“X-raying the Threat: Detecting Covid-19 using Chest X-ray”

Mini Project submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (AI&ML)

Under the esteemed guidance of

K. Padmaja
Sr. Assistant Professor, CSE(AI&ML) Department

Submitted by

JUVVAGANI HARITEJ 21R11A66B5

POOJALA TANISHQ 21R11A66C9

SANGAM ARYAVEER 21R11A66D6

Department of Computer Science and Engineering CSE(AI&ML)
Accredited by NBA

Geethanjali College of Engineering and Technology
(UGC Autonomous)
(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.

2024-2025

Geethanjali College of Engineering & Technology
(UGC Autonomous)
(Affiliated to JNTUH, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara(M), Medchal Dist.-501 301.
Department of Computer Science and Engineering(AI&ML)

CERTIFICATE

This is to certify that the Mini Project Report entitled **"X-Raying the Threat: Detecting Covid-19 using Chest X-ray"** is a bonafide work done and submitted by

JUVVAGANI HARITEJ 21R11A66B5
POOJALA TANISHQ 21R11A66C9
SANGAM ARYAVEER 21R11A66D6

during the academic year **2024 - 2025**, in partial fulfilment of requirement for the award of Bachelor of Technology degree in **"Computer Science and Engineering (AI&ML)"** from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to my best of the knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

Project Guide
K. Padmaja
Sr. Assistant Professor CSE(AI&ML)

Project Co-Ordinator
Mr. Shaik Akbar
Associate Professor CSE(AI&ML)

Head of the Department CSE(AI&ML)
Dr. G. Bindu Madhavi

External Examiner

DECLARATION

We hereby declare that the Mini Project report entitled **"X-raying the Threat: Detecting Covid-19 using Chest X-ray"** is an original work done and submitted to **Computer Science and Engineering (AI&ML)** Department, **Geethanjali College of Engineering & Technology**, affiliated to Jawaharlal Nehru Technological University Hyderabad, in partial fulfilment of therequirement for the award of Bachelor of Technology in Computer Science and Engineering (AI&ML) and it is a record of bonafide project work carried out by us under the guidance of **K. Padmaja** , Sr. Assistant Professor, Department of Computer Science and Engineering (AI&ML).

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma.

Signature of the Student
Juvvagani Haritej
21R11A66B5

Signature of the Student
Poojala Tanishq
21R11A66C9

Signature of the Student
Sangam Aryaveer

21R11A66D6

ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and foremost, we thank the Chairman, Principal, Vice Principal for availing infrastructural facilities to complete the major project in time.

We offer our sincere gratitude to our Project guide **K. Padmaja**, Sr. Assistant Professor, Computer Science and Engineering (AI&ML) Department, Geethanjali College of Engineering & Technology for her immense support, timely co- operation and valuable advicethroughout the course of our Mini Project work.

We would like to thank the Project Co-Ordinator, **Mr. Shaik Akbar**, Associate Professor, Computer Science and Engineering (AI&ML) for his valuable suggestions in implementing the project.

We would like to thank the Head of Department CSE(AI&ML), **Dr. K. Bindu Madhavi,** for her meticulous care and cooperation throughout the project work.

JUVVAGANI HARITEJ 21R11A66B5
POOJALA TANISHQ 21R11A66C9
SANGAM ARYAVEER 21R11A66D6

ABSTRACT

The COVID-19 pandemic has had a dreadful impact on public health, economy and social life since 2019. The Polymerase Chain Reaction (PCR) tests are widely used to detect COVID-19 due to their high success rate in detection. However, there are many challenges associated with this detection method, such as it relies on human involvement, which can increase the chances of COVID-19 spread. To overcome the above problem, our project focused on detecting COVID-19 from Chest X-rays using deep learning techniques. This project involved collecting and curating the datasets from COVID-19 positive and negative cases. Deep learning models like Convolutional Neural Networks were trained on this dataset to identify the abnormal patterns in chest X-ray images which provided valuable information in detecting the disease. These findings helped the model to early detect the COVID-19 by overcoming the limitations of using Polymerase Chain Reaction tests and contribute to fight against COVID-19.

Keywords: Polymerase Chain reaction (PCR), Convolutional Neural Networks (CNN), Deep learning, Neural Networks.

LIST OF FIGURES

TABLE OF CONTENTS

1. INTRODUCTION

1.1 About Covid-19

COVID-19, also called coronavirus disease 2019, is an illness caused by a virus. The virus is called severe acute respiratory syndrome coronavirus 2, or more commonly, SARS-CoV-2. It started spreading at the end of 2019 and became a pandemic disease in 2020.

During the initial outbreak in Wuhan, the virus and disease were commonly referred to as "coronavirus" and "Wuhan coronavirus", with the disease sometimes called "Wuhan pneumonia" . In January 2020, the World Health Organization (WHO) recommended 2019-nCoV and 2019-nCoV acute respiratory disease as interim names for the virus and disease per 2015 guidance and international guidelines against using geographical locations or groups of people in disease and virus names to prevent social stigma. The official names COVID-19 and SARS-CoV-2 were issued by the WHO on 11 February 2020 with COVID-19 being shorthand for "coronavirus disease 2019".The WHO additionally uses "the COVID-19 virus" and "the virus responsible for COVID-19" in public communications.

The virus that causes COVID-19 spreads most commonly through the air in tiny droplets of fluid between people in close contact. Many people with COVID-19 have no symptoms or mild illness. But for older adults and people with certain medical conditions, COVID-19 can lead to the need for care in the hospital or death.

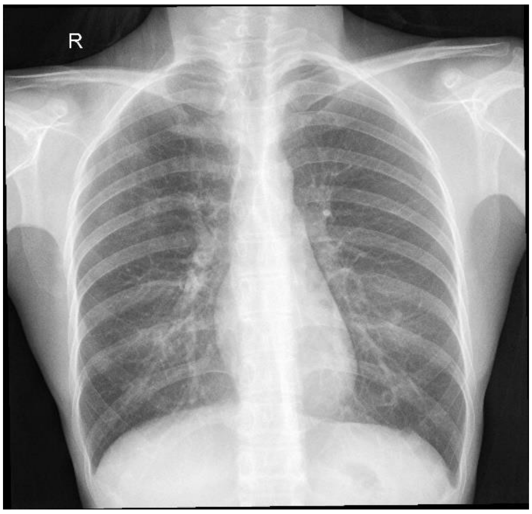

Figure 1.1(a) Chest X-ray of Covid Negative

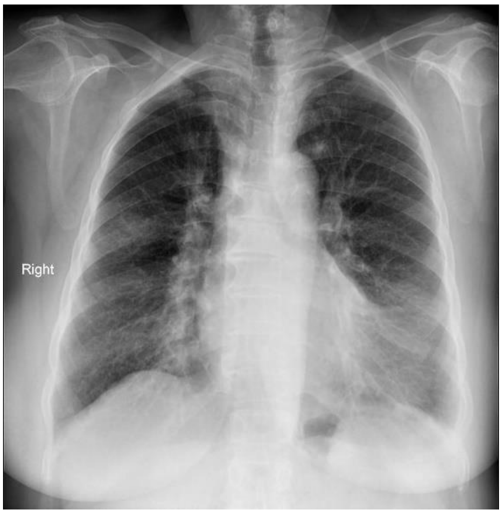

Figure 1.1(b) Chest X-ray of Covid Positive

1.2 Causes of Covid-19

1. **Transmission**

   COVID-19 is mainly transmitted when people breathe in air contaminated by droplets/aerosols and small airborne particles containing the virus. Infected people exhale those particles as they breathe, talk, cough, sneeze, or sing. Transmission is more likely the closer people are. However, infection can occur over longer distances, particularly indoors. The transmission of the virus is carried out through virus-laden fluid particles, or droplets, which are created in the respiratory tract, and they are expelled by the mouth and the nose.

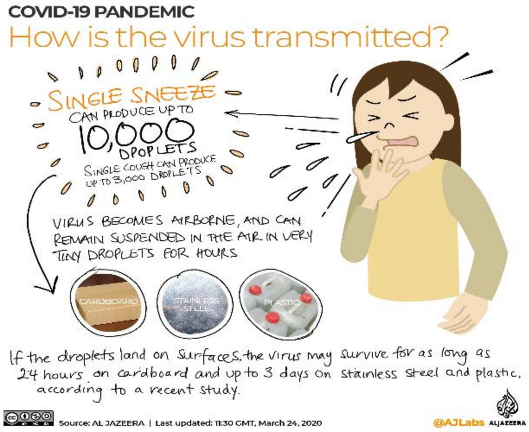

Figure 1.2.1 Transmission of Covid-19

2. **SARS-CoV-2**

   Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is a novel severe acute respiratory syndrome coronavirus. It is a strain of coronavirus that causes COVID-19, the respiratory illness responsible for the COVID-19 pandemic.

