Practice 4: A Study of Segmentation and Classification of 3D Lung CT Images in LUNA16 Dataset

Hoang Khanh Dong - 22BA13072
February 2026

***Abstract*—Lung cancer remains the leading cause of cancer-related mortality worldwide, and early detection of pulmonary nodules on computed tomography (CT) scans is critical to improving patient survival. This practice presents a two-stage deep learning pipeline for automated lung nodule detection using the LUNA16 challenge dataset.**

I. Introduction

Pulmonary nodules are small, round lesions found in the lungs that may indicate early-stage lung cancer. Detecting these nodules in chest CT scans is a challenging task due to the large volumetric data, diverse nodule morphologies, and the overwhelming number of non-nodule structures that can mimic nodules.

The LUNA16 (Lung Nodule Analysis 2016) challenge provides a standardized benchmark for evaluating nodule detection algorithms. It consists of 888 CT scans (10 subsets) with expert-annotated nodule locations drawn from the LIDC-IDRI dataset. The primary evaluation metric is the FROC score, defined as the average sensitivity at seven predefined false-positive rates (1/8, 1/4, 1/2, 1, 2, 4, and 8 FP/scan).

This work adopts just **5 subsets** of the dataset for training and testing, using a two-stage approach:

- **Stage 1 — Lung Segmentation:** A lightweight 3D U-Net (4-level encoder-decoder, channels 16–128) segments the lung region from the raw CT volume.
- **Stage 2 — Nodule Classification:** A 3D ResNet-18 (channels 32–256) classifies each candidate as a true nodule or false positive.

II. Dataset

This study uses 5 out of 10 official subsets (subset 0–4) from the LUNA16 challenge, comprising a total of 445 CT scans. Each scan is stored in MetaImage (`.mhd/.raw`) format. Table I summarizes the subset statistics.

TABLE I
Summary of LUNA16 subsets used in this study.

Subset	Scans	Avg. Slices
0	89	256.97
1	89	243.28
2	89	278.17
3	89	268.61
4	89	262.99
Total	**445**	**261.80**

Nodule annotations are provided in `annotations.csv`, which contains the world coordinates (x, y, z) and diameter (in mm) for each nodule identified by at least 3 out of 4 radiologists. Figure 1 shows the distribution of nodule diameters across the 5 subsets.

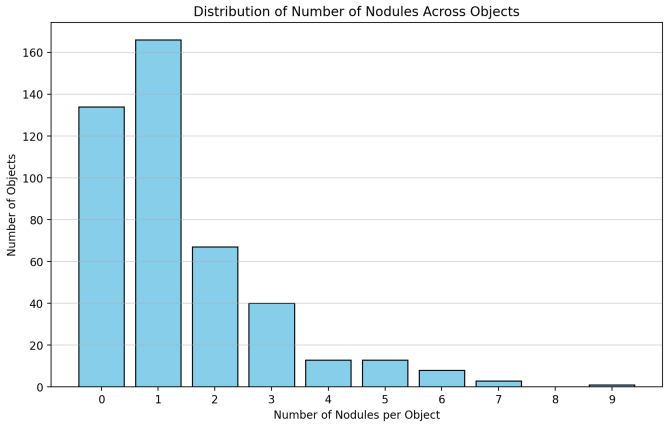

Fig. 1. Distribution of nodule diameters in the selected LUNA16 subsets.

III. Methodology

A. Data Preprocessing

All CT scans are loaded from MetaImage (`.mhd`) format using SimpleITK. The raw Hounsfield Unit (HU) intensities are clipped to the range $[-1000, 400]$ and linearly normalized to $[0, 1]$. Each volume is then resampled to 1 mm isotropic spacing via trilinear interpolation (`scipy.ndimage.zoom`), ensuring consistent physical resolution across scans with varying slice thickness.

Stage 1 (Segmentation): The resampled volumes are divided into non-overlapping chunks of $64 \times H \times W$ along the depth axis. The spatial dimensions H and W are resized to 256×256 using bilinear interpolation. Chunks shorter than 64 slices are zero-padded. Ground-truth lung masks from the `seg-lungs-LUNA16` directory are resampled with nearest-neighbor interpolation and binarized.

