....	73
7.4. Lessons Learnt.....	73
7.5. Future Enhancements/Recommendations	74
Appendices.....	75
Appendix A: Information / Promotional Material	76
Reference and Bibliography	78
Index	80

List of Figures

1.1	Caption of first figure of first chapter.....	6
1.2	Caption of second figure of first chapter.....	7
2.1	Caption of first figure of second chapter.....	14
2.2	Caption of second figure of second chapter.....	22
2.3	Caption of third figure of second chapter.....	26
5.1	Caption of first figure of fifth chapter.....	49
5.2	Caption of second figure of fifth chapter.....	49

List of Tables

1.1	label of first table of first chapter.....	6
1.2	label of second table of first chapter.....	7
2.1	label of first table of second chapter.....	14
2.2	label of second table of second chapter.....	22
2.3	label of third table of second chapter.....	26
5.1	label of first table of fifth chapter.....	49
5.2	label of second table of fifth chapter.....	49

Chapter 1

Introduction

Chapter 1: Introduction

A pneumonia detection mobile and web app is a software application that can help users detect the presence of pneumonia in individuals by analyzing symptoms and other medical data. The app can be accessed through a mobile device or web browser, making it convenient for users to use from anywhere at any time.

The app would use artificial intelligence (AI) and machine learning (ML) algorithms to analyze the user's data and provide an accurate diagnosis. The algorithms would take into account factors such as age, medical history, symptoms, and other relevant information to make a diagnosis.

The app could be used by healthcare professionals, caregivers, and individuals who are concerned about their health. Healthcare professionals could use the app to quickly and accurately diagnose pneumonia, which could lead to earlier treatment and better outcomes for patients. Caregivers could use the app to monitor the health of their loved ones and detect potential cases of pneumonia. Individuals could use the app to monitor their own health and seek medical attention if needed.

Some of the features of the pneumonia detection app could include:

Detector: Users could enter their chest x-ray into the app, and the AI algorithm would analyze the data to determine the likelihood of pneumonia.

Medical history tracker: Users could input their medical history into the app, which would be taken into account when making a diagnosis.

Chatbot: The app could include a chatbot feature that could provide users with real-time advice and guidance on how to proceed based on their symptoms.

In-app doctor consultation: The app could offer users the option to connect with a doctor through the app for further consultation and treatment.

Overall, a pneumonia detection mobile and web app could be a valuable tool for detecting and diagnosing pneumonia, particularly in areas where access to healthcare is limited. By leveraging AI and ML, the app could provide accurate and timely diagnoses, potentially leading to improved health outcomes for patients.

We use in this project this technique

- Algorithm [DL Model]
- CNN [Flask]
- Framework [python]
- TensorFlow,Keras[HTM,CSS]
- Tensor flow lite[flutter,dart]

1.1. Background

This document explains the complete project report of our Final Year Project, which is developed by the 3 team members of this project i.e. **Malaika Imran, Allah Mafi and Usama Afzal** This document is aimed to explain every process involved in the implementation of our Final Year Project. It provides the deep background study of the deep learning and python also with the reasons for not considering the alternative ways of creating this App and justification of the decision for our project to build a App. We make a detailed description of “**Pycharm**” the main tool used in the creation of our App for web application and use android studio for mobile app.

1.2. Motivations and Challenges

Motivations:

Early detection: One of the key motivations for developing a pneumonia detector app is to enable early detection of pneumonia. Early diagnosis and treatment of pneumonia can lead to better health outcomes for patients and can help prevent the spread of the disease.

Accessibility: Another motivation is to increase access to healthcare for people who may not have access to a doctor or medical facility. A pneumonia detector app can be accessed from anywhere, making it easier for people in remote or underserved areas to receive a diagnosis.

Efficiency: The use of an AI-powered pneumonia detector app can help healthcare professionals to save time and resources by automating the diagnosis process. This can lead to quicker and more accurate diagnoses, enabling prompt treatment and reducing the burden on healthcare systems.

Challenges:

Accuracy: One of the key challenges of developing a pneumonia detector app is ensuring that it is accurate and reliable. The app needs to be able to accurately diagnose pneumonia and distinguish it from other conditions with similar symptoms.

Data privacy and security: The app will need to comply with data privacy regulations to protect users' sensitive health data. Additionally, the app will need to implement robust security measures to prevent unauthorized access to user data.

Limited access to technology: In some areas, limited access to technology or internet connectivity may make it difficult for people to use the app. This challenge can be addressed by developing an app that is accessible offline and can be used on low-end devices.

Integration with existing healthcare systems: Integrating the app with existing healthcare systems can be challenging due to differences in data formats and data-sharing protocols. However, overcoming this challenge is important for the app to be used effectively in clinical settings.

Overall, while a pneumonia detector app has the potential to provide significant benefits, addressing these challenges will be crucial to its success. By focusing on accuracy, data privacy and security, accessibility, and integration with existing healthcare systems, developers can create an app that can improve early detection and treatment of pneumonia while maintaining high standards of quality and safety.

1.3. Goals and Objectives

It is crucial to have quick and accurate detection of pneumonia so patients can receive treatment in a timely manner especially in impoverished regions.

Only focus on the classification of normal chest and affected by pneumonia.

The goals of our project include:

- Strategies and challenges that will be enjoyable for both doctors and patients.
- Making this available for web devices as well as mobile devices.
- Strategies are supposed to teach the value of money and time management when running a small business.
- Difficulty level increases with the passage of time making it more challenging.
- **Make this as fast as possible**
 - The goal is to classify only pneumonia-causing bacteria and covid-19.
 - Import the libraries.
 - Import the dataset.
 - Import the code.
 - Testing
 - Deployment stage

1.4. Literature Review/Existing Solutions

In existing system nurse read the x-ray. Manually identifications are subject to potential misinterpretations and time-consuming issues owing to the visual similarity of pneumonia lesions.

1.5. Gap Analysis

Since the major concern of our detector project is to make our detector faster than Reading an x-ray manually User run this app on their Windows and mobile phone. Hence, the actual performance of the detector is matched with the desired performance of the detector. This is done by the right choice of tool, feasible project plan, and skilled development team members. (details of which will be discussed later in this documentation)

1.6. Proposed Solution

The proposed solution for a pneumonia detector app would be to develop a user-friendly mobile and web-based application that uses artificial intelligence (AI) to diagnose pneumonia. The app would use machine learning algorithms trained on large datasets of chest X-rays and other clinical data to accurately diagnose pneumonia.

1.7. Project Plan

Here's a high-level project plan for developing a pneumonia detector app with a web app in Python and a mobile app in Flutter:

Requirements gathering: Define the app's requirements by conducting user research, analyzing the market, and identifying the features and functionalities needed.

Design: Create wireframes, user flows, and visual designs for both the web and mobile apps. This includes designing the UI, incorporating necessary features, and ensuring that the app is user-friendly and accessible.

Back-end development: Build the back-end of the web app using Python and relevant libraries such as Flask or FastAPI. Develop an AI algorithm that can accurately diagnose pneumonia by analyzing chest X-rays and other clinical data. Ensure that the app is scalable and can handle a large volume of requests.

Front-end development: Develop the front-end of the web app using HTML, CSS, and integrate it with the back-end. Ensure that the web app is responsive, user-friendly, and accessible.

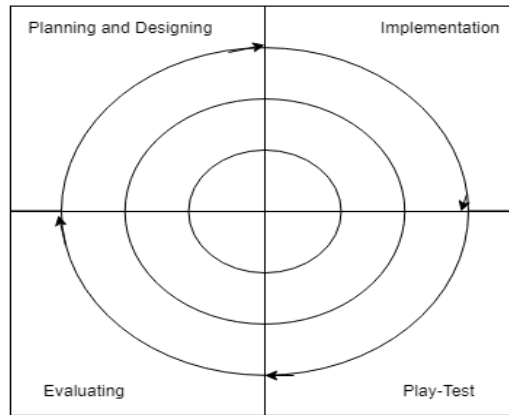
Mobile app development: Develop the mobile app using Flutter, a popular cross-platform mobile app development framework. Ensure that the app's design is consistent with the web app and that it can integrate with the back-end.

Testing: Conduct rigorous testing of the app to ensure that it's reliable, accurate, and user-friendly. This includes conducting unit tests, integration tests, and user acceptance tests.

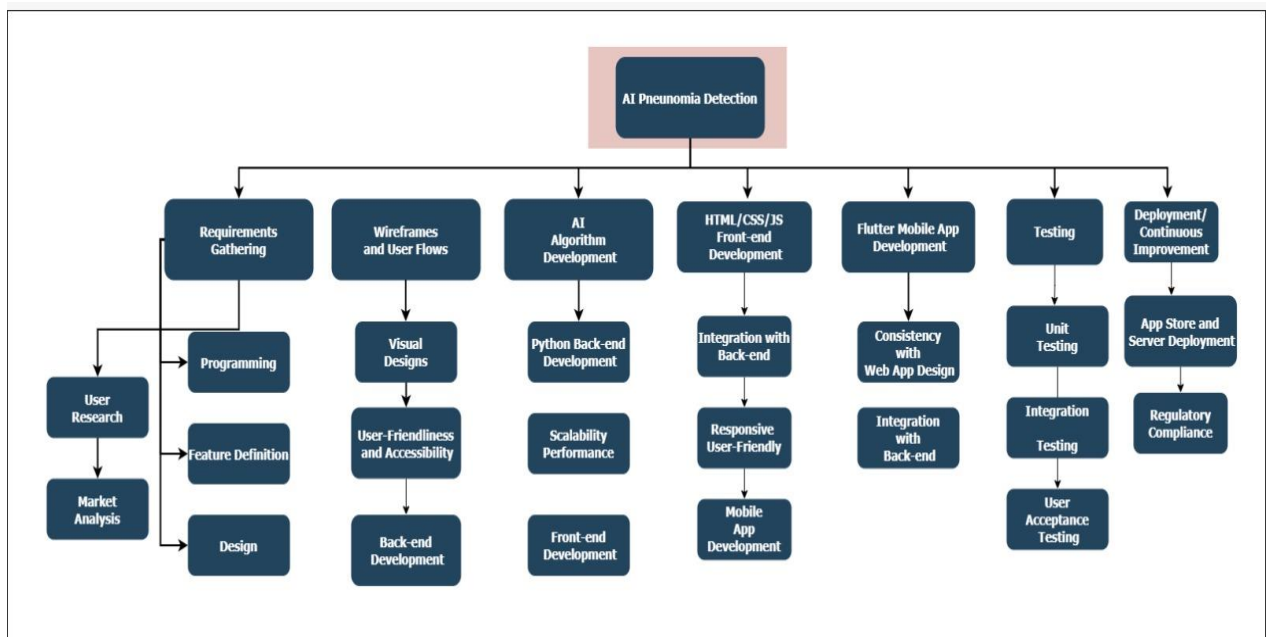
Deployment: Deploy the web app and mobile app to their respective app stores and servers. Ensure that the app meets all the necessary regulatory requirements, such as data privacy regulations.

Continuous improvement: Continuously monitor the app's performance, collect user feedback, and make improvements to ensure that the app remains accurate, reliable, and effective over time. This includes incorporating new developments in AI and machine learning and adding new features as needed.

Overall, this project plan is iterative and requires close collaboration between the development team, designers, and stakeholders. Effective project management tools and methodologies, such as Agile or Scrum, can help ensure that the project is completed on time and within budget while meeting all the necessary requirements.



1.7.1. Work Breakdown Structure



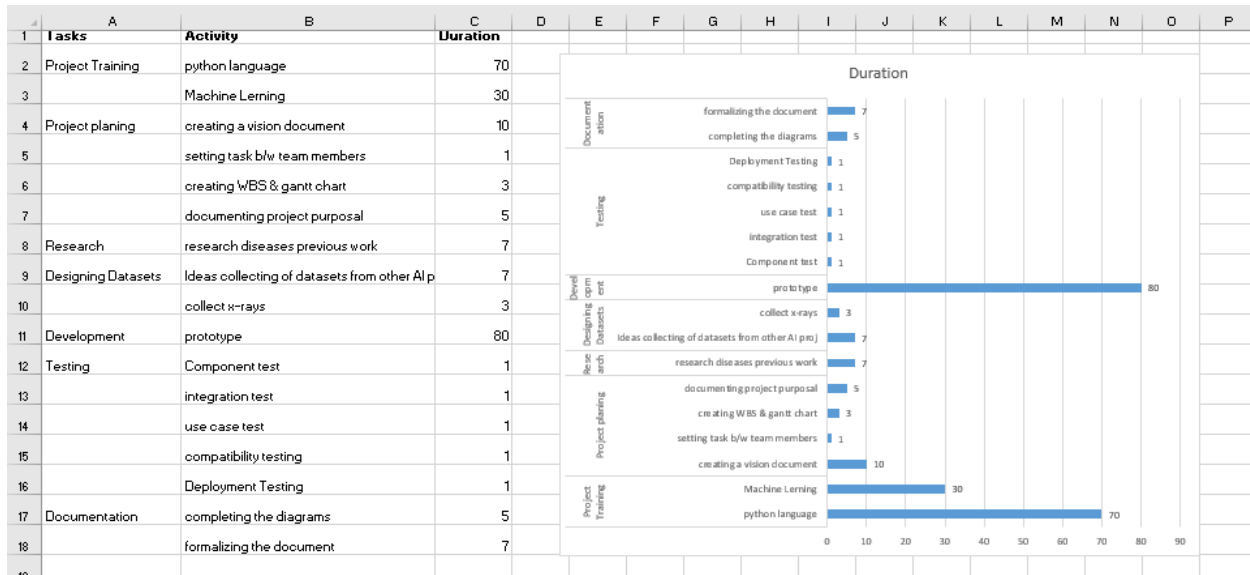
1.7.2. Roles & Responsibility Matrix

The purpose of roles & responsibility matrix is to identify who will do what.

