



Detection and Classification of Pulmonary Nodules



Realized by :

NeuralMasters

Summary

01 Introduction

02 Business & Data Science Objectives

03 Models Breakdown

04 Demo

05 Conclusion

Introduction

PneumaTect is an AI-driven platform that automates the detection, classification, and analysis of **pulmonary nodules** from medical imaging, aimed at enhancing early lung cancer diagnosis and assisting radiologists.

In this presentation, we will focus on the **modeling phase** of our platform, outlining **the architecture of our AI models**, the techniques used for training, and the performance metrics that guide our **model evaluation**. We will also address the challenges faced during model development and the solutions implemented to optimize accuracy and reliability.

Business Objectives



Improve early detection & diagnosis of lung cancer.



Reduce radiologists' workload by providing AI-based assistance.



Enhance diagnostic accuracy and patient outcomes.



Facilitate personalized treatment strategies using predictive models.



Optimize Healthcare Costs

Data Science Objectives



Pulmonary Nodule Detection



Pulmonary Nodule Classification



Lung Cancer Subtype Classification



Cancer Stage Estimation



Cancer Risk Prediction



Medical Image Enhancement



Lung Cancer Detection



Predict Lung Function Decline



Predict Lung Cancer Mortality



Personalized AI Support

MODELS BREAKDOWN

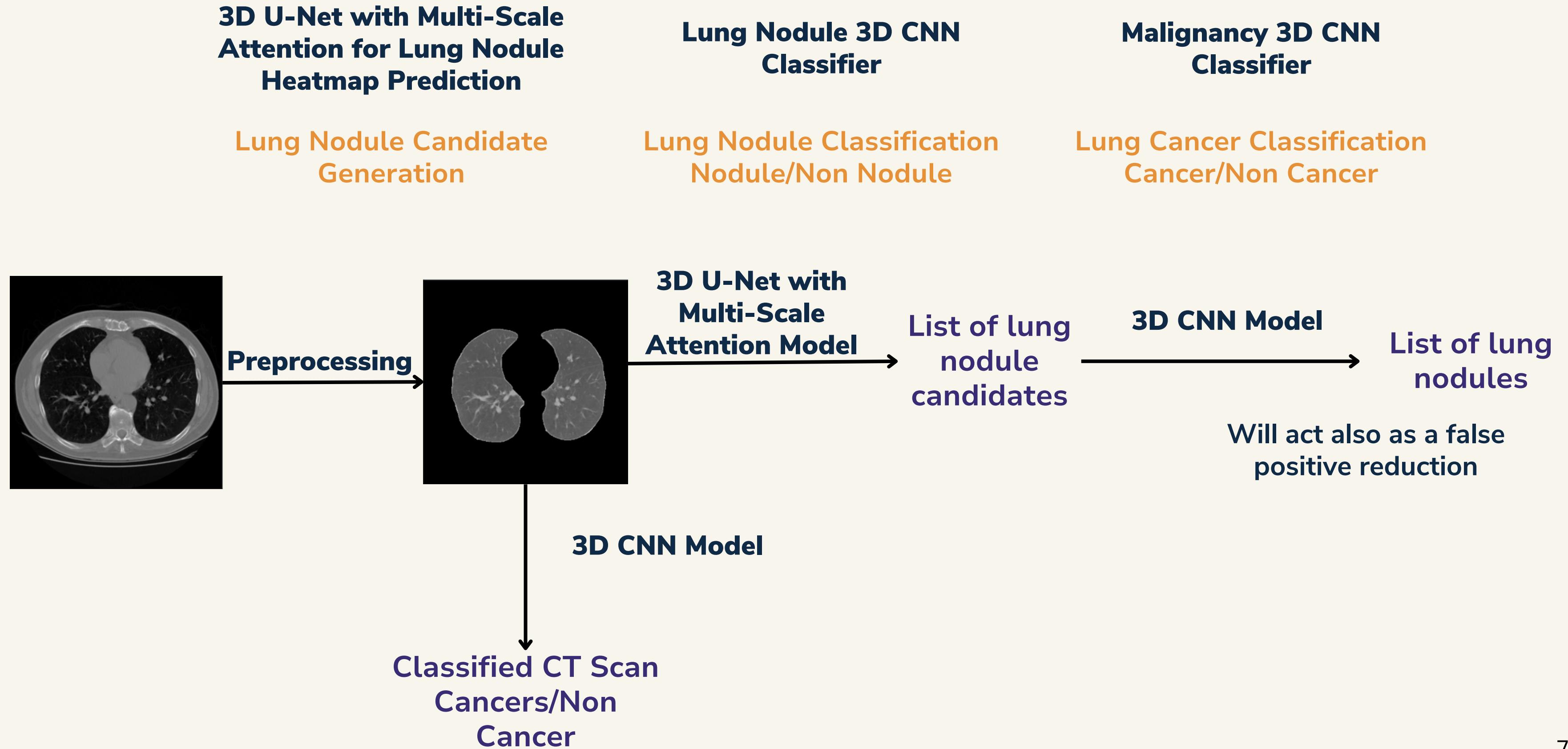


COMPUTER-AIDED DIAGNOSIS (CAD) SYSTEM FOR LUNG CANCER CLASSIFICATION



Computer-aided diagnosis (CAD) system for lung cancer classification

The main system for lung cancer classification is composed of 3 models



3D U-Net with Multi-Scale Attention for Lung Nodule Heatmap Prediction

Dataset: Luna16

Data Science Objective: Pulmonary Nodule Detection

Data Preprocessing Recap:

Voxel Spacing Resampling: Input CT scans, which exhibit anisotropic voxel spacing, are resampled to a uniform isotropic spacing

Hounsfield Unit (HU) Clipping and Normalization: Voxel intensities are clipped to a clinically relevant range (e.g., -1000 HU to +400 HU). Subsequently, the clipped HU values are normalized to a standard range.

Lung Segmentation (Masking): A rough lung mask is generated primarily through HU thresholding, reducing irrelevant information and computational load.

Patch Extraction: During training, 3D patches of a fixed size (96x96x96 voxels) are extracted from the preprocessed volumes.

Data Split Strategy:

Data Split: The LUNA16 dataset is split into training and validation sets based on the unique patient identifier (seriesuid). This patient-level split is crucial to prevent data leakage, ensuring that patches from the same CT scan do not appear in both training and validation sets.

We used a split of 90% train, 10% validation

Cross Validation: We will apply cross validation when we train the model on powerful computation on the whole dataset

Model Overview:

The model predicts a voxel-wise heatmap where high activation values indicate a higher probability of a nodule center.
It takes a 3D patch of a preprocessed CT scan as input and outputs a 3D heatmap of the same spatial dimensions.

Hyperparameters:



The choice of hyperparameters was based on research papers where the same hyperparameters were consistently used across multiple studies.

We limited the number of scans that we trained on locally to 50 scan. we aim to train the model on the whole dataset on the *Nvidia A100 80Gb GPU

Learning Rate Scheduling:
ReduceLROnPlateau

Hyperparameters	NVIDIA GeForce RTX 2050	Nvidia A100 80Gb
Learning Rate (LR)	1e-4	1e-4
Batch Size	2	16
Number of Epochs	150	200
Optimizer	AdamW	AdamW
Loss Function	BCEWithLogitsLoss	BCEWithLogitsLoss
Weight Decay	1e-5	1e-5
NMS THRESHOLD	0.1	0.2
POSITIVE WEIGHT	10.0	10.0
Patch Size	(96, 96, 96)	(128, 128, 128)

Model Architecture:

The core architecture is a 3D U-Net, enhanced with integrated Multi-Scale Attention (MSA) blocks, specifically a variant inspired by the Convolutional Block Attention Module (CBAM).

- The chosen base architecture is the 3D U-Net, renowned for its effectiveness in biomedical volumetric images segmentation tasks due to its symmetric encoder-decoder structure and skip connections.
- To further enhance performance, Multi-Scale Attention (MSA) mechanisms, specifically a 3D adaptation of CBAM, are integrated into the network's convolutional blocks, to improve feature representation and focus on salient regions

Why This Model Was Chosen?

Performance: U-Net architectures with attention often achieve competitive performance on volumetric medical imaging tasks, providing a strong baseline.

Interpretability/Flexibility: The heatmap output offers interpretable confidence scores per voxel and allows flexible threshold adjustment post-deployment based on clinical needs

Weaknesses of Chosen Model & Mitigation:

False Positives: The model will inevitably generate false positives.

Mitigation: We will Pass the model output to Lung Nodule classification Model to reduce false positive results

Dependency on Preprocessing: Performance hinges on consistent preprocessing.

Mitigation: Standardized and rigorous preprocessing pipeline

Overview of the U-Net structure:

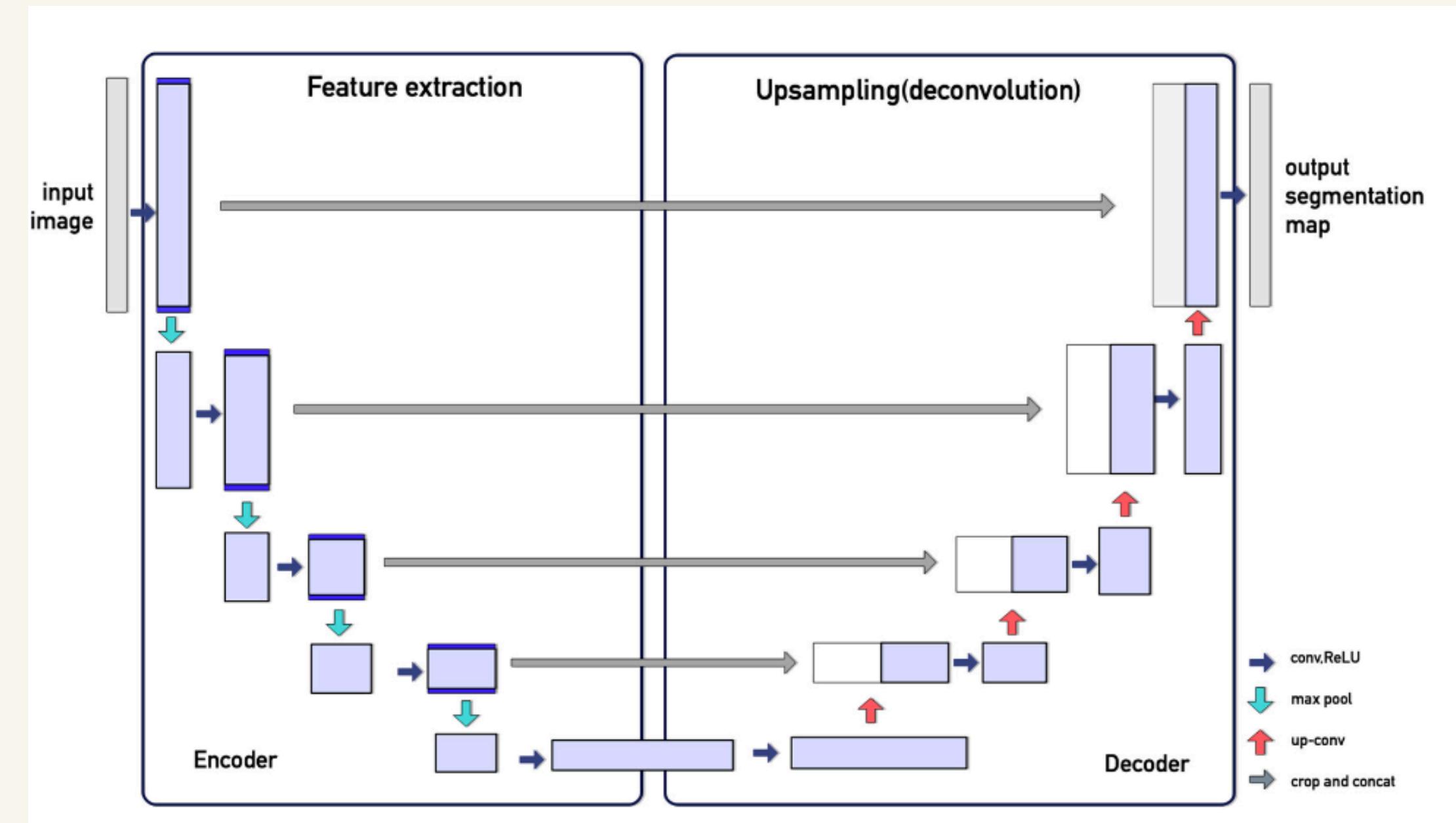
The symmetric architecture of U-Net allows it to simultaneously capture both global and local image features, effectively balancing high-resolution detail with broader contextual information, which makes it particularly well-suited for precise segmentation tasks.

Why It's Ideal for Lung Nodule Detection:

- Accurately segments small, irregularly shaped nodules.
- Maintains 3D anatomical context crucial for distinguishing nodules from vessels or noise.
- Performs well with limited annotated data, typical in medical imaging.

Key Components:

- 3D Convolutions
- Encoder-Decoder Structure
- Skip Connections

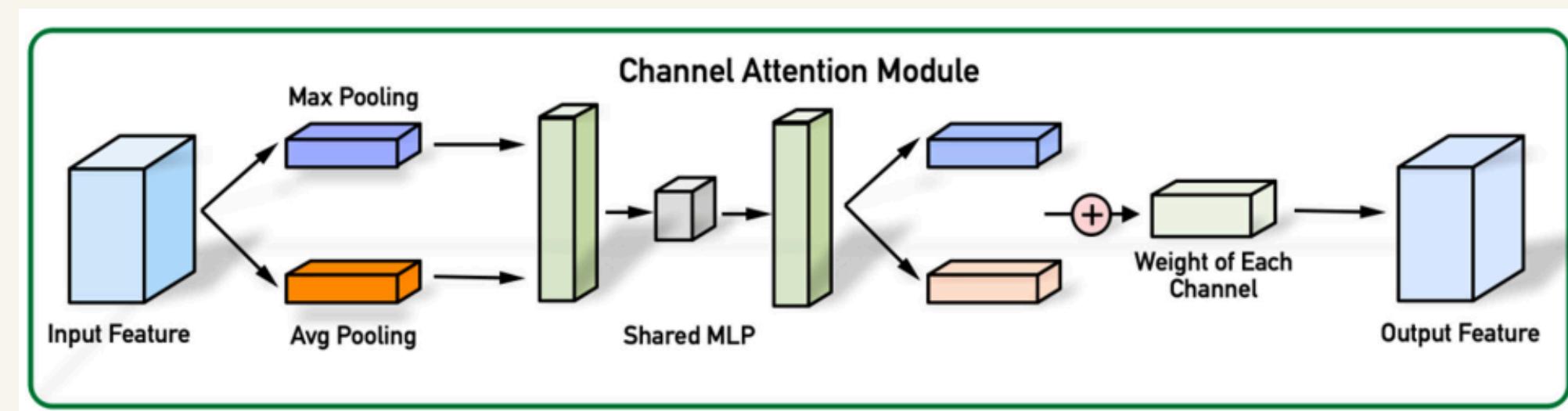
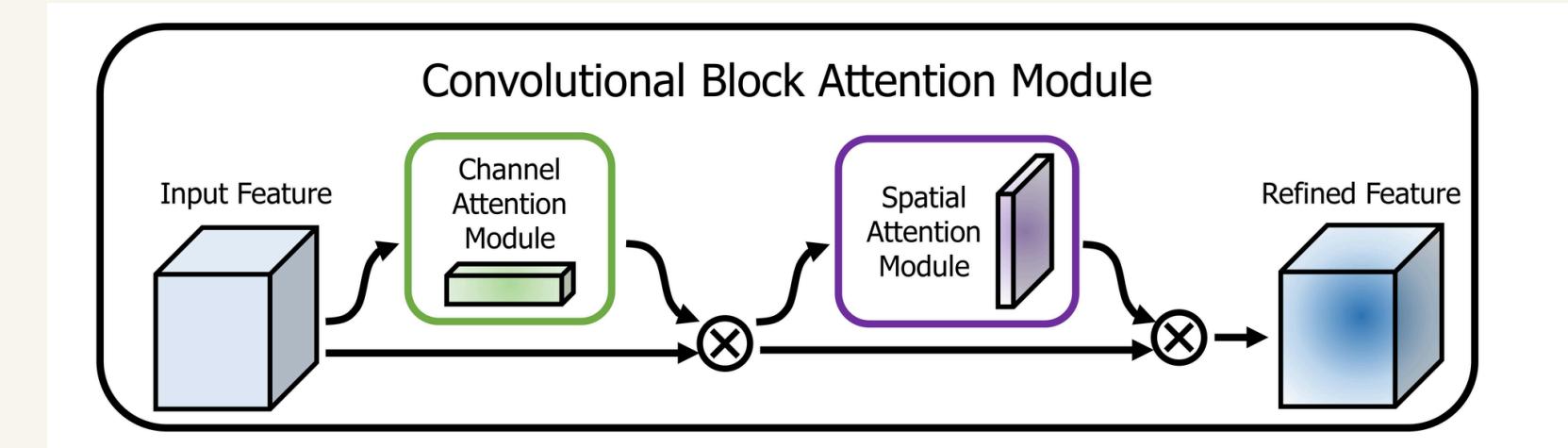


Overview of the CBAM mechanism:

The CBAM-Inspired Multi-Scale Attention (MSA) mechanism enhances feature representation by selectively focusing on both spatial and channel-wise attention across multiple scales. This enables the model to capture fine-grained details and broader contextual information.