Stage 2 (Classification): Candidate locations from `candidates_V2.csv` are converted from world coordinates to voxel coordinates in the isotropic volume. A 32^3-voxel patch is cropped around each candidate center, with

zero-padding applied at volume boundaries. During training, random flips along all three axes (Z, Y, X) are applied as data augmentation.

B. Stage 1: Lung Segmentation

1) Model Architecture: The segmentation model is a lightweight 3D U-Net with a 4-level encoder-decoder architecture and skip connections. It takes a single-channel input of size $64 \times 256 \times 256$ and produces a binary lung mask of the same size. The total parameter count is approximately 1.40 M. Figure 2 illustrates the architecture.

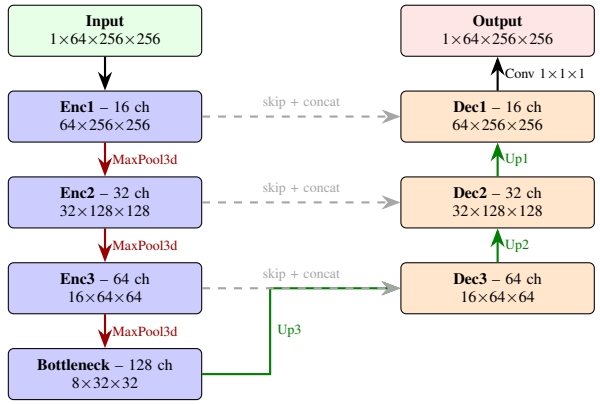

Fig. 2. 3D U-Net architecture for lung segmentation (1.40 M parameters).

Each encoder block consists of two consecutive 3D convolutions ($3 \times 3 \times 3$) followed by GroupNorm and ReLU. Max-pooling ($2 \times 2 \times 2$, stride 2) halves the spatial dimensions between levels. The decoder mirrors the encoder using transposed convolutions for upsampling, with skip connections concatenating encoder features before each decoder block. The final $1 \times 1 \times 1$ convolution produces per-voxel logits.

2) Training and Validation Strategy: I employ 5-fold cross-validation, where each fold holds out one subset for validation and trains on the remaining four. Table II summarises the data split across folds. Each CT volume is divided into non-overlapping chunks of depth 64 (after resampling to 1.0 mm isotropic spacing), with H and W resized to 256×256.

TABLE II
STAGE 1: 5-FOLD CROSS-VALIDATION DATA SPLITS.

Fold	Train Subsets	Train Chunks	Val Subset	Val Chunks
1	{1, 2, 3, 4}	1930	{0}	475
2	{0, 2, 3, 4}	1926	{1}	479
3	{0, 1, 3, 4}	1927	{2}	478
4	{0, 1, 2, 4}	1922	{3}	483
5	{0, 1, 2, 3}	1915	{4}	490

Training. The 3D U-Net is trained for 10 epochs with a batch size of 1 (due to large volume sizes) using AdamW optimizer (weight decay 10^{-4}) and OneCycleLR scheduler (max LR $= 10^{-3}$, 10% warmup, cosine annealing). The loss function combines Dice loss and BCE loss with equal weighting. Automatic mixed-precision (AMP) is used throughout, and gradient norms are clipped at 5.0.

Validation. At the end of each epoch, the model is evaluated on the held-out validation subset. I compute per-voxel Dice coefficient and Intersection-over-Union (IoU) for the lung region. The best checkpoint is selected based on the lowest validation loss. Figure **??** shows the training curves.