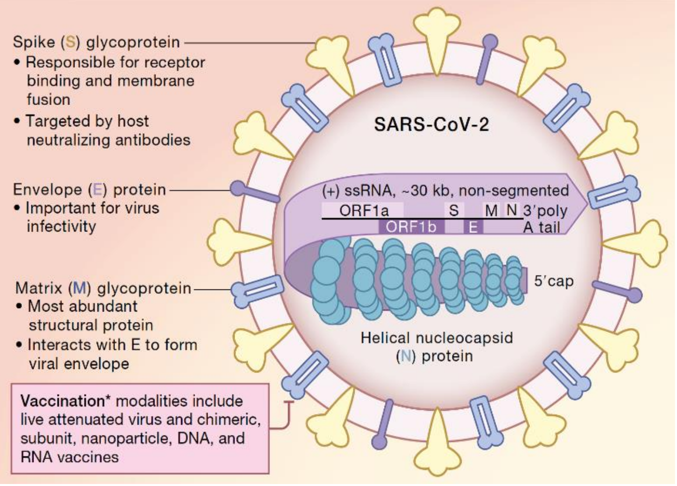

Figure 1.2.2 SARS-CoV-2

1.3 Risk Factors

1. **Age**

   People age 65 and older and babies younger than 6 months have a higher than average risk of serious COVID-19 illness. Those age groups have the highest risk of needing hospital care for COVID-19. Babies younger than 6 months aren't eligible for the COVID-19 vaccine, which adds to their risk. For older people, the challenge is that the immune system is less able to clear out germs as people age. Also, as people age, medical conditions that raise the risk of severe COVID-19 are more likely. In the U.S. as of March 2024, about 76% of all deaths from COVID-19 have been among people age 65 and older.

2. **COPD Lung Disease**

   Having moderate to severe asthma raises some risks of serious COVID-19 illness. It raises the risk of needing care in the hospital, including intensive care, and needing mechanical help breathing. The risk of serious COVID-19 illness also is higher for people who have conditions that damage lung tissue over time. Examples are tuberculosis, cystic fibrosis, interstitial lung disease, bronchiectasis or COPD, which stands for chronic obstructive pulmonary disease. These diseases raise the risk of needing care in the hospital for COVID-19. Depending on the condition, the risk of needing intensive care and the risk of death from COVID-19 also may go up.

3. **Cancer**

   In general, people with cancer have a greater risk of getting serious COVID-19. People who have or had blood cancer may have a higher risk of being sick for longer, or getting sicker, with COVID-19 than people with solid tumors. Having cancer raises the risk of needing care in the hospital, intensive care and the use of breathing support. Having blood cancer and getting COVID-19 raises the risk of death from the illness.

4. **Other Conditions**

   If an organ or body system is already weakened by disease, infection with the COVID-19 virus can cause further damage. In other cases, medicine for the original condition can lower the immune system's response to the virus that causes COVID-19. Many different diseases can raise the risk of severe COVID-19 illness such as brain and nervous system diseases.

1.4 Symptoms of Covid-19

COVID-19 affects different people in different ways. Most infected people will develop mild to moderate illness and recover without hospitalization. Typical COVID-19 symptoms often show up 2 to 14 days after contact with the virus.

Symptoms can include:

- **Dry cough.**
- **Shortness of breath.**
- **Loss of taste or smell.**
- **Extreme tiredness, called fatigue.**
- **Digestive symptoms such as upset stomach, vomiting or loose stools, called diarrhea.**
- **Pain, such as headaches and body or muscle aches.**
- **Cold-like symptoms such as congestion, runny nose or sore throat.**
- **Fever or chills.**

Fig. 1.4.1 symptoms of Covid-19

1.5 Existing System

1. **Polymerase Chain Reaction Tests (PCR)**

   The polymerase chain reaction (PCR) test for COVID-19 is a molecular test that analyzes your upper respiratory specimen, looking for genetic material (ribonucleic acid or RNA) of SARS-CoV-2, the virus that causes COVID-19. Scientists use the PCR technology to amplify small amounts of RNA from specimens into deoxyribonucleic acid (DNA), which is replicated until SARS-CoV-2 is detectable if present. The PCR test has been the gold standard test for diagnosing COVID-19 since authorized for use in February 2020. It's accurate and reliable.

2. **Antigen Tests**

   An antigen is a substance that can trigger an immune response when present in the body. It may be a virus, bacterium, toxin, chemical, or other substance from outside of the body. Tissues and cells in the body also contain antigens that can cause immune responses. An antigen test detects antigens in a sample taken from the body. These antigens could be the proteins that make up a virus, as with SARS-CoV-2, which is the virus that causes COVID-19.

3. **Antibody Tests**

   Antibodies are developed by the body in response to an infection or after vaccination. SARS-CoV-2 is the virus that causes COVID-19. SARS-CoV-2 antibody tests detect antibodies to the SARS-CoV-2 virus. SARS-CoV-2 antibody tests can help identify people who may have been infected with the SARS-CoV-2 virus or have recovered from COVID-19.

4. **Saliva Tests**

   Saliva tests typically require patients to spit into a tube, making them far less invasive than the current nose and throat swab collection methods for COVID-19.

1.6 Limitations of Existing Systems

1. **Lack of Specialized Expertise**

   PCR testing depends on the availability of skilled personnel and sophisticated lab Equipment. The procedure to process samples and interpret results is complex and Requires expertise, which may cause delays in testing, particularly in under-resourced Areas or during periods of high demand. This reliance on specialized skills limits its rapid deployment for large-scale testing.

2. **Restricted Access to Testing Facilities**

   Antigen tests are quicker and more cost-effective but still dependent on healthcare centers or clinics for administration. In remote or underprivileged regions, the limited availability of testing services restricts timely access to COVID-19 detection, potentially hindering efforts to control outbreaks in such areas.

3. **Concerns Around Data Protection**

   For antibody testing, when results are stored or shared via digital health platforms or databases, the privacy and security of patient information become a significant concern. Handling sensitive data, such as test results and medical history, requires strong safeguards to ensure compliance with privacy laws and protect against potential breaches.

4. **Compatibility Issues**

   Other alternative detection methods, for example saliva-based tests will also not be easy to integrate in well -structured and well-set health care delivery systems. Such results produced by these methods may have to be delivered and processed easily across different platforms of healthcare service delivery. This may prove very challenging since some problems with data compatibility could limit integration and effective management of public health information.

5. **Testing Accessibility and Convenience**

   Many testing methods require individuals to visit healthcare facilities, which can be a barrier for those with mobility issues, lack of transportation, or fear of exposure to the virus in clinical settings. This can lead to delays in testing and treatment.

6. **Cost Implications**

   While some tests are cheaper than others, the overall cost can still be a barrier for many individuals and healthcare systems, particularly when considering the need for repeated testing. This economic factor can hinder access to timely diagnosis and treatment.

7. **Latency Periods**

   For antibody tests, there is often a delay in antibody production after infection. This latency means that individuals who have recently been infected may test negative, leading to missed cases and potentially allowing the virus to spread undetected.

8. **Public Perception and MisInformation**

   The public's understanding of different testing methods can influence their willingness to get tested. Misinformation about the reliability and purpose of certain tests such as antibody tests, can lead to hesitance in seeking testing or following the public health guidelines.

9. **Concerns Over Test Sensitivity**

   Antigen testing, despite its speed and affordability, often lacks the high accuracy seen with PCR tests. This can result in false negatives, particularly in people with no symptoms or early-stage infections. While suitable for mass testing scenarios, lower sensitivity levels may cause missed cases, delaying treatment and increasing the risk of community transmission.

10. **Limited Scope of Detection**

    Many tests are designed to detect only SARS-CoV-2. If a patient is co-infected with another virus, such as the flu, the test may not provide comprehensive information about their overall health, limiting the utility of the results.

1.7 Proposed System

The proposed system seeks to enhance the detection of COVID-19 through the analysis of chest X-ray images, focusing on improving diagnostic accuracy and facilitating timely treatment. This innovative approach employs Deep Convolutional Neural Networks (DCNNs) to extract critical features from the X-ray images, allowing for effective classification of COVID-19 cases. A tailored preprocessing pipeline is implemented to optimize image quality and ensure consistent input for the model.

By leveraging DCNNs, the system can differentiate between COVID-19 positive patients and healthy individuals, enabling early identification of the disease. Such early differentiation lead to better health outcomes and play a very crucial role in more successful public health interventions, especially in primary healthcare settings where the use of traditional testing means is unavailable due to remoteness. Generally, it will revolutionize the diagnostics of COVID-19 and offer critically important inputs for doctors.