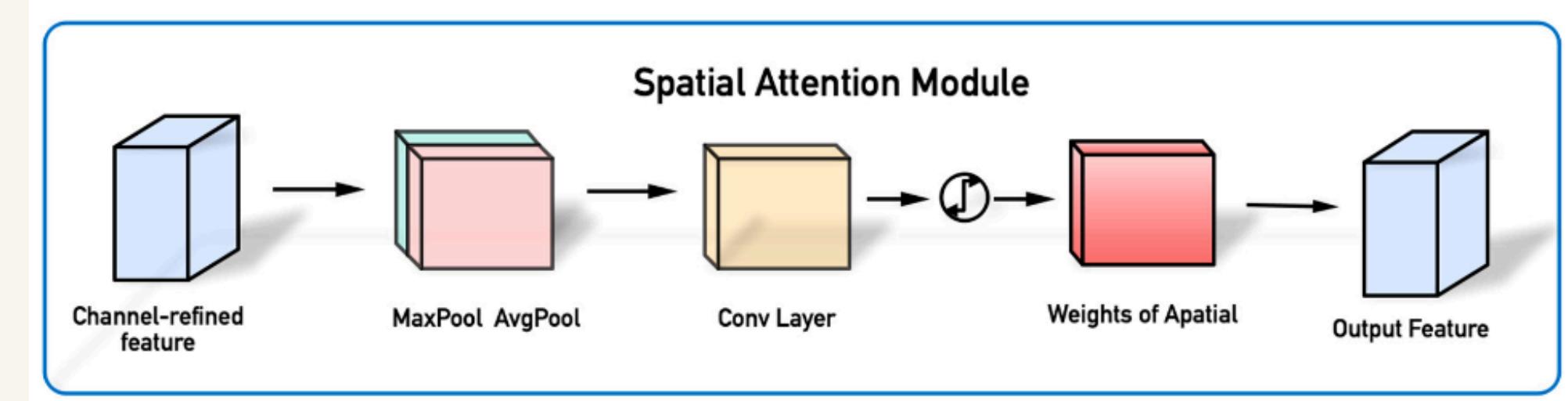
Why It's Ideal for Lung Nodule Detection:

- Refines U-Net outputs by focusing on Nodule structures.
- Improves sensitivity to small and subtle nodules.
- Light weight and easy to integrate. improves performance without excessive computation.



Key Components:

- Channel Attention
- Spatial Attention
- Multi-Scale Fusion



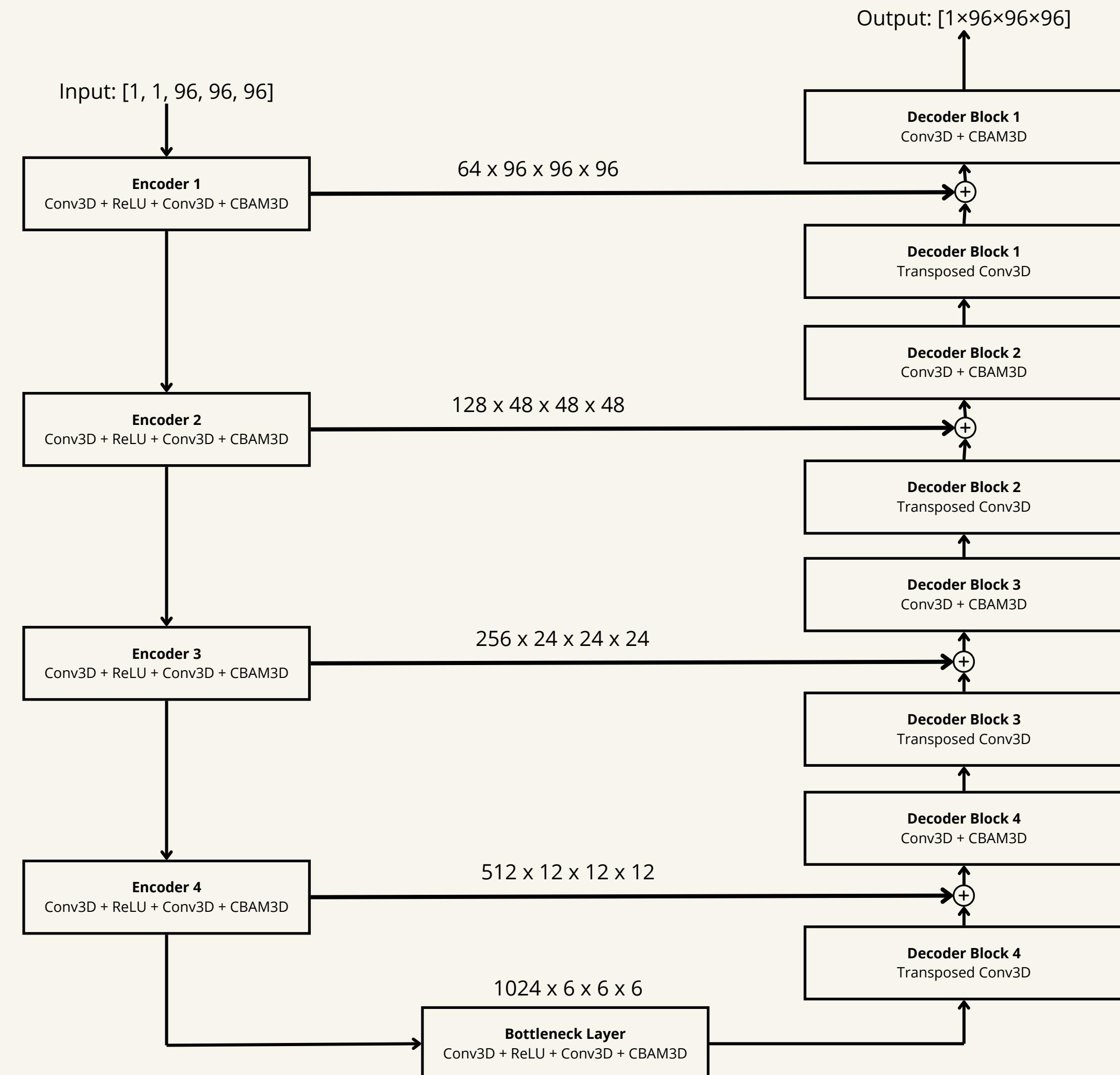
Overview of our 3D UNet-CBAM Architecture

The model architecture is based on an enhanced 3D U-Net framework, integrating Convolutional Block Attention Modules (CBAM3D) at every stage to improve feature refinement. It consists of four encoder blocks and four decoder blocks, connected through skip connections to preserve spatial information across different scales.

Each encoder block performs two 3D convolutions followed by instance normalization, ReLU activation, and CBAM3D attention, while the decoder path employs transposed convolutions for upsampling and similarly structured convolutional blocks. A deep bottleneck with 1024 feature channels captures high-level volumetric representations, enabling the model to accurately localize lung nodules.

Total trainable parameters:
90,514,199*

- **Where most parameters are?**
- Bottleneck (Middle 1024 feature block):
42 Million parameters
- Deeper Layers (512, 256 channels):
Each deep layer is very heavy compared to shallow ones (64 channels).



Evaluation Metrics:

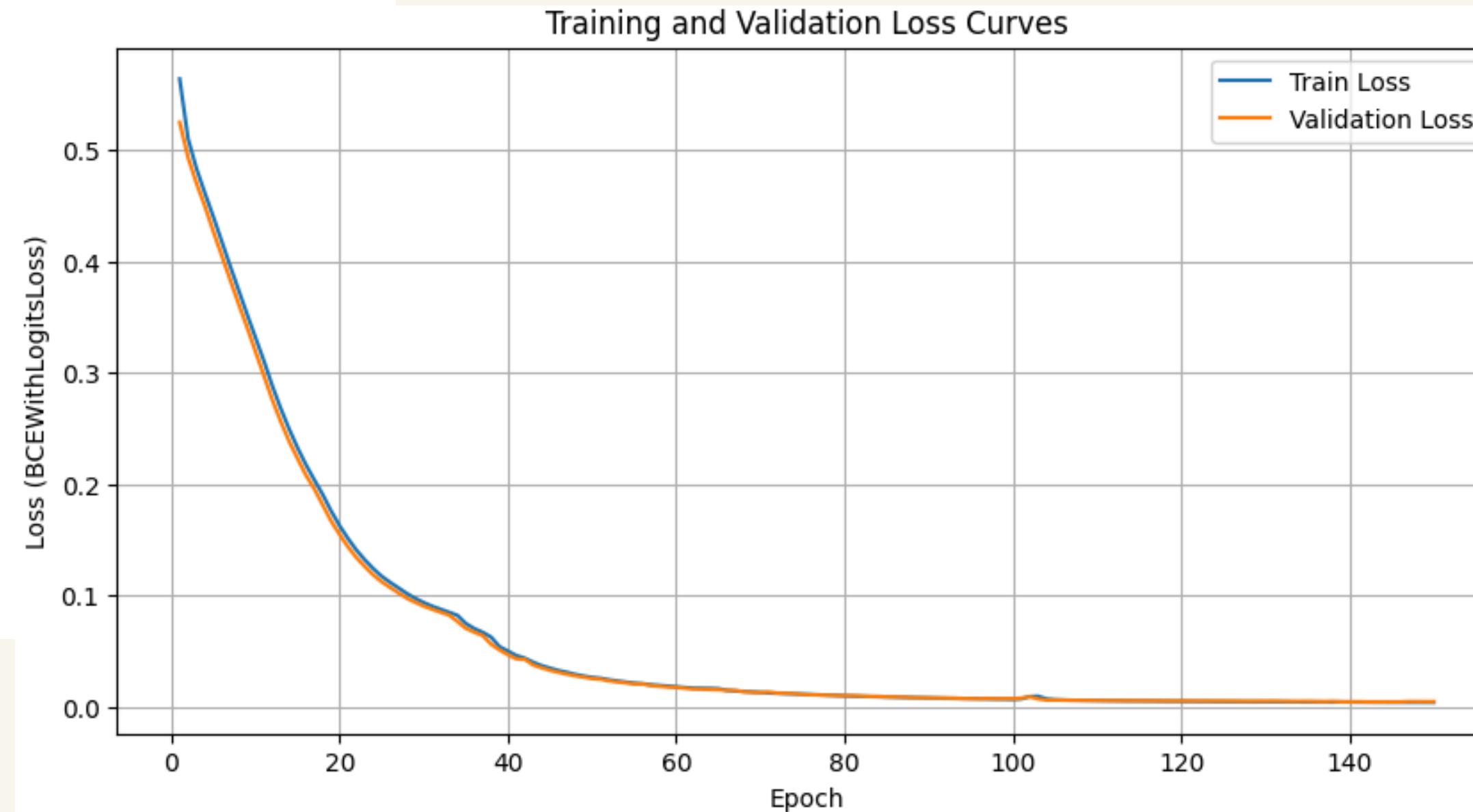
Best Validation Loss: 0.004618

Sensitivity: 0.4286

```
--- Evaluating at fixed threshold: 0.1 ---  
Starting evaluation on 4 scans with 7 GT nodules and 718 predictions.  
Matching GT and Predictions: 0% | 0/4 [00:00<?, ?it/s]
```

Evaluation Summary:
total_gt_nodules: 7
total_predictions: 718
true_positives: 3
false_positives: 715
false_negatives: 4
sensitivity: 0.4286
average_fp_per_scan: 178.7500
distance_threshold_mm: 5.0000
num_scans_evaluated: 4

Confusion Matrix (Candidate Level):
True Positives (Detected Nodules): 3
False Positives (Incorrect Detections): 715
False Negatives (Missed Nodules): 4



This Model Was Trained on 45 scans (162 total patch samples. with Approx. 57 patches centred on nodules) and Validated on 5 scans(57 total patch samples. with Approx. 7 patches centred on nodules).

We will compare our model metrics against those of other models after training it on the full dataset.
+ Cross-Validation

Model Inference:

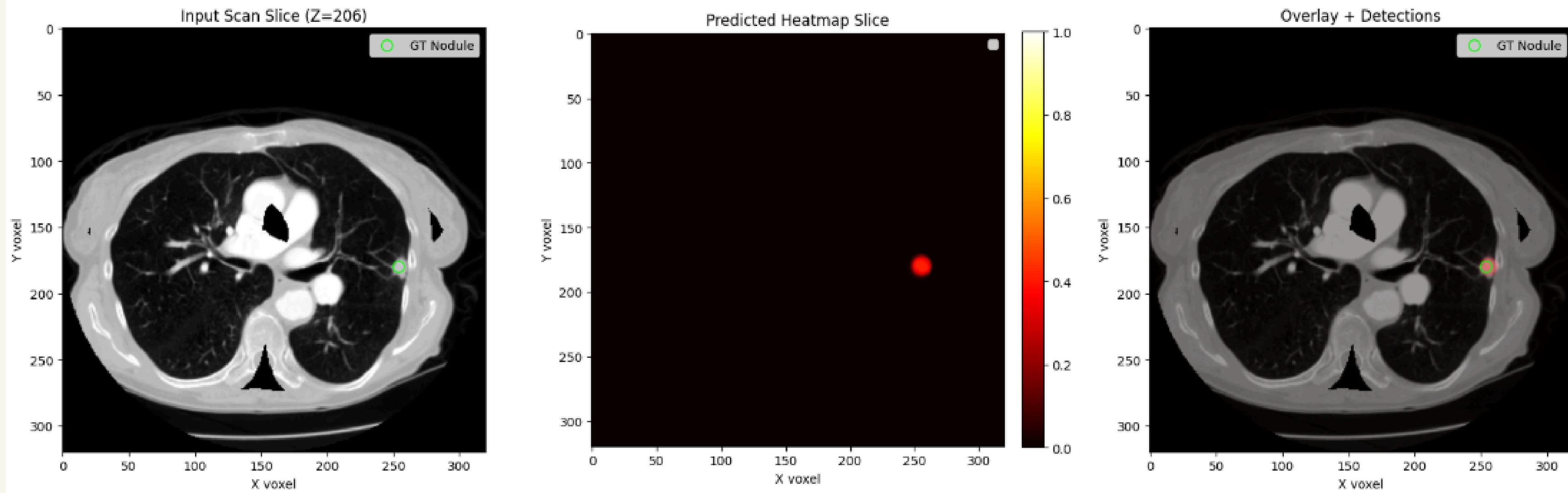
Found 158 candidate nodules in 1.3.6.1.4.1.14519.5.2.1.6279.6001.219909753224298157409438012179.

Inference took 7.64 seconds.

Example detections:

Coords (X,Y,Z): (117.90, 21.00, -154.00), Prob: 0.4114
Coords (X,Y,Z): (182.90, -65.00, -100.00), Prob: 0.4109
Coords (X,Y,Z): (182.90, 159.00, -100.00), Prob: 0.4104
Coords (X,Y,Z): (-41.10, 159.00, -100.00), Prob: 0.4104
Coords (X,Y,Z): (-41.10, 159.00, -357.00), Prob: 0.4104

Inference Example - Scan: 1.3.6.1.4.1.14519.5.2.1.6279.6001.219909753224298157409438012179



The coordinates are based on the input CT scan, which will assist in accurately visualizing the nodule's location during model inference.

For instance, this allows us to show the doctor the precise location of the lung nodule, enabling them to conduct their own analysis.

*The number of candidate nodules can be reduced by clustering nodules with very close coordinates.
And augmenting the threshold.

Nodule 3D CNN Classifier

Dataset: Luna16

Data Science Objective: Pulmonary Nodule Classification

Data Preprocessing Recap:

Voxel Spacing Resampling: Input CT scans, which exhibit anisotropic voxel spacing, are resampled to a uniform isotropic spacing

Hounsfield Unit (HU) Clipping and Normalization: Voxel intensities are clipped to a clinically relevant range (e.g., -1000 HU to +400 HU). Subsequently, the clipped HU values are normalized to a standard range.

Lung Segmentation (Masking): A rough lung mask is generated primarily through HU thresholding, reducing irrelevant information and computational load.

Patch Extraction: During training, 3D patches of a fixed size (32x32x32 voxels) are extracted from the preprocessed volumes.

Data Split Strategy:

Data Split:

We used a split of 80% train, 20% validation

Nodule 3D CNN Classifier Model:

A relatively standard 3D CNN architecture designed for volumetric image classification. It follows a common pattern of convolutional blocks followed by fully connected layers.

```
Simple3DCNN(  
    conv_layers): Sequential(  
        (0): Conv3d(1, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))  
        (1): ReLU()  
        (2): MaxPool3d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
        (3): Conv3d(32, 64, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))  
        (4): ReLU()  
        (5): MaxPool3d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
        (6): Conv3d(64, 128, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))  
        (7): ReLU()  
        (8): MaxPool3d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    )  
    fc_layers): Sequential(  
        (0): Flatten(start_dim=1, end_dim=-1)  
        (1): Linear(in_features=8192, out_features=256, bias=True)  
        (2): ReLU()  
        (3): Dropout(p=0.5, inplace=False)  
        (4): Linear(in_features=256, out_features=1, bias=True)  
    )
```

Hyperparameters:

Input Shape: (32, 32, 32)

Batch Size: BATCH_SIZE = 16.