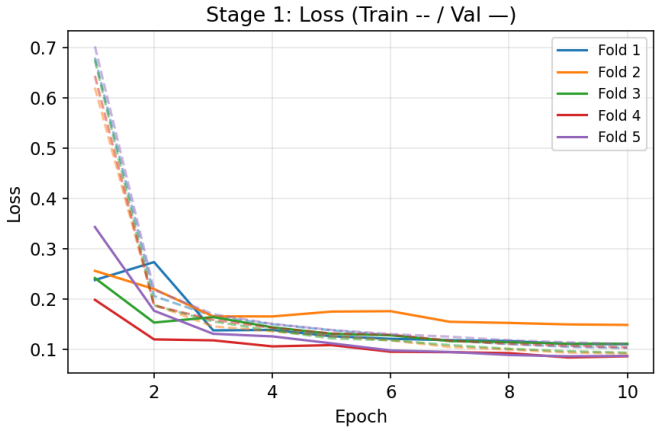

Fig. 3. Stage 1: Training and Validation Loss.

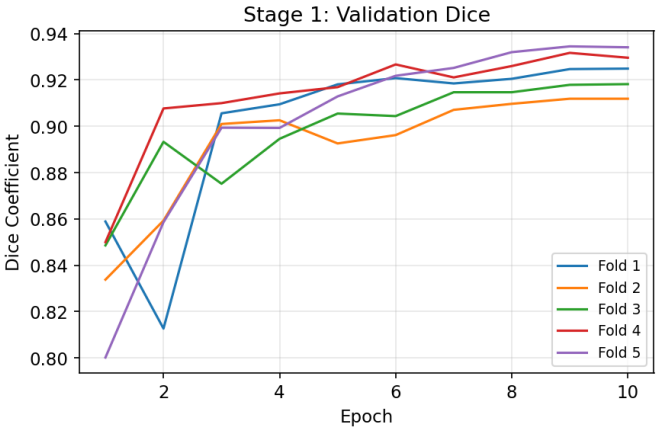

Fig. 4. Stage 1: Validation Dice Coefficient.

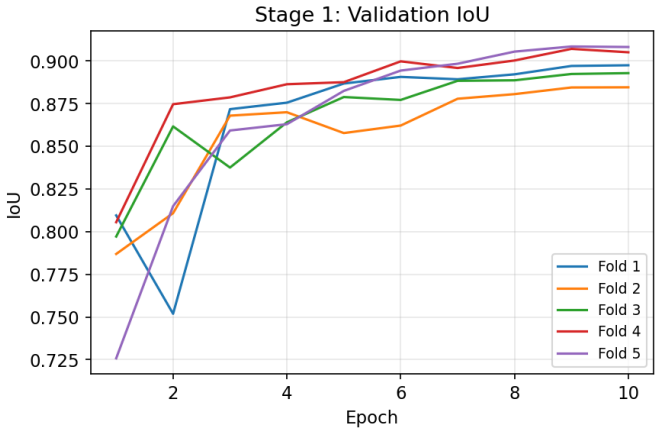

Fig. 5. Stage 1: Validation IoU.

C. Stage 2: Nodule Classification

1) Model Architecture: The classification model is a 3D ResNet-18 adapted for volumetric input. It takes a single-channel 32×32×32 isotropic patch and outputs a single nodule probability logit. The network uses narrower channels (32–256) for efficiency, totalling approximately 3.60 M parameters. Figure 6 illustrates the architecture.

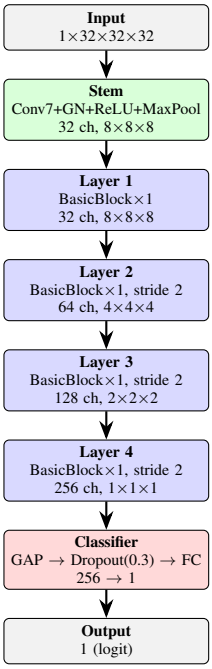

Fig. 6. 3D ResNet-18 architecture for nodule classification (3.60 M parameters).

The stem consists of a 7×7×7 convolution (stride 2) followed by GroupNorm, ReLU, and 3×3×3 max-pooling (stride 2), reducing the input from 32^3 to 8^3. Each of the four residual layers contains one BasicBlock with two 3×3×3 convolutions and a skip connection; layers 2–4 use stride-2 downsampling. The classifier head applies global average pooling, dropout (0.3), and a fully-connected layer to produce the final logit.

