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

SANDHYA BELAVI

sandhyabelavi1@gmail.com

# Table of Content

# Project Description

The aim of this project is to develop an AI-powered system capable of automatically detecting and localizing lung abnormalities from chest X-ray (CXR) images using advanced deep learning techniques. The system is designed to support radiologists and healthcare providers by identifying a wide range of pathological findings such as opacities, consolidations, fibrosis, nodules, masses, and potentially classifying associated pulmonary diseases like pneumonia or tuberculosis.

To ensure clinical reliability and deployment readiness, the model will be trained on a large-scale, annotated chest X-ray dataset and optimized for:

- Accuracy in abnormality detection
- Interpretability (via heatmaps and bounding boxes)
- Robustness across diverse imaging conditions
- Real-world diagnostic integration
-

The overall workflow for the project includes:

1. Data Acquisition
2. Image Preprocessing
3. Model Development (Detection & Classification)
4. Training and Evaluation
5. Visualization & Interpretation (Grad-CAM)
6. Deployment

# 1. INTRODUCTION

Chest X-ray images captured across hospitals and X-ray machines vary significantly in:

- brightness
- contrast
- noise levels
- image dimensions
- presence of artifacts

Deep learning models like **YOLO** (for object detection) and **EfficientNet** (for classification) require **clean, consistent, and high-quality input images** to learn accurately. Thus, preprocessing is a crucial step that greatly impacts model performance.

Notebook-1 performs all necessary preprocessing steps to transform raw medical images into a standardized dataset ready for model training.

# 2. TOOLS AND TECHNOLOGIES USED

## 2.1 Platform

Kaggle Notebook Environment
For GPU/CPU computing, efficient dataset handling, and reproducibility.

## 2.2 Programming Language

Python 3

## 2.3 Libraries

| **Libraries** | **Purpose** |
|---|---|
| NumPy | Numerical operations, pixel manipulation |
| Pandas | Handling CSV annotation files |
| OpenCV (cv2) | Image processing (grayscale, crop, resize, denoise, CLAHE) |
| Matplotlib | Visualization of before/after images |
| pydicom | Reading DICOM medical images (if applicable) |
| tqdm | Showing progress bars |
| sklearn.model_selection | Creating train/val/test split |

# 3. PREPROCESSING WORKFLOW

This pipeline was designed using medical imaging best practices and deep learning requirements.

### Step 1 — Load Dataset & Setup Paths

- Define input directories (DICOM or PNG)
- Create preprocessing output folders
- Set image size to 640×640
- Set limit for testing (e.g., 500 images)

**Purpose:**
Organizes files properly, ensures reproducibility and scalability.

### Step 2 — Convert DICOM → PNG (if required)

If using DICOM (medical format), images are decoded using pydicom and saved as grayscale PNG.

**Purpose:**
PNG format is lightweight, universally readable, and compatible with deep learning frameworks.

### Step 3 — Convert Images to Grayscale

All images are loaded as single-channel grayscale.

**Significance:**

- Medical X-rays inherently contain no color
- Reduces computation by 3×
- Stabilizes training
- Removes artificial RGB noise

### Step 4 — Crop Borders

Removes black/blank margins around the lungs.

**Significance:**

- Prevents model from learning irrelevant regions
- Focuses on lung anatomy
- Faster training
- Higher detection accuracy

### Step 5 — Intensity Normalization

Performs Min-Max scaling:

pixel = (pixel − min) / (max − min)

**Significance:**

- Standardizes brightness
- Reduces variability between X-ray machines
- Helps the model generalize

### Step 6 — Apply CLAHE (Contrast Enhancement)

CLAHE = Contrast Limited Adaptive Histogram Equalization

**Significance:**

- Enhances visibility of abnormalities
- Improves edge clarity
- Works well on low-quality chest X-rays
- Avoids over-amplification of noise

### Step 7 — Denoising

Uses Gaussian and Median filtering.

**Significance:**

- Smooths image
- Reduces granular noise
- Preserves important structures like ribs and nodules

### Step 8 — Resize & Pad to 640×640

Maintains aspect ratio and pads remaining space.

**Significance:**

- Required by YOLO models
- Prevents anatomical distortion
- Ensures consistent model input

### Step 9 — Save Pre-processed Images

Final images stored in:
processed_640/

- These become the training input for detection/classification.

### Step 10 — Train/Validation/Test Split

- A standard:
  - 70% — Train
  - 15% — Validation
  - 15% — Test
- **Train (Training Set)**
  Model learns patterns and features.
- **Val (Validation Set)**
  Used during training to tune model and avoid overfitting.
- **Test (Testing Set)**
  Used only at the end to evaluate real performance.

## 4. Why Each Step Matters (Summary Table)

| Step | What It Does | Why It Is Important |
|---|---|---|
| Grayscale | Removes artificial color, reduces channels | X-rays are naturally grayscale; improves speed + clarity |
| Cropping | Removes black borders | Removes irrelevant pixels; focuses on anatomy |
| Normalization | Standardizes intensity | Reduces equipment differences |
| CLAHE | Enhances contrast | Highlights abnormalities and soft tissues |
| Denoising | Removes noise | Cleaner images → better learning |
| Resize + Pad | Standardizes size | Required by YOLO and CNN models |
| Split | Creates train/val/test sets | Ensures fair evaluation and prevents overfitting |

# 5. Conclusion

Notebook-1 successfully converts raw chest X-ray images into a **clean, standardized, enhanced** dataset ready for deep learning.
This preprocessing pipeline improves:

- Model performance
- Stability during training
- Accuracy in detecting abnormalities
- Clinical interpretability