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Project Training	1	Training for the gain of skills in Python development tool “ Pycharm ”. And flutter in development tool android studio.	40	
		2	Training for the gain of skills in ML tool “ Tensor flow ” and tensor flow lite.	30	
2	Project Planning	1	Create Vision Document	10	
		2	Setting tasks between the team members	1	
		3	Creating WBS & Gantt Chart to manage time efficiently	3	
		4	Documenting Project proposal	5	

3	Research	1	Research diseases Symptoms, Treatment.	7	
4	Designing Datasets	1	Ideas collection of datasets by the Inspiration from other AI based projects.	7	
		2	Collect X-rays	3	
5	Development	1	Prototyping	65	
6	Testing	1	Component Testing	1	
		2	Integration Testing	1	
		3	Use case Testing	1	
		4	Compatibility Testing	1	
		5	Deployment Testing	1	
7	Documentation	1	Completing the Diagrams	5	
		2	Formalizing the documentation	7	

1.7.3. Gantt Chart



1.8. Report Outline

Here's an outline for a report on the development of a pneumonia detector app with a Python web app and a Flutter mobile app:

Introduction

Background and context of the project

Motivations and challenges of developing a pneumonia detector app

Goals and objectives of the project

Methodology

Overview of the development process, including the use of Agile or Scrum methodology

Description of the tools and technologies used in the development process

Discussion of the development team and their roles and responsibilities

Requirements and Design

User research findings and how they informed the app's features and functionalities

Wireframes and user flows for the web and mobile apps

Visual designs for the web and mobile apps

Discussion of the app's user-friendliness and accessibility

Implementation

Back-end development, including the development of an AI algorithm and the use of Python libraries such as Flask.

Front-end development, including the use of HTML/CSS/JS for the web app and Flutter for the mobile app

Discussion of integration between the front-end and back-end

Testing and Deployment

Overview of the testing process, including unit testing, integration testing, and user acceptance testing

Discussion of the deployment process, including deployment to app stores and servers

Regulatory compliance considerations

Results and Evaluation

Discussion of the app's accuracy in detecting pneumonia

Discussion of user feedback and how it was incorporated into the app's development

Evaluation of the app's performance and effectiveness in achieving the project's goals and objectives

Conclusion and Future Work

Summary of the project's outcomes and accomplishments

Discussion of the limitations of the app and opportunities for future improvement

Reflection on the project as a whole and lessons learned

References

List of sources consulted during the development process

Citations for any external libraries or technologies used in the development process

Appendices

Additional materials that support the report, such as technical documentation, sample code, or user testing results.

Ch#2: In Ch. 2 we can define the features of our project and define functional and non-functional requirements and internal and external interface of our project

- **Ch#3:** we can implementation and define actors and use case analysis in our project with the help of diagrams.
- **Ch#4** we can define our project through diagrams
- **Ch#5** we can implement our project in this chapter and tell what tools and techniques are used.

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

The Software specification and Requirements of pneumonia detector for the successful deployment are.

- Loading Required Libraries and setting Configurations
- Downloading the Data sets from Kaggle
- Loading the Data into Memory
- Representing Images in Computer Vision
- Defining the Model
- Transfer Learning
- Training the Model
- Deploying” the Model
- Model Performance
- Deploying the Application
- Sending Data to the Application
- Defining the App
- Running the App Locally

2.1.1. Purpose

Develop a web or mobile app to detect the initial bacteria and virus of pneumonia from patient's x-ray.

2.1.2. Document Conventions

This document follows the font Calibri and the font size is 12 for paragraph text. The bold-faced text has been used to emphasize sections and italicized text is used to label and recognize diagrams. And the all text is Justified.

2.1.1 Intended Audience and Reading Suggestions

The document is mainly created for the approval of our Final Year Project of Information Technology and hence the main intended audience is the approval panel of the Superior University that will declare the status of our project. Also, the documentation may help the team members (Allah Mafi & Usama Afzal) to keep a track of all aspects (features, processes used, model used, risks identified, risks reasons, processes used to remove risks) of the

detector application and that will help for future improvements and updates in the ML and may also be essential when the team members may create a similar project. It may also help the user to understand the project in detail.

2.1.3. Product Scope

It is crucial to have quick and accurate detection of pneumonia so patients can receive treatment in a timely manner especially in impoverished regions.

Only focus on the classification of normal chest and affected by pneumonia.

The goal is to classify only pneumonia-causing bacteria and covid-19.

- Import the libraries.
- Import the dataset.
- Import the code.
- Testing
- Deployment stage

2.2. Overall Description

This project is running on Pycharm tool use html to creates templates which is running on web browser create a flask app by using python to give predictions by using deep learning train model.

2.2.1. Product Perspective

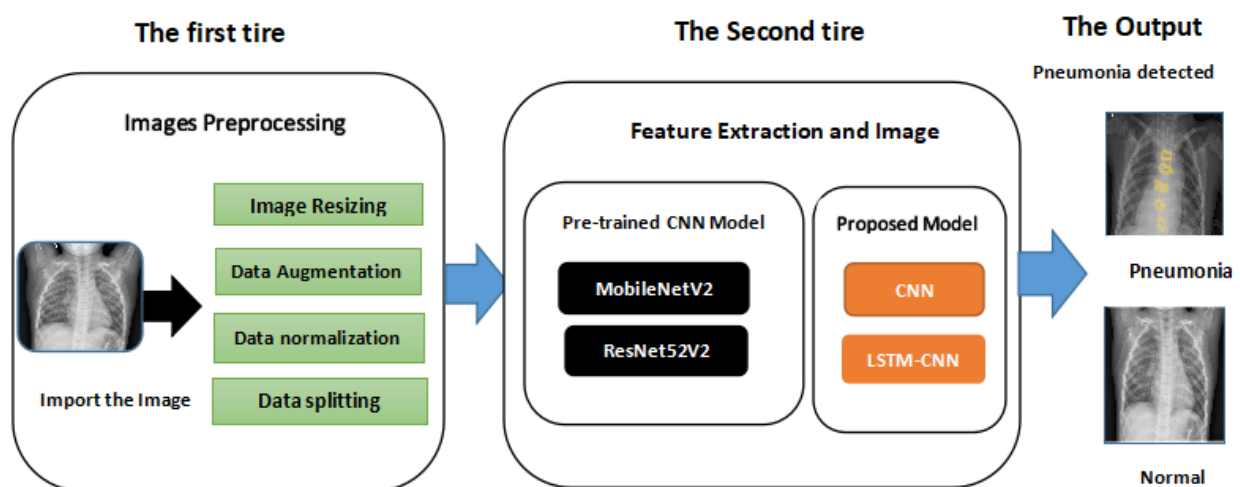


Figure 1

2.2.2. User Classes and Characteristics

2.2.1 User Classes

Following are the major classes for our project users:

- **Web users**

Web app develop in python users can use this app by web link.

- **iPhone users:**

The app will be available for the iPhone users, hence will be available on Apple Store.

- **iPad users:**

As described above, the app will be available on Apple Store, also the resolution and graphics will be in a wise that will make able to run efficiently on big screens iPads too.

- **Android Phone users:**

The app will be available for the Android phones users, hence will be available on Play Store.

- **Android Tab users:**

As described above, the app will be available on Play Store, also the resolution and graphics will be in a wise that will make able to run efficiently on big screens Android Tabs too.

2.2.3. Operating Environment

For a pneumonia detector app, the operating environment may include the following:

Mobile Devices:

iOS and Android smartphones and tablets

Minimum hardware requirements for running the app, such as processor speed, memory, and storage

Web Browser:

Chrome, Firefox, Safari, or other modern web browsers

Internet connection with sufficient speed and bandwidth

Server-Side Environment:

Python and relevant libraries and frameworks for web app development, such as Flask

Appropriate operating system, such as Linux or Windows

Web server, such as Apache or Nginx

Database management system, such as MySQL or PostgreSQL

Development Environment:

Integrated Development Environment (IDE) for developing web app and mobile app

Emulator or simulator for testing the mobile app

Unit testing frameworks and tools for testing the web app

Other Requirements:

Relevant APIs and SDKs for accessing device hardware such as camera, microphone, and storage

Secure Socket Layer (SSL) certificates for secure data transfer

Tools for error reporting and analysis

Overall, the operating environment of a pneumonia detector app will depend on the chosen development framework is flutter, programming language is python for web and dart for mobile, and target platforms sdk 29 and above. It is essential to ensure that the app runs smoothly on all targeted platforms and devices to provide a seamless user experience.

2.2.4. Design and Implementation Constraints

Following are the constraints related to the implementation and design of our Final Year Project:

- Web app run via link
- This App would not be able to run on an android device having less than Android OS Version 4.0 installed.
- This App would not be able to run on an iOS device having less than iOS Version 6 installed.
- The app must be done on Vs code of at least Version 1.73 or above, installed in a Windows 10 laptop.

The app must complete its development, testing and documentation till the end of 8th Semester (May 2023).

2.2.5. Assumptions and Dependencies

We will be using labeled Chest X-Ray images to train a model for pneumonia detection. The dataset is hosted on Kaggle and consists of 5,863 X-Ray images. The pneumonia images are further categorized as viral or bacterial. Ignoring this secondary categorization, our model will classify images as pneumonia or normal.

Kaggle API

The Kaggle API makes it easy to download Kaggle datasets programmatically/through the command line interface.

1. Create a Kaggle account if you don't have one already
2. Generate a new token at `kaggle.com/USERNAME/accounts`. This will prompt you to download a `kaggle.json` file which contains credentials for accessing the API
3. Export credentials from the json file as environment variables. The easiest way to do this is by running

Install Python Dependencies

Run the good old `pip install -r requirements.txt` to download the Python dependencies.

2.3. External Interface Requirements

2.3.1. User Interfaces

User interact with default web browser.

2.3.2. Hardware Interfaces

Windows 10 OS.

<u>Software used</u>	<u>Description</u>
Operating system	We have chosen Windows 10 operating system for its best support and user-friendliness. Or a mobile phone android/ iOS
Tools	Pycharm, Google colab, android studio and teachable machine.
Techniques	X-ray images, CNN, AL(Artificial Intelligence)

2.3.3. Software Interfaces

Pycharm is a platform having the capability for **web app** devices.

Android studio is a platform for android and iOS apps.

- First, for the **development of the app**:

-a Windows 10 laptop is required

- with Pycharm installed in it

- with VS code installed in it

With a android studio installed in it

- with pip installed in it

- After the making the web or app to be able to run on Android devices

-An android phone with the latest version of Android OS installed.

-A Mac computer running OS X 10.14 or later

-X Code installed in Mac computer

-The latest version of Pycharm installed in Mac

-An iOS device with the latest version installed.

2.4. System Features

2.2.2 System Features

The system features of a pneumonia detector app can be categorized into the following:

Authentication and User Management:

User registration and login functionality to access the app's features.

User profile management to allow users to update their information and preferences.

Secure authentication and authorization mechanisms to ensure the confidentiality and integrity of user data.

Image Capture and Processing:

Ability to capture medical images of the lungs, such as X-rays or CT scans.

Image processing algorithms to analyze the images and detect signs of pneumonia.

Image enhancement techniques to improve the quality of the images and increase the accuracy of the analysis.

Diagnosis and Reporting:

Automatic diagnosis of pneumonia based on the results of the image analysis.

Provision of detailed diagnostic reports with recommendations for further medical evaluation and treatment.

Ability to store and retrieve diagnostic reports for future reference.

Chabot:

A chatbot for pneumonia detection can be a useful tool to assist users in identifying potential signs and symptoms of pneumonia. The chatbot can be programmed with a series of questions to help identify potential risk factors and symptoms, such as coughing, fever, and chest pain. processing (NLP) to understand and respond to user inquiries, providing information on pneumonia, its causes, and how it can be diagnosed and treated. a chatbot for pneumonia detection can be a valuable tool to assist users in identifying potential symptoms and risk factors for pneumonia and provide them with the necessary information and guidance to seek appropriate medical attention.

Medicine purchase/Payment method:

This app allows you to purchase medicines and healthcare products online and have them delivered to your doorstep.

Reviews for improvement

feedback or suggestions for improvement

User Interface and Experience:

A user-friendly interface that allows for easy navigation and interaction with the app's features.

Intuitive design and layout that is accessible to users of different ages and technical abilities.

Support for multiple languages and localization features to accommodate users from different regions and cultures.

Integration and Compatibility:

Ability to integrate with third-party APIs and services, such as cloud storage or image processing APIs.

Compatibility with different hardware configurations and operating systems, such as iOS and Android.

Security and Privacy:

Encryption of user data and secure data transfer to protect against unauthorized access or data breaches.

Implementation of appropriate security measures, such as multi-factor authentication, to prevent hacking attempts.

pneumonia detector app is designed to provide a reliable and efficient tool for the early detection and diagnosis of pneumonia, helping to improve patient outcomes and reduce healthcare costs.

2.2.2.1 Description and Priority

It is crucial to have quick and accurate detection of pneumonia so patients can receive treatment in a timely manner especially in impoverished regions. Human welfare is our first priority.

2.2.2.2 Stimulus/Response Sequences

The application consists of a few simple steps the user can use the camera to take a photo or import a certain image from the photo gallery for analysis once the user selects the image, the application processes it and displays the result with the prediction and the percentage of confidence offered by that result

Stimulus: User launches the pneumonia detector app.

Response: The app displays the main screen with options to perform a test, view previous test results, and access other features.

Stimulus: User selects the option to perform a test.

Response: The app prompts the user to provide information such as their age, gender, and any symptoms they are experiencing.

Stimulus: User provides the requested information and starts the test.

Response: The app guides the user through the process of taking a chest X-ray or a CT scan, or provides other instructions for performing the test.

Stimulus: User completes the test.

Response: The app analyzes the test results and provides an interpretation based on the user's input and the analysis.

Stimulus: App provides a result indicating the likelihood of pneumonia or another respiratory illness.