2. AIM and OBJECTIVES

2.1 AIM of proposed System

The primary aim of the proposed system is to address the urgent need for effective COVID-19 detection through the analysis of chest X-ray images. This system seeks to enhance the diagnostic process by leveraging deep learning technology to identify the presence of COVID-19 in patients quickly and accurately.

By employing advanced deep learning models, the system analyzes chest X-ray images to detect characteristic patterns associated with COVID-19 infection. This enables healthcare providers to make informed decisions based on precise diagnostic results, facilitating early intervention and appropriate treatment.

The system is designed with user accessibility in mind, allowing individuals to upload their chest X-ray images for analysis. This approach empowers users to take proactive steps in managing their health and encourages timely consultations with healthcare professionals when COVID-19 is detected.

While the proposed system offers significant support in COVID-19 detection and management, it is crucial to emphasize that it does not substitute for the expertise of healthcare professionals. Engaging with medical experts remains essential for accurate diagnosis and personalized treatment planning. The system aims to complement existing healthcare services, enhancing the overall process of managing COVID-19 and ensuring that individuals receive optimal care and guidance.

2.2 Objectives of proposed system

1. **Integrate Advanced Image Analysis Techniques**

   The first goal focuses on the integration of advanced deep learning models specifically designed for analyzing chest X-ray images. These models will be trained on a diverse dataset to recognize patterns and anomalies indicative of COVID-19 infections. By harnessing the power of deep learning, the system can enhance diagnostic accuracy, reducing the chances of false negatives or positives. This not only aids in confirming the presence of the virus but also helps differentiate COVID-19 from other respiratory conditions, providing healthcare professionals with reliable data to inform their decisions.

2. **Enable Instantaneous Result Processing**

   A significant goal of the system is to implement instantaneous result processing for the submitted chest X-rays. This feature will ensure that users receive immediate feedback regarding their X-ray analysis, thereby expediting the overall diagnostic process. Quick result turnaround times are crucial in managing infectious diseases like COVID-19, as early detection can lead to timely intervention and better patient outcomes. By providing real-time results, the system aims to reduce wait times and enhance the user experience.

3. **Design an Intuitive User Interface**

   A crucial goal is to design a user-friendly interface that facilitates seamless interaction with the system. This interface will feature clear navigation pathways, X-ray image submission, and accessing diagnostic results. The design will prioritize usability, catering to individuals with varying levels of technical expertise. An intuitive user experience can significantly enhance user engagement and satisfaction, making the technology accessible to a broader audience.

4. **Encourage Early Detection and Proactive Management**

   The overarching goal is to foster early detection and proactive management of COVID-19. By successfully implementing the previous objectives, the system aims to mitigate the impact of COVID-19 through accurate diagnostics that encourage timely intervention. Early detection can lead to better health outcomes, as individuals can receive the necessary care before the virus progresses. This proactive approach is vital in managing public health during the pandemic.

5. **Deliver Comprehensive Health Insights**

   The fifth goal is to provide users with comprehensive health insights based on their reported symptoms and the analysis of their chest X-rays. This feature will include educational content about COVID-19, information on potential treatment options, and guidelines on preventive measures. By equipping users with knowledge about the virus and its implications, the system promotes informed decision-making, empowering individuals to take proactive steps in managing their health and seeking appropriate medical care when necessary.

3. LITERATURE SURVEY

3.1 Introduction

Detection of COVID-19 using chest X-ray radiography has advanced significantly with the application of deep learning techniques, enabling automated analysis for identifying COVID-19 cases from chest X-ray images. These deep learning models process image data to detect specific patterns indicative of COVID-19 infection, distinguishing them from other respiratory conditions. Despite the effectiveness of traditional machine learning methods in medical image analysis, there is significant room for improvement through the adoption of more advanced neural network architectures. Current systems can benefit from the enhanced feature extraction and classification capabilities offered by deep learning models such as convolutional neural networks (CNNs) and transfer learning approaches. By leveraging these models, automated COVID-19 detection can improve the speed and accuracy of diagnosis, reduce dependency on manual interpretation, and provide valuable insights for healthcare professionals. Deep learning architectures not only streamline the diagnostic process but also contribute to more precise identification of COVID-19 indicators in chest X-ray images. This advancement supports timely medical interventions, optimizes resource allocation, and enhances overall healthcare management during the ongoing COVID-19 pandemic.

3.2 Covid-19 Detection using various Techniques

1. **Leveraging Chest X-ray Images for COVID-19 Diagnosis**
    - In the proposed system, chest X-ray images play a crucial role in diagnosing COVID-19 by identifying abnormalities in the lungs associated with the disease.
    - Instead of relying on standard pre-trained models like VGG-16 or ResNet, our system employs a custom-built deep learning model tailored to efficiently detect COVID-19-related patterns in X-ray images.

- Advanced feature extraction techniques are used to isolate important characteristics from the images, improving the accuracy and robustness of the detection process without depending on traditional CNN architectures.

2. **Deep Learning Approaches for COVID-19 Diagnosis**
    - Our approach to deep learning involves developing a lightweight model architecture designed specifically for COVID-19 detection, which offers high accuracy while minimizing computational requirements.
    - Unlike approaches using models like AlexNet or Mask-RCNN, our system uses a streamlined neural network optimized for binary classification (COVID-19 positive or negative) based on chest X-ray scans.
    - This architecture has been rigorously trained on a curated dataset of COVID-19 chest X-rays to improve the generalization and ensure reliable predictions across different cases.

3. **Streamlined X-ray Image Submission and Processing**
    - The system includes functionality for users to submit their chest X-ray images directly through an intuitive interface.
    - Real-time image processing is implemented to ensure that users receive immediate feedback, promoting faster decision-making regarding further medical consultation or testing.
    - This feature is crucial in reducing delays in diagnosis and facilitating early intervention, which is critical for controlling the spread of COVID-19.

4. **Aras M Ismael** Aras M. Ismael and his team conducted a study focused on detecting COVID-19 using chest X-ray images, applying three different deep CNN approaches. They experimented with two transfer learning techniques: deep feature extraction and fine-tuning, alongside a newly developed end-to-end CNN model. The study also employed Support Vector Machines (SVM) with various kernel functions for classifying the deep features.
5. **Mesut Toğaçar** Mesut Toğaçar and colleagues focused on distinguishing individuals with lung damage caused by COVID-19 from those with healthy lungs or pneumonia The research employed deep learning models for COVID-19 detection, emphasizing the importance of rapid and accurate identification using AI techniques due to the global spread of the virus.
6. **Tulin Ozturk.:** In this study, he proposed a deep learning-based model aimed at detecting and classifying COVID-19 cases using chest X-ray images. The system he developed operates in an end-to-end manner, eliminating the need for manual feature extraction, making it fully automated. His model is capable of handling both binary classification (COVID-19 vs. non-COVID) and multi-class classification (COVID-19, normal, and other chest-related diseases), achieving an impressive 98.08% accuracy for binary tasks and 87.02% accuracy for multi-class tasks.
7. **Ferhat Ucar.:** Ferhat Ucar and colleagues emphasize the urgent need for rapid diagnostic methods to combat infectious diseases such as COVID-19, especially given the limitations of the widely-used RT-PCR nucleic acid-based tests. These limitations include methodological constraints, dependency on disease progression, challenges in specimen collection, and slow response times. To address these issues, the study proposes an AI-based decision-making system designed for the rapid diagnosis of COVID-19 using X-ray images

4. SOFTWARE / HARDWARE REQUIREMENT

4.1 Functional Requirements

1. **Image Upload and Analysis**
    - Users should be able to upload chest X-ray images for analysis.
    - The system will employ deep learning models to classify the uploaded images and identify COVID-19 cases.
    - No manual feature extraction is required, as the model will be fully automated with end-to-end processing
2. **Binary Classification**
    - The model will perform binary classification (COVID-19 vs. non-COVID).
    - The system should achieve high accuracy for both tasks to ensure reliable diagnosis
3. **User Friendly Application**
    - Design an intuitive and user-friendly web application for users to easily upload images and view results.
    - Ensure the application has a clean interface with simple navigation and clear instructions.
4. **Real -Time Image Processing**
    - The system should process the images quickly, providing real-time or near-real-time feedback to users.
    - Implement efficient algorithms to minimize processing time while maintaining accuracy.
5. **Model Accuracy and Robustness**
    - Regularly update the deep learning model using the latest datasets to enhance its accuracy and robustness.
    - Address the limitation of using a small dataset by incorporating larger, diverse datasets as they become available
6. **Data Privacy and Security**
    - Ensure user data, particularly X-ray images and diagnostic results, are securely stored and protected.
    - Implement data encryption and access control measures to safeguard privacy.
    - Comply with data privacy regulations to ensure the confidentiality of sensitive information

4.2 Non – Functional Requirements

1. **Performance**
   - The system must be able to process and analyze uploaded X-ray images within 10 seconds for a seamless user experience.
   - The deep learning model should maintain high accuracy, with at least 95% accuracy for binary classification.
2. **Scalability**
   - The application should support multiple users simultaneously uploading X-ray images and receiving results without performance degradation.
   - The backend should be scalable to handle increased data loads as the number of users grows.
3. **Reliability**
   - The system should provide consistent, accurate results even when handling a large volume of image uploads and predictions.
   - Implement backup and recovery mechanisms to prevent data loss..
4. **Security**
   - Implement end-to-end encryption to ensure the secure transfer of X-ray images and diagnostic results.
   - Protect the system against unauthorized access or potential security breaches to maintain data confidentiality and integrity.
5. **Portability**
   - The application should be compatible across different platforms, including desktop web browsers (Chrome, Firefox, etc.) and mobile web browsers.
   - Ensure compatibility with various operating systems such as Windows, macOS, and Linux.
6. **Extensibility**
   - The system should be designed in a way that allows for future expansions, such as adding more disease classifications (e.g., tuberculosis, lung cancer) or integrating with hospital systems.
   - The architecture should support the addition of new models or algorithms without requiring a complete system redesign.