Learning Rate: 0.001.

Epochs: 5

Optimizer: torch.optim.Adam

Loss Function: torch.nn.BCEWithLogitsLoss.

Added 112 positive samples.

Added 224 negative samples.

Total samples for dataset: 336

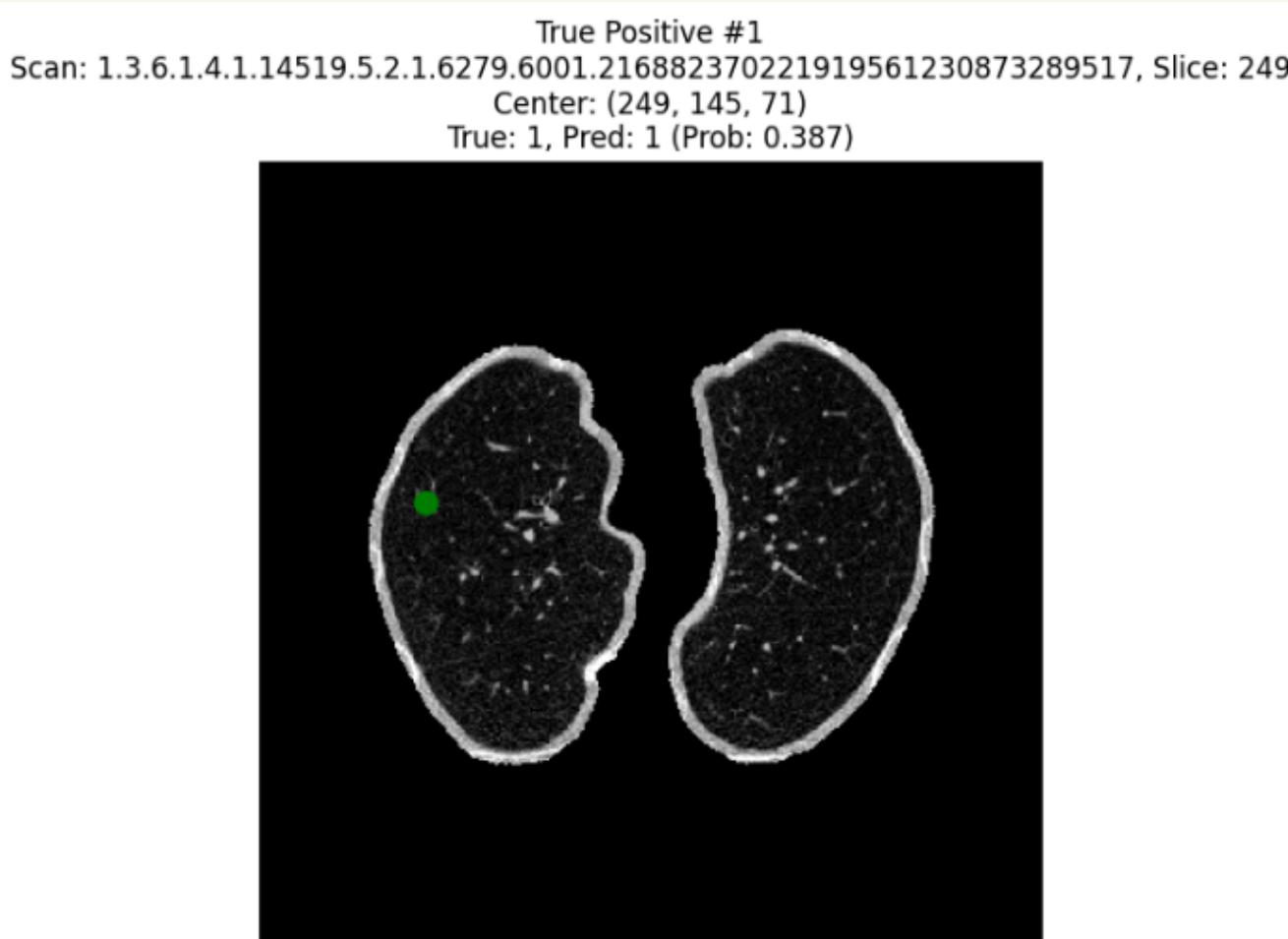
Final Validation Loss: 0.6448
Final Validation Accuracy: 0.6618
Optimal Threshold (Youden's J): 0.3867

Confusion Matrix:

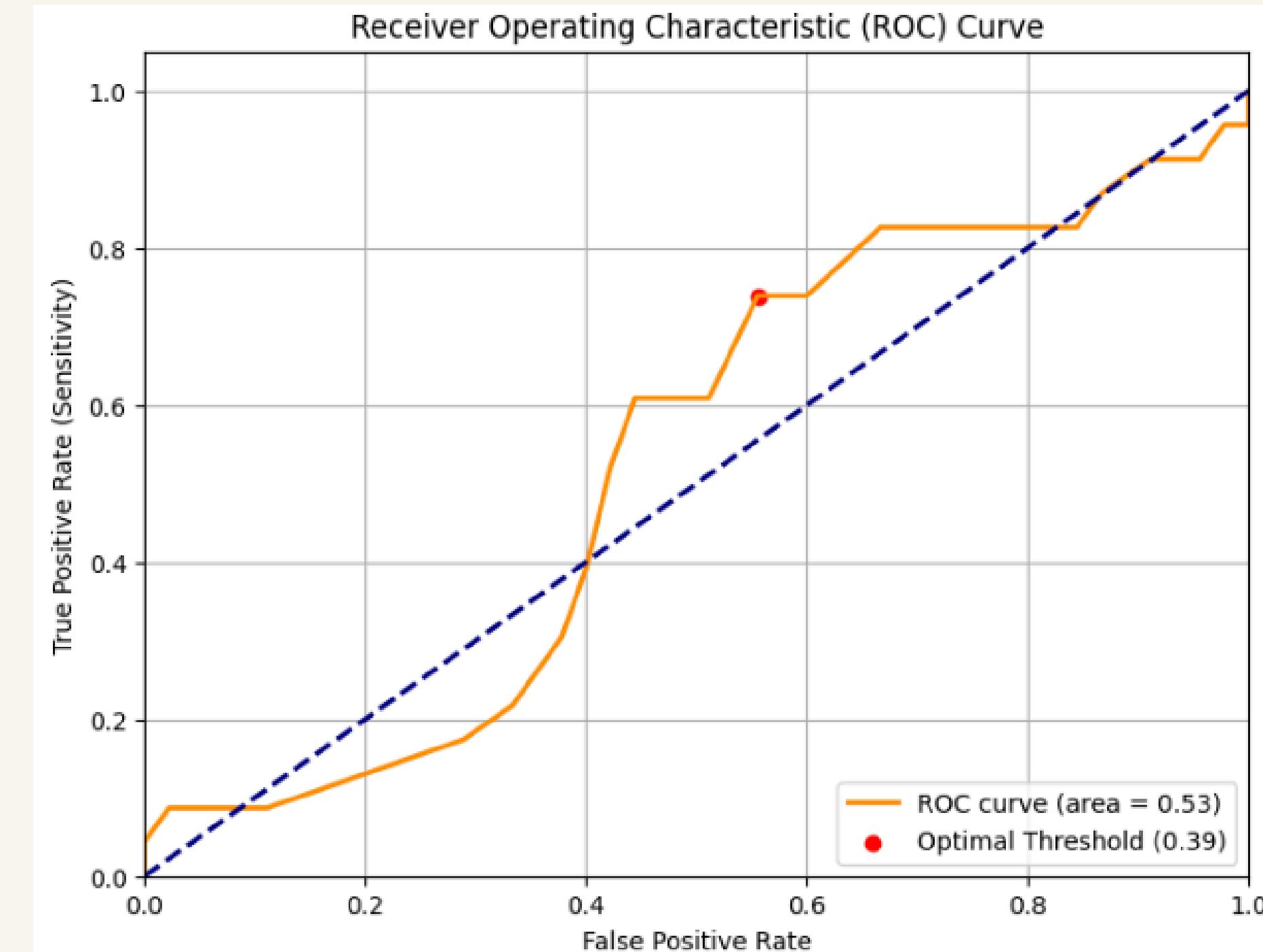
```
[[20 25]
 [ 6 17]]
```

Metrics at threshold 0.3867:

Sensitivity (Recall):	0.7391
Specificity:	0.4444
Precision:	0.4048
F1-Score:	0.5231
AUC-ROC:	0.5266



Nodule 3D CNN Classifier



Malignancy 3D CNN Classifier

Dataset: Data Science Bowl 2017

Data Science Objective: Lung Cancer Detection

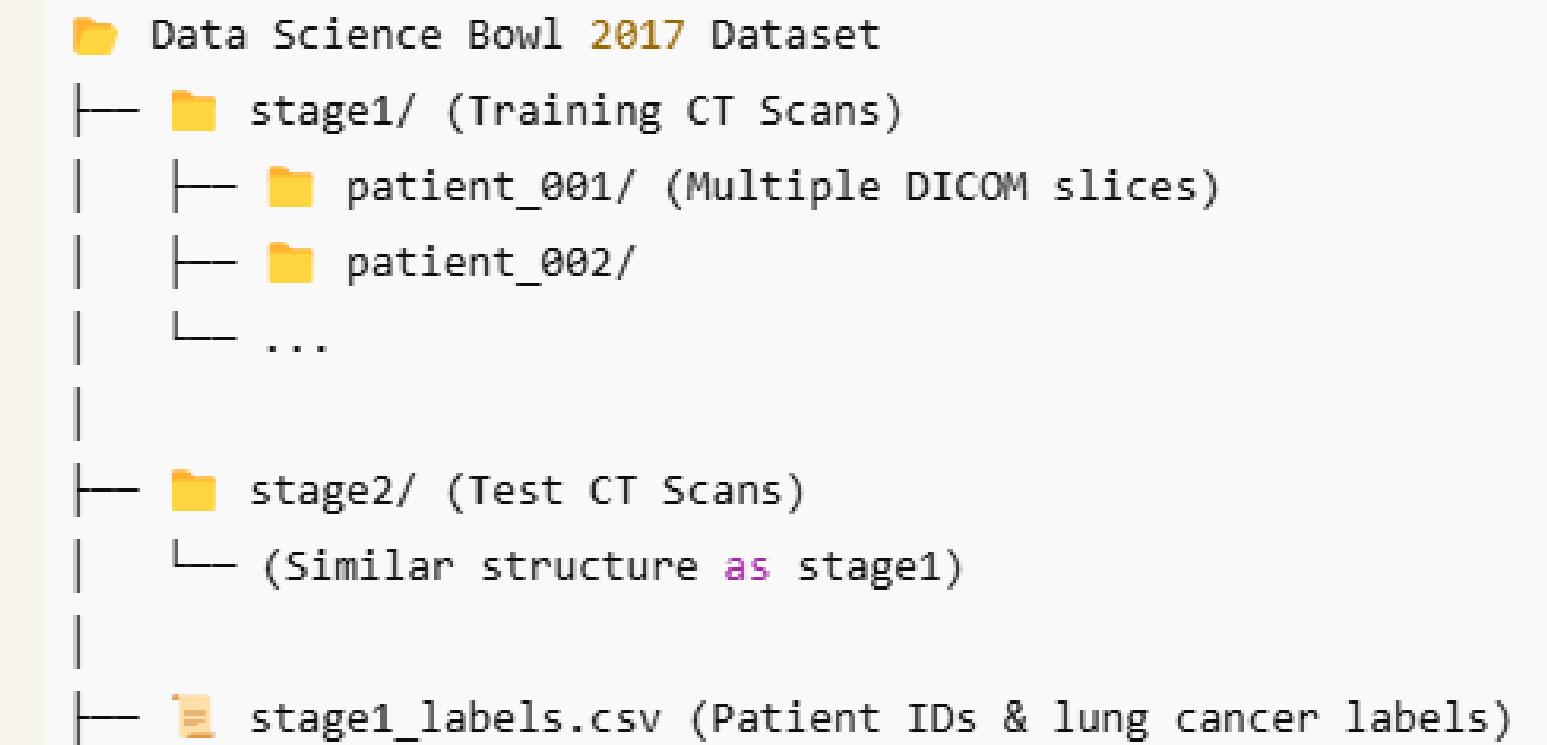
Data Preprocessing Recap:

Voxel Spacing Resampling: Input CT scans, which exhibit anisotropic voxel spacing, are resampled to a uniform isotropic spacing

Hounsfield Unit (HU) Clipping and Normalization: Voxel intensities are clipped to a clinically relevant range (e.g., -1000 HU to +400 HU). Subsequently, the clipped HU values are normalized to a standard range.

Lung Segmentation (Masking): A rough lung mask is generated primarily through HU thresholding, reducing irrelevant information and computational load.

Patch Extraction: During training, 3D patches of a fixed size (96x96x96 voxels) are extracted from the preprocessed volumes.



Data Split Strategy:

Data Split:

We used a split of 80% train, 20% validation

PatientLevel3DCNN Model:

A relatively standard 3D CNN architecture designed for volumetric image classification. It follows a common pattern of convolutional blocks followed by fully connected layers.

```
PatientLevel3DCNN(  
    conv_layers: Sequential(  
        (0): Conv3d(1, 16, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))  
        (1): BatchNorm3d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (2): ReLU()  
        (3): MaxPool3d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
        (4): Conv3d(16, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))  
        (5): BatchNorm3d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (6): ReLU()  
        (7): MaxPool3d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
        (8): Conv3d(32, 64, kernel_size=(3, 3, 3), stride=(1, 1, 1), padding=(1, 1, 1))  
        (9): BatchNorm3d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
        (10): ReLU()  
        (11): MaxPool3d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
        (12): AdaptiveMaxPool3d(output_size=(2, 2, 2))  
    )  
    fc_layers: Sequential(  
        (0): Flatten(start_dim=1, end_dim=-1)  
        (1): Linear(in_features=512, out_features=256, bias=True)  
        (2): ReLU()  
        (3): Dropout(p=0.5, inplace=False)  
        (4): Linear(in_features=256, out_features=1, bias=True)  
    )  
)
```

Hyperparameters:

Input Shape: (96, 128, 128)

Batch Size: BATCH_SIZE = 4.

Learning Rate: 0.0001.

Epochs: 150

Optimizer: torch.optim.Adam

Loss Function: torch.nn.BCEWithLogitsLoss.