Response: The user can view the test results and interpret the outcome or seek medical advice. it is important to note that the results of a pneumonia detector app are not a substitute for professional medical advice and should only be used as a screening tool. It is always recommended to seek medical advice from a healthcare professional if you have any concerns about your health

2.2.2.3 Functional Requirements

2.1.4 Functional Requirements

This is functional requirement our project

Images P reprocessing

- Image Resizing
- Data Augmentation
- Data normalization
- Data splitting

Feature Extraction and image

- Pre-trained CNN model
 - 1) MobileNetV2
 - 2) ReesNet52V2
- **Proposed Model**
 - 1) CNN
 - 2) LSTM-CNN

Functional requirements for a pneumonia detector flutter mobile app could include:

User Authentication: The app should allow users to register or login to their account to access the app's features.

User Input: The app should allow users to input their personal information, such as age, gender, and any relevant medical history, to provide accurate test results.

Image Capture: The app should allow users to capture chest X-rays or CT scans using the device's camera or upload images from their device's storage.

Image Analysis: The app should be able to analyze the uploaded images and identify any signs of pneumonia or other respiratory illnesses.

Test Results: The app should provide the user with a clear and easy-to-understand report of the test results, indicating the likelihood of pneumonia or other respiratory illnesses.

Share Test Results: The app should allow users to share their test results with healthcare professionals, family members, or others as needed.

Notification: The app should be able to send push notifications to users, reminding them to perform regular tests and alerting them of any changes in their test results.

Secure Data Management: The app should ensure that all user data, including personal information and test results, is stored securely and is only accessible by authorized users.

Customer Support: The app should provide users with access to customer support, either through a chatbot, email, or phone support, to answer any questions or concerns they may have.

The specific functional requirements may vary depending on the app's design, features, and target audience. Additionally, it is important to ensure that the app complies with all applicable laws and regulations regarding medical devices and healthcare applications.

2.5 Other Nonfunctional Requirements

Following is an explanation of **non-functional attributes** of the project pneumonia detection.

2.1.1 Performance Requirements

Based on the capabilities of current phones and tablets, performance should not be an issue. However, phones and tabs with weaker hardware may incur some difficulties and potentially run slower. The app design is tailored in order to give an enjoyable experience on all Android, iOS and Windows phones, regardless of hardware. The functionality of the app is simplistic enough

2.1.2 Safety Requirements

The Application is completely environmentally friendly and does not cause any safety violations. The menu will have a flexible font that can be zoomed so as to not over constrain the eyes.

2.1.3 Security Requirements

There is a need for a proper and encrypted login authentication for head doctor and nurse as patient sensitive information as well as record should be protected from hacking. Information transmission should be securely transmitted to dataset without any changes in information to avoid disturbances in payments

Testability:

All the requirements are fulfilled, response time is low, and all functions are working perfectly.

Usability:

Interface of the software must be easy to use, so interface should be simple.

Chapter 3

Use Case Analysis

3.1. Use Case Model

Model-1

Title	Signup and Login
Primary Actors	Patients
Secondary Actors	Doctor, Nurses
Stake Holders	Management, Development team
Goal	To create a new account and login to the system.
Pre-conditions	<ul style="list-style-type: none"> • The user should have IOS or Android Device • The user must download the game from Apple Games Store/Play Store. • The user has access to the internet and a web browser. • The user is not currently logged in to the system.
Main Success Scenario:	<ul style="list-style-type: none"> • The user navigates to the system's homepage. • The user clicks on the "Signup" button. • The system displays a signup form. • The user enters their email address, username, password, and other required information. • The user clicks the "Submit" button. • The system validates the user's information and creates a new account. • The system sends a verification email to the user's email address. • The user checks their email and clicks on the verification link. • The system verifies the user's email address and confirms the account creation. • The user navigates back to the system's homepage. • The user enters their username and password. • The user clicks the "Login" button. • The system verifies the user's credentials and logs them in. • The system redirects the user to their account dashboard.

Post conditions:	The user is logged in to the system and can access their account dashboard.
Alternate/Extended Flow	<ul style="list-style-type: none"> • If the user enters invalid information during signup, the system displays an error message and prompts the user to correct their information. • If the user's email address is already associated with an existing account, the system displays an error message and prompts the user to log in instead. • If the user forgets their password, the system provides a password reset mechanism.

Model-2

Title	Pneumonia Detection
Primary Actors	Patients
Secondary Actors	Doctor, Nurses
Stake Holders	Management, Development team
Goal:	To detect and diagnose pneumonia in patients using a machine learning algorithm.
Pre-conditions	<ul style="list-style-type: none"> • The user should have IOS or Android Device • The user must download the game from Apple Games Store/Play Store.
Main Success Scenario:	<ul style="list-style-type: none"> • The healthcare professional or patients opens the Pneumonia Detector software. • They select the option to "Start" after signup. • The system displays instructions on how to take the X-ray image of the patient's chest. • They take an X-ray image of the patient's chest. • The healthcare professional uploads the X-ray image to the

	<p>system.</p> <ul style="list-style-type: none"> • The system processes the image using a machine learning algorithm to detect the presence of pneumonia. • The system provides a diagnosis based on the results of the analysis. • The healthcare professional reviews the diagnosis and determines the appropriate treatment.
Post-conditions	The healthcare professional has made a diagnosis and determined a course of treatment for the patient.
Alternate/Extended Flow	<ul style="list-style-type: none"> • If the healthcare professional encounters technical issues with the software, they can contact technical support for assistance. <ul style="list-style-type: none"> • If the system cannot process the X-ray image, the healthcare professional may need to retake the image and try again. • If the system's diagnosis is inconclusive, the healthcare professional may need to order additional tests or seek a second opinion.

Model-3

Title	Chatbot
Primary Actors	Patients
Secondary Actors	Doctor, Nurses
Stake Holders	Management, Development team
Goal:	To detect and diagnose pneumonia in users through a chatbot interface.
Pre-conditions	<ul style="list-style-type: none"> • The user has access to the Pneumonia Detector app and has chosen to use the chatbot interface. • The user is willing to provide necessary information to complete the diagnosis.

Main Success Scenario:	<ul style="list-style-type: none"> • The user opens the Pneumonia Detector app and selects the chatbot option. • The chatbot greets the user and prompts the user to provide their demographic information and medical history. • The user enters the required information into the chatbot interface. • The chatbot asks the user to provide a photo of their chest X-ray. • The user takes a photo of their chest X-ray and uploads it to the chatbot interface. • The chatbot processes the image using a machine learning algorithm to detect the presence of pneumonia. • The chatbot provides a diagnosis based on the results of the analysis. • The chatbot offers recommendations for further action, such as seeking medical attention or scheduling a follow-up appointment.
Post-conditions	The user has received a diagnosis and guidance on what to do next.
Alternate/Extended Flow	<ul style="list-style-type: none"> • If the user encounters technical issues with the chatbot interface, they can contact technical support for assistance. • If the chatbot cannot process the X-ray image, the user may need to retake the image and try again. • If the Chatbot's diagnosis is inconclusive, the user may need to seek a second opinion from a healthcare professional.

Model-4

Title	User profile
Primary Actors	Patients
Secondary Actors	Doctor, Nurses
Stake Holders	Management, Development team
Goal:	To maintain a user profile for the Pneumonia Detector app.
Pre-conditions	The user has registered for an account in the Pneumonia

	Detector app.
Main Success Scenario:	<ul style="list-style-type: none"> • The user opens the Pneumonia Detector app and navigates to their profile. • The user can view and edit their personal information, such as their name, email address, and phone number. • The user can view and edit their medical information, such as their medical history and any relevant allergies. • The user can view their previous diagnosis and treatment history. • The user can view their appointment schedule with healthcare professionals, if applicable. • The user can choose to opt in or opt out of notifications related to their health and appointments.
Post-conditions	The user has updated their personal and medical information, as needed.
Alternate/Extended Flow	<ul style="list-style-type: none"> • If the user encounters technical issues with the user profile interface, they can contact technical support for assistance. • If the user wants to delete their account and all associated data, they can request to do so through the app.

Model-5

Title	Recommended Doctors/hospital
Primary Actors	Patients
Secondary Actors	Doctor, Nurses
Stake Holders	Management, Development team
Goal:	To recommend healthcare professionals and hospitals to users based on their diagnosis.
Pre-conditions	The user has received a diagnosis of pneumonia through the Pneumonia Detector app.
Main Success	<ul style="list-style-type: none"> • The Pneumonia Detector app analyzes the user's diagnosis and

Scenario:	<p>recommends healthcare professionals and hospitals in their area with expertise in treating pneumonia.</p> <ul style="list-style-type: none"> • The app displays a list of recommended healthcare professionals and hospitals with their contact information, location, and ratings. • The user can select a healthcare professional or hospital from the list and view more information about them. • The user can contact the healthcare professional or hospital to schedule an appointment. • The user can view their appointment schedule within the app.
Post-conditions	The user has scheduled an appointment with a recommended healthcare professional or hospital.
Alternate/Extended Flow	<ul style="list-style-type: none"> • If the user does not want to use the recommended healthcare professionals or hospitals, they can choose to search for their own providers within the app. • If the recommended healthcare professional or hospital is not available, the user can search for other options within the app. • If the user encounters technical issues with the recommendation or scheduling interface, they can contact technical support for assistance.

Model-6

Title	Medicine purchase /Payment method
Primary Actors	Patients
Secondary Actors	Doctor, Nurses
Stake Holders	Management, Development team
Goal:	To purchase medicine and pay for it through the Pneumonia Detector app.
Pre-conditions	The user has received a prescription for medicine through the Pneumonia Detector app.
Main Success	<ul style="list-style-type: none"> • The Pneumonia Detector app analyzes the user's diagnosis and

Scenario:	<p>generates a prescription for the required medication.</p> <ul style="list-style-type: none"> • The app displays a list of pharmacies or online medicine delivery services in the user's area that have the medication in stock. • The user selects a pharmacy or online medicine delivery service from the list. • The app displays the total cost of the medication and the payment options available. • The user selects a payment method and enters their payment information. • The app processes the payment and sends a confirmation to the user. • The user receives the medication from the selected pharmacy or delivery service.
Post-conditions	The user has received the required medication.
Alternate/Extended Flow	<ul style="list-style-type: none"> • If the selected pharmacy does not have the requested medicine in stock, the user may need to choose a different pharmacy or medicine. • If the payment is declined, the user will be prompted to update their payment method or choose a different payment method. • If the user needs to cancel or modify their order, they can do so within the app. • If the user encounters technical issues with the purchase or payment interface, they can contact technical support for assistance.

3.2. Use Case Descriptions

Sign Up Use Case:

- The user opens the Pneumonia Detector app and selects the sign-up option.
- The user is prompted to enter their personal details, such as name, email address, and password.
- The user agrees to the app's terms and conditions.
- The user submits the sign-up form.
- The app validates the user's inputs and sends a verification email to the user's email address.
- The user verifies their email address by clicking on the verification link.
- The app confirms the verification and redirects the user to the login page.

Login Use Case:

- The user opens the Pneumonia Detector app and selects the login option.
- The user enters their email address and password.
- The app validates the user's credentials.
- If the credentials are valid, the app grants access to the user's account.
- The user is redirected to the home screen of the app.
- The Pneumonia Detector app allows users to sign up and create an account, which they can then use to log in to the app and access its features. Once logged in, users can use the app to detect pneumonia in chest X-rays and receive an instant diagnosis. The app also allows users to track their diagnosis history and access educational resources related to pneumonia.

Pneumonia Detection Use Case:

- The user opens the Pneumonia Detector app and logs in to their account.
- The user selects the "Detect Pneumonia" option from the app's home screen.
- The app prompts the user to upload a chest X-ray image from their device.
- The user selects an X-ray image from their device and uploads it to the app.

- The app analyzes the X-ray image and uses a machine learning algorithm to detect pneumonia.
- The app displays the results of the analysis to the user, indicating whether pneumonia is present or not.
- If pneumonia is detected, the app provides additional information and resources to the user, such as treatment options and recommended medical professionals.
- The user can choose to save the analysis results to their account for future reference.
- This use case allows users to detect pneumonia using the Pneumonia Detector app's machine learning algorithm. Users can upload a chest X-ray image to the app and receive an instant diagnosis, along with additional information and resources to help them manage the condition if it is detected. The app can also store analysis results for future reference, allowing users to track their diagnosis history and monitor their health over time.

Chatbot Use Case:

- The user opens the Pneumonia Detector app and logs in to their account.
- The user selects the "Chatbot" option from the app's home screen.
- The app prompts the user to type in their question or concern about pneumonia.
- The user types in their question or concern, and the app's chatbot provides a response.
- The chatbot can provide information about pneumonia, such as its causes, symptoms, and treatment options.
- The chatbot can also answer questions about the Pneumonia Detector app's features and how to use them.
- If the chatbot is unable to provide a satisfactory response, it can escalate the user's question to a human customer support representative.
- The user can end the chat session at any time by typing "exit" or a similar command.
- This use case allows users to get answers to their questions and concerns about pneumonia and the Pneumonia Detector app through the app's chatbot feature. Users can type in their questions or concerns, and the chatbot will provide a response with relevant information or instructions. The chatbot can also escalate complex or unresolved issues to a human customer support representative for further assistance. The chat session can be ended by the user at any time, giving them control over the interaction.

User Profile Use Case:

- The user opens the Pneumonia Detector app and logs in to their account.

- The user selects the "Profile" option from the app's home screen.
- The app displays the user's profile information, such as their name, email address, and profile picture.
- The user can edit their profile information by selecting the "Edit Profile" button.
- The user can update their profile picture, name, and contact information.
- The user can view their diagnosis history, which includes the results of any previous pneumonia detection tests performed using the app.
- The user can access educational resources related to pneumonia and its treatment options.
- The user can delete their account by selecting the "Delete Account" button.
- This use case allows users to manage their profile information and view their diagnosis history and educational resources related to pneumonia. Users can update their profile information, such as their contact information and profile picture, and delete their account if they no longer wish to use the app. The diagnosis history feature allows users to track their pneumonia detection test results over time, while the educational resources provide information about pneumonia and its treatment options. Overall, the User Profile feature gives users control over their account and provides valuable information to help them manage their health.