.

4.3 Software Requirements

1. Python

Fig, 4.3.1 Python logo

Python is a widely-used high-level programming language known for its simplicity and readability. It features clean and easy-to-understand syntax, making it a favorite among developers. Python is interpreted, cross-platform, and comes with an extensive standard library, reducing the need for custom code. It supports dynamic typing and object-oriented programming principles and is highly interoperable with other languages. Python has a thriving community and offers versatility, making it suitable for web development, data analysis, machine learning, and more. Its open-source nature and ease of learning have contributed to its widespread adoption.

2. Tensorflow

Fig.4.3.2 Tensorflow logo

TensorFlow is an open-source machine learning framework developed by Google. It's designed to facilitate the creation, training, and deployment of machine learning models, particularly deep learning models. TensorFlow offers a flexible and scalable platform for building various AI applications, including image and speech recognition, natural language processing, and more. Its key features include a high-level API for quick model development (Keras), support for both CPU and GPU acceleration, and a large ecosystem of pre-built models and tools. TensorFlow's versatility and robust community support make it a powerful tool for machine learning and artificial intelligence projects.

3. Flask

Fig. 4.3.3 Flask logo

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

4. ReactJs

Fig. 4.3.4 ReactJs logo

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on components by Facebook Inc. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.

React can be used to develop single-page, mobile, or server-rendered applications with frameworks like Next.js. Because React is only concerned with the user interface and rendering components to the DOM, React applications often rely on libraries for routing and other clientside functionality. A key advantage of React is that it only rerenders those parts of the page that have changed, avoiding unnecessary rerendering of unchanged DOM elements. It was first launched on 29 May 2013.

5. VScode

Fig. 4.3.5 VS code logo

Visual Studio Code (VSCode) is a widely-used integrated development environment (IDE) known for its lightweight and versatile nature. It's developed by Microsoft and is favored by developers for its efficiency and extensibility. VSCode is highly customizable, allowing developers to tailor it to their specific needs with the help of numerous extensions available in its marketplace. Its key features include a powerful code editor with syntax highlighting, debugging support, and integrated version control.

6. Colab and Jupyter

Fig. 4.3.6 jupyter, colab logo

Jupyter is an open-source interactive computing environment that allows users to create and share documents containing live code, equations, visualizations, and narrative text. It supports various programming languages, including Python, R, and Julia. Jupyter notebooks, which are a key component of Jupyter, enable users to combine code execution with explanatory text, making it an excellent tool for data analysis, scientific research, and educational purposes.

Jupyter notebooks run in a web browser and can integrate with data visualization libraries, making it easier to explore and present data-driven insights.

Google Colab (Colaboratory) is a cloud-based platform provided by Google that allows users to create and run Jupyter notebooks without the need for local installation or powerful hardware. Colab offers free access to GPU and TPU resources, making it well-suited for machine learning and deep learning tasks. Users can collaborate on Colab notebooks in real-time, share them with others, and access pre-installed libraries and packages, simplifying the setup process for data analysis and machine learning projects. Colab notebooks are stored on Google Drive, ensuring easy access and version control.

7. ExpressJs

Fig. 4.3.7 ExpressJs logo

Express.js, or simply Express, is a backend web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

4.4 Hardware Requirements

- **CPU:** A multi-core processor (e.g., Intel Core i5 or better) for efficient handling of requests and serving the model.
- **RAM:** At least 8GB, preferably 16GB or more, for running the model and handling multiple requests concurrently.
- **GPU (Graphics Processing Unit):** To accelerate deep learning model training, consider using one or more GPUs. NVIDIA GPUs are commonly used for this purpose. High-end GPUs with CUDA support can significantly speed up the training of complex models.
- **Storage:** SSD for faster access to the model weights and for storing user-uploaded X-ray images. At least 256GB, but depends on how many X-ray images are processed or stored.
- **Compatibility:** Ensure that the user interface of your application is responsive and compatible with a variety of devices and screen sizes.
- **Network Bandwidth:** Adequate network bandwidth is essential, especially if your system handles a large number of image uploads and downloads. High-speed internet connectivity and sufficient internal network bandwidth are required for efficient data transfer.

4.5 Tech stack

- **Programming language** → Python
- **Development Framework** → Tensorflow
- **IDE** → VScode, jupyter
- **Data visualization** → matplotlib, seaborn
- **Operating System** → Window 11
- **Model Development** → keras
- **Interface Development** → ReactJs

5. SOFTWARE DESIGN

5.1 Process Cycle

In this application, users begin their interaction on the Home Page, where they can upload a chest X-ray image for analysis. Once the user uploads the image, it is immediately sent to the backend for further processing. Upon receiving the image, the backend performs essential preprocessing steps, such as resizing or enhancing the quality of the X-ray. After preprocessing, the image is fed into the deep learning model designed to predict the classification of the input. The model analyzes the X-ray and predicts whether the case falls under categories like Normal or Covid. Once the prediction is generated, it is sent back to the flask server. The server then communicates the predicted result to the React frontend, where the output is displayed to the user in a clear and understandable format. This entire process is streamlined to ensure a fast and efficient user experience, allowing the user to receive their diagnosis quickly and easily.

In summary, the process cycle enables the user to interact effortlessly with the system, from uploading the X-ray to receiving a diagnosis, providing a user-friendly and reliable way to detect COVID-19 using advanced AI-based models.

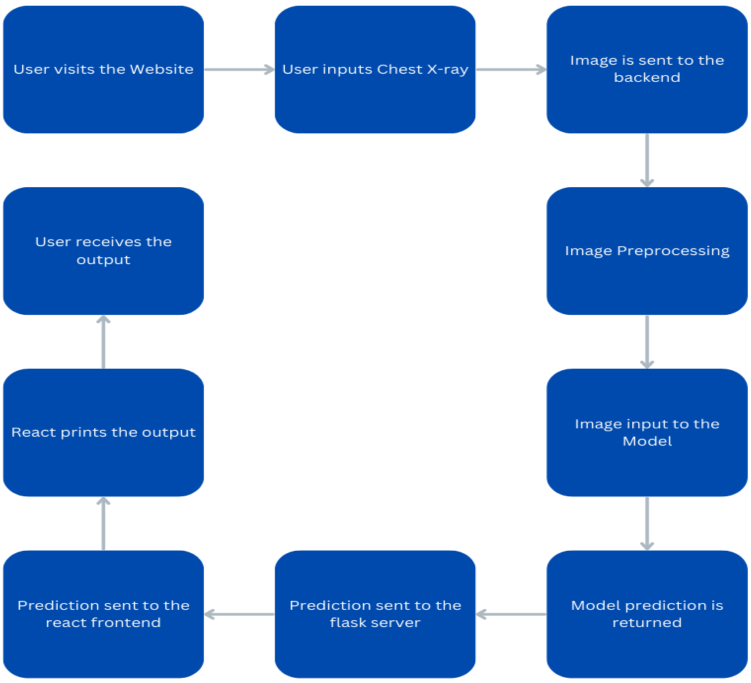

Fig. 5.1.1 Process cycle

6. SYSTEM IMPLEMENTATION

6.1 FRONTEND IMPLEMENTATION

The user interface (UI) is designed using React.js to create a Single Page Application (SPA) that ensures smooth and dynamic user interaction. The key features of the frontend include.

- **User Friendly Interface**
  The UI is responsive and intuitive, allowing users to upload chest X-ray images with minimal effort. A simple form is provided where users can select and upload X-ray images from their device. Image validation ensures that users upload files in acceptable formats (JPEG, PNG, etc.), and error messages are displayed for incorrect file type.

