CENG 407

Innovative System Design and Development I

2025-2026 Fall

MULTI-LABEL CHEST X-RAY DISEASE CLASSIFICATION AND EXPLAINABILITY WITH DEEP LEARNING

DATASET DESCRIPTION & PREPROCESSING

YAREN AKDOĞAN 202111046

SEZİN KÖROĞLU 202211046

İBRAHİM TAŞPINAR 202111072

CEVDET ALP ÖZGÜN 202211406

İPEK BAŞ 202211013

SALİH BARKIN AKKAYA 202211004

Table of Contents

1. Dataset Description

1.1. NIH ChestX-ray14

The NIH ChestX-ray14 dataset contains 112,120 frontal chest X-ray images labelled for 14 different thoracic diseases, and it has become one of the most commonly used datasets in this research field. The dataset is comprised of posterior-anterior (images taken with the patient facing detector) and anterior–posterior (images taken with the patient facing machine) X-ray images along with various details on each patient. All images in the dataset have a standard format of grayscale 1024x1024 images [1].

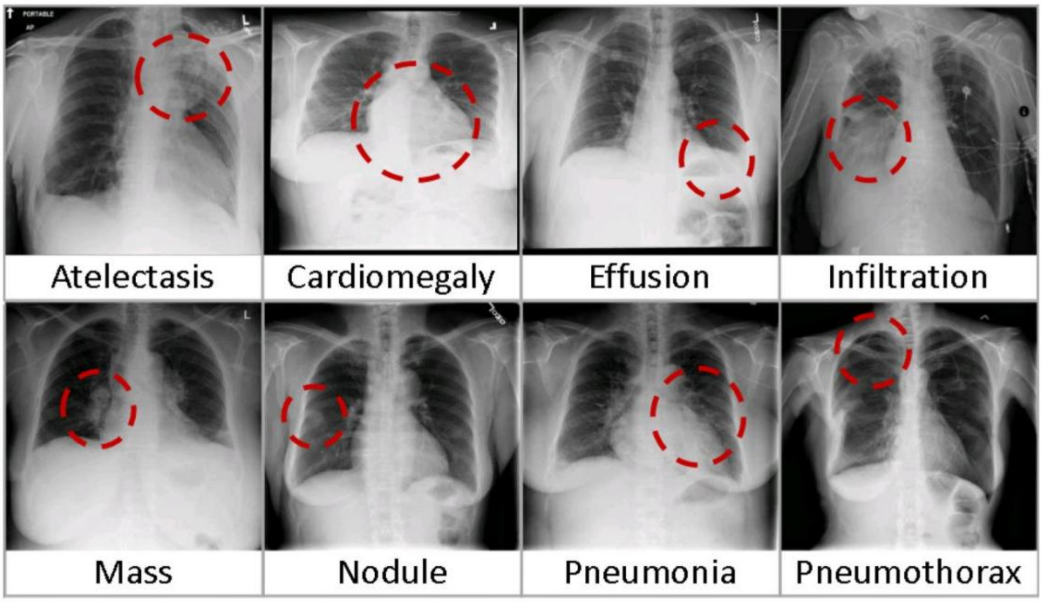


Figure 1. Eight visual examples of common thorax diseases

1.2. Label Information

The dataset has a multi-label structure for 14 thoracic diseases including Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural Thickening, Cardiomegaly, Nodule, Mass and Hernia. The labelling for the images has been done with the use of NLP techniques on real radiologist reports. Due to the uncertain nature of NLPs, the dataset contains some uncertain labels. However, the overall label accuracy of the dataset is >90%. The dataset also contains an issue due to the class imbalance of images. Although some classes have over 20,000 labels associated with them, there are also those with only around 2000 labels. To get around this issue we are planning to utilize weighted learning in our model [1].