Recommended Doctors/Hospital Use Case:

- The user opens the Pneumonia Detector app and logs in to their account.
- The user selects the "Recommended Doctors/Hospital" option from the app's home screen.
- The app displays a list of recommended doctors and hospitals in the user's area who specialize in pneumonia treatment.
- The user can filter the list of recommended doctors and hospitals based on various criteria, such as location, specialty, and insurance accepted.
- The user can view detailed profiles of each recommended doctor or hospital, including their contact information, office hours, and patient reviews.
- The user can schedule an appointment with a recommended doctor or hospital directly from the app by selecting the "Schedule Appointment" button.
- The app can also provide directions to the recommended doctor or hospital via a map and GPS navigation.
- This use case allows users to find and schedule appointments with recommended doctors and hospitals who specialize in pneumonia treatment. The app provides a list of recommended

doctors and hospitals in the user's area, which the user can filter based on various criteria to find the best fit for their needs. Users can view detailed profiles of each recommended doctor or hospital, including their contact information, office hours, and patient reviews. Users can also schedule appointments directly from the app, saving time and hassle. Finally, the app can provide directions to the recommended doctor or hospital via a map and GPS navigation, making it easy for users to find their way to their appointment. Overall, this feature helps users to find and access high-quality medical care for pneumonia treatment.

Medicine Purchase/Payment Method Use Case:

- The user opens the Pneumonia Detector app and logs in to their account.
- The user selects the "Medicine Purchase" option from the app's home screen.
- The app displays a list of medicines commonly used to treat pneumonia.
- The user can select the medicine they need and add it to their cart.
- The user can view their cart and adjust the quantity of medicine they wish to purchase.
- The user can select their preferred payment method, such as credit card or PayPal, and enter their payment information.
- The app securely processes the payment and sends a confirmation of the purchase to the user's email address.
- The user can track the status of their order and receive updates on its delivery status.
- This use case allows users to purchase medicine to treat pneumonia directly from the app. The app provides a list of medicines commonly used to treat pneumonia, which the user can add to their cart and purchase with their preferred payment method. The app securely processes the payment and sends a confirmation of the purchase to the user's email address. Users can also track the status of their order and receive updates on its delivery status. Overall, this feature makes it easy and convenient for users to obtain the medicine they need to manage their pneumonia symptoms.

Chapter 4

System Design

4.1. Architecture Diagram

From the architecture point of view, maybe the most specific feature of this Application Therefore, the Application uses only internal data, which are construct both with internal information both Application and website from the logic. This is emphasizing by the following explanations and diagrams.

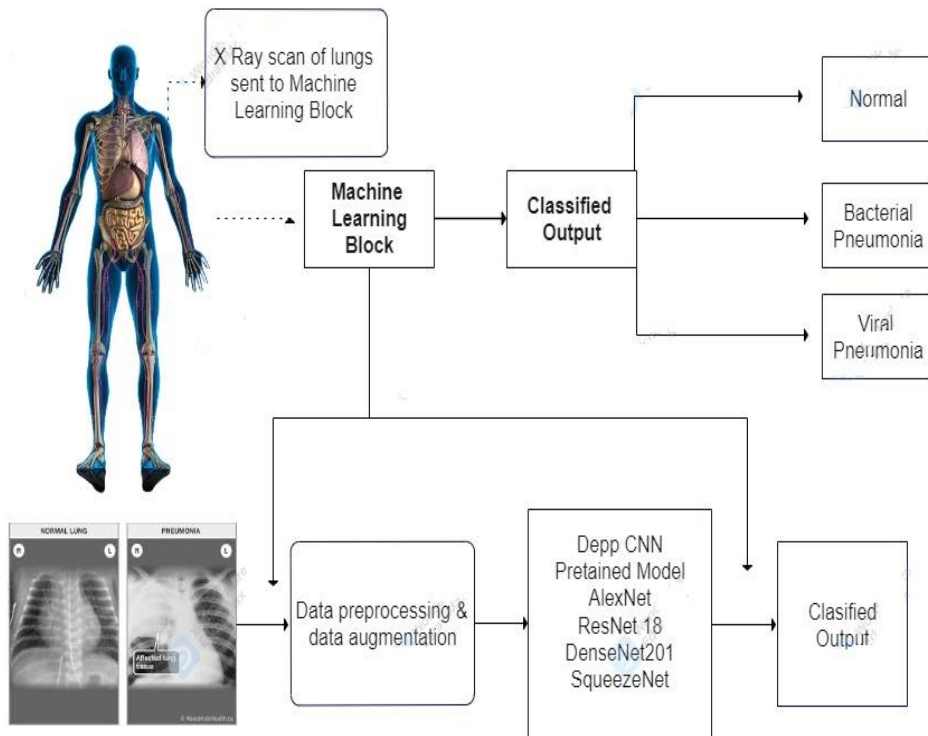


Figure.3 diagram Architecture Diagram

4.2. Domain Model

The domain model is the collection of all of the objects in that system. This domain contains all of the data and a representation of how that data behaves and interacts with each other.

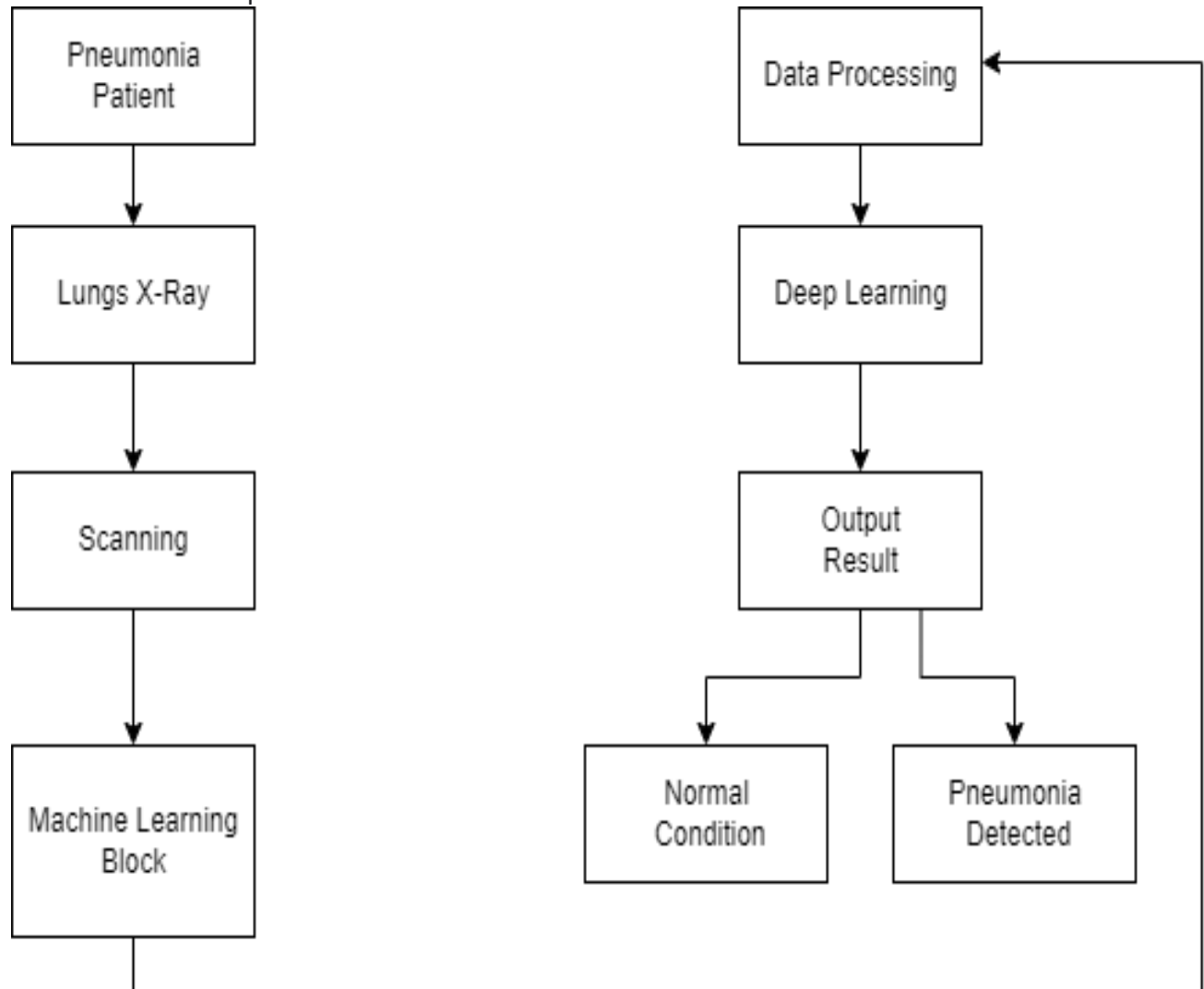
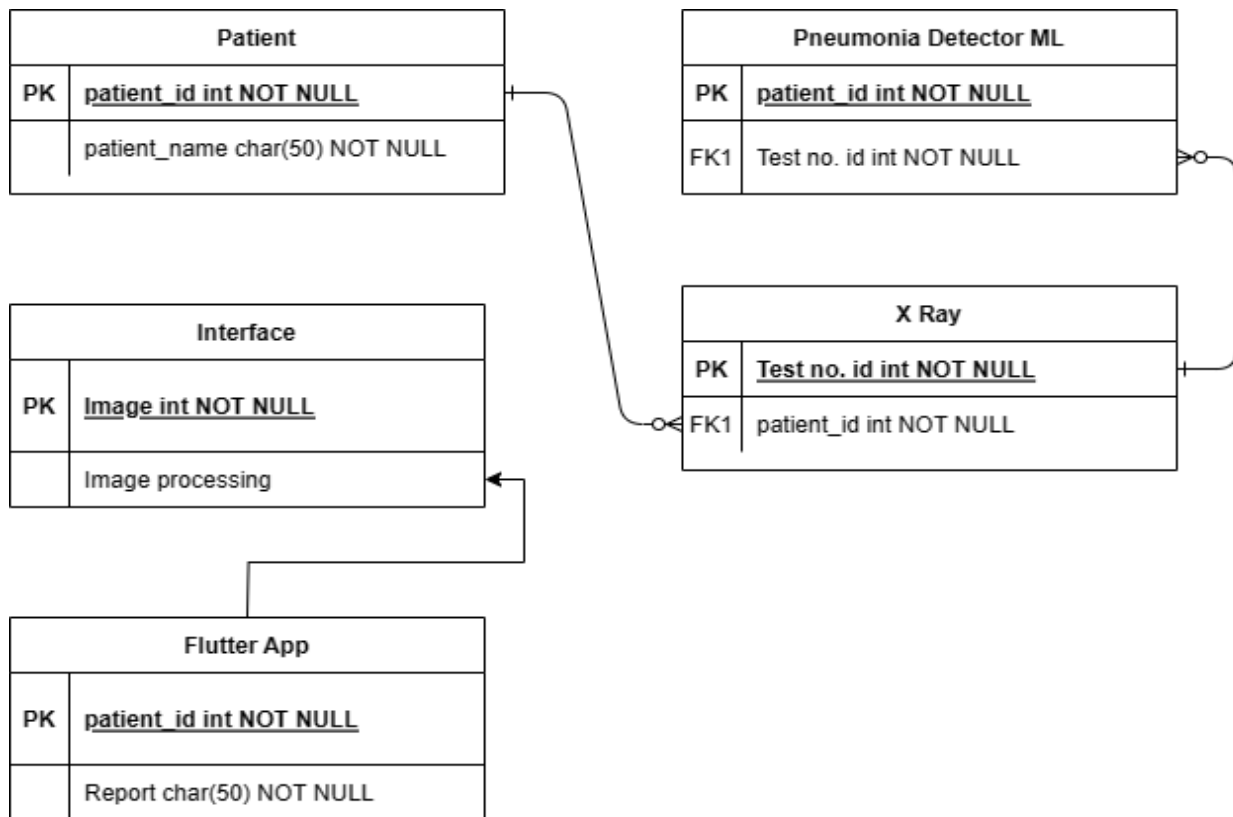


Figure.3 Domain Model

ERD



User Interface: Represents the users of the Flutter app who interact with the Pneumonia Detector system.

X-Ray Image: Represents the X-ray images that are used for detecting pneumonia. Each X-ray image has attributes such as an identifier, image file, and timestamp.

Pneumonia ML Report: Represents the reports generated by the machine learning (deep learning) model, indicating whether pneumonia is detected or not for a specific X-ray image. It includes attributes like an identifier, report result (positive/negative), timestamp, and any additional relevant details.

Now let's define the relationships between these entities:

A User can upload multiple X-ray Images (one-to-many relationship).

Each X-ray Image is associated with one User (many-to-one relationship).

Each X-ray Image generates one Pneumonia Report (one-to-one relationship).

