Lightweight XNet-Inspired Convolutional Neural Network for Lung Segmentation in Chest X-Ray Images Using a Small-Scale Public Dataset

Azhaff Khalid (22i-1895)
Hashir Ahmed (22i-1988)

FAST NUCES

December 11, 2025

Abstract

Medical image segmentation is a core task in diagnostic imaging, enabling automated analysis and clinical decision support. Traditional deep learning segmentation models require large, diverse annotated datasets, which are not always available in real clinical environments. Inspired by the XNet architecture—designed for small medical datasets—this research implements and evaluates a lightweight U-Net–style model for lung segmentation using a small Kaggle dataset of chest X-ray images with lung masks.

The dataset is modest in size and includes a limited number of high-quality paired X-ray/mask samples, making it suitable for exploring small-dataset learning. The proposed model integrates skip connections, multi-stage encoding/decoding, and a combined Binary Cross-Entropy (BCE) + Dice loss function to address class imbalance and segmentation boundary complexity. Experiments were conducted using PyTorch in Google Colab.

The model achieves strong performance despite the limited dataset size, showing that small, well-designed architectures can produce competitive segmentation results. The trained network reaches a validation Dice score of **0.9369** and a test Dice score of **0.9370**. Limitations and opportunities for future enhancement—such as using Tversky loss, attention mechanisms, or self-supervised pretraining—are discussed to guide continued research.

1 Introduction

Chest X-ray imaging is one of the most widely used modalities in clinical practice because it is fast, accessible and relatively inexpensive. Accurate segmentation of lung structures in X-ray images is important for computer-aided diagnosis, disease quantification, and preprocessing for downstream machine learning models.

Deep learning has transformed medical image segmentation with convolutional neural network (CNN) architectures such as U-Net [1] and SegNet [2]. However, these architectures typically assume access to large annotated datasets. In many medical settings, especially when pixel-wise annotations must be created by radiologists, the data available for training is limited.

Bullock et al. introduced XNet [3], a convolutional neural network specifically designed for X-ray image segmentation using small datasets. Their architecture uses a multi-stage encoder–decoder structure to balance feature extraction depth with preservation of fine-grained detail, and was shown to outperform classical image processing approaches and some generic segmentation networks on three-class segmentation (open beam, soft tissue and bone).

This work takes inspiration from the XNet architecture and adapts its design philosophy to a simpler yet clinically relevant task: binary lung segmentation in chest X-ray images. We use a small, publicly available dataset from Kaggle containing chest X-rays and corresponding lung masks. The main objectives of this research are:

- To determine whether a lightweight XNet-inspired U-Net model can achieve robust lung segmentation performance on a small dataset.
- To evaluate the impact of a combined BCE + Dice loss function on segmentation quality.
- To identify limitations and potential improvements relative to the original XNet approach.

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 describes the dataset and preprocessing. Section 4 details the proposed model architecture and training methodology. Section 5 presents experimental results. Section 6 discusses findings and limitations, and Section 8 concludes with directions for future work.

2 Related Work

Early X-ray segmentation methods largely relied on classical image processing techniques such as histogram-based thresholding, clustering, and edge detection. Pakin et al. used clustering for bone and soft tissue segmentation in radiographic images [4], while Bandyopadhyay et al. developed entropy-based methods for automatic bone segmentation [5]. Spectral clustering and active shape models have also been applied to robust X-ray segmentation [6]. These approaches often require careful parameter tuning for each anatomical region and may struggle with noise and large anatomical variability.

In contrast, deep learning methods learn hierarchical representations of the input directly from data. U-Net [1] introduced a symmetric encoder–decoder architecture with skip connections and quickly became a standard architecture in biomedical image segmentation. SegNet [2] proposed an encoder–decoder design that stores pooling indices to improve memory efficiency. Many variants of U-Net and other CNN architectures have since been proposed, including attention-based models and architectures optimised for specific organs or modalities.

However, most of these architectures are parameter-heavy and designed with large datasets in mind. In many medical scenarios, only a small number of annotated images are available. XNet [3] directly addresses this constraint by designing an architecture suitable for training on a small X-ray dataset. The authors report strong segmentation performance compared to classical techniques and a simplified SegNet baseline.

Recent work has also focused on loss functions tailored to segmentation with class imbalance. Dice loss and related variants (e.g. Tversky loss) have been widely adopted to improve training stability and overlap metrics [7]. These loss functions are particularly relevant when the target structures occupy a small proportion of the image.

Our work lies at the intersection of these ideas: we adopt a lightweight XNet-inspired architecture, apply it to a small public dataset, and use a combined BCE + Dice loss to improve performance on an imbalanced segmentation task.