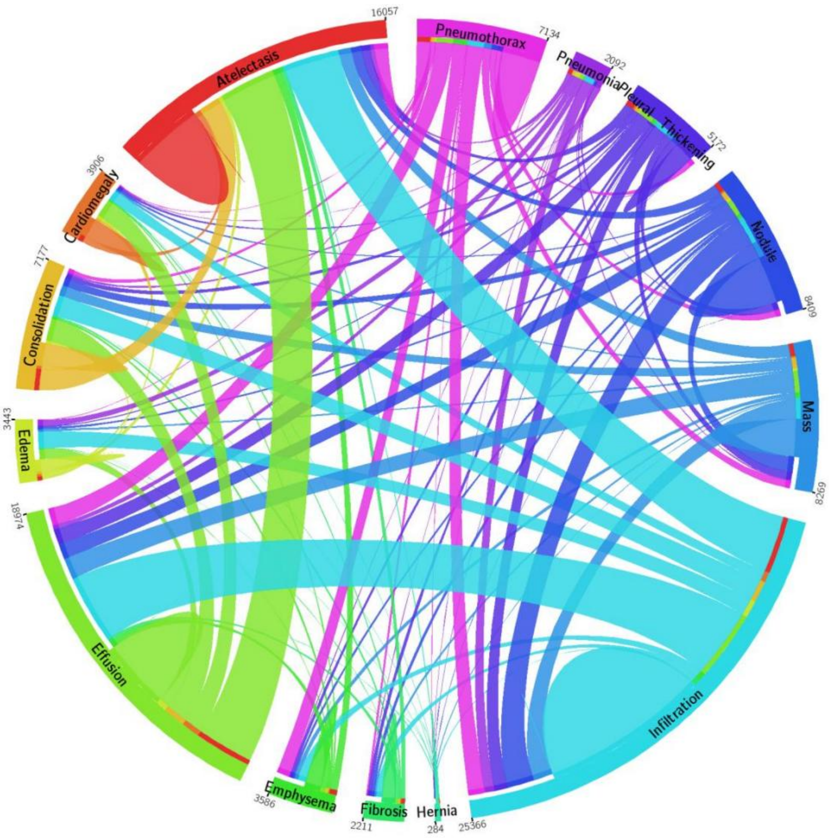


Figure 2. Chord diagram displaying the distributions of 14 disease categories with co-occurrence statistics

2. Frontal View–Based Filtering

First, we limit the dataset to chest X-ray images in posterior–anterior (PA) view. We read the original metadata file and select only the rows where the "View Position" column is "PA". Next, we scan the image directory tree and copy these PA images into a new folder called FinalDataset_PA. After this step, we save a filtered metadata file (metadata_pa_only.csv) that includes only the copied images. We also save a simple CSV file (image_names_pa_only.csv) that contains only the PA image names.

Then, we rearrange the original train and test splits for this PA-only subset. We read the PA metadata and create a set from the "Image Index" column. Using this set, we filter the original train_val_list.txt and test_list.txt files. For both of the file, we keep only the lines whose image name is in the PA set and write these lines into train_list_pa.txt and test_list_pa.txt. As a result of this way, every image in the train and test lists is a PA image that exists in the FinalDataset_PA directory.

