AI-CliniScan: Pneumonia Detection using Chest X-Ray Imaging

Intern: Drashti Patel

Intern: Drashti Patel

Introduction

Chest X-ray imaging is one of the most commonly used diagnostic tools for analyzing lung diseases. Pneumonia, if not detected early, can lead to severe respiratory problems and may become life-threatening. Manual interpretation of chest X-ray images requires experienced radiologists and can be time-consuming and subjective.

With the growing availability of medical imaging data, there is a strong need for automated and accurate computer-aided detection systems. The AI-CliniScan project aims to build such a system by automating the preprocessing, classification, and later, detection of pneumonia-affected regions using artificial intelligence.

Milestone-1 focuses only on understanding the dataset and building a standardized, clean preprocessing pipeline before moving to model training.

Intern: Drashti Patel

Project Description

The goal of this project is to build an AI-based model that can analyze chest X-ray images, classify whether pneumonia is present, and eventually detect the specific regions of infection using deep learning. This milestone is dedicated to dataset handling, environment setup, and preprocessing of X-ray images to make them suitable for training.

Dataset Note

Due to repeated technical issues while loading the originally provided dataset, a similar publicly available dataset, *Chest X-Ray Pneumonia (Kaggle – Paul Timothy Mooney)*, was temporarily used to complete preprocessing.

⚠ The same preprocessing pipeline can be directly applied to the original dataset later without changes.

Dataset Used

The dataset contains two types of chest X-ray images:

- **Normal**
- **Pneumonia**

The images vary in contrast, size, brightness, and quality, making preprocessing essential before model training and later bounding-box based detection

Intern: Drashti Patel

Environment Setup

Tools used for preprocessing:

- **Python**: Primary programming language
- **NumPy**: Converts images into numerical arrays
- **Pandas**: Handles metadata/paths of images
- **Matplotlib**: Visualizations during exploration
- **OpenCV**: Image reading, filtering, contrast enhancement
- **Image Processing Utilities**: Data augmentation generators in training

These libraries help standardize X-ray data, remove noise, and prepare for later model development.

Data Exploration

Before preprocessing:

- Counted number of images in each class
- Checked image file formats (mostly JPG/PNG)
- Verified grayscale channels
- Observed varying pixel intensity, lighting differences, and size variations

Exploration confirmed the need for resizing, normalization, noise cleaning, and safe augmentation

Image Resizing

All X-rays are resized to fixed size (e.g., 224×224 or 256×256).
Uniform resolution reduces computation and helps the model learn consistently.

Grayscale Handling & Normalization

Images are converted to a uniform grayscale or channel format.
Pixel values are scaled (0-1 or -1-1) to speed up model learning and reduce instability.

Noise Reduction

Chest X-rays may contain noise due to sensors.
Gaussian blur or similar filters remove noise while preserving lung edges.

CLAHE (Contrast Enhancement)

CLAHE improves visibility of local lung patterns without over-enhancing noise.
It helps highlight pneumonia-related opacities and textures.

Train–Validation–Test Split

Dataset is divided into:

- **Training set** – model learns
- **Validation set** – fine-tuning

- **Test set** – final evaluation

This avoids overfitting and ensures real performance testing.

Data Augmentation

Since X-ray data is limited, valid transformations help increase sample diversity:

- Small rotations
- Horizontal flips (only medically appropriate)
- Zooming within safe range
- Height/width shifts

Augmentation is applied **only to the training set**, not validation or test sets

Detection Extension + Conclusion

Planned Extension: Pneumonia Region Detection

In later milestones, the model will not just classify normal/pneumonia but will mark **pneumonia-affected regions with bounding boxes**.
This can be achieved using advanced architectures like:

- **YOLO**
- **Faster R-CNN**

- **Region-based detectors**

These require additional annotation data and use the same preprocessing pipeline created in Milestone-1.

Conclusion

A complete preprocessing pipeline was created using a substitute Kaggle dataset, but it remains fully compatible with the original dataset.
The project is now ready to progress toward:

- Classification modeling
- Region-detection using bounding box algorithms