2) Training and Validation Strategy: The same 5-fold cross-validation scheme is used, with each fold holding out one subset for validation. Table III summarises the candidate data statistics across folds, highlighting the extreme class imbalance (∼1:440 positive-to-negative ratio in the raw data).

TABLE III
STAGE 2: 5-FOLD CROSS-VALIDATION CANDIDATE STATISTICS.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train subsets	{1,2,3,4}	{0,2,3,4}	{0,1,3,4}	{0,1,2,4}	{0,1,2,3}
Val subset	{0}	{1}	{2}	{3}	{4}
Train scans	356	356	356	356	356
Val scans	89	89	89	89	89
Train total candidates	298,003	306,126	302,680	301,188	300,555
Train positives	679	647	636	659	647
Train negatives	297,324	305,479	302,044	300,529	299,908
Per-epoch samples	1,358	1,294	1,272	1,318	1,294
Val total candidates	79,135	71,012	74,458	75,950	76,583
Val positives	138	170	181	158	170
Val negatives	78,997	70,842	74,277	75,792	76,413
Val samples (10:1)	1,518	1,870	1,991	1,738	1,870

Training. The 3D ResNet-18 is trained for 30 epochs with a batch size of 64 using AdamW optimizer (weight decay 10^{-4}) and OneCycleLR scheduler (max LR $= 10^{-3}$, 10% warmup, cosine annealing). The loss function is binary cross-entropy (BCE). To handle the severe class imbalance, I adopt the following sampling strategy for each epoch:

1) All positive candidates (n_{pos}) are included.
2) An equal number of negatives (n_{pos}) are randomly sampled *without replacement* from the full negative pool, using a different random seed per epoch.
3) The balanced set ($2 \times n_{\text{pos}} \approx 1{,}300$ samples) is ordered by scan ID (scan-level shuffling) to improve cache locality during I/O, then fed through the DataLoader.

Since each epoch sees only ∼0.2% of all negatives, approximately 440 epochs would be needed to cycle through the entire negative pool; however, 30 epochs already provide sufficient diversity. Data augmentation consists of random 3D flips along Z, Y, and X axes (each with 50% probability).

Validation. After each epoch, the model is evaluated on a *fixed* validation subset (deterministic seed per fold). To approximate a realistic class distribution while keeping evaluation tractable, I subsample negatives at a 10:1 ratio relative to positives (e.g., Fold 1: 138 pos + 1,380 neg = 1,518 total). The following metrics are computed using scikit-learn:

- **Accuracy**: overall correct predictions.
- **Precision** and **Recall**: binary metrics with threshold 0.5.
- **F1 Score**: harmonic mean of precision and recall.
- **AUC-ROC**: area under the receiver operating characteristic curve, computed from continuous probabilities (sigmoid output).

The best checkpoint is selected based on the highest validation AUC-ROC, as it is threshold-independent and more robust for imbalanced datasets than loss-based selection. Figures 7–**??** show the training curves.

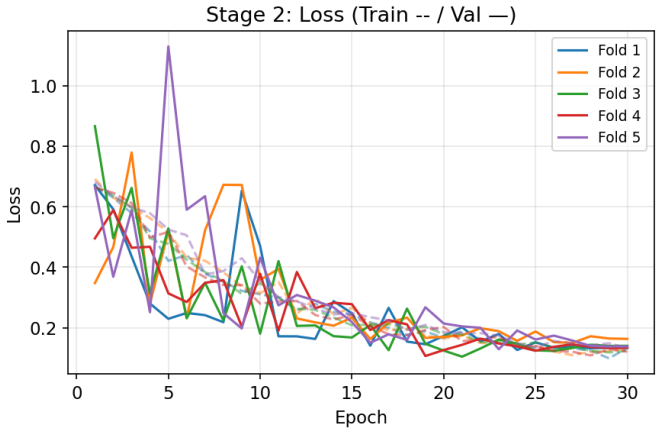

Fig. 7. Stage 2: Training and validation loss.