3. Image Processing Scenarios

3.1. Scenario 1:

Figure 3. Processed version of the Chest X-Ray

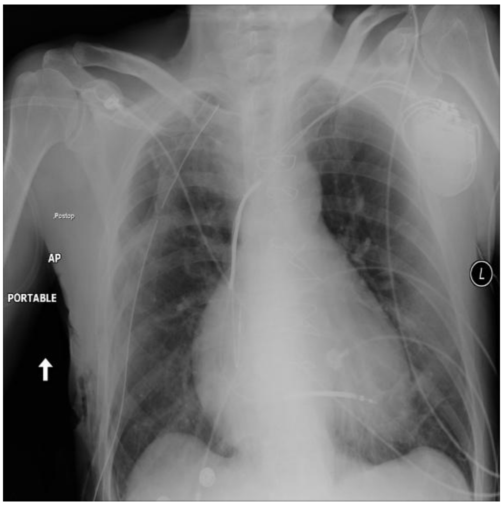


Figure 4. Unprocessed version of the Chest X-Ray

We plan to apply an intensity enhancement pipeline to chest X-ray images in the dataset. First, we will normalize each image to the range of [0,1]. Then, we will keep that range as a single grayscale channel. Later, we will apply gamma correction ($\gamma = 1.2$), CLAHE with a low clip limit, and an unsharp mask for sharpening [3][5]. All of these steps are planned to improve contrast and make the images more understandable for our system.

For each image, we plan to load it as a grayscale image with PIL and convert 16-bit images to 8-bit if necessary. We will then apply the enhancement pipeline and resize the result to 224×224 pixels using Albumentations (this will be needed for our deep learning model). After that, we will create a 3-channel RGB image by repeating the same grayscale channel three times, scale the values to [0, 255], convert to 8-bit, and save the result. This will give us images in a standard 224×224 RGB format as desired.

3.2. Scenario 2:

Figure 5. Processed version of the Chest X-Ray

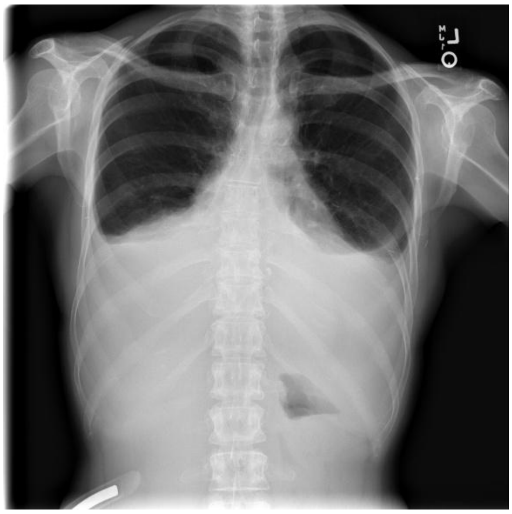


Figure 6. Unprocessed version of the Chest X-Ray

We plan to use a second preprocessing pipeline based on OpenCV and NumPy. First of all, we will take each input image and convert them to 8-bit grayscale in the range of [0, 255]. Then, we will make sure the image has a single channel and apply a median filter [4]. This method will reduce noise in images while keeping their edges. After that, we will use CLAHE to increase local contrast and then apply a simple sharpening kernel to highlight important structures [3]. Finally, we will normalize the processed image back to the range of [0, 1] in float32 format.

For each image, we plan to load it as a grayscale image with PIL and run this pipeline. We will resize the processed output to 224×224 pixels with Albumentations. After resizing, we will create a 3-channel RGB image by repeating the grayscale channel three times. Finally, we will scale the values to [0, 255], convert them to 8-bit, and save the result. This will give us images in a standard 224×224 RGB format that our deep learning model can use.

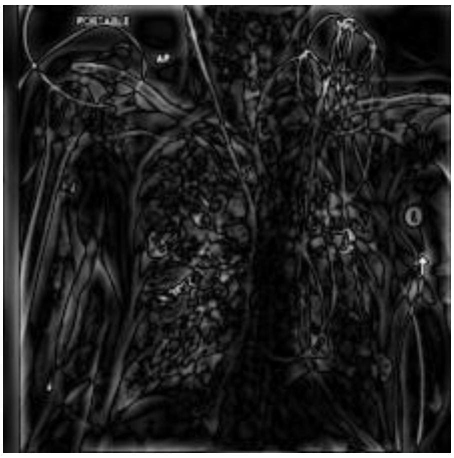
3.3. Scenario 3:

Figure 7. Processed version of the Chest X-Ray ***Figure 8. Unprocessed version of the Chest X-Ray***

In this scenario, we plan to use a simple bone suppression step for chest X-ray images. First of all, we will take each input image and convert them to 8-bit grayscale in the range of [0, 255]. Then we will make sure the image has a single channel and apply a Gaussian blur to obtain a "bone-like" structure. Next, we will subtract a weighted part of this blurred image(original) to reduce the level of brightness of bones. After that, we will apply CLAHE to increase local contrast [3]. Finally, we will normalize the result to the [0, 1] range as a float32 image. The goal is to suppress bones and make soft tissue regions (lungs), more visible.