- **Image Upload Mechanism**
  The React frontend sends the X-ray image to the backend using HTTP POST requests through RESTful API communication. The form data includes the image and any necessary metadata. The API request is handled asynchronously, allowing the UI to remain interactive and responsive while the backend processes the image

6.2 BACKEND IMPLEMENTATION

The backend is responsible for handling the communication between the frontend, the deep learning model, and any data storage services (if needed). The backend implementation includes two primary technologies: Express.js for routing and API handling, and Flask for model inference.

- **Flask for Model Inference**

  The deep learning model is hosted on a Flask server. Flask is lightweight and ideal for running the model inferences. Flask is responsible for receiving the image from the frontend and preprocessing it. After preprocessing, the image is passed to the trained deep learning model, which analyzes the input and makes a prediction about the likelihood of COVID-19 based on the X-ray. The prediction (either Covid or Normal) is returned as a categorical output, which is passed back to the frontend.

6.3 Machine Learning Model

The deep learning model is the core of the COVID-19 detection system. It was trained on a dataset of chest X-ray images to classify them as COVID-19 positive or COVID-19 negative.

- **Model Architecture**

The model is built using a convolutional neural network (CNN) architecture, which is well-suited for image classification tasks. CNNs extract hierarchical features from the image data, which help the model identify key patterns associated with COVID-19 in X-ray images.

Model: "sequential"

Layer (type)	**Output Shape**	**Param #**
conv2d (Conv2D)	(None, 254, 254, 32)	896
conv2d_1 (Conv2D)	(None, 252, 252, 64)	18,496
max_pooling2d (MaxPooling2D)	(None, 126, 126, 64)	0
dropout (Dropout)	(None, 126, 126, 64)	0
conv2d_2 (Conv2D)	(None, 124, 124, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0
dropout_1 (Dropout)	(None, 62, 62, 64)	0
conv2d_3 (Conv2D)	(None, 60, 60, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 128)	0
dropout_2 (Dropout)	(None, 30, 30, 128)	0
flatten (Flatten)	(None, 115200)	0
dense (Dense)	(None, 64)	7,372,864
dropout_3 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Total params: 7,503,105 (28.62 MB)
Trainable params: 7,503,105 (28.62 MB)
Non-trainable params: 0 (0.00 B)

Figure 6.3.1 Model Architecture

7. TRAINING and TESTING

7.1 Training

- The training phase of the deep learning model for COVID-19 detection from chest X-rays plays a crucial role in the overall system's accuracy and reliability
- The first step in training the model was to collect a dataset of chest X-ray images labeled as COVID-19 positive and COVID-19 negative. The dataset forms the backbone of the entire training process, as the quality and variety of data directly impact the model's ability to generalize
- The dataset was compiled from public sources such as Kaggle, medical research repositories, and online image databases. These datasets provided labeled X-ray images of patients diagnosed with COVID-19, as well as those who tested negative.
- Before feeding the images into the model, preprocessing was applied to standardize the inputs. This involved resizing the images to a consistent dimension, typically 224x224 pixels, which is the standard input size for many convolutional neural network (CNN) models.
- For COVID-19 detection, a convolutional neural network (CNN) was chosen due to its strong performance in image classification tasks.
- Once the architecture was designed, the model was compiled and trained on the dataset. Since this is a binary classification task, the binary cross-entropy loss function was chosen.
- The Adam optimizer was selected due to its adaptive learning rate, which helps in faster convergence and efficient training

7.2 Model Fit and epochs

```
Epoch 1/10
C:\Users\harit\anaconda3\Lib\site-packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call `s
uper().__init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use_multiprocessing`, `max_queue_size`. Do not pass these arguments to
`fit()`, as they will be ignored.
  self._warn_if_super_not_called()
8/8 ━━━━━━━━━━━━━━━━━━━━ 27s 2s/step - accuracy: 0.5320 - loss: 2.8519 - val_accuracy: 0.5417 - val_loss: 0.6875
Epoch 2/10
8/8 ━━━━━━━━━━━━━━━━━━━━ 11s 1s/step - accuracy: 0.4951 - loss: 0.6919 - val_accuracy: 0.6667 - val_loss: 0.6384
Epoch 3/10
C:\Users\harit\anaconda3\Lib\contextlib.py:158: UserWarning: Your input ran out of data; interrupting training. Make sure that your dataset or generator
can generate at least `steps_per_epoch * epochs` batches. You may need to use the `.repeat()` function when building your dataset.
  self.gen.throw(typ, value, traceback)
8/8 ━━━━━━━━━━━━━━━━━━━━ 5s 453ms/step - accuracy: 0.6250 - loss: 0.7348 - val_accuracy: 0.5000 - val_loss: 0.6777
Epoch 4/10
8/8 ━━━━━━━━━━━━━━━━━━━━ 16s 1s/step - accuracy: 0.6175 - loss: 0.6729 - val_accuracy: 0.7500 - val_loss: 0.6628
Epoch 5/10
8/8 ━━━━━━━━━━━━━━━━━━━━ 13s 2s/step - accuracy: 0.7337 - loss: 0.6007 - val_accuracy: 0.9167 - val_loss: 0.5147
Epoch 6/10
8/8 ━━━━━━━━━━━━━━━━━━━━ 3s 244ms/step - accuracy: 0.5352 - loss: 0.6618 - val_accuracy: 1.0000 - val_loss: 0.4663
Epoch 7/10
8/8 ━━━━━━━━━━━━━━━━━━━━ 17s 2s/step - accuracy: 0.6468 - loss: 0.6521 - val_accuracy: 0.9375 - val_loss: 0.5218
Epoch 8/10
8/8 ━━━━━━━━━━━━━━━━━━━━ 12s 1s/step - accuracy: 0.8209 - loss: 0.4272 - val_accuracy: 0.9167 - val_loss: 0.2146
Epoch 9/10
8/8 ━━━━━━━━━━━━━━━━━━━━ 4s 445ms/step - accuracy: 0.8125 - loss: 0.3166 - val_accuracy: 0.9375 - val_loss: 0.2327
Epoch 10/10
8/8 ━━━━━━━━━━━━━━━━━━━━ 16s 2s/step - accuracy: 0.9050 - loss: 0.3278 - val_accuracy: 0.9167 - val_loss: 0.1412
```

Figure 7.2.1 Model Epochs

7.3 Testing and Evaluation

After completing the model training, the next crucial step is testing the model to evaluate its performance on unseen data. This phase helps to ensure that the model generalizes well and can make accurate predictions on real-world X-ray images. Model testing involves using a separate dataset that was not involved in the training process to objectively assess the model's accuracy.

7.3.1 Testing

- **Dataset for Testing:** For testing, we used a reserved portion of the dataset, typically 15% of the total data, which was completely isolated from both the training and validation datasets. This test set contained chest X-ray images labeled as COVID-19 positive or COVID-19 negative, ensuring the model's predictions could be directly compared to the ground truth.
- **Model Training:** Once the model was trained, it was evaluated on the test set.

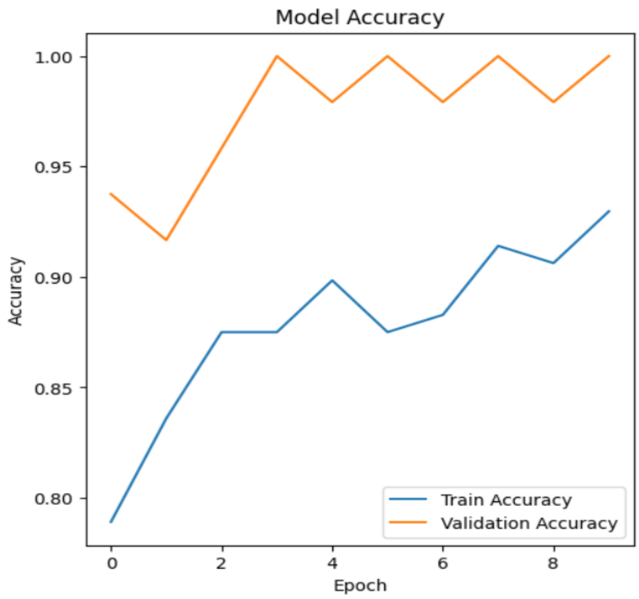

Fig. 7.3.1 Accuracy Comparison

8. USER INTERFACE INTEGRATION

The integration of a user-friendly interface is crucial in making the COVID-19 detection system accessible to end-users, such as medical professionals or the general public. The user interface (UI) allows users to interact with the deep learning model, upload chest X-ray images, and receive real-time predictions regarding COVID-19 status

- **ReactJs:** The frontend was developed using React.js, a popular JavaScript library for building interactive user interfaces
- **Flask**: The deep learning model, which was trained for COVID-19 detection, is served using a Flask API. The Flask server is responsible for:
    - Receiving the chest X-ray image sent from the frontend
    - Preprocessing the image (resizing, normalization) to match the input format of the trained model.
    - Feeding the preprocessed image to the deep learning model
    - Returning the prediction (COVID-19 positive or negative) as JSON data.