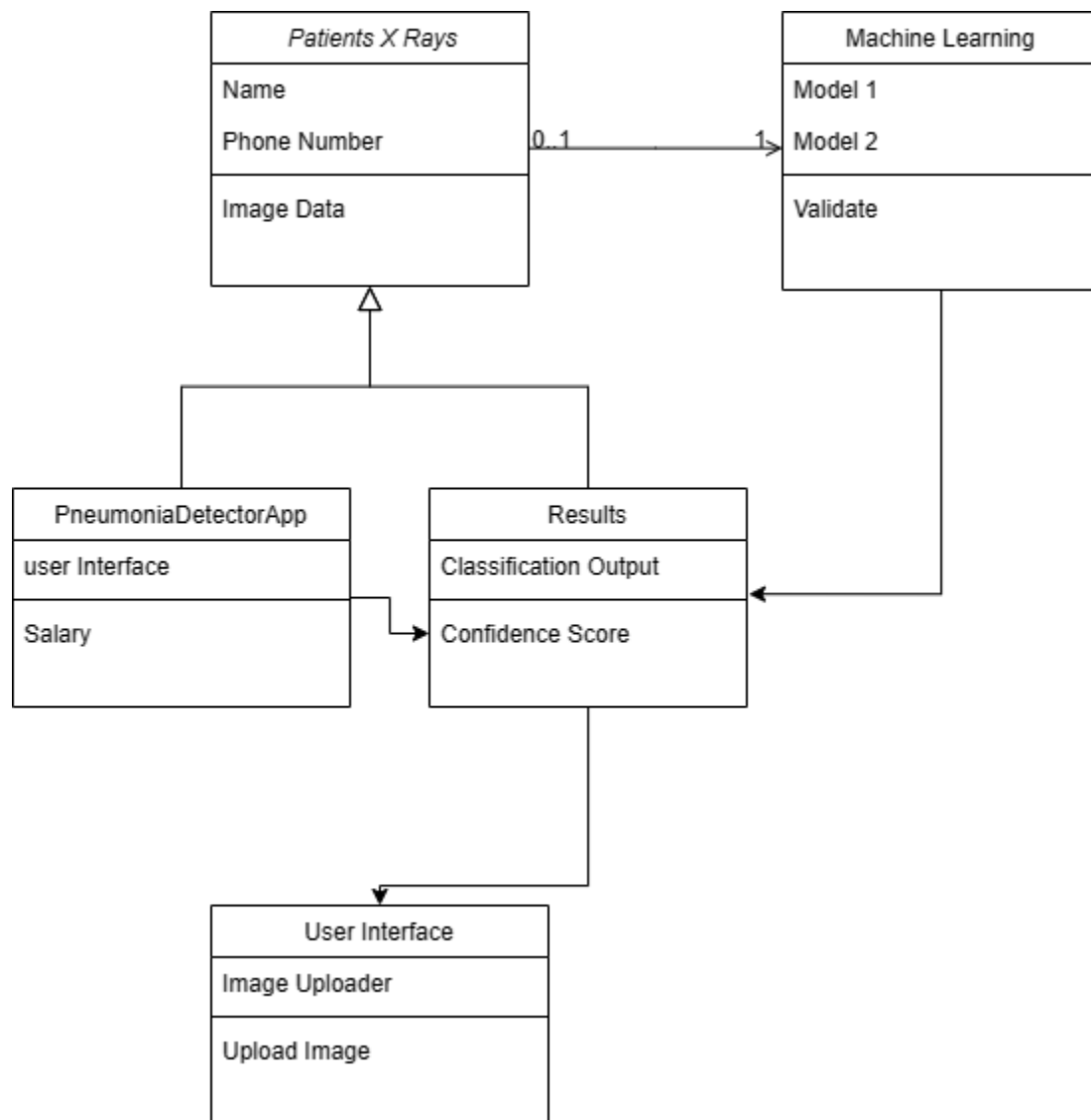
Each Pneumonia Report is linked to one X-ray Image (one-to-one relationship).

The Patient entity now includes additional attributes such as name, ID which provide more information about the user interacting with the system.

Pneumonia detector is introduced to represent the machine learning model used for pneumonia detection. It has attributes such as **model_id**, **name**, **description**, **accuracy**, and **training_date** to store information about the model.

The **Pneumonia Report** entity remains the same, but is now associated with the **Model** entity through a one-to-many relationship. This allows you to track which model was used to generate each report.

Class Diagram



Each class is represented by a box, which includes the class name at the top.

Class attributes are listed below the class name.

The relationships between the classes are depicted using lines with arrows indicating the direction of the relationship.

The relationships include:

User has a one-to-many association with X-Ray Image.

X-Ray Image has a one-to-one association with Pneumonia Report.

Pneumonia Report has a many-to-one association with X-Ray Image and Model.

Model and Label have no direct associations with other classes in this diagram.

Note that this class diagram represents the relationships between classes based on the expanded ERD. You can further extend or modify the diagram based on your specific requirements and the methods and behaviors associated with each class in your system.

4.3. Methodology

CNN models have been created from scratch and trained on Chest X-Ray Images (Pneumonia) dataset on Kaggle. Keras neural network library with Google colab back end has been used to implement the models. Dataset consists of 5216 training images, 624 testing images and 16 validation images. Data augmentation has been applied to achieve better results from the dataset. The four models have been trained on the training dataset, each with different number of convolutional layers. Each model was trained for 20 epochs, with training and testing batch sizes of 32 and 1, respectively.

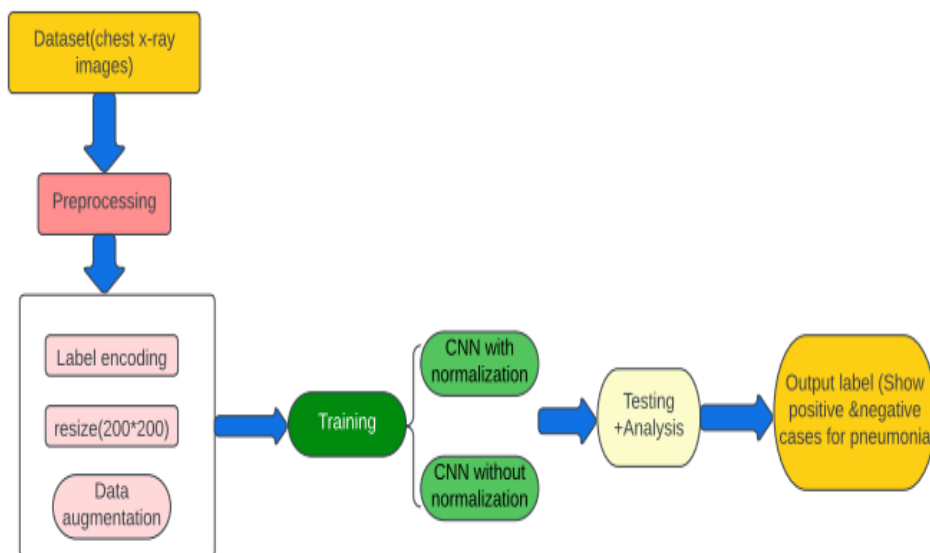


Figure.4 Methodology

4.4. flowchart Diagram

The following Flowchart Diagram of our project gives an overview of pneumonia detector by displaying how to operation operations. The diagram will help in developing an understanding of app what different transformations take place on data and what are final outputs generated by these transformations.

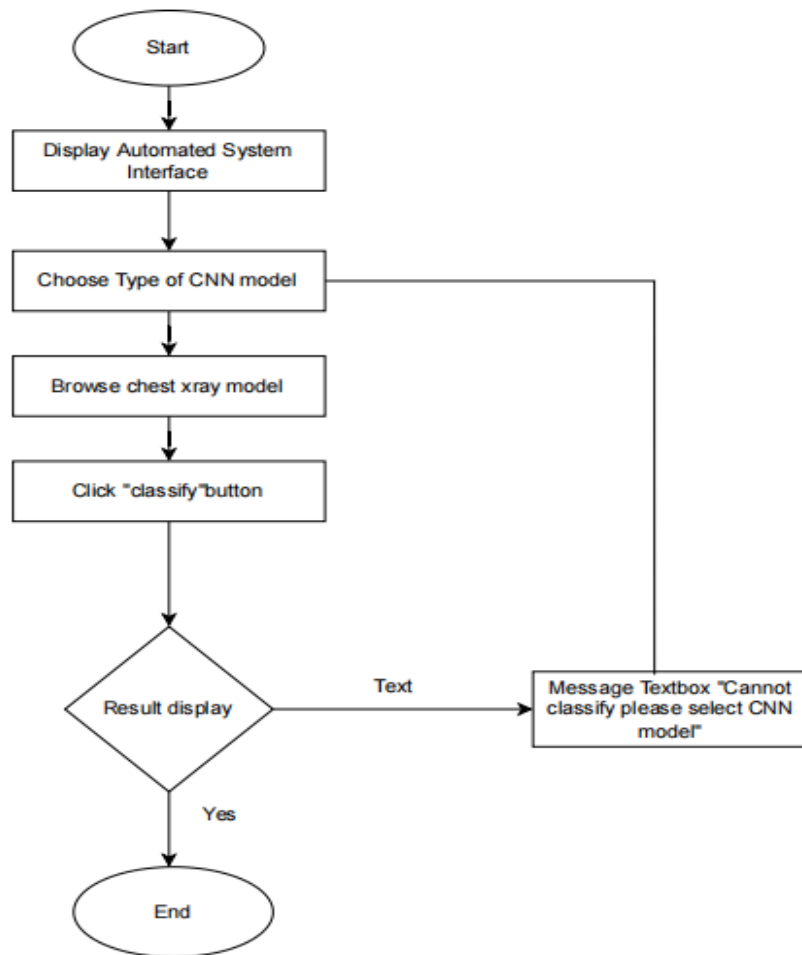


Figure.5 Flowchart Diagram

4.5. Sequence / Collaboration Diagram

The following Sequence diagram show the result how to detect the pneumonia patient through the x-ray detect pneumonia germs and classifiers data are used to obtain final output use the Machine Learning Model

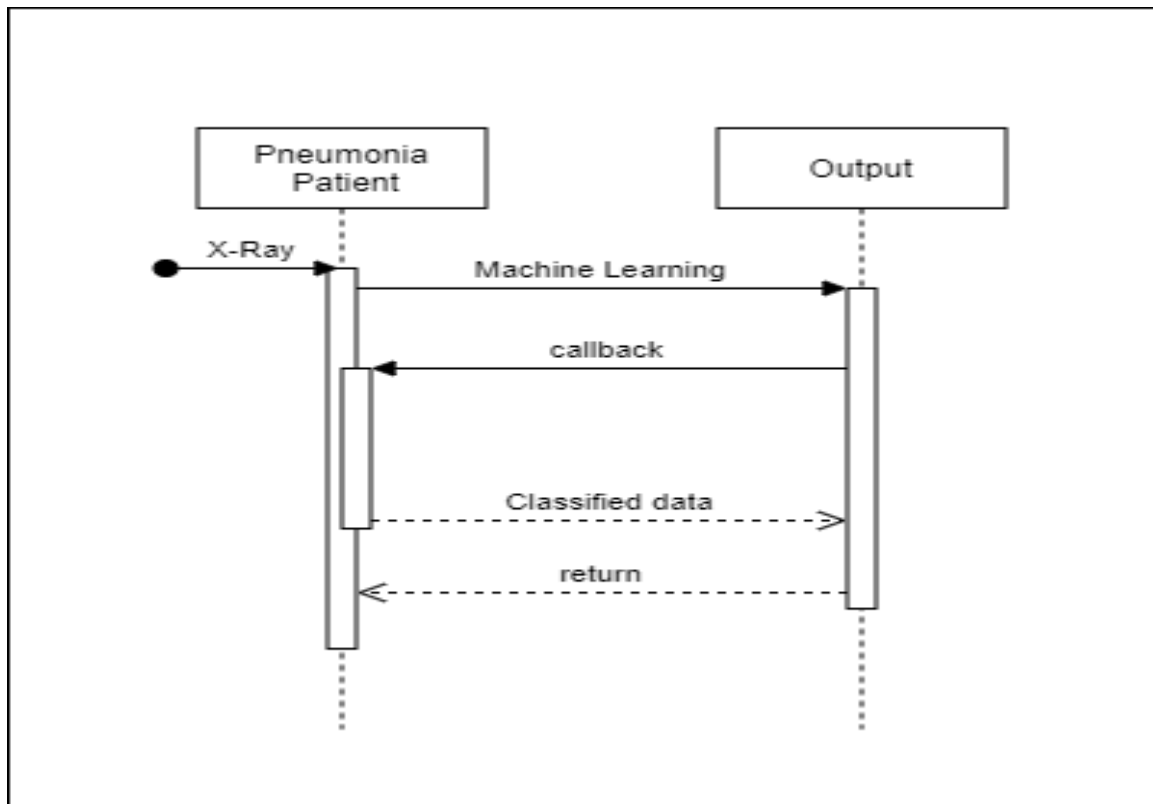


Figure.6 Sequence Collaboration Diagram

4.6. Operation contracts

We have described -

Use Cases –

Domain Model –

Sequence Diagrams

Now we describe Operation Contracts | After wards, we go into the Design Model Design
Operation Contracts | Use Cases often fully describe the behavior of a system | But they may not
be enough | Operation Contracts describe how the internal state of the concepts in the Domain
Model may change | Operation Contracts are described in terms of preconditions and post
conditions

4.7. Activity Diagram

The following **activity diagram** visually presents a series of major actions and the flow of control of our Website. It explains the dynamic behavior of our website.