3 Dataset

3.1 Source

The dataset used in this study is the "CRD: Chest X-Ray Images with Lung Segmented Masks" dataset, available on Kaggle [8]. It consists of frontal chest X-ray images with corresponding binary lung masks indicating the lung regions.

The dataset is downloaded programmatically in Google Colab using the `kagglehub` Python library, which simplifies accessing Kaggle datasets without manually uploading API credentials.

3.2 Structure

After download and extraction, the data is organised into the following structure:

```
CXR_Combined/
    images/
        DARCXR_2327.png
        DARCXR_4495.png
        ...
    masks/
        DARCXR_2327.png
        DARCXR_4495.png
        ...
```

Each image file has a corresponding mask file with the same filename in the `masks/` directory. This allows straightforward pairing of images and masks by matching stems of filenames.

3.3 Preprocessing

All images and masks are converted to grayscale and resized to 256×256 pixels. The X-ray images are normalised to approximately $[-1, 1]$ using a simple linear transformation after conversion to tensors.

The masks are loaded as single-channel images and binarised with a threshold of 0.5:

$$M_{\text{bin}}(x, y) = \begin{cases} 1, & \text{if } M(x, y) > 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

3.4 Data Split

The paired image–mask samples are split into training, validation and test sets as follows:

- 70% training
- 15% validation
- 15% test

The split is performed randomly using a fixed random seed for reproducibility. All splits are mutually exclusive at the image level.

4 Methods

4.1 Overview

The overall pipeline consists of:

1. Downloading and extracting the dataset via KaggleHub.
2. Building PyTorch `Dataset` and `DataLoader` objects for train/validation/test splits.
3. Implementing an XNet-inspired U-Net model.
4. Training using a combined BCE + Dice loss.
5. Evaluating performance on validation and test sets.

4.2 Dataset and DataLoader

A custom PyTorch `Dataset` class is implemented to load image–mask pairs. Each sample returns:

- **Input**: a $(1, 256, 256)$ tensor representing a normalised chest X-ray.
- **Target**: a $(1, 256, 256)$ binary tensor representing the lung mask.

An optional horizontal flip is applied as a simple data augmentation technique on the training set with probability 0.5.

4.3 Model Architecture

The proposed architecture is a lightweight U-Net-style encoder–decoder, inspired by XNet [3]. It includes:

- **DoubleConv blocks**: each block consists of two consecutive convolutional layers with 3×3 kernels, batch normalisation and ReLU activation.
- **Encoder**: two DoubleConv blocks with intermediate max-pooling for downsampling.
- **Bottleneck**: a deeper DoubleConv block to capture higher-level features.
- **Decoder**: transposed convolutions for upsampling followed by DoubleConv blocks. Skip connections from encoder to decoder layers are used to preserve spatial detail.
- **Output layer**: a 1×1 convolution to map to a single-channel logit map.

Formally, let E_1 and E_2 denote encoder feature maps, B the bottleneck, and D_2, D_1 decoder feature maps. The architecture can be summarised as:

$$E_1 = \text{DoubleConv}_1(X), \quad E_2 = \text{DoubleConv}_2(\text{Pool}(E_1)),$$

$$B = \text{DoubleConv}_3(\text{Pool}(E_2)),$$

$$D_2 = \text{DoubleConv}_4(\text{Concat}(\text{Up}(B), E_2)),$$

$$D_1 = \text{DoubleConv}_5(\text{Concat}(\text{Up}(D_2), E_1)),$$

$$\hat{Y} = \text{Conv}_{1\times1}(D_1),$$

where X is the input image and $\hat{Y}$ are the output logits.

4.4 Loss Functions

We employ a combined BCE + Dice loss to improve segmentation quality under class imbalance.

Binary Cross-Entropy (BCE) Loss For a batch of N pixels, BCE loss is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \left(y_i \log p_i + (1 - y_i) \log(1 - p_i) \right),$$

where p_i is the predicted probability and $y_i \in \{0, 1\}$ the ground truth.

Dice Loss Dice loss is derived from the Dice similarity coefficient:

$$\text{Dice} = \frac{2\sum_i p_i y_i + \epsilon}{\sum_i p_i + \sum_i y_i + \epsilon},$$

and

$$\mathcal{L}_{\text{Dice}} = 1 - \text{Dice}.$$

Here ϵ is a small constant for numerical stability.

Combined Loss We combine both losses as:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{BCE}} + (1 - \alpha)\mathcal{L}_{\text{Dice}},$$

with $\alpha = 0.5$ in our experiments.

4.5 Optimisation and Training

The model is trained using the Adam optimiser with a learning rate of 10^{-4}. We use a batch size of 8 and train for 10 epochs on a GPU provided by Google Colab.