9. SAMPLE CODE

```
import numpy as np
import matplotlib.pyplot as plt
import keras
from keras.layers import Dense,Conv2D,MaxPool2D,Dropout,Flatten
from keras.models import Sequential
from keras.preprocessing import image #This not working in new version
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.metrics import BinaryAccuracy

train_datagen=ImageDataGenerator(
   rescale=1/255,horizontal_flip=True, zoom_range=0.2,shear_range=0.2
)
train_data=train_datagen.flow_from_directory(directory="D:\Projects\Minor project - COVID-19
detection\Dataset\Train",target_size=(256,256),batch_size=16,
                                             class_mode='binary')

train_data.class_indices

test_datagen=ImageDataGenerator(
   rescale=1/255
)
test_data=test_datagen.flow_from_directory(directory="D:\Projects\Minor project - COVID-19
detection\Dataset\Val",target_size=(256,256),batch_size=16,
                                             class_mode='binary')

model=Sequential()
model.add(Conv2D(filters=32,kernel_size=(3,3),activation='relu',input_shape=(256,256,3)))  # 3 because
it is an rgb image

model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu'))  # 3 because it is an rgb image
model.add(MaxPool2D())
model.add(Dropout(rate=0.25))

model.add(Conv2D(filters=64,kernel_size=(3,3),activation='relu'))  # 3 because it is an rgb image
model.add(MaxPool2D())
model.add(Dropout(rate=0.25))

model.add(Conv2D(filters=128,kernel_size=(3,3),activation='relu'))  # 3 because it is an rgb image
model.add(MaxPool2D())
```

```
model.add(Dropout(rate=0.25))

model.add(Flatten())
model.add(Dense(units=64,activation='relu'))
model.add(Dropout(rate=0.50)) # if not dropped out model will overfit.
model.add(Dense(units=1,activation='sigmoid'))#  why units=1 ??

model.compile(loss=keras.losses.binary_crossentropy,optimizer="adam",metrics=['accuracy'])

model.summary()

model.fit(train_data,steps_per_epoch=8,epochs=10,validation_steps=3,validation_data=test_data)

path=r"D:\Projects\Minor project - COVID-19 detection\Dataset\Val\Covid\covid-19-pneumonia-53.jpg"
#r to tell python to treat string as raw because of backslashes
img=image.load_img(path,target_size=(256,256))

img=image.img_to_array(img)/255
img=np.array([img])
img.shape

ar=model.predict(img)
ar=np.round(ar).astype(int)
print(ar)
if(ar[0][0] == 0):
    print("Covid")
else:
    print("Normal")


import matplotlib.pyplot as plt

# Assuming 'history' is the variable storing the training results
history = model.fit(train_data, steps_per_epoch=8, epochs=10, validation_steps=3,
validation_data=test_data)

# Plot training & validation accuracy and loss
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))

# Accuracy plot
ax1.plot(history.history['accuracy'], label='Train Accuracy')
ax1.plot(history.history['val_accuracy'], label='Validation Accuracy')
ax1.set_title('Model Accuracy')
ax1.set_xlabel('Epoch')
```

```
ax1.set_ylabel('Accuracy')
ax1.legend(loc='lower right')

# Loss plot
ax2.plot(history.history['loss'], label='Train Loss')
ax2.plot(history.history['val_loss'], label='Validation Loss')
ax2.set_title('Model Loss')
ax2.set_xlabel('Epoch')
ax2.set_ylabel('Loss')
ax2.legend(loc='upper right')

# Save and show the plots
plt.savefig('training_history.png')
plt.show()
```