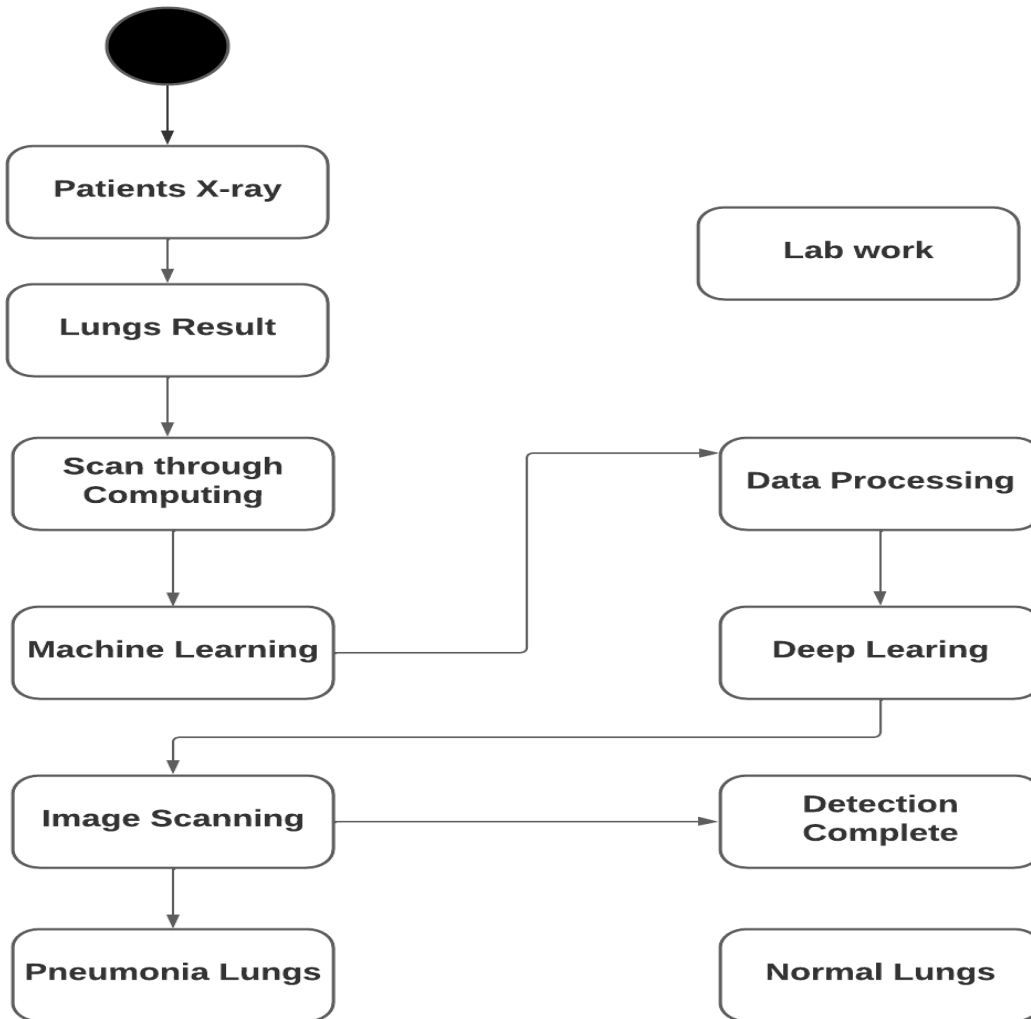


Figure.7 Activity Diagram

4.8. State Transition Diagram

The following State Transition Diagram explains the states of the Pneumonia detector. The user will upload an Image via a webpage and after submitting the image the image will go to the Flask server where our trained model will be used to make predictions. After the predictions are made the response will be sent back to the webpage via the server.

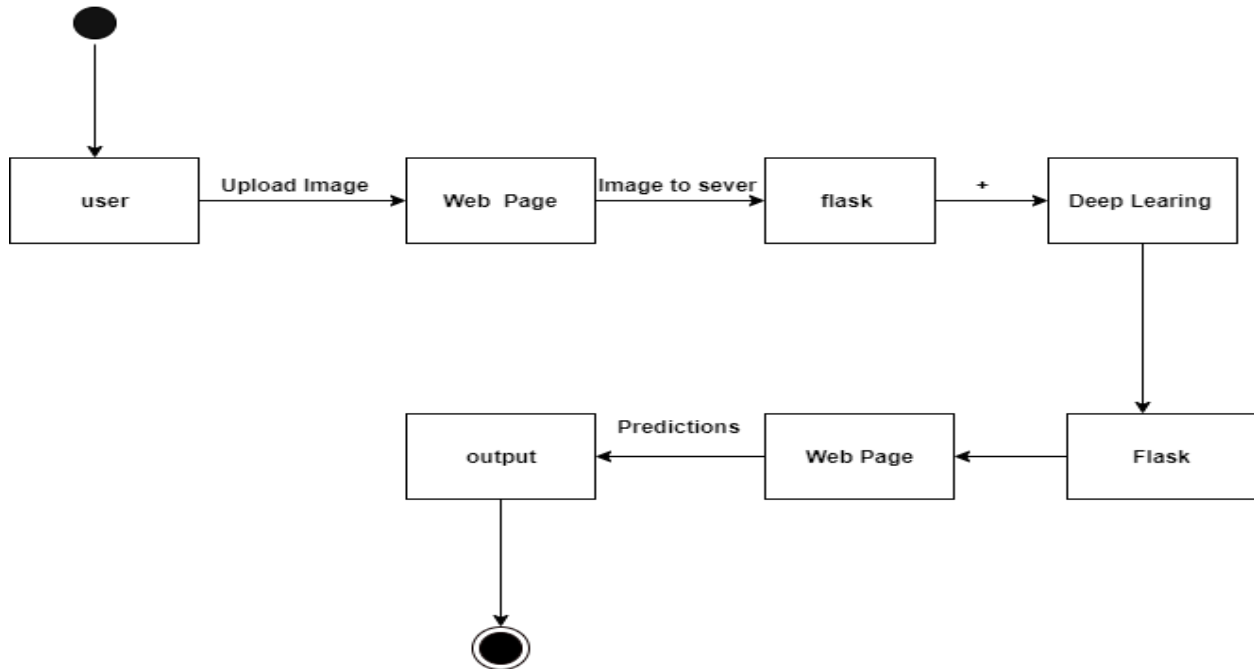


Figure.8 State Transition

4.9. Component Diagram

The following Component Diagram shows the dependencies and interactions between hardware and software components used in our project Pneumonia detector. Every component represents a modular part of our app, for example, includes the libraries files for the train, the hardware used, the setups used, etc.

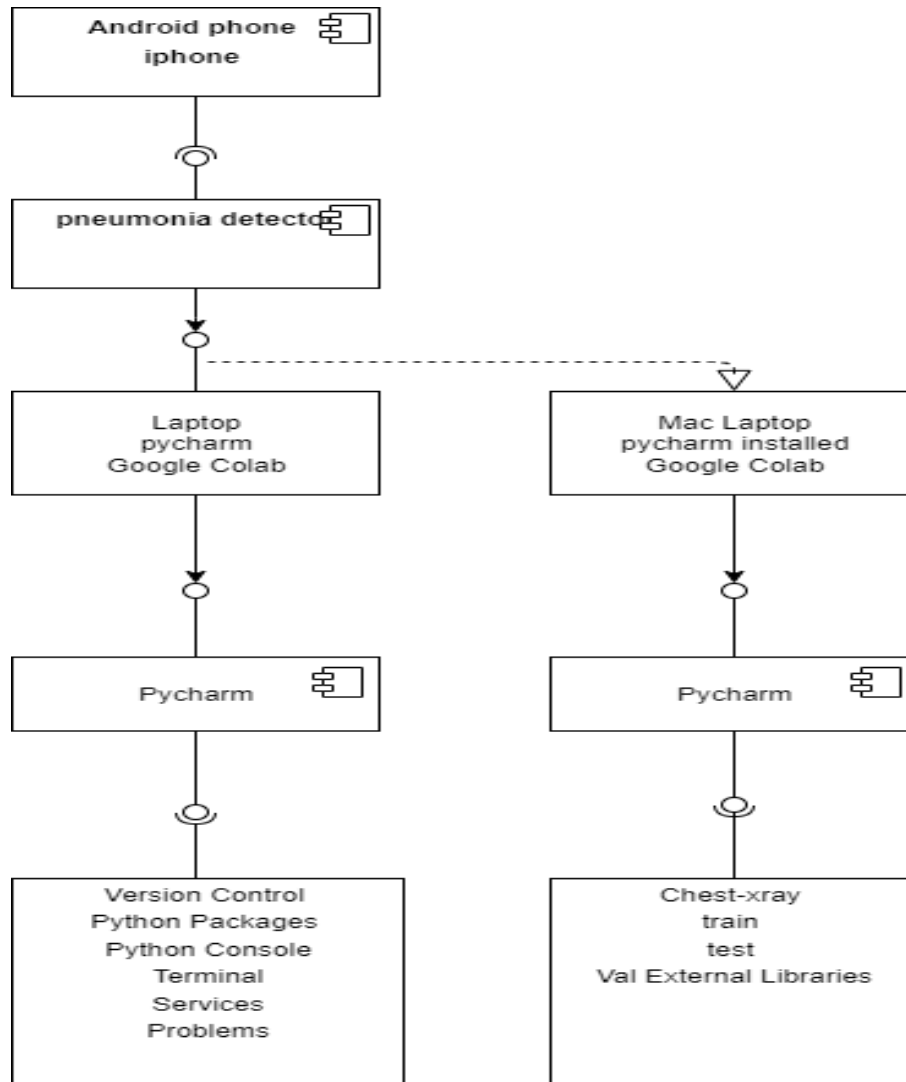


Figure.9 Component Diagram

4.10. Deployment Diagram

The following diagram shows the structure of our Application project in its deployed environment i.e. the physical deployment of the components. CNN models trained on a large dataset such as ImageNet, which consists of more than 14 million images, are frequently used for biomedical image classification tasks.

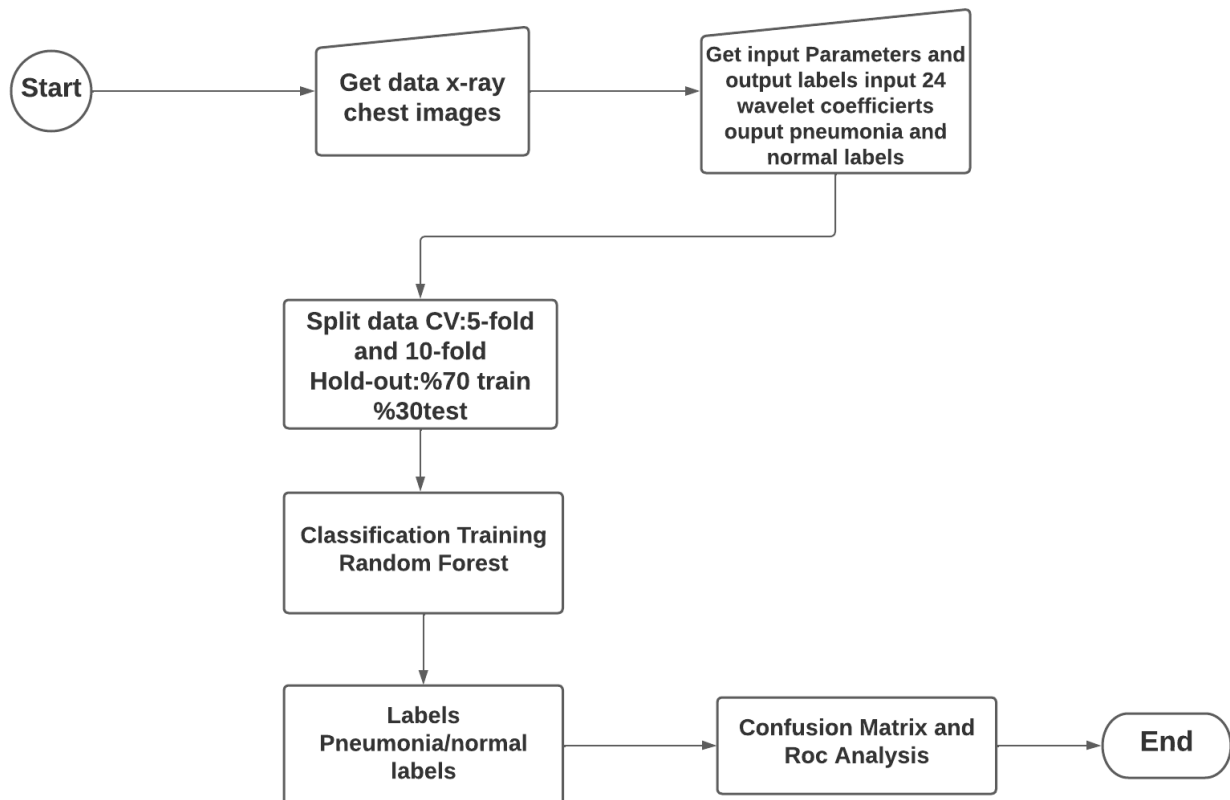


Figure.10 Deployment Diagram

Chapter 5

Implementation

Chapter 5: Implementation

Implementing a Pneumonia Detector web app and a Flutter mobile app involves different steps and technologies.

Here is an overview of what is involved in implementing both:

Pneumonia Detector Python Web App:

web development framework: Python has several web development frameworks such as Flask. the one that best suits the project requirements.

Develop the backend: Create a server-side application that handles requests from the client, processes data, and returns a response. Use Python libraries for data processing and analysis such as NumPy, Pandas to develop the pneumonia detection algorithm.

Design the frontend: Create a user interface that allows users to interact with the application. Use HTML, CSS, and JavaScript to create a user-friendly interface that provides a smooth user experience.

Integrate the frontend with the backend: Use web development frameworks to connect the frontend with the backend. Flask have built-in web server support that can run locally during the development process.

Deploy the web app: Once the development is complete, deploy the web app to a server. we can use cloud hosting services like Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure to deploy the app in future.

Pneumonia Detector Flutter Mobile App:

Development environment: Install the Flutter SDK and choose an IDE like Android Studio to develop the app.

Design the user interface: Use Flutter widgets and Material Design to create a visually appealing user interface.

Develop the app logic: Use Dart to write the app logic and integrate it with the pneumonia detection algorithm.

Test the app: Use the Flutter testing framework to run unit and integration tests to ensure the app works as expected.

Deploy the app: Publish the app to the App Store and Google Play Store.

Both the web app and mobile app require data management and storage, and we can use database management systems like firebase database or SQLite. we can also use third-party libraries and APIs to add features like user authentication, email notifications, and payment gateways.

WEB APP

The user will upload an Image via a webpage and after submitting the image, the image will go to the Flask server where our trained model will be used to make predictions. After the predictions are made the response will be sent back to the webpage via the server.

Initializing the Flask app and Making a Basic Template

we initialize our flask app we just write the following to create our app.

```
app = Flask(__name__,template_folder = 'templates')
```

We also tell our flask app to look for the templates folder for rendering it on the web page. Run the following

```
python app.py
```

You will get a link to <http://127.0.0.1:5000/> i.e. your localhost. Copy this and paste it on your browser. You will get web page like this.