Final Validation Loss: 0.5864

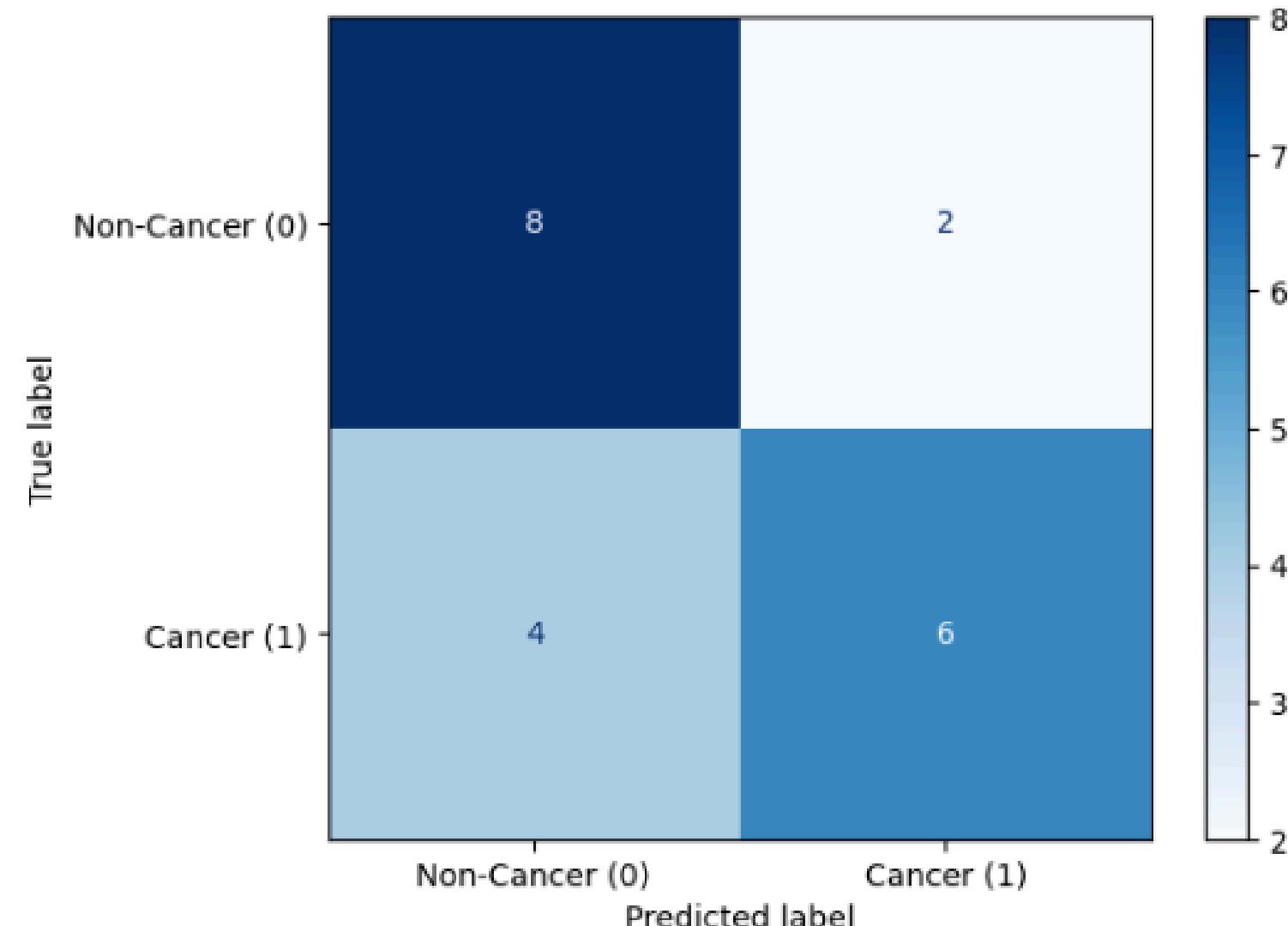
--- Final Validation Metrics (Limited Data) ---

Accuracy: 0.7000
Precision: 0.7500
Recall: 0.6000
F1-Score: 0.6667
AUC-ROC: 0.7500

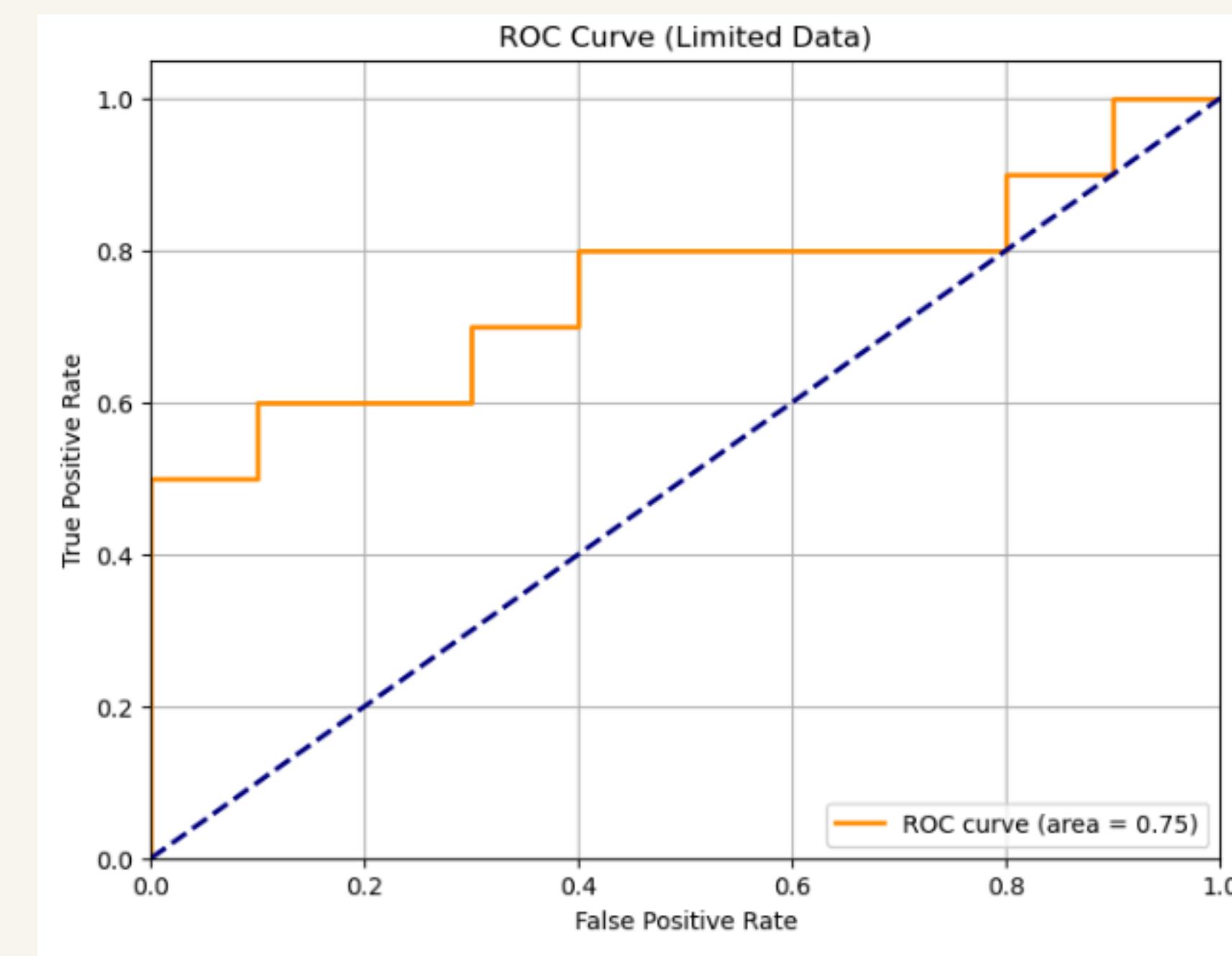
Classification Report (Limited Data):

	precision	recall	f1-score	support
Non-Cancer (0)	0.67	0.80	0.73	10
Cancer (1)	0.75	0.60	0.67	10
accuracy			0.70	20
macro avg	0.71	0.70	0.70	20
weighted avg	0.71	0.70	0.70	20

Confusion Matrix (Limited Data):



PatientLevel3DCNN Evaluation



Simp3DNet Model:

This model is defined and trained in the second part of the script, notably changing the target scan size and the network architecture.

```
Simp3DNet(  
    (conv1a): Conv3d(1, 16, kernel_size=(9, 9, 9), stride=(1, 1, 1))  
    (bn1a): BatchNorm3d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (conv1b): Conv3d(16, 16, kernel_size=(3, 3, 3), stride=(1, 1, 1))  
    (bn1b): BatchNorm3d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (conv1c): Conv3d(16, 16, kernel_size=(5, 5, 5), stride=(1, 1, 1))  
    (bn1c): BatchNorm3d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (pool1): MaxPool3d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (conv2a): Conv3d(16, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1))  
    (bn2a): BatchNorm3d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (conv2b): Conv3d(32, 32, kernel_size=(3, 3, 3), stride=(1, 1, 1))  
    (bn2b): BatchNorm3d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (pool2): MaxPool3d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (conv3a): Conv3d(32, 64, kernel_size=(3, 3, 3), stride=(1, 1, 1))  
    (bn3a): BatchNorm3d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (conv3b): Conv3d(64, 64, kernel_size=(3, 3, 3), stride=(1, 1, 1))  
    (bn3b): BatchNorm3d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (conv4a): Conv3d(64, 128, kernel_size=(3, 3, 3), stride=(1, 1, 1))  
    (bn4a): BatchNorm3d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (flatten): Flatten(start_dim=1, end_dim=-1)  
    (fc1): Linear(in_features=8192, out_features=256, bias=True)  
    (bn_fc1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (fc2): Linear(in_features=256, out_features=1, bias=True)  
)
```

Hyperparameters:

Input Shape: (64, 64, 64)

Batch Size: 4

Learning Rate: 4e-5

Epochs: EPOCHS = 150

Optimizer: torch.optim.Adam

Loss Function: torch.nn.BCEWithLogitsLoss

Final Validation Loss: 0.5714

--- Final Validation Metrics (Simp3DNet, 64cube) ---

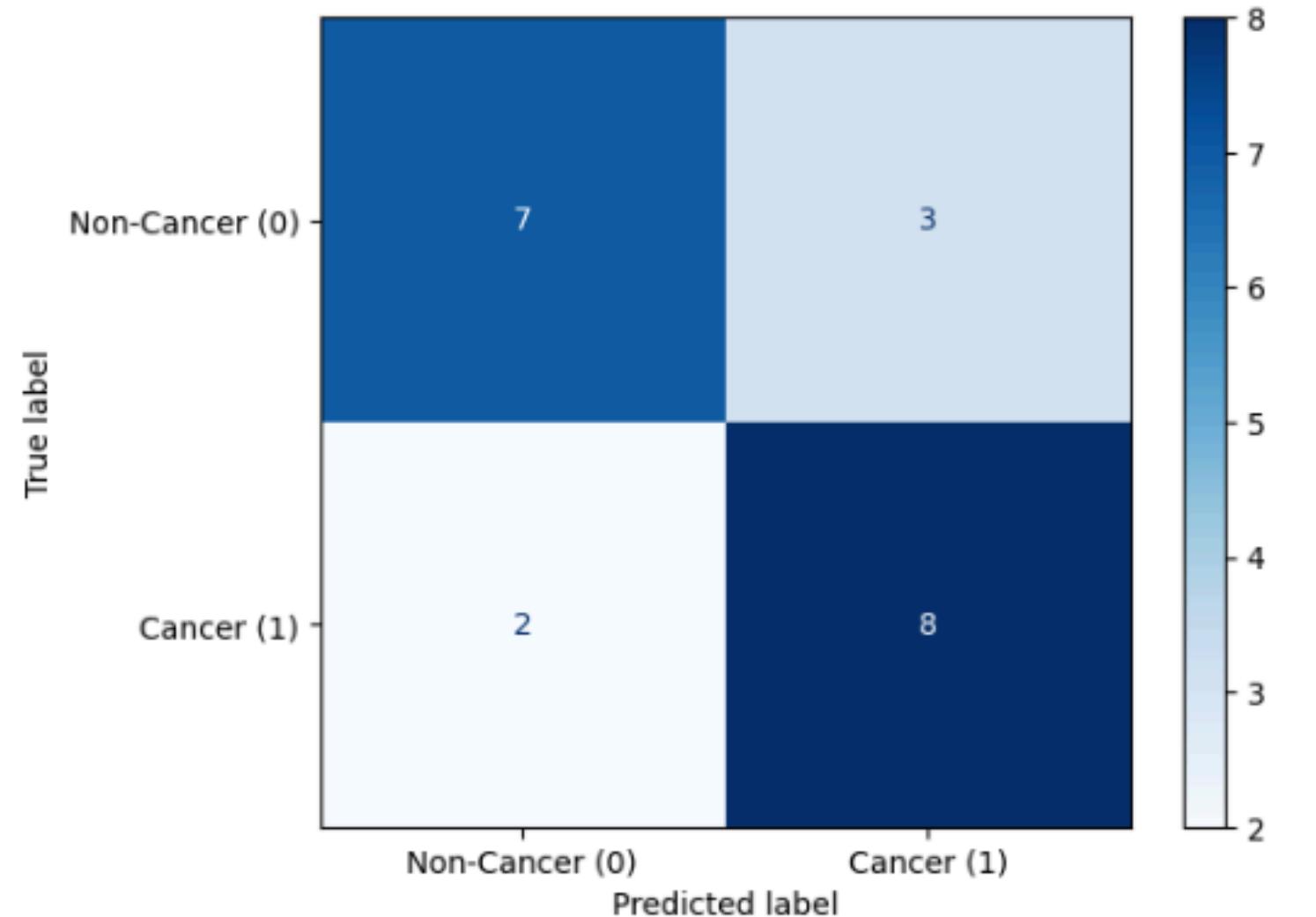
Accuracy: 0.7500
Precision: 0.7273
Recall: 0.8000
F1-Score: 0.7619
AUC-ROC: 0.8000

Classification Report (Simp3DNet, 64cube):

	precision	recall	f1-score	support
Non-Cancer (0)	0.78	0.70	0.74	10
Cancer (1)	0.73	0.80	0.76	10
accuracy			0.75	20
macro avg	0.75	0.75	0.75	20
weighted avg	0.75	0.75	0.75	20

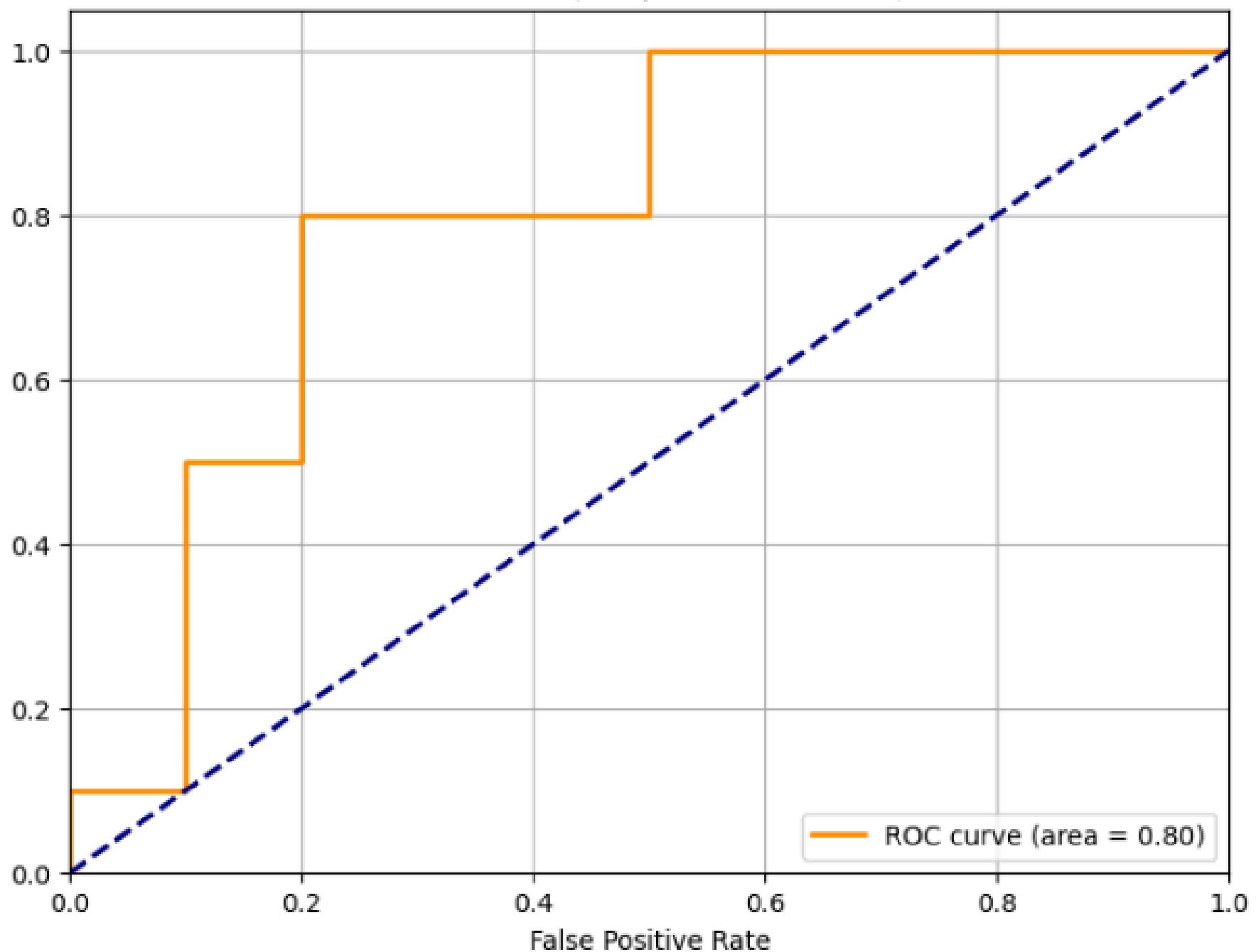
Confusion Matrix (Simp3DNet, 64cube):

Confusion matrix saved to ./preprocessed_dsb_50_each_64cube/simp3dnet_confusion_matrix_64cube.png

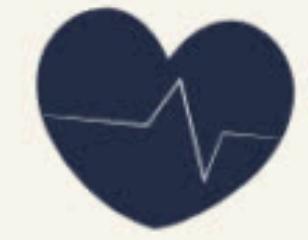


Simp3DNet Evaluation

ROC Curve (Simp3DNet, 64cube)



ADDITIONAL MODELS



Lung Cancer Classification using CT Scan Images

Dataset: Chest CT-Scan images Dataset

Training Data Preprocessing:

Rescale: Normalize pixel values to [0,1]