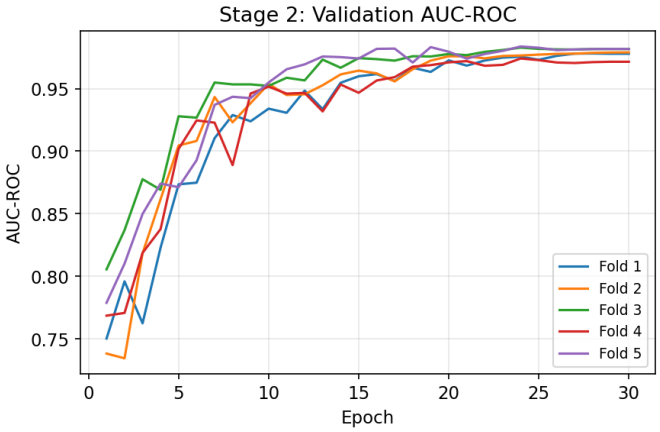

Fig. 8. Stage 2: Validation AUC-ROC.

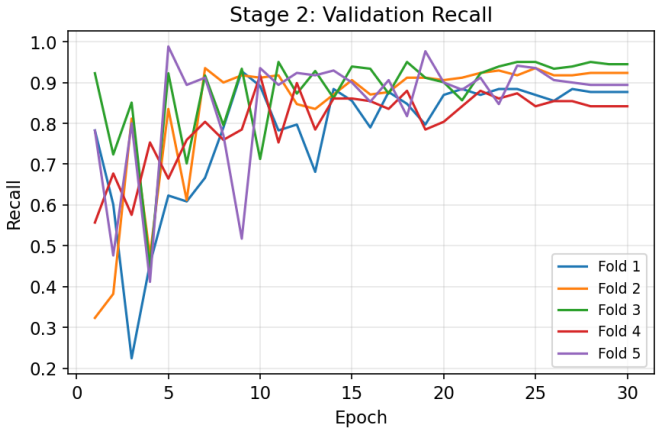

Fig. 11. Stage 2: Validation Recall.

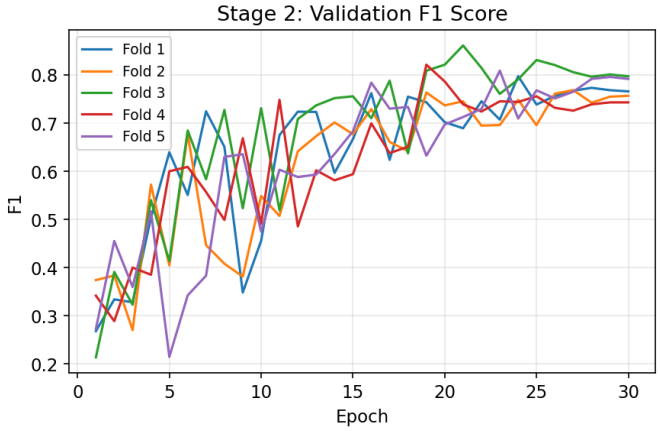

Fig. 9. Stage 2: Validation F1 Score.

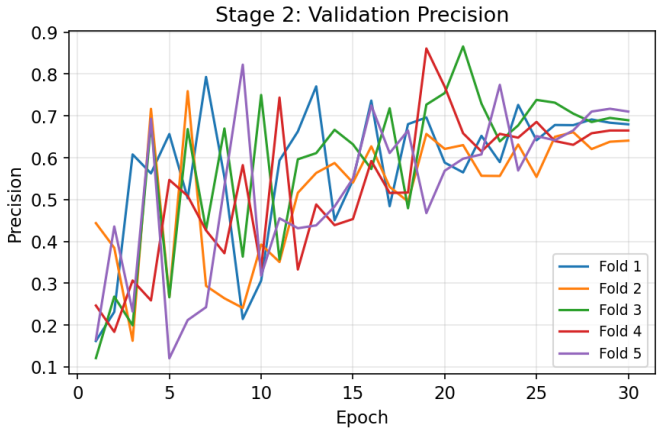

Fig. 10. Stage 2: Validation Precision.