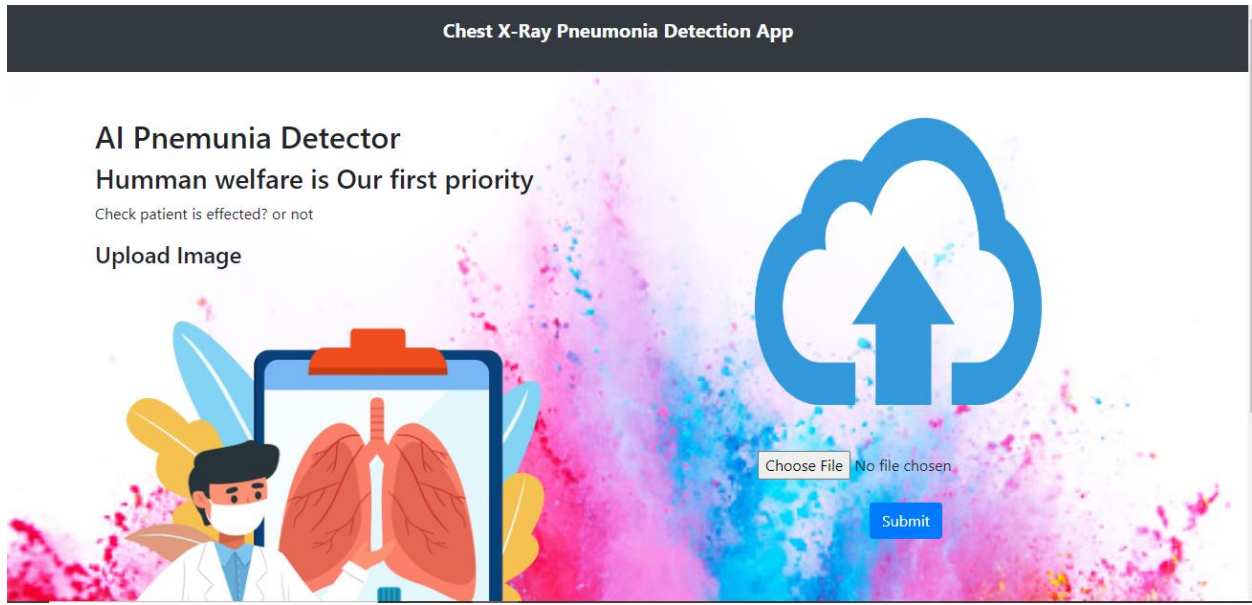
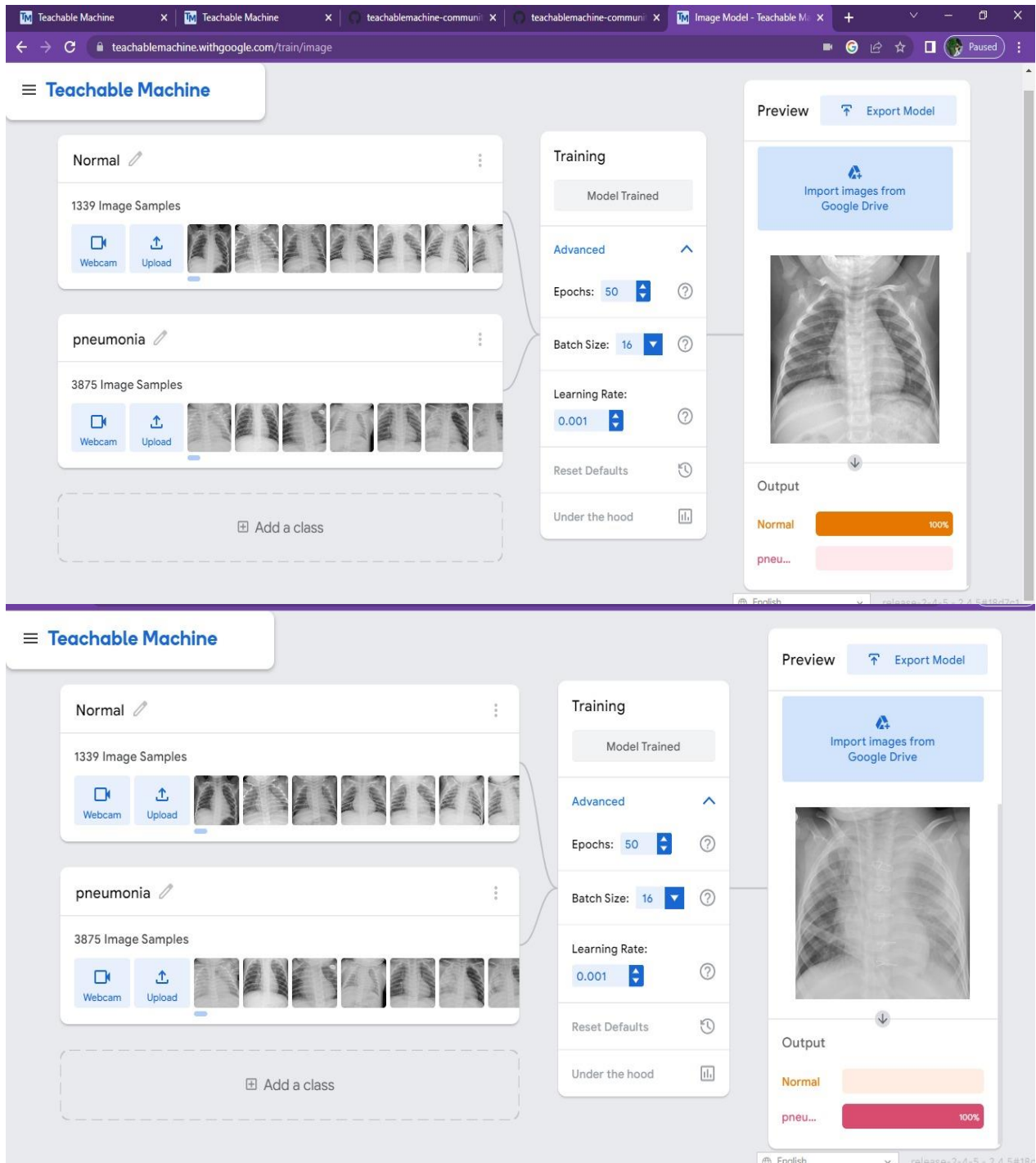


Figure 11

Mobile APP

For providing a facility to end user we also develop a mobile app in flutter by the help of teachable machine.



We train our model and test, then we download the model in tensor flow lite for android and iOS apps.

Tensorflow.js ⓘ Tensorflow ⓘ **Tensorflow Lite ⓘ**

Android Coral [Contribute on Github](#) ⓘ

For this Teachable Machine example, the *Quantized tflite* model is being used. It is using the [TFLite Android example](#), note that the example only supports models with 3 or more classes, even though the classifier itself in the example supports 2.

1. Get the Android app example from [Github](#)
2. Unpack the *converted_tflite_quantized.zip* archive exported from Teachable Machine
3. Copy *converted_tflite_quantized* folder to the example asset folder
examples/lite/examples/image_classification/android/app/src/main/assets/
4. Open [examples/lite/examples/image_classification/android/app/src/main/java/org/tensorflow/lite/examples/classific](#)
5. Modify `getModelPath()` and `getLabelPath()` to

```
@Override
protected String getModelPath() {
    return "converted_tflite_quantized/model.tflite";
}

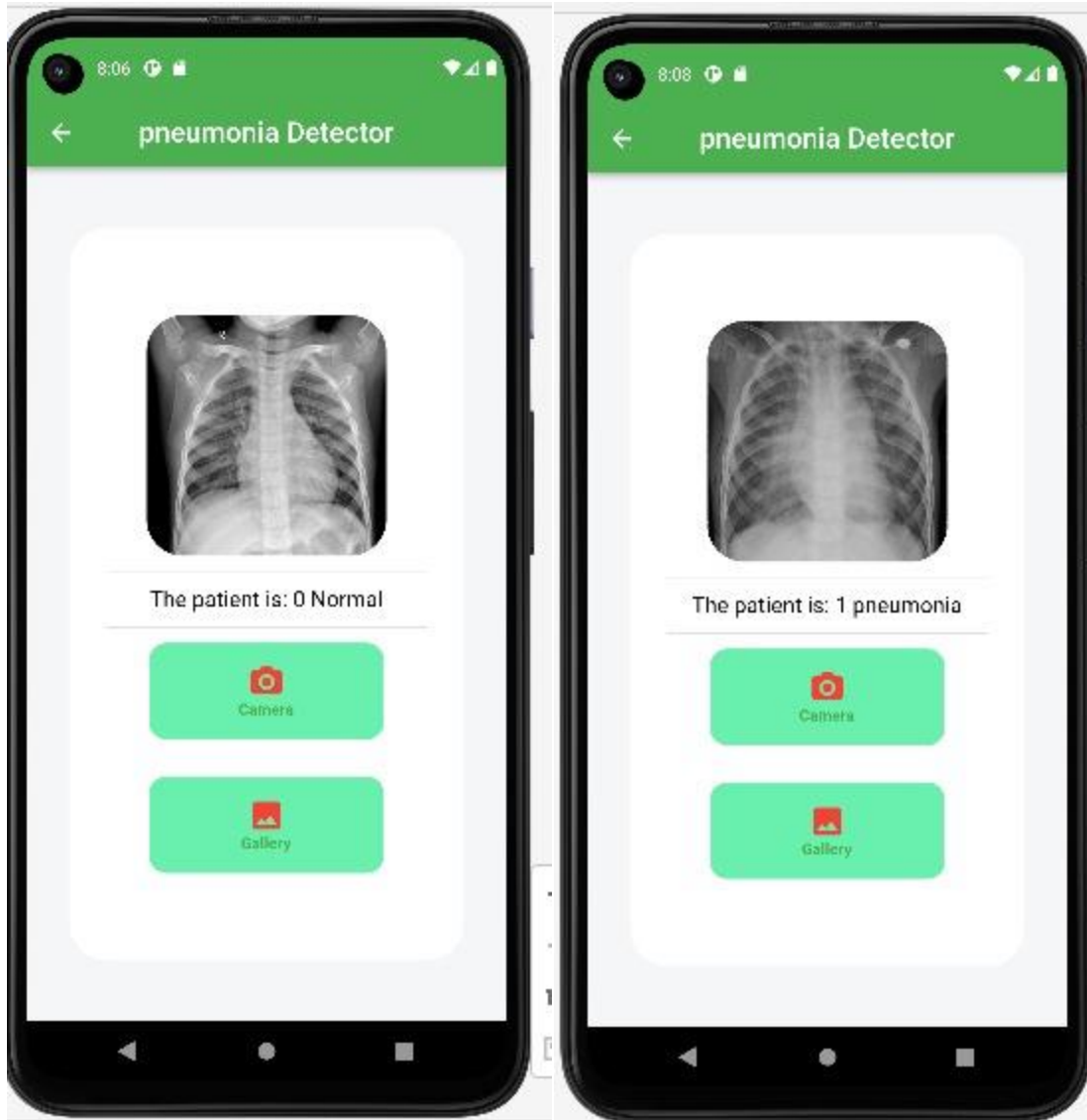
@Override
protected String getLabelPath() {
    return "converted_tflite_quantized/labels.txt";
}
```

[Copy](#) ⓘ

You can now build the app using *Android Studio* as described in the [README](#).

To enable your model in the app, switch the active model to *Quantized MobileNet*

After that we use that model in our flutter app, run app and test.



5.1. Important Flow Control/Pseudo codes

START

- When we run our app it will be show in our default browser.
- Show a button to upload images.
- User can click the button.
- Chose images from gallery.
- If images are not .png extension there will be show an error message.
- To upload an image with .png extension.

- When user upload an x-ray image.
- After processing it will be show a message healthy or pneumonic.
- If an image from effected person pneumonic message will be display.
- If an image from normal person healthy message will be display.

END

5.2. Components, Libraries, Web Services and stubs

Components

- Flask==1.1.2
- Keras-Preprocessing==1.1.2
- h5py==2.10.0
- tensorflow==2.3.1

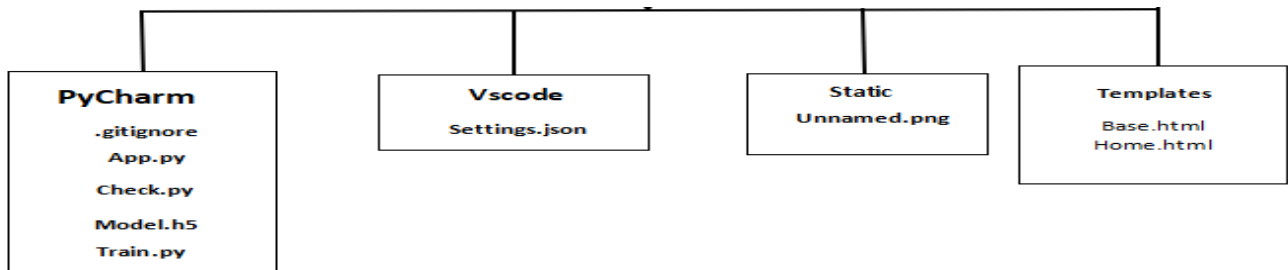


Figure1 2

Mobile App Components

```

sdk: flutter
image_picker: ^0.8.6+1
tflite: ^1.1.2
lottie: ^2.3.0
easy_splash_screen: ^1.0.4
introduction_screen: ^3.1.4
http: ^0.13.5
cupertino_icons: ^1.0.2
  
```

5.3. Deployment Environment

Since our project Pneumonia detector is a detector, hence in the deployment environment, there has been a:

a Web browser, cell phone (iPhone/ Android Phone/ Windows Phone) or a Tablet (iPad/ Android Tablet)

The detector will be open from the web link. And mobile app downloaded from play and app store.

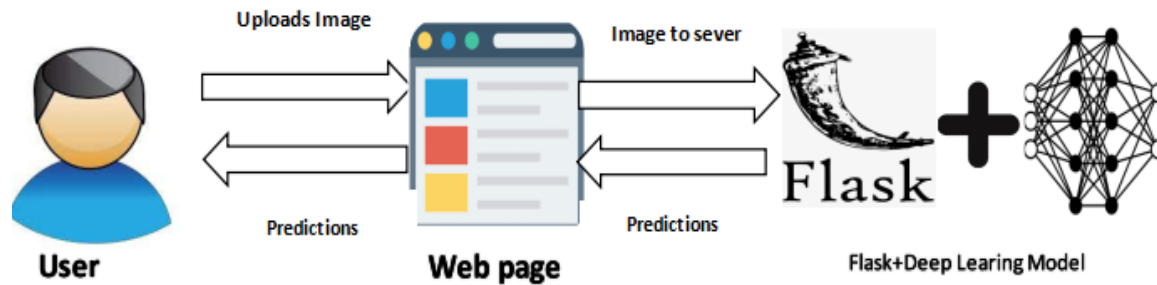


Figure 13

5.4. Tools and Techniques

Operating system	We have chosen Windows 10 operating system for its best support and user-friendliness.
Tools	Google colab, Pycharm, Android Studio.
Techniques	ML Model, Flask App, teachable machine.
Languages	Python, html/css, flutter, Dart.