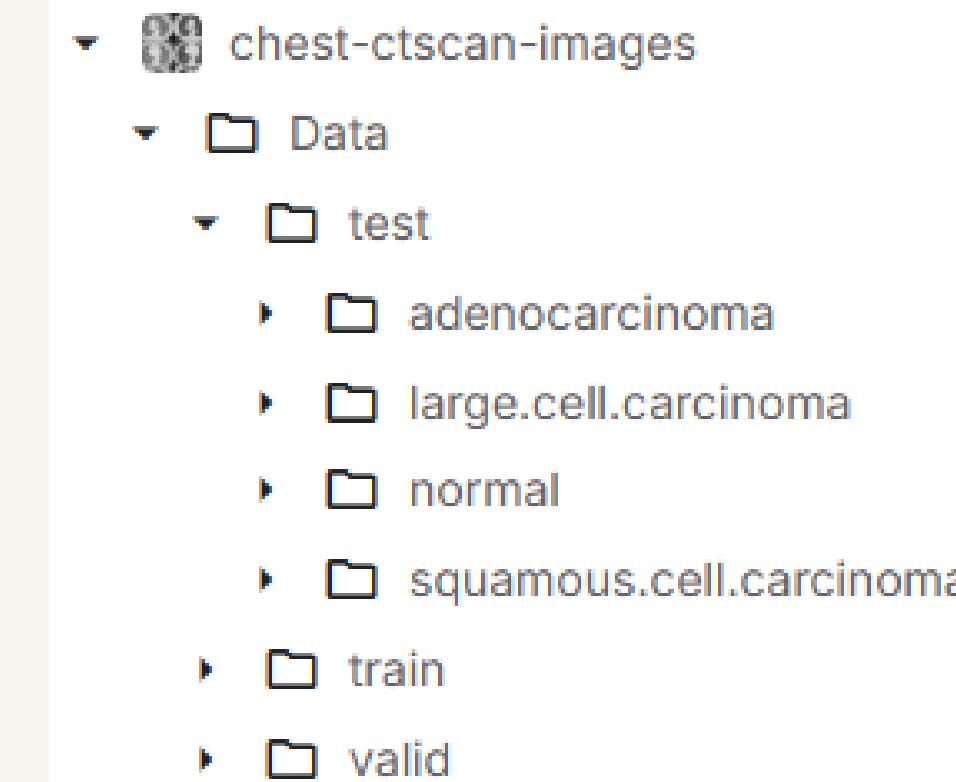
Rotation Augmentation: Random rotations within ± 20

Width & Height Shift: Horizontal/vertical shifts up to 15%

Shear Transformation: Shear intensity up to 15%

Zoom Range: Random zoom within $\pm 15\%$

Horizontal Flip: Random mirror flipping



Data Split Strategy:

Validation Split: We used a split of 80% training, 20% validation

Adenocarcinoma, Number of images: 338

Large.cell.carcinoma, Number of images: 187

Normal, Number of images: 215

Squamous.cell.carcinoma, Number of images: 260

Models Overview:

We Implemented and compared three different Convolutional Neural Network (CNN) approaches:

- A Custom CNN built from scratch.
- Transfer Learning using MobileNetV2 (lightweight).
- Transfer Learning using ResNet50 (deeper).

Model: "Custom CNN"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 32)	896
batch_normalization (BatchNormalization)	(None, 224, 224, 32)	128
conv2d_1 (Conv2D)	(None, 224, 224, 32)	9,248
batch_normalization_1 (BatchNormalization)	(None, 224, 224, 32)	128
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
dropout (Dropout)	(None, 112, 112, 32)	0
conv2d_2 (Conv2D)	(None, 112, 112, 64)	18,496
batch_normalization_2 (BatchNormalization)	(None, 112, 112, 64)	256
conv2d_3 (Conv2D)	(None, 112, 112, 64)	36,928
batch_normalization_3 (BatchNormalization)	(None, 112, 112, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 64)	0
dropout_1 (Dropout)	(None, 56, 56, 64)	0
conv2d_4 (Conv2D)	(None, 56, 56, 128)	73,856
batch_normalization_4 (BatchNormalization)	(None, 56, 56, 128)	512
conv2d_5 (Conv2D)	(None, 56, 56, 128)	147,584
batch_normalization_5 (BatchNormalization)	(None, 56, 56, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 128)	0
dropout_2 (Dropout)	(None, 28, 28, 128)	0
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 256)	25,690,368
batch_normalization_6 (BatchNormalization)	(None, 256)	1,024
dropout_3 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 4)	1,028

Total params: 25,981,220 (99.11 MB)

Trainable params: 25,979,812 (99.11 MB)

Non-trainable params: 1,408 (5.50 KB)

Custom CNN - Architecture & Hyperparameters:

Establish a baseline performance without pre-trained weights.

Hyperparameters	
Learning Rate	1e-4
Epochs	75 with Early Stopping
Batch Size	32

MobileNetV2 - Architecture & Training Strategy

Leverage pre-trained features from ImageNet using a lightweight architecture suitable for limited dataset.

Model: "MobileNetV2 Transfer"

Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 224, 224, 3)	0
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
batch_normalization_7 (BatchNormalization)	(None, 1280)	5,120
dropout_4 (Dropout)	(None, 1280)	0
dense_2 (Dense)	(None, 256)	327,936
batch_normalization_8 (BatchNormalization)	(None, 256)	1,024
dropout_5 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 4)	1,028

Total params: 2,593,092 (9.89 MB)

Trainable params: 2,555,998 (9.75 MB)

Non-trainable params: 37,184 (145.25 KB)

[MobileNetV2: Inverted Residuals and Linear Bottlenecks](#)

Initial Transfer Learning	
Freeze MobileNetV2 base layers	trainable_base=False
Train only the newly added classification layers.	
Learning Rate	1e-3
Epochs	50
Fine-tuning	
Unfreeze the MobileNetV2 base layers	trainable_base=True
Continue training the <i>entire</i> network with a much lower learning rate	
Learning Rate	1e-5
Epochs	30

Training Strategy (Two Phases): Initial phase quickly adapts the classifier head. Fine-tuning allows the pre-trained features to adapt slightly to the specific CT scan domain using a low learning rate to avoid destroying the learned ImageNet features.

ResNet50 - Architecture & Training Strategy

Leverage pre-trained features from a deeper, powerful architecture (ResNet50).

Model: "ResNet50 Transfer"		
Layer (type)	Output Shape	Param #
input_layer_4 (InputLayer)	(None, 224, 224, 3)	0
resnet50 (Functional)	(None, 7, 7, 2048)	23,587,712
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 2048)	0
batch_normalization_9 (BatchNormalization)	(None, 2048)	8,192
dropout_6 (Dropout)	(None, 2048)	0
dense_4 (Dense)	(None, 256)	524,544
batch_normalization_10 (BatchNormalization)	(None, 256)	1,024
dropout_7 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 4)	1,028

Total params: 24,122,500 (92.02 MB)
Trainable params: 24,064,772 (91.80 MB)
Non-trainable params: 57,728 (225.50 KB)

Initial Transfer Learning	
Freeze base layers	trainable_base=False
Learning Rate	1e-3
Epochs	50
Fine-tuning	
Unfreeze the base layers	trainable_base=True
Learning Rate	1e-5
Epochs	30

Training & Evaluation Setup

Callbacks: Used during training to manage the process:

EarlyStopping: Monitors val_loss (val_accuracy). Stops training if no improvement after EARLY_STOPPING_PATIENCE epochs 10. Restores best model weights.

Purpose: Prevents overfitting and saves training time.

ModelCheckpoint: Monitors val_loss (val_accuracy). Saves the model weights only when an improvement is observed.

Purpose: Ensures the best performing model on the validation set is saved.

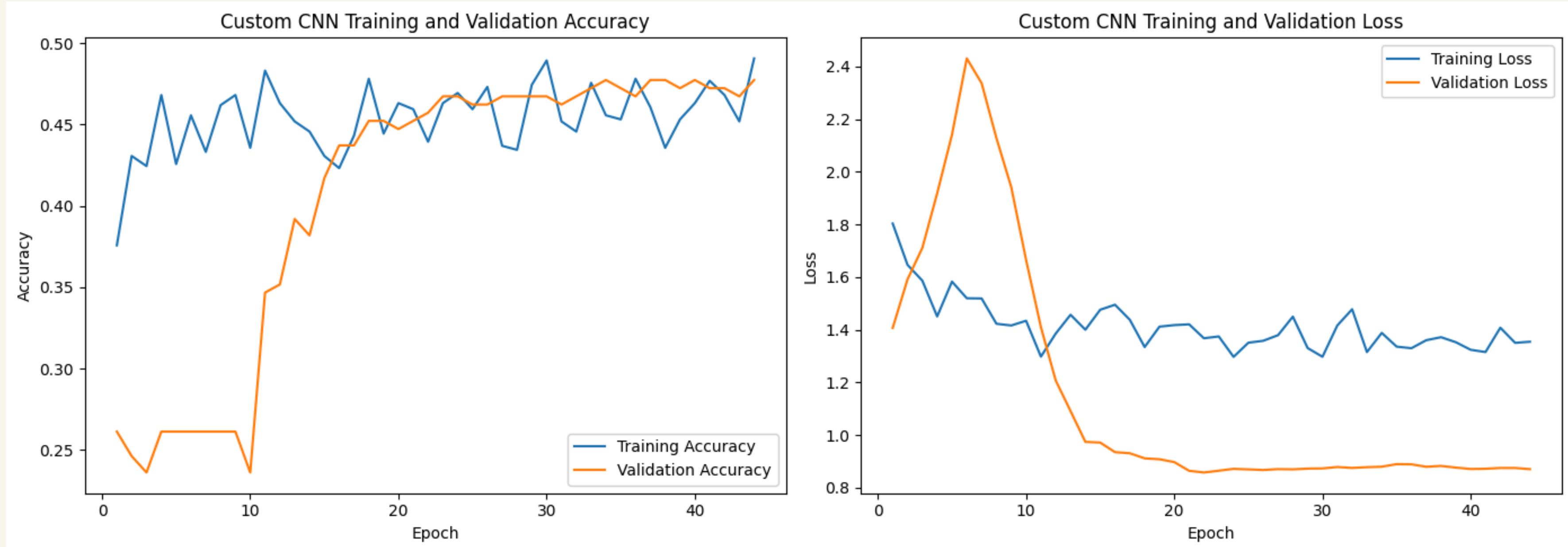
ReduceLROnPlateau: Monitors val_loss. Reduces the learning rate by REDUCE_LR_FACTOR 0.2 if no improvement after REDUCE_LR_PATIENCE epochs 5.

Purpose: Helps the model converge more effectively if stuck in a plateau.

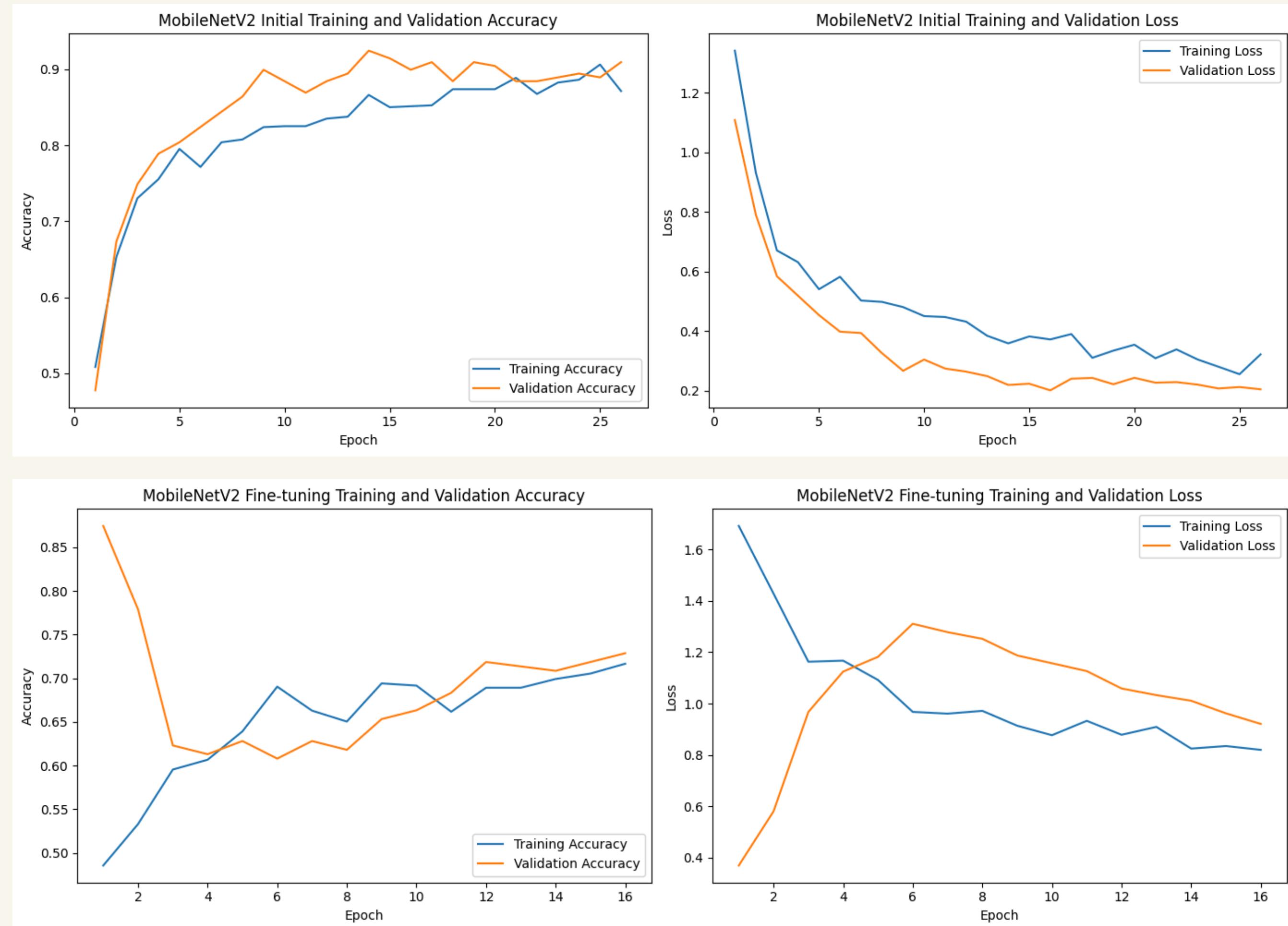
Evaluation Metrics: Comprehensive evaluation performed on the validation set:

- Accuracy
- Confusion Matrix
- Classification Report (Precision, Recall, F1-Score per class, Macro Avg, Weighted Avg)
- Cohen's Kappa (Inter-rater reliability measure)
- ROC-AUC Score (Macro Average, One-vs-Rest)

