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

Intern Name: Varun Khandelwal

Table of Contents

1. Project Overview

1.1 Objective

CliniScan aims to develop an AI model that automatically detects and classifies abnormalities in chest X-ray images. The model identifies conditions like consolidation, infiltration, pneumothorax, nodule, mass, pleural effusion, and other pathological findings.

1.2 Significance

This project assists radiologists in diagnostic processes by providing automated, rapid, and accurate analysis of chest X-rays. Early detection of lung abnormalities enables timely medical intervention and improves patient outcomes.

2. Dataset Details

2.1 Dataset Information

Name: VinBigData Chest X-ray Abnormalities Detection

Source: /kaggle/input/vinbigdata-chest-xray-abnormalities-detection

Format: DICOM files with annotations and bounding boxes

2.2 Dataset Characteristics

- Large-scale collection of chest X-ray images in DICOM format
- Comprehensive annotations with bounding boxes for abnormalities
- Diverse patient populations and clinical scenarios
- Multiple abnormality classes for classification
- High-resolution medical images suitable for deep learning

3. Environment Setup

3.1 GPU Configuration

GPU acceleration is checked using **torch.cuda.is_available()** to determine if CUDA-capable GPUs are available. This enables 10-100x faster training compared to CPU processing.

3.2 Key Libraries

- **PyTorch:** Deep learning framework for model building and training
- **OpenCV:** Computer vision library for image processing and resizing
- **Pydicom:** Library for reading and handling DICOM medical image files
- **Pandas & NumPy:** Data manipulation and numerical computations
- **Matplotlib:** Visualization of X-ray images and results
- **PIL:** Image I/O and format conversion
- **tqdm:** Progress bars for tracking long-running operations

4. Data Exploration

4.1 DICOM File Reading

DICOM is the standard medical image format. Using pydicom, we extract pixel arrays from DICOM files. The pixel values (typically 16-bit integers) represent X-ray attenuation through different tissue types.

4.2 Visualization

Images are visualized using matplotlib with the bone colormap, which is standard for radiological interpretation. This colormap displays bones as dark areas and soft tissues as lighter areas.

4.3 Custom PyTorch Dataset

A custom **VinBigDataDataset** class is implemented to handle efficient data loading. It scans directories for DICOM files, reads them on-demand (lazy loading), and applies transformations.

5. Data Preprocessing

5.1 Preprocessing Pipeline

- **1. Load DICOM:** Read image using pydicom
- **2. Resize:** Standardize to 512x512 resolution
- **3. Normalize:** Min-max normalization to [0, 1] range
- **4. CLAHE:** Contrast Limited Adaptive Histogram Equalization for contrast enhancement
- **5. Denoise:** Gaussian blur (3x3 kernel) to remove noise
- **6. Crop & Pad:** Remove borders and pad back to 512x512
- **7. Convert:** Grayscale to RGB for model compatibility
- **8. Normalize (ImageNet):** Apply ImageNet statistics

5.2 Why These Steps

CLAHE: Enhances local contrast and improves visibility of subtle abnormalities. **Min-max normalization:** Ensures consistent pixel value ranges. **Grayscale to RGB:** Pre-trained models expect 3-channel input. **ImageNet normalization:** Matches statistics of pre-trained models for better transfer learning.

6. Data Augmentation

6.1 Why Augmentation

Data augmentation increases effective dataset size and improves model generalization. It must preserve medical validity and anatomical correctness.

6.2 Augmentation Techniques

- **Horizontal Flip (40%):** Simulates patient positioning variations. Clinically valid as chest anatomy is symmetric.

- **Rotation (-5 to +5°, 30%):** Simulates minor positioning changes without severe distortion.
- **Contrast Variation (30%):** Simulates equipment differences and exposure variations across facilities.

6.3 Implementation

The **safe_augment** function applies transformations probabilistically. Each original image is saved as PNG, and an augmented version is created. Processing 3000 images generates 6000 training samples, doubling the effective dataset size.

6.4 Storage Optimization

Images are saved as PNG (not DICOM) because PNG loads faster during training. Preprocessing is done offline before training, reducing GPU idle time and ensuring consistency across epochs.