For each file, we will load the image in grayscale mode using PIL and convert it to a NumPy array. We will then apply the bone suppression processing and prepare the output for resizing by adding a channel dimension. The processed image will be resized to 224×224 pixels with Albumentations. After resizing, we will create a 3-channel RGB image by repeating the same grayscale channel three times. Last, we will scale the pixel values to [0, 255], convert them to 8-bit, and save the result to disk. This will give us bone-suppressed images in a standard 224×224 RGB format that our deep learning model can use.

3.4. Scenario 4:



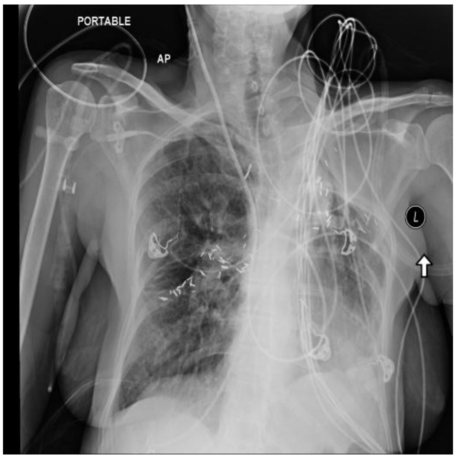


Figure 9. Processed version of the Chest X-Ray ***Figure 10. Unprocessed version of the Chest X-Ray***

In this method, we plan to combine CLAHE and a Butterworth band-pass filter for preprocessing. First of all, convert each image to 8-bit grayscale and apply CLAHE to improve local contrast [3]. Then we will move to the frequency domain with a 2D FFT and apply a Butterworth band-pass filter to emphasize structural details while reducing very low and very high frequencies. This idea follows previous work on frequency-domain enhancement of chest X-rays [2].

For each image, we will load it as a grayscale image with PIL, apply the CLAHE + Butterworth pipeline, and then resize the result to 224×224 pixels using Albumentations. After resizing, we will create a 3-channel RGB image by repeating the grayscale channel three times, scale the values to [0, 255], convert to 8-bit, and save the result. This will give us a 224×224 RGB format that our deep learning model can use.

4.References

[1] **National Institutes of Health, "NIH Chest X-rays Dataset," Kaggle, [Online].**

Available: https://www.kaggle.com/datasets/nih-chest-xrays/data.

Accessed: 9 Dec. 2025.

[2] **A. Giełczyk, A. Marciniak, M. Tarczewska & Z. Lutowski, "Pre-processing methods in chest X-ray image classification,"** ***PLOS ONE*****, vol. 17, no. 4, e0265949, Apr. 2022.**

Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0265949

Accessed: 9 Dec. 2025.

[3] IEEE Xplore, "Article (document no. 10955260)," [Online].

Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10955260.

Accessed: 9 Dec. 2025.

[4] "Median Filter – an overview," ***ScienceDirect Topics*****, Elsevier, [Online].**

Available: https://www.sciencedirect.com/topics/engineering/median-filter.

Accessed: 9 Dec. 2025.

[5] N. F. Sahib and Z. A. Hashim, "Contrast Image Enhancement by Gamma Correction," ***Computer Engineering and Intelligent Systems*****, vol. 9, no. 7, 2018.**

Available: https://scispace.com/pdf/contrast-image-enhancement-by-gamma-correction-3upr34p2k1.pdf.

Accessed: 9 Dec. 2025.