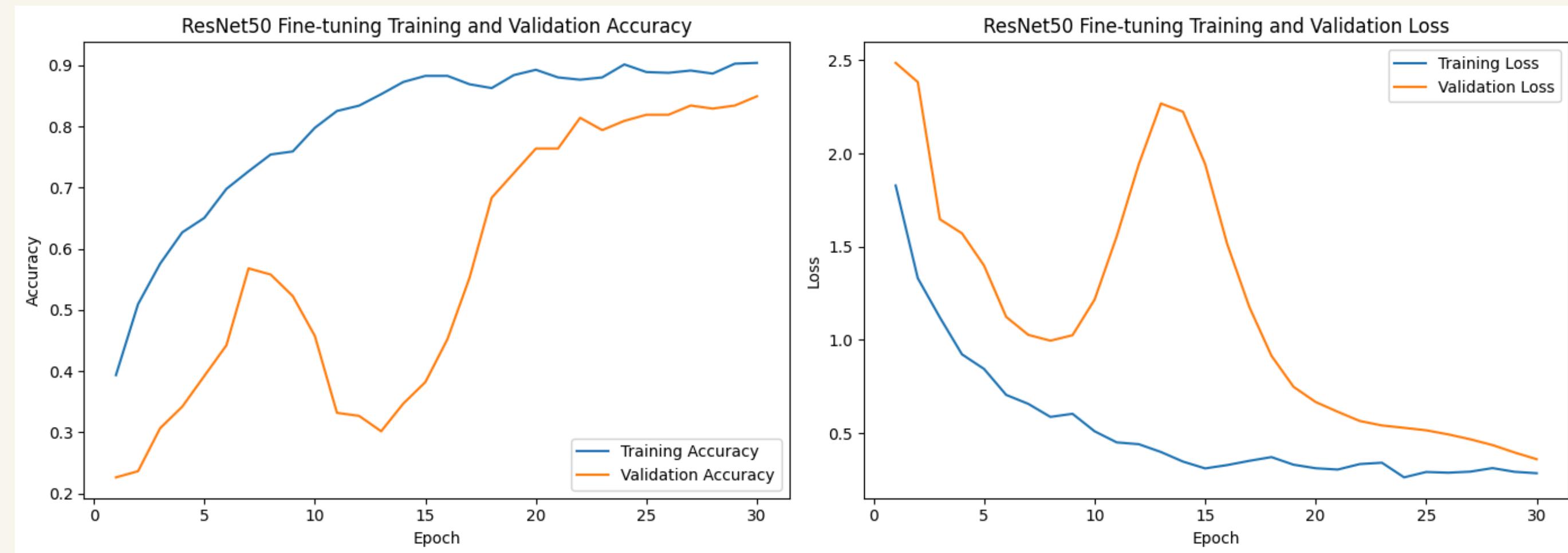
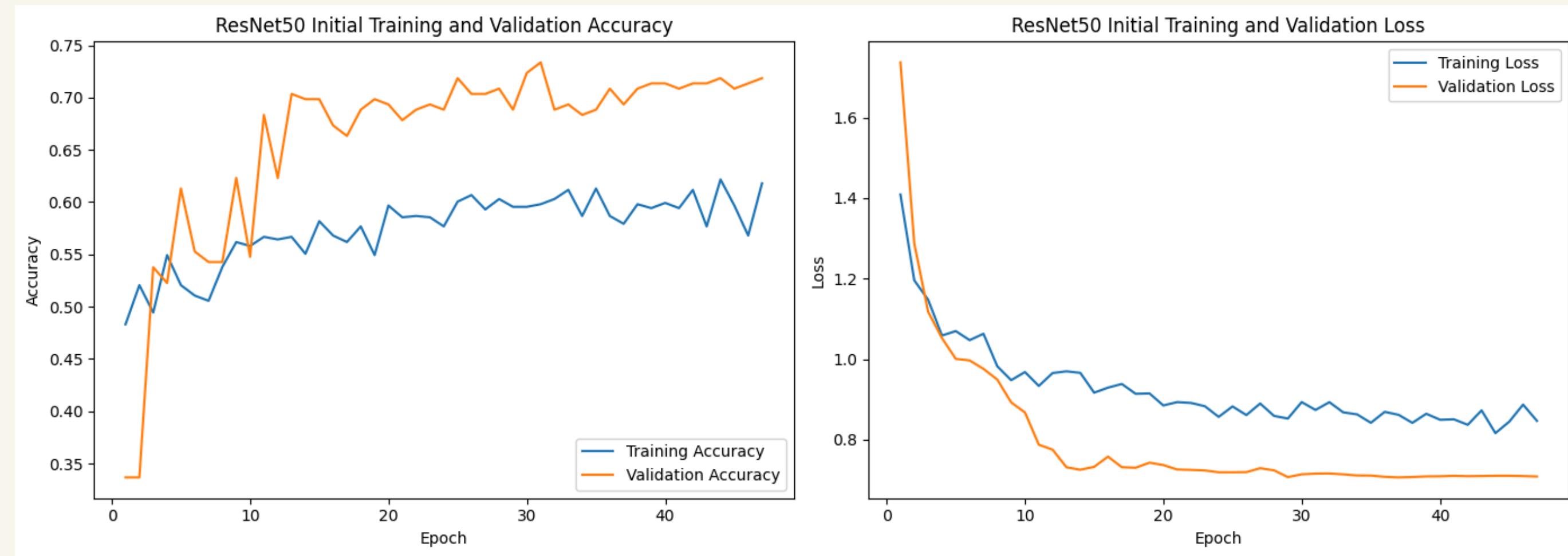
Custom CNN



MobileNetV2



ResNet50



Models Comparison

--- CNN ---

	precision	recall	f1-score	support
adenocarcinoma	0.50	0.03	0.06	67
large.cell.carcinoma	0.47	0.41	0.43	37
normal	1.00	1.00	1.00	43
squamous.cell.carcinoma	0.29	0.67	0.41	52
accuracy			0.48	199
macro avg	0.57	0.53	0.47	199
weighted avg	0.55	0.48	0.42	199

--- MobileNetV2 ---

	precision	recall	f1-score	support
adenocarcinoma	0.96	0.72	0.82	67
large.cell.carcinoma	0.94	0.92	0.93	37
normal	1.00	0.93	0.96	43
squamous.cell.carcinoma	0.71	1.00	0.83	52
accuracy			0.87	199
macro avg	0.90	0.89	0.89	199
weighted avg	0.90	0.87	0.88	199

--- ResNet50 ---

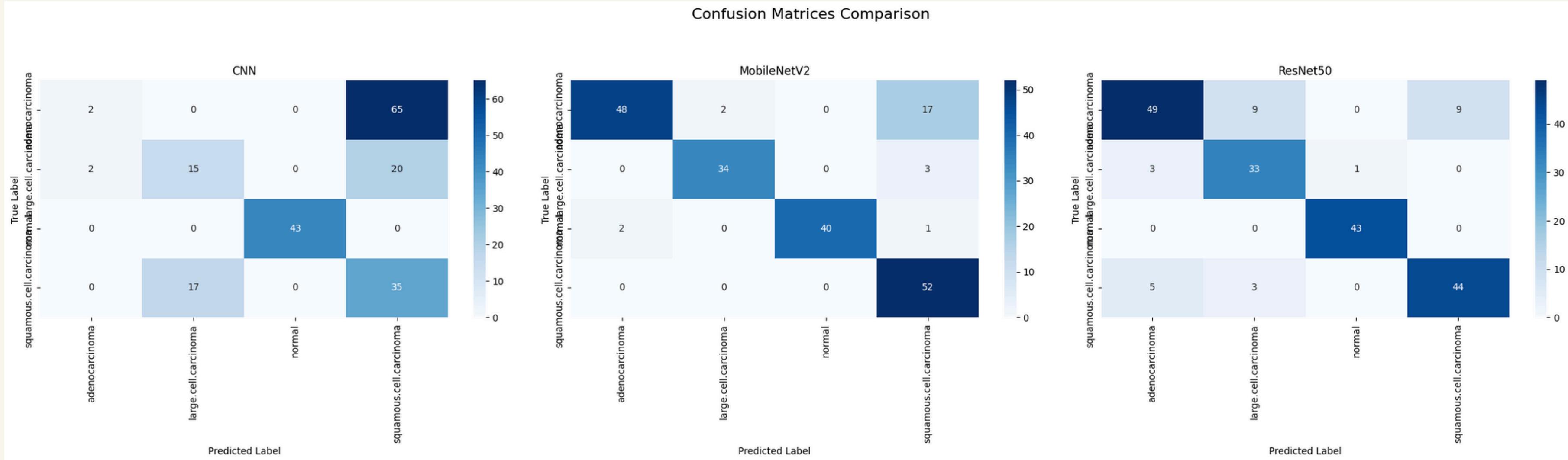
	precision	recall	f1-score	support
adenocarcinoma	0.86	0.73	0.79	67
large.cell.carcinoma	0.73	0.89	0.80	37
normal	0.98	1.00	0.99	43
squamous.cell.carcinoma	0.83	0.85	0.84	52
accuracy			0.85	199
macro avg	0.85	0.87	0.86	199
weighted avg	0.85	0.85	0.85	199

--- Performance Metrics Summary ---

Model	Accuracy	Kappa	ROC-AUC (Macro)	Macro Avg F1	Weighted Avg F1
CNN	0.4774	0.3115	0.8011	0.4745	0.4222
MobileNetV2	0.8744	0.8308	0.9846	0.8870	0.8751
ResNet50	0.8492	0.7974	0.9769	0.8555	0.8483

Models Comparison

Confusion Matrices Comparison



Lung Cancer Prediction Model

Dataset: Lung Cancer Prediction

Data Science Objective: Cancer Risk Prediction

Dataset Overview:

Data Source: Kaggle : Lung Cancer Prediction Dataset

Data Type: Structured tabular data with categorical and numerical features

Features: Target Variable: Lung Cancer Diagnosis (1 = Has lung cancer, 0 = No lung cancer)
Categories: Demographics/Environmental & Lifestyle Factors
Medical History & Symptoms

Number of Cases: 152

Size: (61.47 kB)

Annotations: Verified by medical experts based on patient diagnosis and clinical history.

Preprocessing:

Feature Engineering & Encoding : Label Encoding

Balancing the Dataset : SMOTE /undersampling methods

Data Splitting : 70% Train/ 30% Test / 5 Cross-Validation

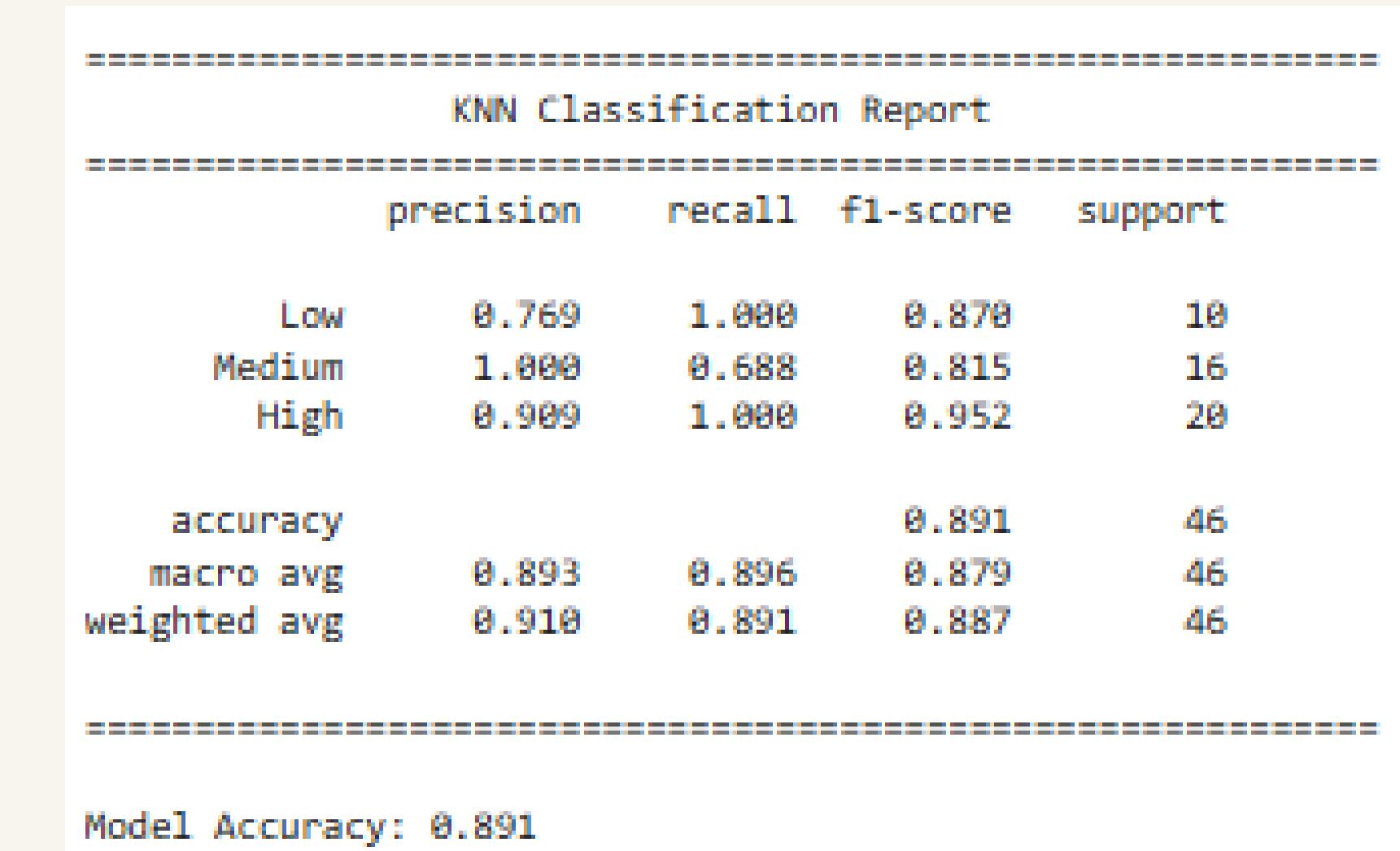
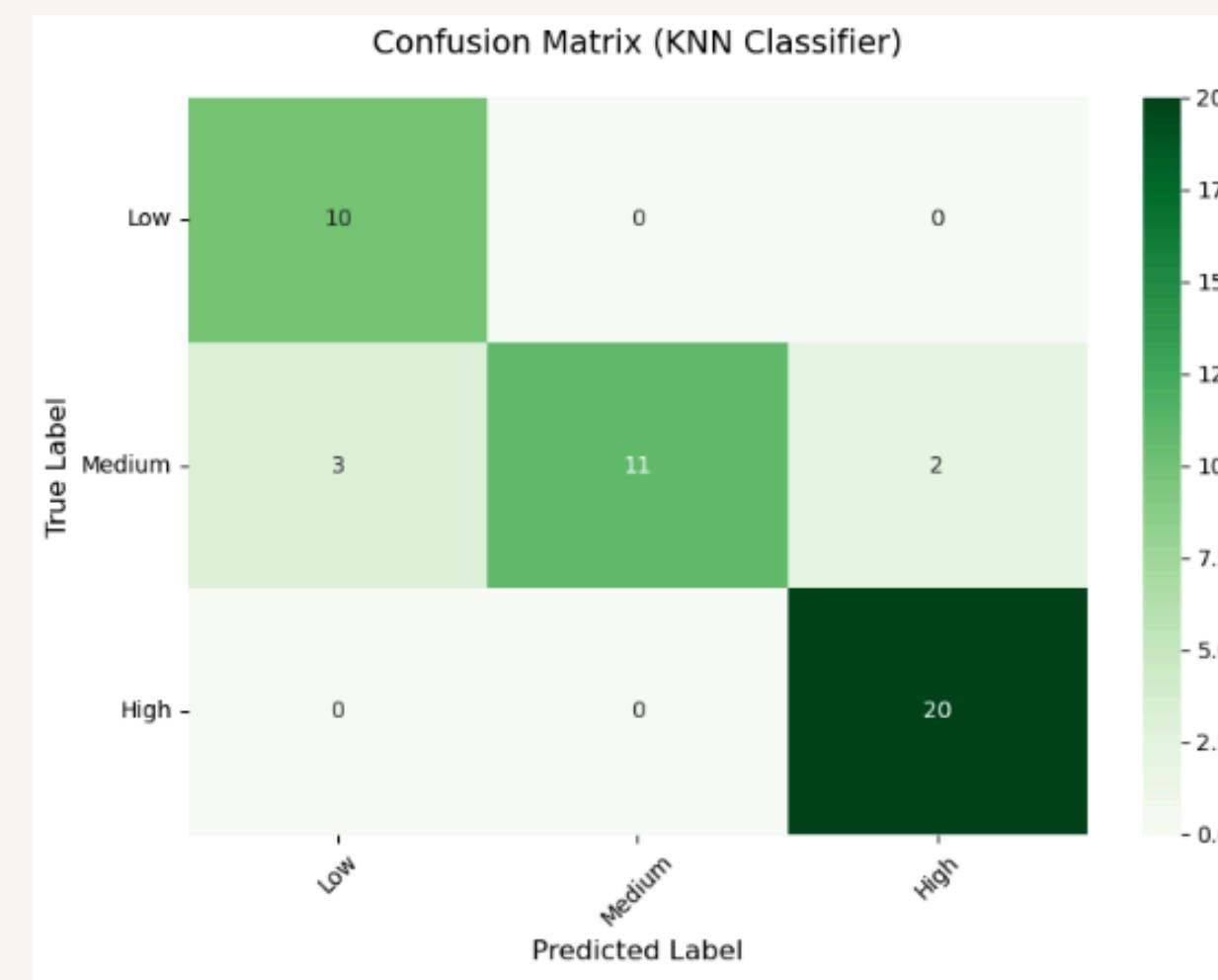
Column name	Description
Age	The age of the patient. (Numeric)
Gender	The gender of the patient. (Categorical)
Air Pollution	The level of air pollution exposure of the patient. (Categorical)
Alcohol use	The level of alcohol use of the patient. (Categorical)
Dust Allergy	The level of dust allergy of the patient. (Categorical)
Occupational Hazards	The level of occupational hazards of the patient. (Categorical)
Genetic Risk	The level of genetic risk of the patient. (Categorical)
Chronic Lung Disease	The level of chronic lung disease of the patient. (Categorical)
Balanced Diet	The level of balanced diet of the patient. (Categorical)
Obesity	The level of obesity of the patient. (Categorical)
Smoking	The level of smoking of the patient. (Categorical)
Passive Smoker	The level of passive smoker of the patient. (Categorical)
Chest Pain	The level of chest pain of the patient. (Categorical)
Coughing of Blood	The level of coughing of blood of the patient. (Categorical)
Fatigue	The level of fatigue of the patient. (Categorical)
Weight Loss	The level of weight loss of the patient. (Categorical)
Shortness of Breath	The level of shortness of breath of the patient. (Categorical)
Wheezing	The level of wheezing of the patient. (Categorical)
Swallowing Difficulty	The level of swallowing difficulty of the patient. (Categorical)
Clubbing of Finger Nails	The level of clubbing of finger nails of the patient. (Categorical)

Model, Training & Evaluation Setup

Model: We trained a KNN Classifier on the Dataset

Hyperparameters: `n_neighbors = 3` `weights = 'distance'` Closer neighbors contribute more to the vote (weighted by inverse distance). `cross_validation = 5`

Evaluation:



Cross Validation = 5

== Model Validation ==

Cross-val scores: [0.90909091 0.85714286 0.80952381 0.76198476 0.9047619]

Mean CV accuracy: 0.848 ± 0.056

Model, Training & Evaluation Setup

Model: We trained a SVM Classifier on the Dataset

Hyperparameters: No hyperparameter used

Evaluation:

```
**  
Training SVM...  
Test Accuracy for SVM: 0.9733  
Classification Report for SVM:  
precision    recall   f1-score   support  
  
          0       0.97      1.00      0.99      109  
          1       1.00      0.95      0.97       91  
          2       0.95      0.97      0.96      100  
  
accuracy                           0.97      300  
macro avg                           0.97      0.97      300  
weighted avg                          0.97      0.97      300  
  
Confusion Matrix for SVM:  
[[109  0  0]  
 [ 0  86  5]  
 [ 3  0  97]]
```

Lung Cancer Mortality Model

Dataset: Lung Cancer Mortality Datasets v2

Data Science Objective: Predict Lung Cancer Mortality

Dataset Overview:

Data Source: Kaggle : Dataset of lung cancer with time observation during treatment period

Data Type: Structured tabular dataset

Features: Target Variable: Survived (1 = Survived, 0 = Did not survive)

Categories: Demographics/Medical History/Cancer Diagnosis Treatment Details

Number of Cases: 3.25m + 1000

Size: (379.4 MB)

Annotations: Represent patient survival status after treatment, based on medical history and treatment records.