5.5. Best Practices / Coding Standards

Since using **Python and html languages** for the coding of our detector in Pycharm.

Naming Conventions and Standards

Following Name Conventions and Standards are used while the coding phase of our detector.

- Pascal casing: the first character of all words is upper case and the other characters are lower case.
- Camel casing: the first character of all words, except the first word, is upper case and other characters are lower case.

Good Programming Practices

We have avoid using too large files. If a file has 300-400 lines of code, we must consider refactoring the code into helper classes. We have avoid writing very long methods. A preferred method of our detector has been of 1-60 lines of code. If we had a method more than 70 lines of code, we had must consider refactoring it into separate methods.

Comments

Not writing comments for every line of code and every variable declared and only wherever required. Good, readable code will require very few comments and that is what we will do.

Exception Handling

We will never do a "catch exception and do nothing." If we happen to hide an exception, we will never know if the exception happened or not. So, in the case of exceptions, we will give a friendly message to the user, but log the actual error with all possible details about the error, including the time it occurred, the method and class name, etc. We will only catch the specific exception, not generic exceptions.

5.6. Version Control

Version control is a method of tracking changes to documents and files to always know which version is the current iteration. By using project software with version control, or version

management, we can efficiently track and control changes to these documents directly within our software.

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

A version control system records all the changes made to a files so a specific version may be call later if needed. Current version we used python 3.10.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

6.1. Use Case Testing

Test Suite ID	TS001
Test Case ID	TC001
Test Case Summary	User must have to signup first.
Related Requirement	No
Prerequisites	No
Test Procedure	<ol style="list-style-type: none"> 1. Select chatbot button. 2. choose x-ray image from gallery. 3. get result.
Test Data	Valid png image: x-ray. Invalid picture: other extention
Expected Result	<ol style="list-style-type: none"> 1. user have normal x-ray image. 2. user have effected x-ray image.
Actual Result	<ol style="list-style-type: none"> 1. If image is valid, the result is as expected. 2. If image is not valid then invalid message displayed. 3. If fields are empty then warning message displayed.
Status	Pass
Remarks	This test case is simple and easy.
Created By	Malaika Imran
Date of Creation	04/01/19
Executed By	Malaika Imran
Date of Execution	06/06/23
Text Environment	OS: Android Studio

6.2. Equivalence partitioning

1	User name is alphabetic.	valid
2	User name is not alphabetic.	invalid
3	Password is equal to 10 or greater than 15 characters in length.	valid
4	Password is 2 to 16 characters in length.	invalid
5	Password include one uppercase letter & one lowercase letter.	valid
6	Password include ' _ '.	invalid
7	Email without '@' and '.com'.	invalid
8	Email with '@' and '.com'.	valid
9	Price only be numeric.	valid
10	Price is alphabetic.	invalid
11	Search product with alphabetic.	valid
12	Search product with special character (/,% ,^,#,@).	invalid
13	No search product name entered	invalid
14	Brand name alphabetic	valid
15	Brand name with special character	invalid
16	Empty field	invalid

6.3. Boundary value analysis

Sr.		Partition 1	Partition 2	Partition 3
1.	Password	Less than 8 character	1 – 8 character	9 – 12
2.	Phone	<=0	1 - 11	9 – 12

6.4. Data flow testing

The relationship between one entity and another while performing a specific task in during data flow. Such as between the sign up and registration process etc.

6.5. Unit testing

In unit testing we have testified our different panel codes individually by performing different tests and by executing them individually, separately on different computers and they were successfully executed and they performed well.

6.6. Integration testing

After unit testing integrate the project and test as a whole.

6.7. Performance testing

In performance testing a particular certain situation is given to the App. let's just say a 100 users try to register at the same time so how the system recovers back gracefully with complete results. So according to our extracted results, around 100+ people were able to register in our App and test their x-rays.

6.8. Stress Testing

Let's say if 100 user check at the same results at the same time which increases the stress on the app at a maximum load how will the website will perform under these circumstances.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

We developed a mobile app to detect the initial bacteria and virus of pneumonia from a patient's X-ray using deep learning techniques. Two development strategies were adopted to evaluate the performance of the deep learning models on different dataset distributions.

The dataset was divided into 80% training and 20% validation. This approach allowed the deep learning model to be exposed to a larger amount of data during optimization. By training on a larger dataset, the model had the opportunity to learn more radiological features associated with each class, thereby improving its performance in detecting pneumonia-causing bacteria and viruses.

Then a more stringent model testing was pursued. The dataset was split into 70% training, 20% validation, and 10% testing. This approach closely resembles the clinical deployment scenario, where the model is tested on unseen data that was not used during training. By evaluating the model's performance on this "unseen" dataset, it provides a more realistic assessment of how well the model can detect pneumonia-causing bacteria and viruses in real-world situations.

Both strategies utilized deep learning techniques to train and test the models. Deep learning is a subset of machine learning that focuses on training artificial neural networks with multiple layers to learn hierarchical representations of data. In the context of pneumonia detection, deep learning models can analyze the complex patterns and features present in x-ray images, enabling accurate identification of bacterial and viral infections associated with pneumonia.

7.2. Achievements and Improvements

Improved accuracy: The project successfully developed machine learning models that demonstrated improved accuracy in detecting pneumonia from x-ray images compared to traditional methods. This achievement is crucial in enhancing diagnostic capabilities and enabling more accurate and efficient detection of pneumonia.

Efficient screening process: By leveraging machine learning techniques, the project streamlined the screening process for pneumonia. Automated analysis of x-ray images using the developed models reduced the time and effort required for manual review, allowing healthcare professionals to focus on critical cases and providing timely interventions.

Early detection and intervention: The project's machine learning models potentially facilitated the early detection of pneumonia. Early identification of pneumonia-causing bacteria and viruses from x-ray images can aid in prompt intervention, leading to better patient outcomes and reducing the risk of complications associated with the condition.

Scalability and accessibility: The web or mobile app developed as part of the project increased the accessibility of pneumonia detection. With the app, healthcare providers could easily upload and analyze x-ray images, making it possible to perform pneumonia screening in various clinical settings, including remote or resource-constrained areas.

Contribution to medical research: The project's findings and the developed models could contribute to medical research in the field of pneumonia detection. The insights gained from the project's analysis of radiological features associated with pneumonia-causing bacteria and viruses could further enhance the understanding of the disease and its diagnostic patterns.

Overall, the project's achievements signify advancements in pneumonia detection through the application of machine learning, improving accuracy, efficiency, and accessibility in diagnosing and managing this respiratory infection.

Improvement of pneumonia detection

Enhanced accuracy: Continued research and development can aim to improve the accuracy of the machine learning models in detecting pneumonia. This could involve refining the algorithms, incorporating advanced deep learning techniques, and leveraging larger and more diverse datasets for training.

Robustness to different conditions: Machine learning models for pneumonia detection should be designed to be robust and effective across different imaging conditions and variations. This includes accounting for variations in image quality, different x-ray machines, and different patient populations. Ensuring the models' generalizability will enable their reliable use in various clinical settings.

Integration with existing healthcare systems: Seamless integration of the pneumonia detection models with existing healthcare systems, such as electronic health records (EHRs), can enhance the overall workflow and facilitate efficient diagnosis. This integration can automate the process of analyzing x-ray images, provide timely alerts to clinicians, and contribute to better patient care.

Real-time detection and telemedicine: With advancements in technology, future improvements could involve developing machine learning models that can perform real-time pneumonia detection. This would be particularly useful in emergency settings and telemedicine applications, where immediate diagnosis and decision-making are crucial. Real-time detection capabilities can facilitate remote consultations and enable healthcare providers to make timely decisions based on the results.

Ethical considerations: As machine learning models become more prevalent in healthcare, it is essential to address ethical considerations. Future improvements should include efforts to ensure transparency, fairness, and interpretability of the models. Research and development can focus on addressing biases, providing clear explanations for the model's decisions, and maintaining patient privacy and data security.

7.3. Critical Review

Strengths: Technological innovation: The project leverages machine learning techniques to automate the detection of pneumonia from x-ray images. This innovative approach has the potential to enhance diagnostic accuracy and efficiency in identifying pneumonia cases.

Potential impact on healthcare: By streamlining the screening process and enabling early detection, the project's outcomes have the potential to improve patient outcomes and reduce the risk of complications associated with pneumonia. The development of a web or mobile app also increases accessibility to pneumonia detection in various healthcare settings.

Ethical considerations: The critical review could delve deeper into the ethical considerations associated with using machine learning models for pneumonia detection. Addressing issues such as bias in the dataset, model transparency, and data privacy is crucial to ensure the responsible and ethical deployment of the developed technology.

7.4. Lessons Learnt

Machine learning in healthcare: The project highlights the potential of machine learning techniques in improving diagnostic capabilities and streamlining healthcare processes. It demonstrates the value of leveraging advanced technologies to enhance medical decision-making and patient care.

These lessons can inform future projects in the field of medical diagnostics using machine learning, guiding researchers and developers towards more robust, reliable, and ethically sound applications that have a positive impact on patient care.

7.5. Future Enhancements/Recommendations

Firstly, more advanced deep learning techniques can be explored to enhance the accuracy and performance of the models. Increasing the size and diversity of the dataset used for training is also important to make the models more reliable and applicable to different patient populations. Collaboration with other research groups and external validation of the models can provide additional insights and credibility. Integrating the models into existing clinical decision support systems can assist healthcare professionals in real-time pneumonia detection and improve patient outcomes.

Continuous refinement and updates of the models should be conducted to keep up with evolving bacteria and viruses. Ethical considerations, such as fairness, transparency, and data privacy, should be prioritized. Finally, assessing the clinical impact of the models in real-world settings is crucial to determine their effectiveness.

By considering these future enhancements and recommendations, the pneumonia detection project can continue to advance and contribute to the field of medical diagnostics, ultimately benefiting patients and healthcare professionals in the detection and management of pneumonia.

Appendices

Appendix A: Information / Promotional Material

A.1. Broacher

Pneumonia detector

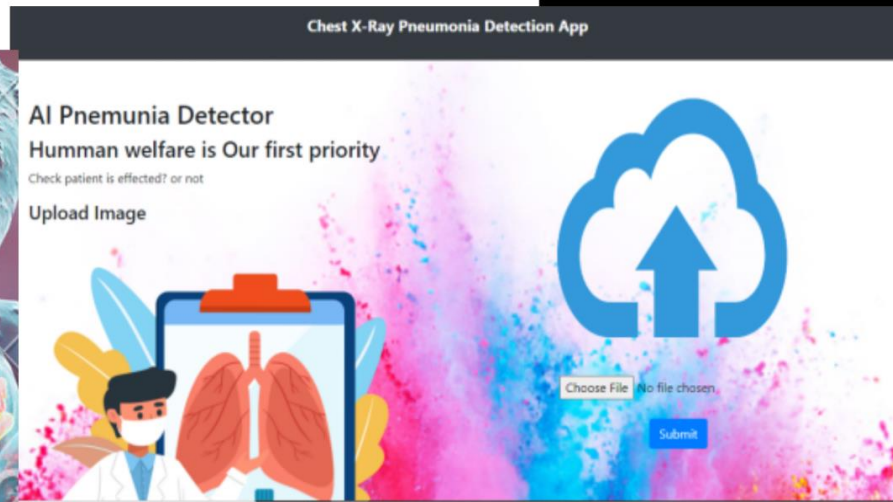
Human Welfare is first priority

Introduction

Pneumonia is a form of acute respiratory infection that affects the lungs. The lungs are made up of small sacs called alveoli, which fill with air when a healthy person breathes. When an individual has pneumonia, the alveoli are filled with pus and fluid, which makes breathing painful and limits oxygen intake.

Human Welfare is our first priority

Human Welfare is our first priority



Contact Us

☎ 03349884345
03413408062
03134781633

✉ amgb922@gmail.com

🏠 Gold Campus Raiwind Road
Lahore

Proposed Solution

Develop a web or mobile app to detect the initial bacteria and virus of pneumonia from patient's x-ray.

Reference and Bibliography

Reference and Bibliography

- [1] Dataset[<https://www.kaggle.com/general/260690>]
- [2] <https://medium.com/camping-with-python/build-a-web-app-to-detect-pneumonia-from-chest-x-ray-images-part-2-building-the-web-app-f26fed2176b6>
- [3] *teachable machine*[<https://teachablemachine.withgoogle.com/train/image>]

Index

Index

[A]

Assumptions and Dependencies	15
Architecture Diagram	27
Activity Diagram	33

[B]

Background	3
------------	---

[C]

Component Diagram	35
Comments	41

[D]

Document conventions	12
Design and Implementation Constraints	14
Description and Priorities	17
Deployment Diagram	36
Domain Model	28
Deployment Environment	39

[E]

Executive Summary	vii
External Interface Requirement	15

[F]		
Functional Requirement		20
[G]		
Goals and Objectives		3
Gap Analysis		4
Gantt Chart		9
[H]		
Hardware Interfaces		16
[I]		
Introduction		3
[J]		
[K]		
[L]		
[M]		
Motivations and Challenges		3
[N]		

[O]

Overall Description	16
Operating Environment	14
Other Non-Functional Requirements	20

[P]

Proposed solution	4
Project Plan	5
Purpose	14
Product Scope	15
Product Perspective	16
Performance Requirements	26
Pseudo Code	37

[Q]**[R]**

Roles and Responsibilities Matrix	7
Reference and Bibliography	46