10. OUTPUT SCREENS

- ## GUI Home Page

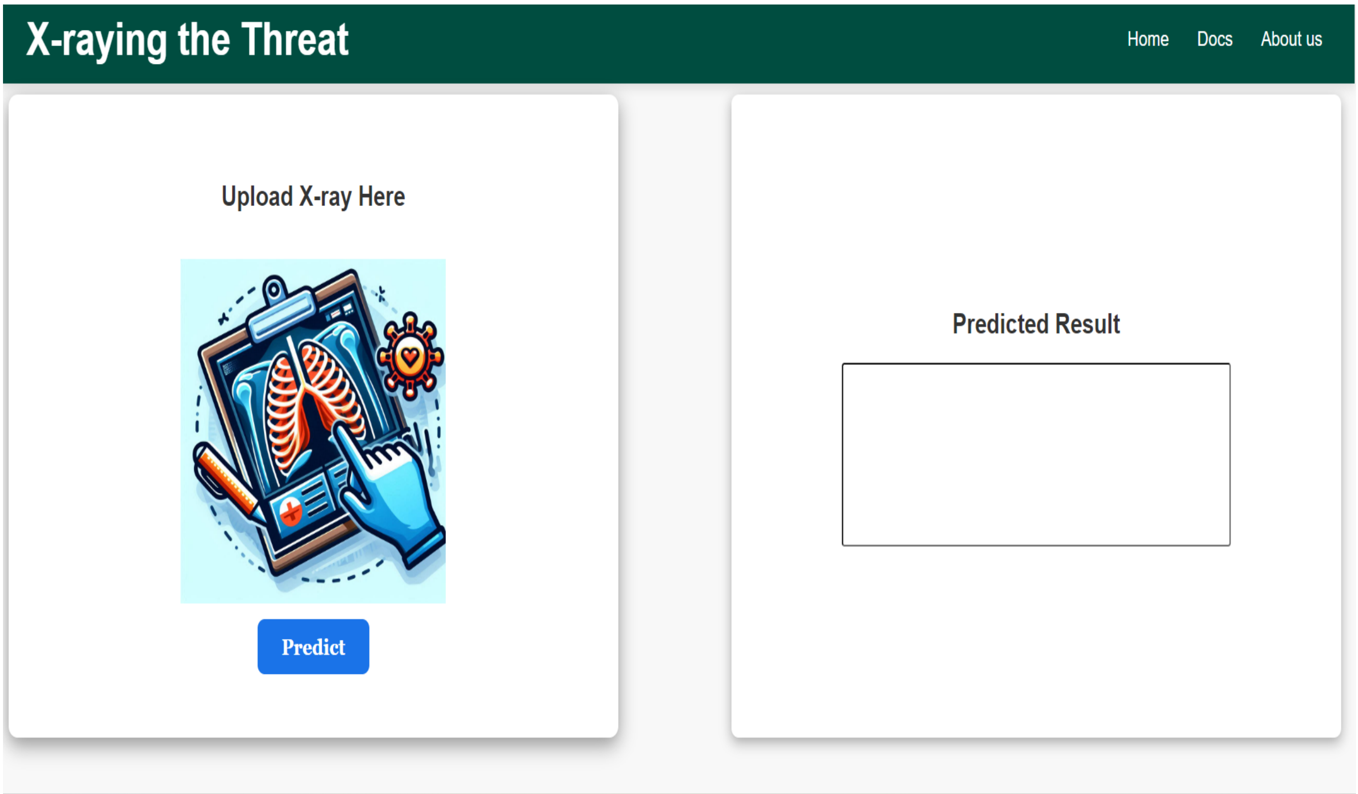

Fig. 10.1 GUI Home Page

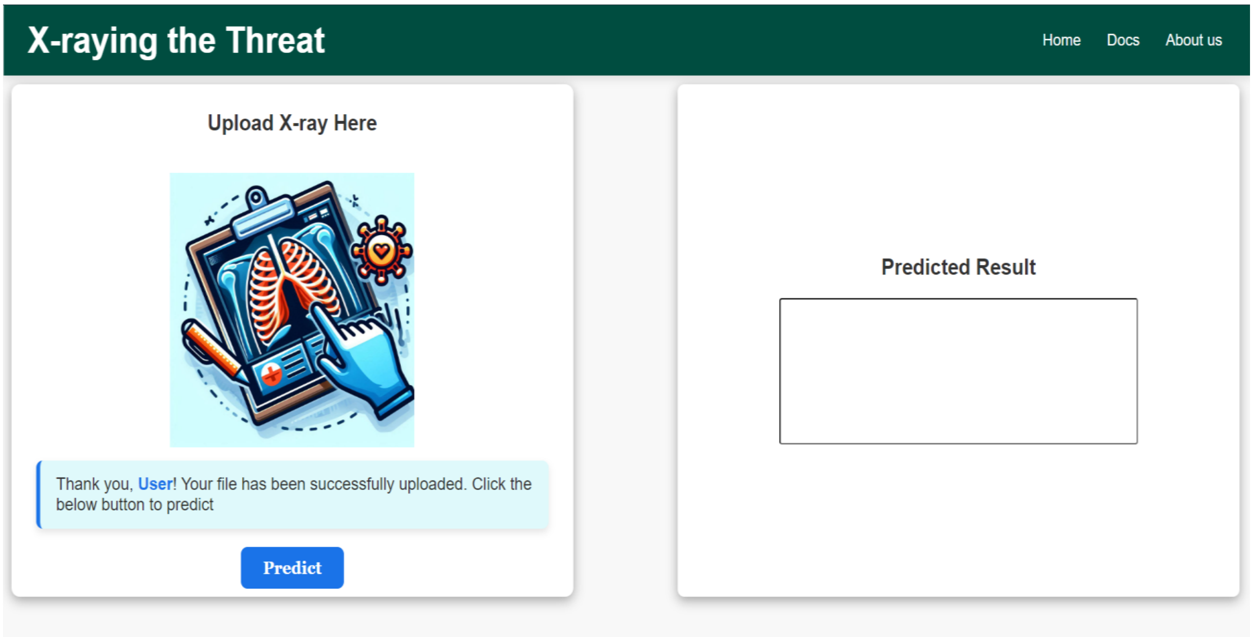

Figure 10.2 Acknowledgement message

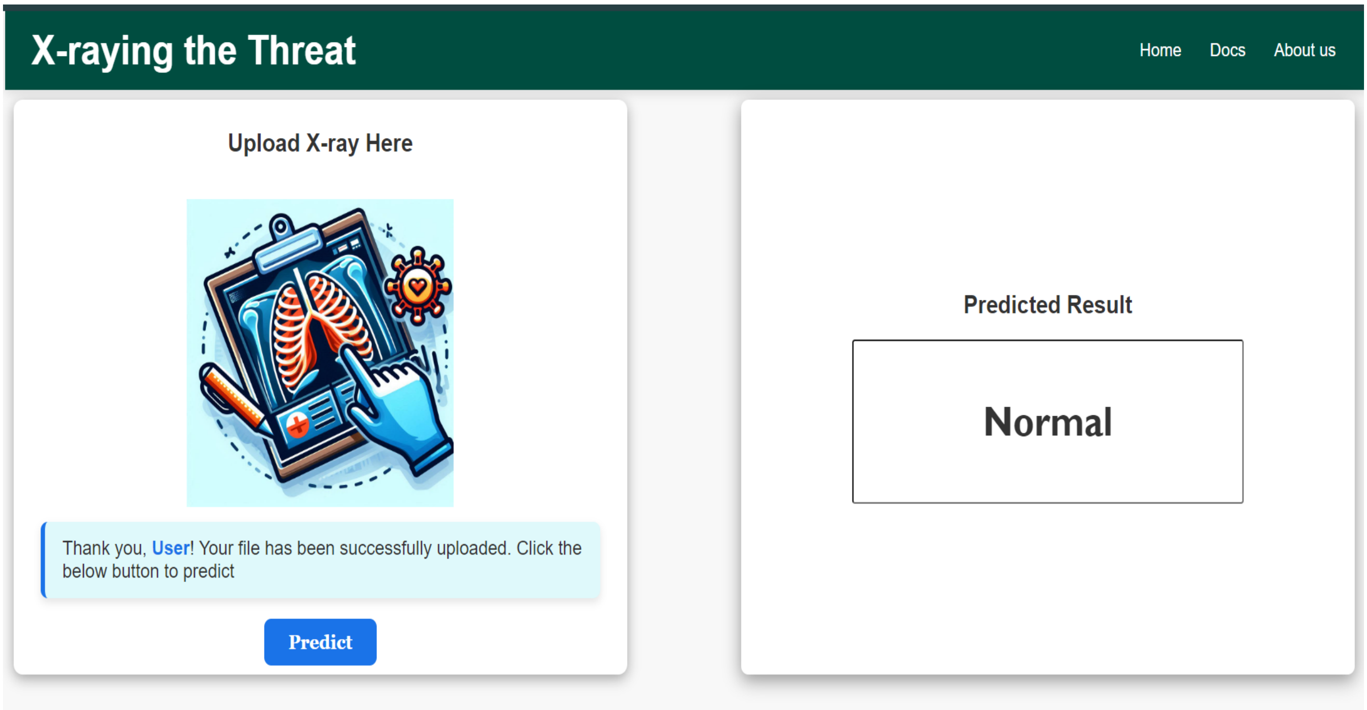

Figure 10.3 Output for Normal X-ray

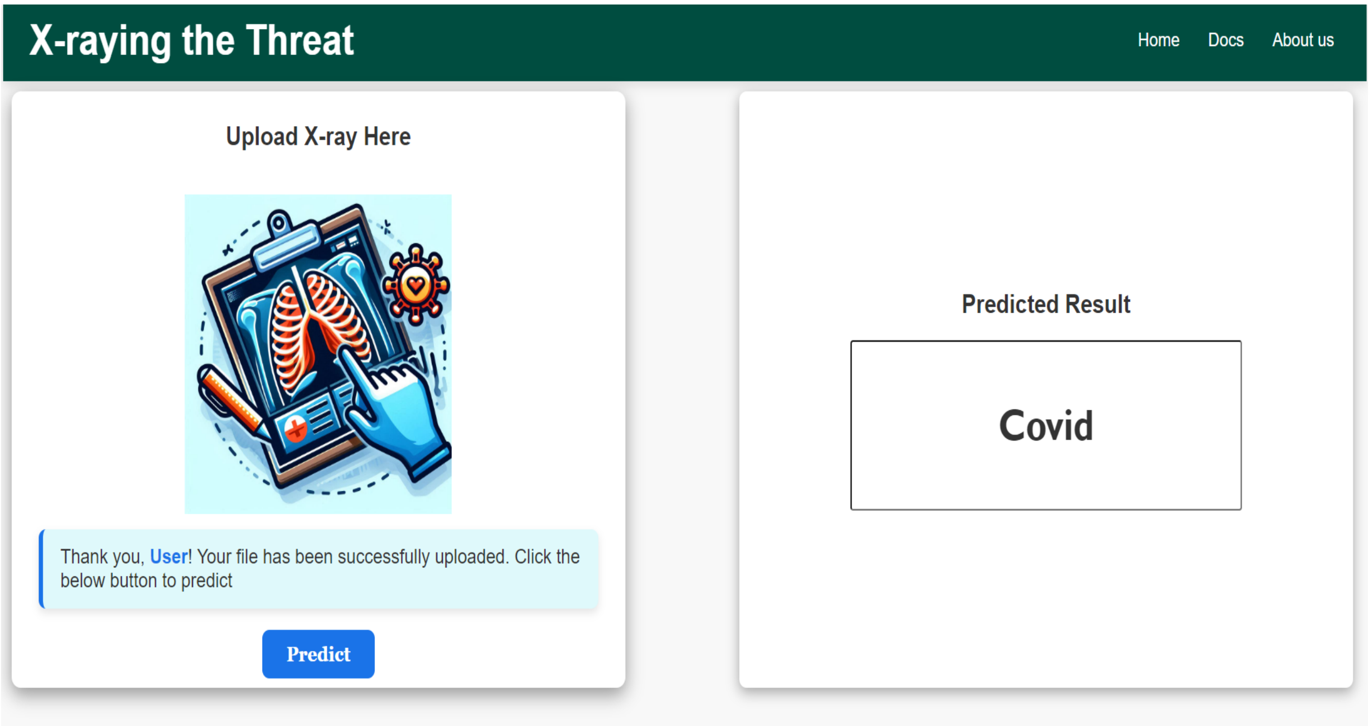

Figure 10.4 Output for Covid positive

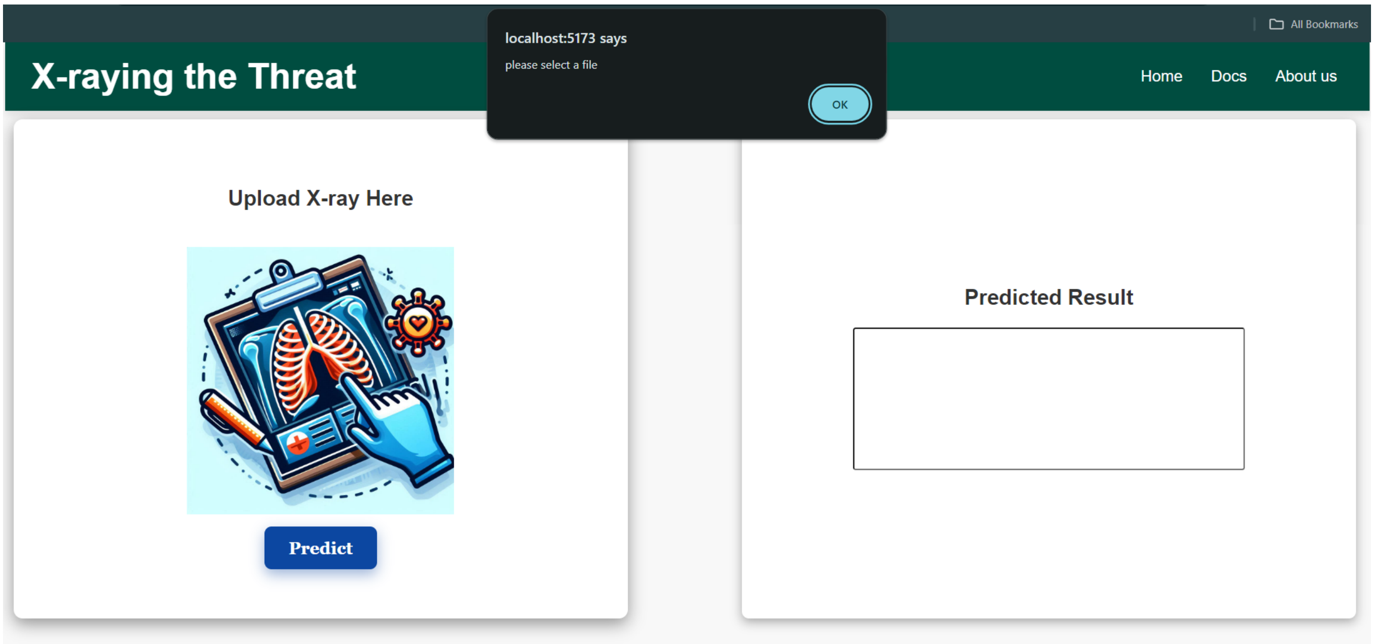

Figure 10.5 Error message 1

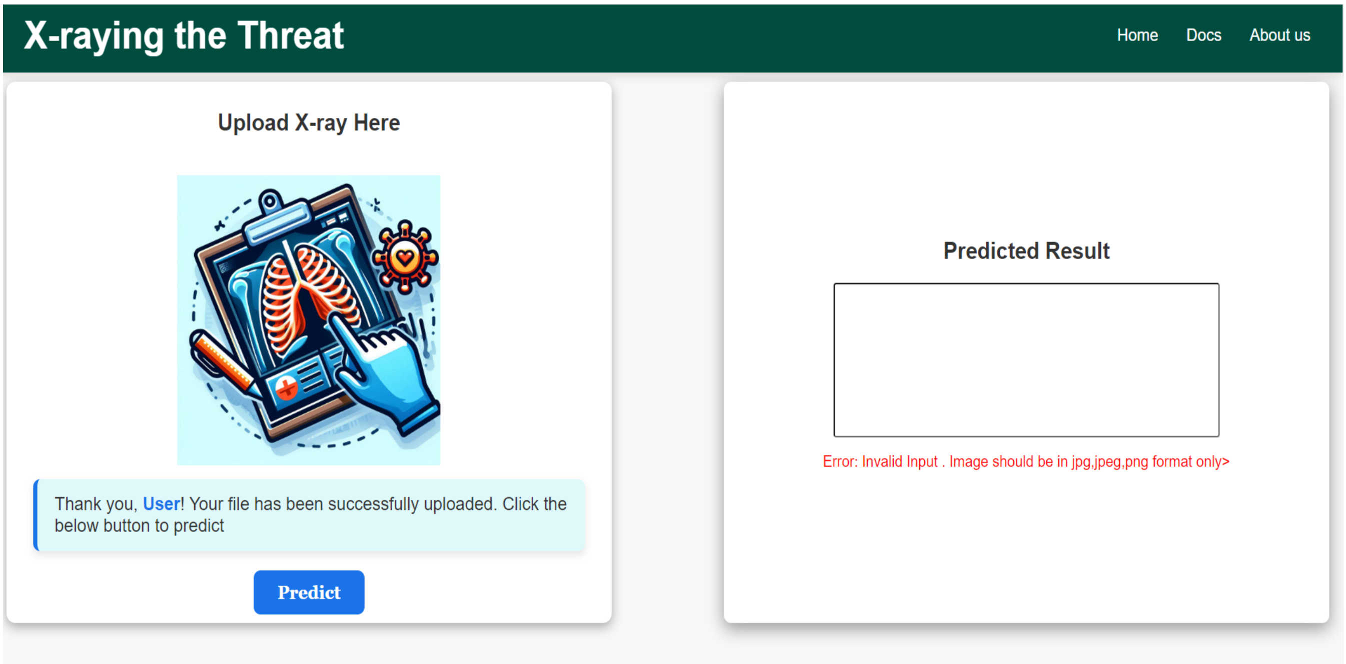

Figure 10.6 Error message 2

11. CONCLUSION

The COVID-19 pandemic has brought about unprecedented challenges for healthcare systems globally, highlighting the urgent need for rapid and accurate diagnostic methods. The integration of deep learning techniques with web-based applications has emerged as a promising solution to enhance the efficiency and accessibility of COVID-19 detection, particularly through the analysis of chest X-ray images. This project demonstrates the successful development of a comprehensive system that leverages state-of-the-art machine learning algorithms to assist in diagnosing COVID-19 through a user-friendly web interface.

The COVID-19 detection system developed in this project comprises a deep learning model for analyzing chest X-ray images, a backend server for processing requests, and a frontend interface for user interactions. The project focuses on several key aspects: data preprocessing, model training, user interface integration, and model evaluation. By systematically addressing each of these components, the project provides a holistic approach to implementing an effective diagnostic tool.The deep learning model was trained using a dataset of chest X-ray images, which included both positive and negative cases of COVID-19. The model architecture was chosen based on its ability to extract features from the images and make predictions with high accuracy.The project's frontend was developed using React.js, a popular framework for building interactive user interfaces. This choice was driven by the need for a responsive and intuitive design that allows users—such as healthcare professionals or patients—to easily upload their chest X-ray images and receive immediate feedback. The upload component was designed to be simple, requiring minimal steps from the user. After the image is uploaded, the user is presented with a clear message indicating whether the prediction is "COVID-19 Positive" or "COVID-19 Negative.". The backend of the system, developed using Node.js and Express.js, serves as the intermediary between the frontend and the deep learning model. This architecture allows for efficient handling of image uploads and seamless communication with the Flask server hosting the model.