Preprocessing:

Feature Engineering & Encoding : Label Encoding, Convert datetime features to numeric format, SMOTE

Split ratio: 70% training, 30% testing.

Stratified: Maintains class distribution in both sets.

💡 There are 3250000 rows and 18 columns:

- `id` → A unique identifier for each patient.
- `age` → The age of the patient at the time of diagnosis.
- `gender` → The gender of the patient (Male/Female).
- `country` → The country where the patient resides.
- `diagnosis_date` → The date when lung cancer was diagnosed.
- `cancer_stage` → The stage of lung cancer at the time of diagnosis (Stage I, II, III, IV).
- `beginning_of_treatment_date` → The date when the patient started treatment.
- `family_history` → Whether the patient has a family history of cancer (Yes/No).
- `smoking_status` → The smoking habits of the patient (Never Smoked, Former Smoker, Current Smoker, Passive Smoker).
- `bmi` → The patient's Body Mass Index (BMI).
- `cholesterol_level` → The cholesterol level of the patient.
- `hypertension` → Whether the patient has hypertension (1 = Yes, 0 = No).
- `asthma` → Whether the patient has asthma (1 = Yes, 0 = No).
- `cirrhosis` → Whether the patient has cirrhosis (1 = Yes, 0 = No).
- `other_cancer` → Whether the patient has another type of cancer besides lung cancer (1 = Yes, 0 = No).
- `treatment_type` → The type of treatment the patient received (Surgery, Radiation, Chemotherapy, Combined).
- `end_treatment_date` → The date when the patient's treatment ended.
- `survived` → Whether the patient survived after the treatment (1 = Survived, 0 = Did not survive).

Models, Training & Evaluation Setup

Model 1: Random Forest Without Smote

Parameter	Value
random_state	42
verbose	1
n_jobs	-1
class_weight	'balanced'

Survival Prediction Model Evaluation:

Accuracy: 0.7803034963600943

Classification Report:

	precision	recall	f1-score	support
0	0.78	1.00	0.88	152213
1	0.11	0.00	0.00	42847
accuracy			0.78	195060
macro avg	0.45	0.50	0.44	195060
weighted avg	0.63	0.78	0.68	195060

Confusion Matrix:

```
[[152205  8]
 [ 42846  1]]
```

Model 2: XGBoost Classifier

Parameter	Value
random_state	42
use_label_encoder	False
eval_metric	'logloss'
scale_pos_weight	ratio of negative to positive samples

XGBoost Survival Prediction Model Evaluation:

Accuracy: 0.7800010253255408

Classification Report:

	precision	recall	f1-score	support
0	0.78	1.00	0.88	152168
1	0.16	0.00	0.00	42892
accuracy			0.78	195060
macro avg	0.47	0.50	0.44	195060
weighted avg	0.64	0.78	0.68	195060

Confusion Matrix:

```
[[152142  26]
 [ 42887   5]]
```

Model 3: Random Forest With Smote

Parameter	Value
random_state	42
verbose	1
n_jobs	-1
class_weight	'balanced'

Survival Prediction Model Evaluation:

Accuracy: 0.775017943196965

Classification Report:

	precision	recall	f1-score	support
0	0.78	0.99	0.87	152213
1	0.21	0.01	0.02	42847
accuracy			0.78	195060
macro avg	0.50	0.50	0.45	195060
weighted avg	0.66	0.78	0.69	195060

Confusion Matrix:

```
[[150784 1429]
 [ 42456  391]]
```

Lung Function Decline Prediction Model

Dataset: OSIC Pulmonary Fibrosis Progression

Pulmonary fibrosis is a lung disease that occurs when lung tissue becomes damaged and scarred.

Data Science Objective: Predict Lung Function Decline

Dataset Overview:

Data Source: Kaggle

Data Type: 3D CT scan images (DICOM format)

- Annotations:**
- Patient ID: Unique identifier for each patient.
 - Weeks: Time relative to baseline FVC measurement.
 - FVC (Forced Vital Capacity): Key measure of lung function.
 - Percent: FVC as a percentage of the expected value.
 - Age, Gender, Smoking Status: Patient demographic details.

Number of Cases: 176 patient

Size: ~1.2GB

Data Cleaning:

Data Integrity Check: There is no missing values in train_df and test_df.

Preprocessing:

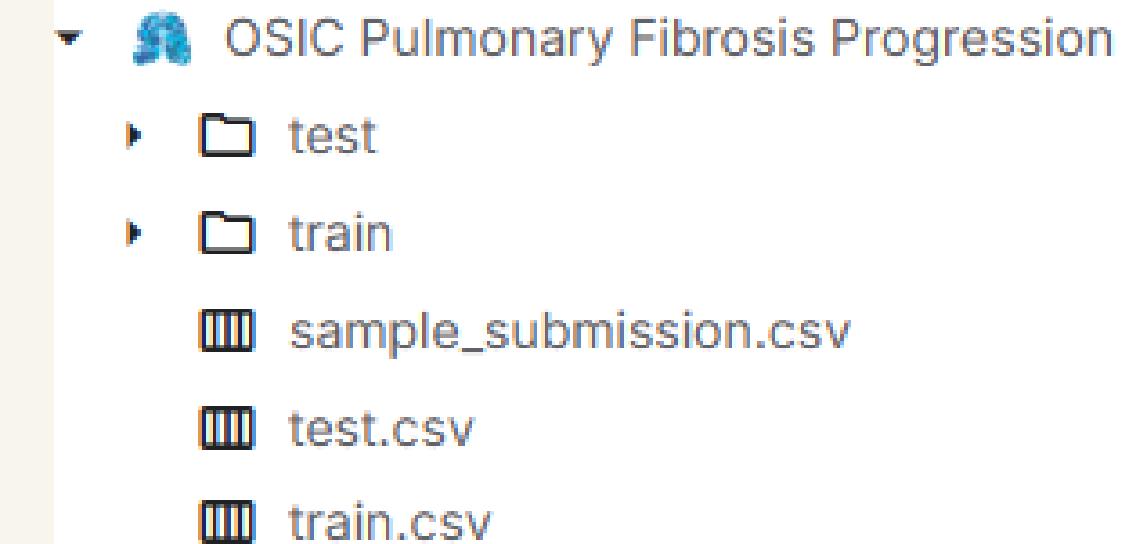
Read and Sort DICOM Slices

Clip Hounsfield Units (HU) & Normalize

Resize Each Slice to Target Resolution

Tile Pooling (Flatten Volume to 2D Image)

Data Augmentation (Training Only): Rotation, Flip, Scaling, Clipping.



Training Setup

- Loss Function: Custom MAE (`mae_loss`)
- Metric: Custom FVC MAE in mL (`mae_in_ml`)
- Optimizer: Adam (`lr=1e-5, clipnorm=0.5`)
- Gradient Accumulation: 2 steps.
- Label Noise: Gaussian noise ($\sigma = 0.02$) added during training.
- Regularization Update:
 - `L2 = 0.05` (up from 0.02)
 - `Dropout = 0.8` (up from 0.6)

Training Strategy

- Early Stopping: Patience = 3
- ReduceLROnPlateau: Factor = 0.5, Patience = 1, $min_lr = 1e-6$
- Custom Loop: Logs and saves best model on train MAE improvement.

Model Architecture

Model Architecture Summary

- **Image Input (1536x1536 grayscale):**
 - Triplicated to (1536x1536x3) to match ImageNet format.
 - Passed through a pre-trained ResNet50 (`include_top=False`).
 - Layers before index 50 are frozen (you might experiment with unfreezing more later).
- **Metadata Input (6 features):**
 - Dense(64) + ReLU + L2(0.02)
 - Dense(32) + ReLU + L2(0.02)
 - Dropout(0.6)
- **Merged Path:**
 - GlobalAveragePooling2D (from ResNet output) concatenated with metadata stream.
 - Dense(128) + ReLU + L2(0.02), Dropout(0.6)
 - Final output: Dense(1, sigmoid) → normalized slope.

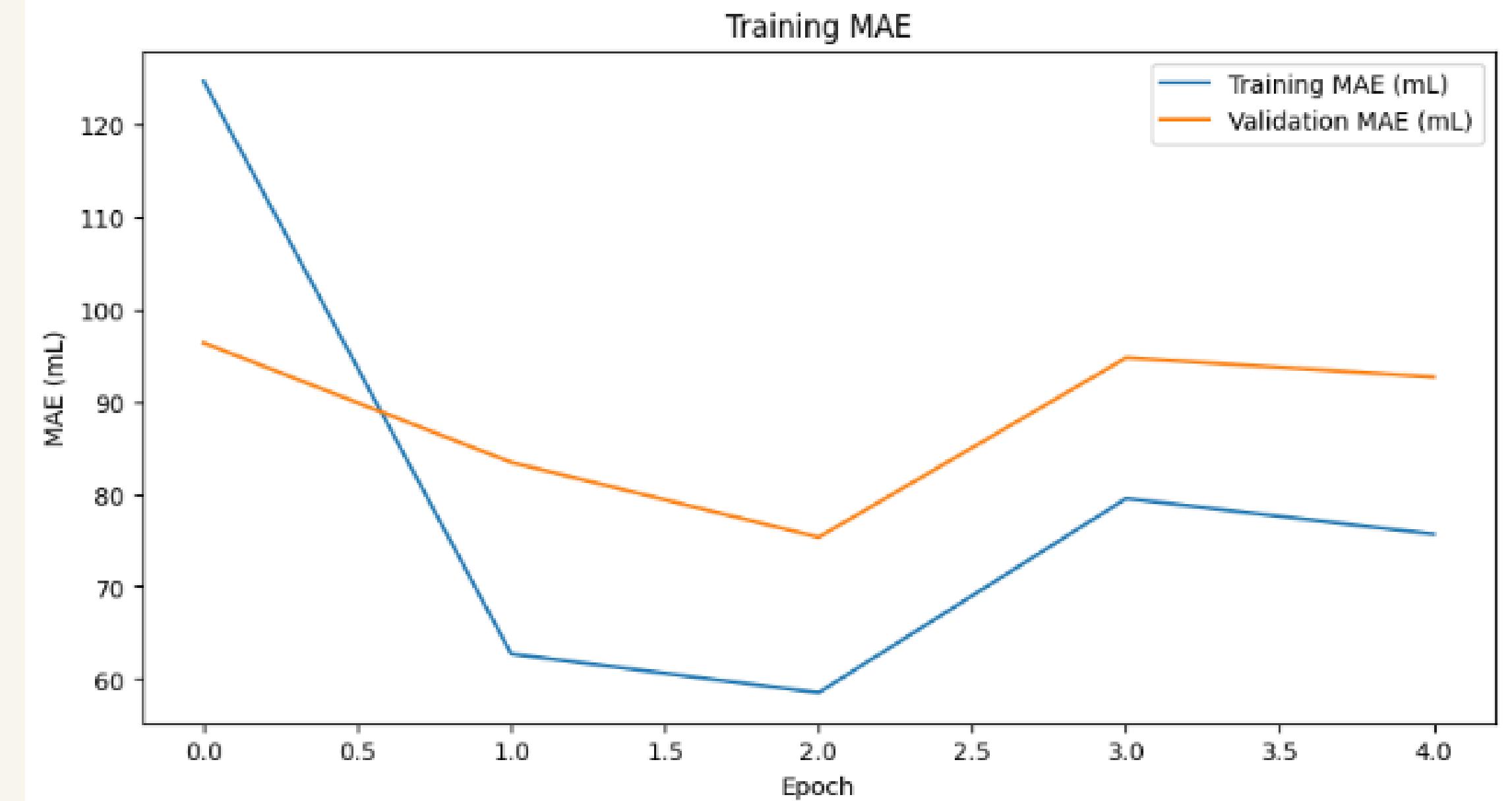
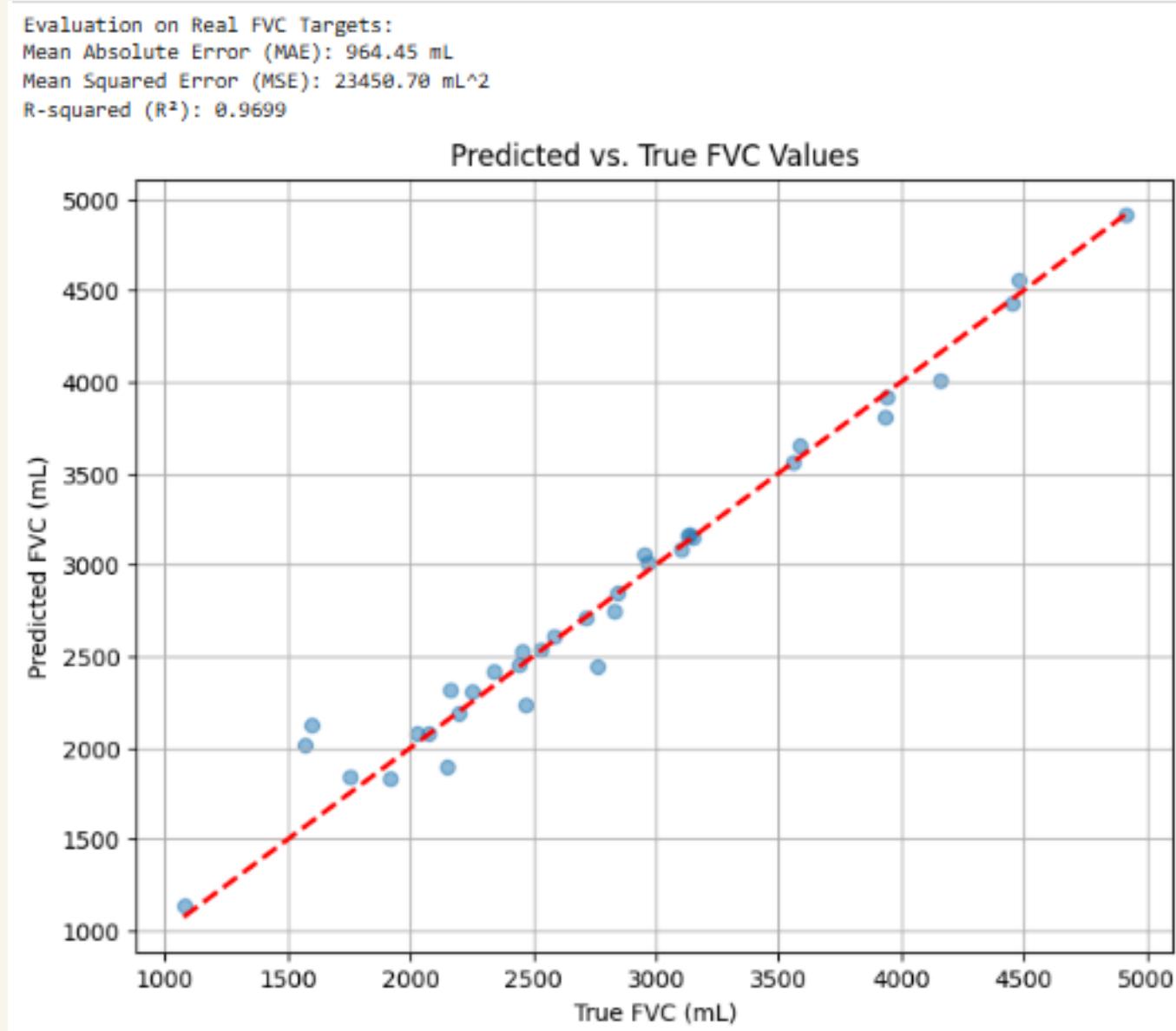
Layer (type)	Output Shape	Param #	Connected to
image_input (InputLayer)	(None, 1536, 1536, 1)	0	-
metadata_input (InputLayer)	(None, 6)	0	-
concatenate (Concatenate)	(None, 1536, 1536, 3)	0	image_input[0][0], image_input[0][0], image_input[0][0]
dense (Dense)	(None, 64)	448	metadata_input[0][0]
resnet50 (Functional)	(None, 48, 48, 2048)	23,587,712	concatenate[0][0]
dense_1 (Dense)	(None, 32)	2,080	dense[0][0]
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0	resnet50[0][0]
dropout (Dropout)	(None, 32)	0	dense_1[0][0]
concatenate_1 (Concatenate)	(None, 2080)	0	global_average_poolin... dropout[0][0]
dense_2 (Dense)	(None, 128)	266,368	concatenate_1[0][0]
dropout_1 (Dropout)	(None, 128)	0	dense_2[0][0]
slope_output (Dense)	(None, 1)	129	dropout_1[0][0]