For each epoch, we compute the average training loss and approximate Dice score on the training set. We also evaluate on the validation set and save the model weights achieving the highest validation Dice score.

4.6 Evaluation Metrics

The primary evaluation metric is the Dice coefficient on the validation and test sets. We also track the combined loss during training.

Given predicted probabilities $\hat{Y}$ and ground-truth masks Y, a hard segmentation is obtained by thresholding at 0.5. The Dice coefficient for a batch is:

$$\text{Dice} = \frac{2\sum_i \hat{y}_i y_i + \epsilon}{\sum_i \hat{y}_i + \sum_i y_i + \epsilon}.$$

5 Results

5.1 Training and Validation Performance

Table 1 summarises the training and validation loss and Dice scores for three representative epochs (1, 5 and 10).

Table 1: Training and validation performance across selected epochs.

Epoch	Train Loss	Train Dice	Val Loss	Val Dice
1	0.3728	0.8110	0.2935	0.8769
5	0.1308	0.9295	0.1223	0.9338
10	0.0919	0.9409	0.0969	0.9369

The validation Dice score increases substantially from epoch 1 to epoch 5 and then stabilises, indicating effective learning without severe overfitting. Both training and validation losses decrease consistently over epochs.

Table 2: Test set performance of the best model.

Metric	Score
Best Validation Dice	0.9369
Test Loss	0.0963
Test Dice	0.9370

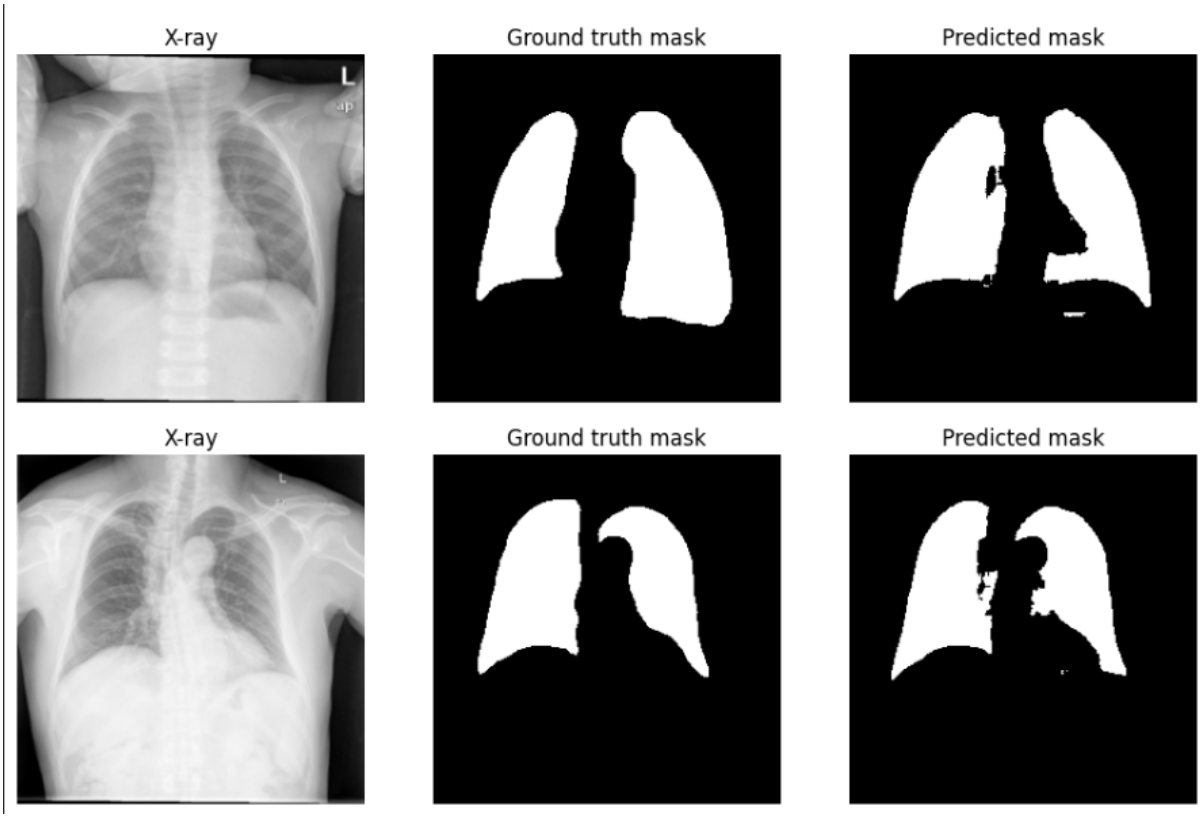

Figure 1: original X-ray, ground-truth lung mask and predicted mask

5.2 Test Set Performance

After training, we load the best-performing checkpoint (based on validation Dice) and evaluate on the held-out test set. The resulting performance is summarised in Table 2.