In summary, the COVID-19 detection system developed in this project represents a significant step forward in the integration of artificial intelligence within the healthcare domain. By combining advanced deep learning techniques with an intuitive web interface, the project provides a practical tool for rapid COVID-19 diagnosis. The lessons learned and methodologies employed in this project can serve as a framework for developing similar diagnostic tools for other medical conditions, illustrating the broader impact of artificial intelligence in enhancing healthcare delivery and patient care

12. FUTURE SCOPE

The future scope of the COVID-19 detection project using chest X-ray images encompasses several exciting possibilities for improvement, expansion, and adaptation to emerging needs within healthcare.

- **Expanding the Dataset:** To enhance the model's generalizability and performance, future work should focus on collecting and curating a larger and more diverse dataset of chest X-ray images. This should include images from different demographics, geographical locations, and varying disease stages, allowing the model to learn from a broader range of cases and reducing biases associated with a limited dataset.
- **Integration of Additional Modalities:** The project can be enhanced by incorporating other diagnostic tools and imaging modalities, such as computed tomography (CT) scans, magnetic resonance imaging (MRI), or ultrasound. Combining multiple data sources can provide a more comprehensive understanding of a patient's condition, potentially leading to improved diagnostic accuracy.
- **Patient Management and Monitoring:** Developing features for tracking patient outcomes over time and integrating them into the system could enhance its utility for healthcare providers. This would allow for a more holistic approach to patient management, including monitoring treatment responses and adjusting care plans based on continuous feedback.
- **Expansion to Other Diseases:** The techniques and methodologies developed in this project can be adapted to detect other diseases with similar diagnostic imaging, such as pneumonia, tuberculosis, or lung cancer. This adaptability could broaden the scope of the system, making it a versatile tool in medical imaging diagnostics.
- **User-Centric Enhancements:** Improving the user interface further by incorporating features like multilingual support, accessibility options for individuals with disabilities, and educational resources about COVID-19 and related health issues can enhance the system's usability and reach. Customizing the interface for different user groups, such as healthcare professionals and patients, can also optimize the user experience

13. BIBLIOGRAPHY

1. https://www.nature.com/articles/s41598-021-99015-3
2. https://ieeexplore.ieee.org/document/9325626
3. Huang, K., Hussain, A., Wang, Q.F., Zhang, R.: Deep Learning: Fundamentals, Theory and Applications., 1st edn. Springer International Publishing, Switzerland (2019)
4. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8286881/
5. https://www.sciencedirect.com/science/article/pii/S247263032100011X
6. https://www.nature.com/articles/s41598-024-61693-0
7. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7448820/