CovidVision: Advanced COVID-19 Detection from Lung X-rays with Deep Learning

1. INTRODUCTION

1.1 Project Idea

CovidVision is an AI-powered web application designed to detect COVID-19 infections from lung X-ray images using deep learning techniques. The system leverages convolutional neural networks (CNNs) to automatically analyze chest X-rays, identifying COVID-19 positive and normal cases with high accuracy. This project aims to assist medical professionals by providing a rapid and reliable diagnostic tool during pandemics.

1.2 Motivation of the Project

During the COVID-19 pandemic, healthcare systems across the world faced a shortage of radiologists and diagnostic delays. Early detection was critical to contain the spread of the virus. This project was motivated by the need to support hospitals and clinics with an AI-based automated solution that can interpret chest X-rays quickly and accurately, even in resource-limited areas.

1.3 Problem Statement

Manual analysis of chest X-rays for COVID-19 diagnosis requires skilled radiologists and is time-consuming. In many regions, especially rural areas, there is a lack of such expertise. This project addresses the challenge of automating the detection process through deep learning, improving efficiency, and reducing dependency on human interpretation.

1.4 Statement of Scope

The project focuses on using deep learning algorithms, particularly CNN models, to classify lung X-ray images as COVID-19 positive or normal. It involves dataset preprocessing, model training, evaluation, and deployment using a Flask web interface that allows users to upload X-ray images and get instant predictions with confidence levels.

1.5 Goals and Objectives

• To design and train a CNN model capable of detecting COVID-19 from chest X-rays. • To develop a user-friendly Flask web application for real-time prediction. • To ensure high accuracy, speed, and reliability of predictions. • To contribute to healthcare diagnostics through AI automation.

2. STUDY PHASE

2.1 Literature Survey

Various studies have explored deep learning for medical image classification. CNN-based models like VGG16, ResNet, and DenseNet have shown promising results in detecting lung diseases. Research confirms that AI-assisted diagnostics can achieve accuracy comparable to human experts, demonstrating potential for large-scale implementation in pandemic situations.

2.2 Existing System

The current diagnostic approach relies on RT-PCR testing, which is time-intensive and may produce false negatives. Manual interpretation of X-rays is limited by human error and availability of specialists. Some AI models exist, but most are not accessible to general healthcare providers or lack real-time functionality.

2.3 Disadvantages of the Existing System

• Delay in results from traditional testing methods. • Human error and inconsistent diagnosis accuracy. • Limited availability of radiologists in rural and underdeveloped areas. • Lack of scalable AI solutions for rapid diagnosis.

2.4 Feasibility Study

Technical feasibility: The system uses pre-trained CNN architectures and Python frameworks (TensorFlow, Keras, Flask). Economic feasibility: Minimal costs due to open-source technologies. Operational feasibility: Easy to deploy in clinics and hospitals. Scheduling feasibility: The system can be developed and trained within a few weeks using publicly available datasets.

3. PROPOSED SYSTEM

3.1 Proposed System

The proposed system, CovidVision, uses a CNN model to predict COVID-19 infection from lung X-rays. Users can upload an image through a web interface, and the system provides the classification result along with confidence percentage. The backend is implemented in Python using Flask, and the model is trained with labeled datasets containing both COVID and normal cases.

3.2 Methodology

Step 1: Data collection and preprocessing. Step 2: Model training using CNN architecture. Step 3: Validation and accuracy testing. Step 4: Integration of model into Flask web app. Step 5: Deployment and user testing. The trained model predicts the probability of COVID-19 based on X-ray features extracted automatically.

3.3 Advantages

• Fast and automated COVID-19 detection. • Reduces dependence on radiologists. • Provides confidence-based results for better decision-making. • Cost-effective and easily deployable solution.

3.4 Approaches

The project uses deep learning for feature extraction and classification, Flask for web deployment, and OpenCV for preprocessing images. The CNN model architecture was chosen for its ability to recognize spatial patterns in X-rays.

4. SYSTEM REQUIREMENTS SPECIFICATION

4.1 Software Requirements

• Operating System: Windows 10/11 • IDE: Visual Studio Code • Programming Language: Python 3.10+ • Frameworks: TensorFlow, Keras, Flask • Libraries: NumPy, Pandas, OpenCV, Matplotlib

4.2 Technologies Used

Deep Learning, Flask Web Framework, HTML, CSS, TensorFlow, and Keras.

4.3 Hardware Requirements

• Processor: Intel i5/i7 • RAM: 8GB or higher • Storage: Minimum 50GB • GPU: Optional for model training acceleration.

5. DESIGN

5.1 System Architecture

The system follows a client-server architecture. The client uploads an X-ray image, which is processed by the Flask backend. The model classifies the image and returns the prediction and confidence level to the user interface.

5.2 Module Description

Module 1: Data Collection Module 2: Image Preprocessing Module 3: Model Training Module 4: Flask Integration Module 5: Web Interface & Result Display

5.3 UML Diagrams

Includes Use Case Diagram, Sequence Diagram, Activity Diagram, and Deployment Diagram to illustrate workflow.

6. IMPLEMENTATION

The model uses a Convolutional Neural Network trained on labeled chest X-ray images. It performs binary classification (COVID Positive or Normal) and integrates with Flask for prediction.

7. CODING

The project uses Python and Flask. The model is implemented using TensorFlow/Keras, while the web interface is designed using HTML and CSS.

8. TESTING

Testing ensures functionality, accuracy, and robustness. • Unit testing: Individual module tests. • Integration testing: Flask and model communication. • User interface testing: Ensures smooth experience.

9. OUTPUT

The system displays the uploaded X-ray image, predicted class (COVID Positive / Normal), and the confidence percentage in a clean web interface.

10. CONCLUSION

CovidVision successfully demonstrates the use of deep learning for rapid COVID-19 detection from X-rays. It reduces diagnosis time, aids early treatment, and supports healthcare workers with AI assistance.

11. FUTURE ENHANCEMENT

Future improvements include adding support for multi-disease detection, mobile integration, and a cloud-based dataset for continuous model retraining.

12. BIBLIOGRAPHY

1. TensorFlow and Keras Documentation 2. COVID-19 Radiography Database (Kaggle) 3. Flask Web Framework Documentation 4. Research Papers on Deep Learning for Medical Imaging