The close agreement between the best validation Dice and the test Dice suggests that the model generalises well to unseen data within this dataset.

5.3 Qualitative Results

Qualitative examples of model predictions show that the network accurately delineates the lung boundaries and produces smooth, coherent masks. Figure 1 illustrates representative examples from the test set, where the original X-ray, ground-truth lung mask and predicted mask are visualised side by side.

6 Discussion

The results demonstrate that a lightweight XNet-inspired U-Net can achieve strong lung segmentation performance on a small public dataset. Despite the limited number of training samples,

the network achieves a high Dice coefficient on both validation and test sets, suggesting that careful architectural design and loss selection can mitigate data scarcity.

The combined BCE + Dice loss plays a key role in stabilising training and improving overlap-based metrics. BCE alone tends to focus on pixel-wise classification accuracy, while Dice loss emphasises region-level overlap. Combining them yields robust gradients and better boundary quality, particularly when the lung region occupies a relatively small portion of the image.

The use of skip connections and multistage encoder–decoder design, inherited conceptually from U-Net and XNet, ensures that fine spatial details are preserved throughout the network, leading to smooth and coherent segmentation masks.

However, due to the limited scale and diversity of the dataset, the model's generalisation to different populations, scanners or acquisition protocols remains uncertain. Moreover, this work considers only binary lung segmentation, whereas the original XNet addressed a more complex three-class segmentation task.

7 Limitations and Future Work

Several limitations of this work suggest directions for future improvement:

- **Dataset size and diversity:** The dataset used is relatively small and may not capture the full variety of clinical cases, pathologies or imaging conditions. Future work should consider cross-dataset evaluation on other public chest X-ray datasets.
- **Binary segmentation only:** This project focuses on binary lung vs. background segmentation. Extending the approach to multi-class segmentation tasks, such as separating different anatomical structures or pathological regions, would more closely align with the original XNet study.
- **Limited augmentation:** Only basic horizontal flipping was used. More advanced augmentations (elastic deformations, intensity variations, noise) could improve robustness.
- **Model capacity:** While the lightweight architecture is suitable for small datasets, exploring slightly deeper or attention-augmented variants (e.g. Attention U-Net) could further enhance segmentation accuracy.
- **Alternative loss functions:** Losses such as Tversky, focal Tversky or focal Dice may better handle edge cases and highly imbalanced regions [7].
- **Self-supervised pretraining:** Pretraining the encoder on large, unlabeled X-ray datasets using self-supervised learning, followed by fine-tuning on the small labelled dataset, could yield better feature representations.

8 Conclusion

This work demonstrates that a lightweight XNet-inspired U-Net model can perform effective lung segmentation on a small chest X-ray dataset. By leveraging a combined BCE + Dice loss and a compact encoder–decoder architecture with skip connections, the model achieves strong Dice scores on both validation and test sets despite limited data.

The study confirms that small-data medical segmentation is feasible when network design, loss function choice and basic augmentation are carefully considered. Future research may focus on more complex segmentation tasks, advanced loss functions, attention mechanisms and self-supervised pretraining to further enhance performance and generalisability.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[3] Joseph Bullock, Carolina Cuesta-Lázaro, and Arnau Quera-Bofarull. Xnet: A convolutional neural network (cnn) implementation for medical x-ray image segmentation suitable for small datasets. *arXiv preprint arXiv:1812.00548*, 2019.

[4] S Kubilay Pakin, Roger S Gaborski, Leonid L Barski, David H Foos, and Kevin J Parker. Clustering approach to bone and soft tissue segmentation of digital radiographic images of extremities. *Journal of Electronic Imaging*, 12(1):12–10, 2003.

[5] Oindrila Bandyopadhyay, Bhabatosh Chanda, and B Bhattacharya. Entropy-based automatic segmentation of bones in digital x-ray images. In *Pattern Recognition and Machine Intelligence*, pages 122–129. Springer, 2011.

[6] Jing Wu and Mohamed R Mahfouz. Robust x-ray image segmentation by spectral clustering and active shape model. *Journal of Medical Imaging*, 3(3):034005, 2016.

[7] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.

[8] Mrunal Shah. Crd: Chest x-ray images with lung segmented masks. https://www.kaggle.com/datasets/mrunalnshah/crd-chest-x-ray-images-with-lung-segmented-masks. Accessed: 2025-12-11.