CliniScan: Lung-Abnormality Detection on Chest X-rays using AI

Intern Name:

Rishika Sharma

Table Contents

1. Project Description

The CliniScan project focuses on building a complete medical-image preprocessing pipeline to prepare radiology images such as X-rays or CT scans for machine learning.
Raw medical images often contain noise, black borders, uneven brightness, different dimensions, and inconsistent intensity levels. These issues reduce the accuracy of deep learning models.

This project aims to clean and standardize images using a sequence of operations such as DICOM-to-RGB conversion, grayscale transformation, resizing, normalization, denoising, contrast enhancement, cropping, padding, annotation conversion, and data augmentation.
The final output is a set of well-processed images suitable for training advanced AI models like YOLO and CNNs.

2. Dataset Used

The dataset provided contained a collection of medical diagnostic images in formats such as **DICOM (.dcm)** and **PNG/JPEG**.
These images belonged to different patients and had variations in:

- Image size and orientation
- Pixel intensity
- Borders and blank areas
- Noise levels
- Colour channels (RGB or grayscale)
- Annotation formats for YOLO training

Due to this inconsistency, preprocessing was essential before building any model.

3. Environment Setup

The following tools and libraries were used for the project:

Software

- Python 3.x
- Jupyter Notebook / Google Colab
- Conda/virtual environment (optional)

Python Libraries

- pydicom – for reading DICOM images
- PIL (Pillow) – for image conversion, cropping, resizing
- opencv-python (cv2) – for normalization, CLAHE, and augmentation
- numpy – for array operations
- os – for directory and file management
- albumentations – for data augmentation
- shutil – for dataset splitting

Folder Structure

- raw_images/
- processed_images/
- annotations/
- train/, val/, test/

This environment ensured smooth preprocessing and dataset preparation.

4.Data Exploration

Before preprocessing, the dataset was carefully explored to understand its characteristics:

- **Image Properties:** dimensions, colour depth, intensity distribution
- **Quality Issues:** noise, blur, unwanted borders
- **Metadata (for DICOM):** patient ID, modality type, pixel spacing
- **Annotation Review:** bounding boxes, label formats
- **Class Distribution:** count of images per category
- **File Formats:** presence of DICOM and PNG/JPEG

This analysis helped in identifying the necessary preprocessing operations.

5.Data Preprocessing

5.1 DICOM to RGB Conversion

Many medical images were in **DICOM (.dcm)** format. These images were converted using pydicom by:

- Reading pixel arrays
- Scaling pixel values
- Converting single-channel data to 3-channel RGB
- Saving them as PNG/JPEG for further processing

This step made the data usable for deep learning models and image libraries.

5.2 Grayscale Conversion

Medical images are commonly analysed in **grayscale** because:

- It reduces computational complexity.
- Important features like bones, tissues, and anomalies are more visible.
- Deep learning models require consistent channels.

Images were converted using Pillow or OpenCV.

5.3 Image Resizing

Images came in different resolutions.
To make them uniform for model training, they were resized to a fixed resolution such as:

- 224×224
- 512×512
- Or YOLO standard sizes

Resizing helps maintain consistency and improves model performance.

5.4 Intensity Normalization

Different medical devices produce images with varying brightness.
Normalization performs:

- Scaling pixel values to 0–1
- Standardizing contrast
- Making images consistent

This step improves the model’s ability to learn meaningful features.

5.5 CLAHE Enhancement

CLAHE (Contrast Limited Adaptive Histogram Equalization):

- Enhances local contrast
- Makes edges and structures more visible
- Prevents over-amplification of noise

It is widely used in radiology to improve feature visibility

5.6 Denoising

Medical images often contain grainy noise.
Denoising was done using:

- Gaussian blur
- Median blur
- Bilateral filter

This reduces unwanted pixels while keeping important details clear.

5.7 Cropping Borders

Many images had **black borders** or blank regions that do not contain medical information.
Border cropping:

- Detects non-black regions
- Removes all unnecessary outer areas
- Focuses the image on the main anatomy

This improves model input quality.

5.8 Padding

After cropping, some images may not be square. Padding was used to maintain aspect ratio while resizing.

- Empty space added using black or constant value
- Ensures uniform input shape
- Prevents image distortion

5.9 YOLO Annotation Conversion

The dataset annotations were converted into YOLO format:

class_id x_center y_center width height

Steps:

- Read bounding box values
- Convert them to normalized YOLO format
- Save .txt files corresponding to each image

This prepares the dataset for YOLO training.

5.10 Data Augmentation

To increase dataset diversity and reduce overfitting, augmentations were applied:

- Rotation
- Flipping
- Scaling
- Brightness/Contrast charges
- Noise addition
- Shift/Crop

Augmented data improves model generalization.

6. Train/Validation/Testing Split

The dataset was divided into:

- **70% Training** – used for learning
- **15% Validation** – used for tuning
- **15% Testing** – used for final evaluation

Splitting ensures that the model learns effectively and its performance is tested fairly on unseen data.

The split was created using Python scripts or libraries like sklearn or shutil.