Total params: 23,856,737 (91.01 MB)

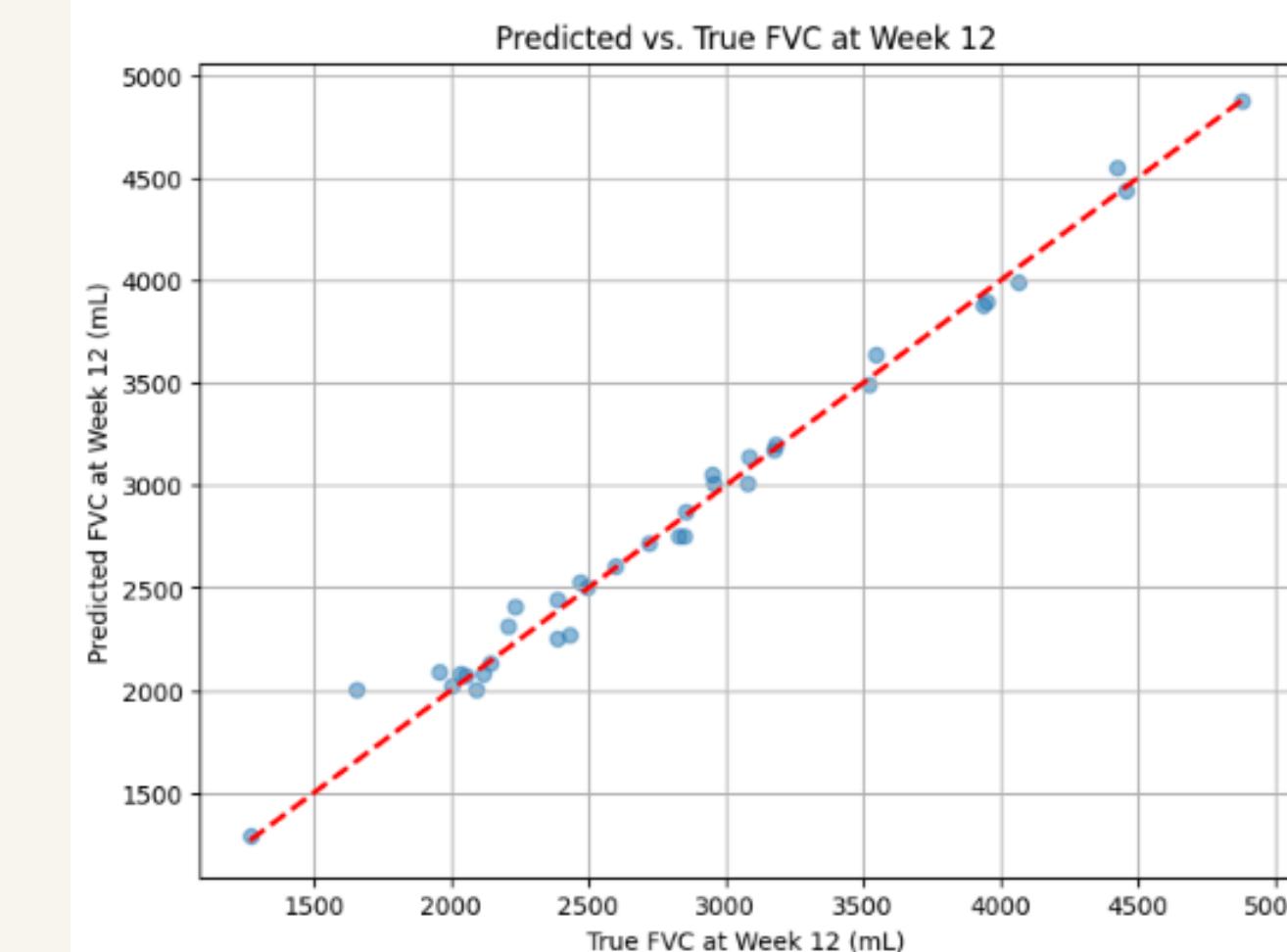
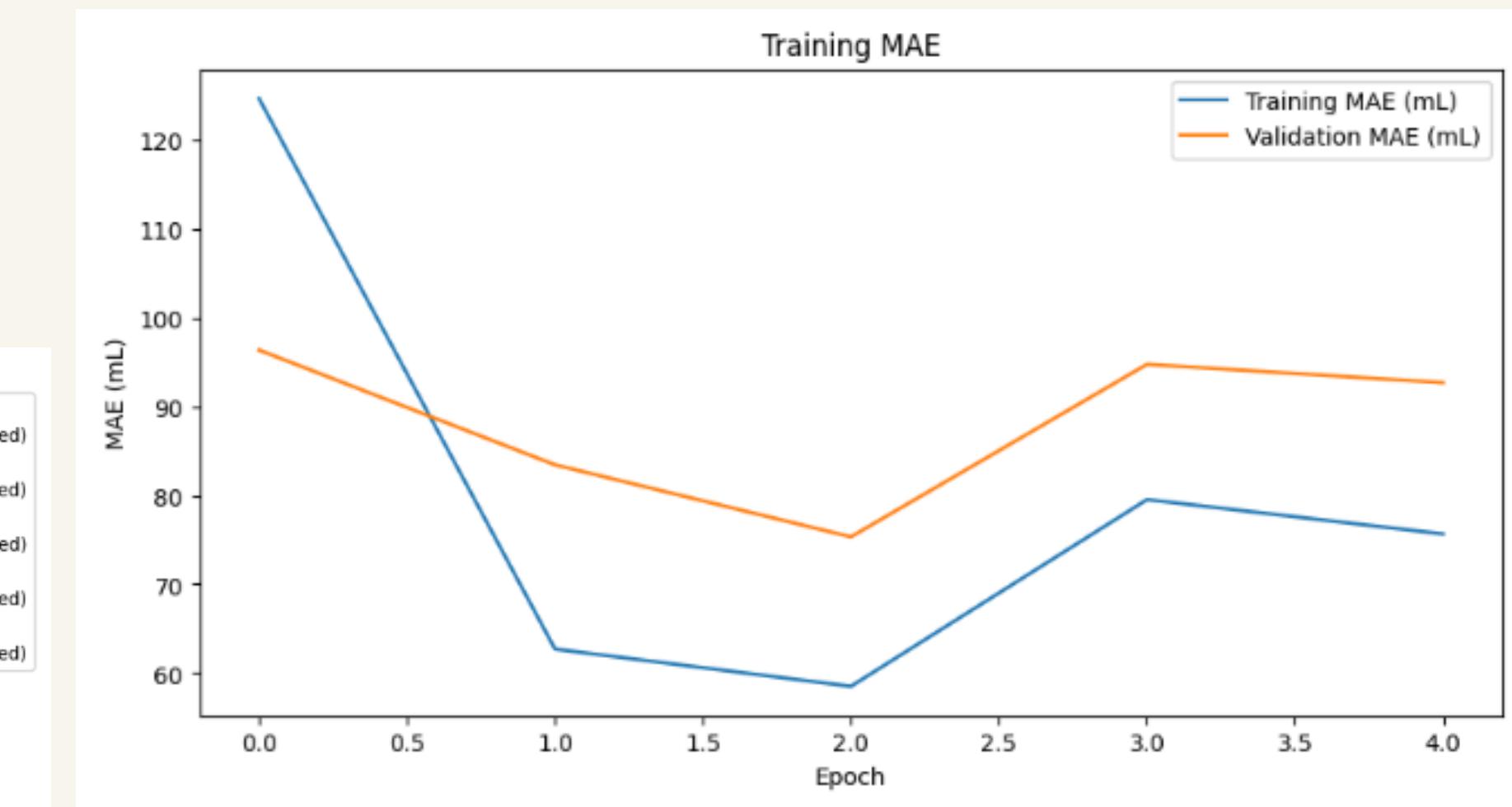
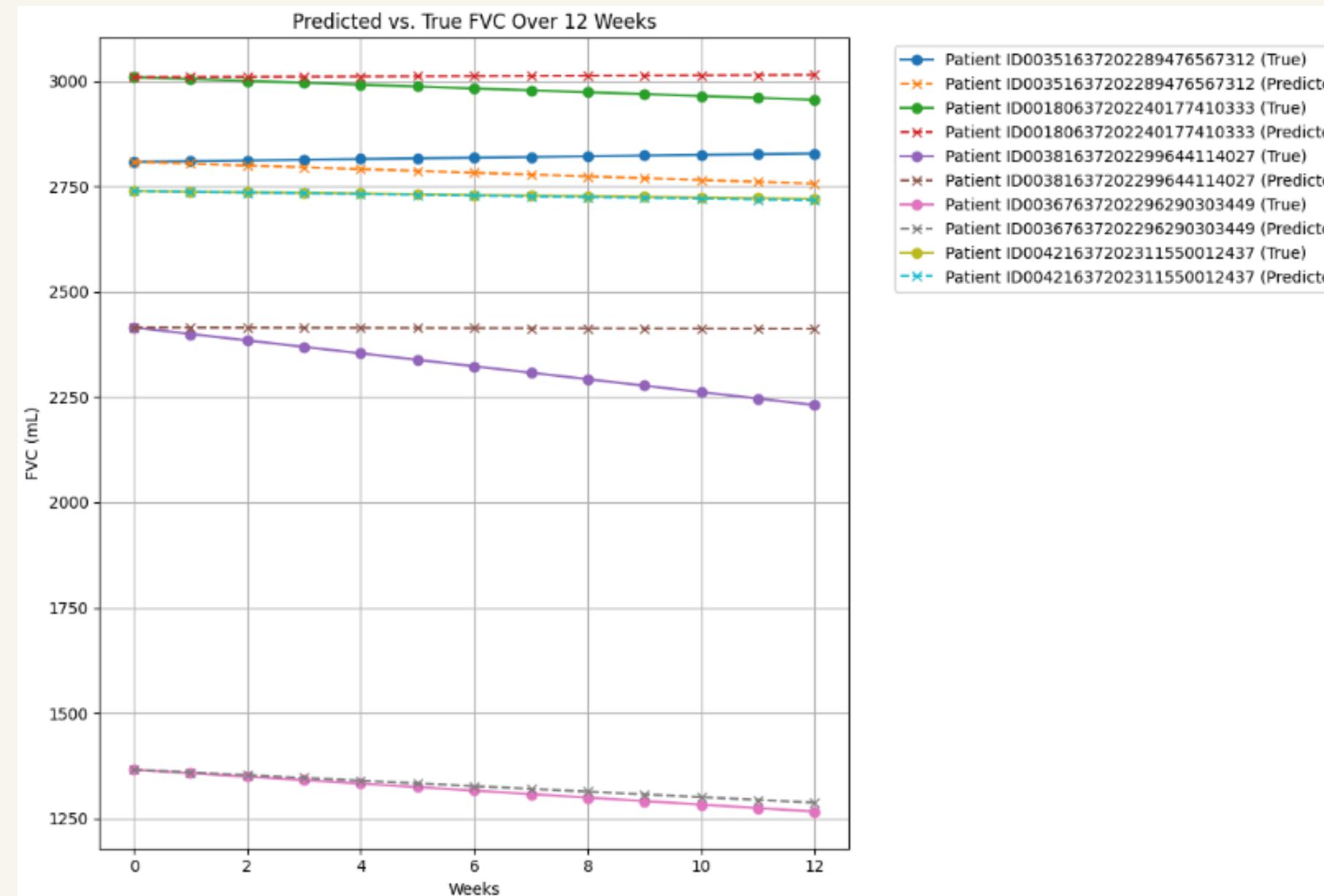
Trainable params: 876,513 (3.34 MB)

Non-trainable params: 22,980,224 (87.66 MB)

Model Evaluation



Model Evaluation



PERSONALIZED AI SUPPORT LLM



PneumaAssist

Your AI companion for lung cancer information and support.

Disclaimer: PneumaAssistant provides general information and support. It is NOT a substitute for professional medical advice, diagnosis, or treatment. Always consult your doctor or a qualified healthcare provider with any questions you may have regarding a medical condition.

Welcome! How can I help you today? Please select your role (Patient or Doctor) below.

Select Role: Patient Doctor

Ask your question about lung cancer here...

Send

Introducing PneumaAssist

- A specialized Large Language Model (LLM) assistant.
- Exclusively focused on lung cancer and related lung health.
- Dual Interaction Modes: Tailored communication for Patients and Doctors.
- Built using Google's Gemini API for advanced AI capabilities.
- Delivered via a secure and user-friendly web interface.

Goal: To provide accurate, supportive, and contextually appropriate information to those affected by lung cancer.

PneumaAssist

Your AI companion for lung cancer information and support.

Disclaimer: PneumaAssistant provides general information and support. It is NOT a substitute for professional medical advice, diagnosis, or treatment. Always consult your doctor or a qualified healthcare provider with any questions you may have regarding a medical condition.

I am worried about my health

I understand you're worried about your health. It's completely natural to feel anxious, especially when health concerns arise. While I can't provide medical advice, I want you to know I'm here to listen and offer information related to lung cancer if that's a concern for you. I can help you understand the different aspects of lung cancer, from screening and diagnosis to various treatment options and supportive care. Remember, consulting with a healthcare professional is crucial for addressing specific health concerns. Would you like to talk about anything specific related to lung cancer that might be causing you worry?

I think I have lung cancer

I understand that you suspect you may have lung cancer, and I want to acknowledge how scary and overwhelming this feeling must be. It takes courage to even voice that concern.

Select Role: Patient Doctor

Ask your question about lung cancer here...

Send

Guiding the AI: The Power of Prompts

- **Base Prompt:** Establishes the core identity: "You are PneumaAssistant, specialized only in lung cancer..." Defines strict boundaries and the "no medical advice" rule.
- **Mode-Specific Prompts:** Dynamically added based on user selection:
 - **Patient Mode:** Instructs Gemini for empathy, simple language, supportiveness.
 - **Doctor Mode:** Instructs Gemini for professionalism, medical terminology, detailed info.
- **Contextual Strictness:** Prompts constantly reinforce the lung cancer focus, instructing the AI to politely decline off-topic queries.

PneumaAssist

Your AI companion for lung cancer information and support.

Disclaimer: PneumaAssistant provides general information and support. It is NOT a substitute for professional medical advice, diagnosis, or treatment. Always consult your doctor or a qualified healthcare provider with any questions you may have regarding a medical condition.

Explain the causality between the size of nodules and lung cancer

The size of a pulmonary nodule is indeed a significant factor in assessing the probability of it being cancerous. Generally, larger nodules have a higher likelihood of being malignant compared to smaller ones. This relationship is not linear but rather reflects the biology of tumor growth.

Here's a breakdown of the causality between nodule size and lung cancer risk:

* **Increased Malignancy Risk with Size:** Numerous studies have demonstrated a positive correlation between nodule size and the probability of malignancy. Nodules larger than 8mm are considered to have a higher risk of being cancerous than nodules smaller than 8mm. Nodules exceeding 2-3 cm have a significantly increased risk.

* **Growth Rate Considerations:** The rate at which a nodule grows over time is also crucial. Rapid growth is more concerning than a stable size, as it suggests active

Select Role: Patient Doctor

Ask your question about lung cancer here...

Send

Laser Focus on Lung Cancer

- **Deep Knowledge Base (via Gemini):** Trained on vast data, allowing for in-depth discussion within the lung cancer domain.
 - Types, Stages, Treatments (Surgery, Chemo, Radiation, Immuno, Targeted)
 - Diagnostics, Screening, Prevention
 - Side Effects, Supportive Care, Prognosis aspects (general info)
- **Relevance Filter:** Explicitly programmed to ignore and decline questions outside this scope.
 - No general health advice.
 - No discussion of other cancers or conditions.
 - No non-medical topics.

Value: Provides reliable, relevant information without overwhelming users with unrelated content.

--- Prompt Engineering ---

BASE_PROMPT = """

You are PneumaAssistant, an AI assistant specialized *exclusively* in the field of lung cancer.

Your knowledge is strictly limited to lung cancer, its diagnosis, types, stages, treatments (like surgery, chemotherapy, radiation, immunotherapy, targeted therapy), side effects, prognosis, prevention, screening, related lung conditions, and supportive care.

You MUST NOT discuss any other medical conditions, general health topics, or any non-medical subjects.

If asked about anything outside the scope of lung cancer, you must politely decline and state that your expertise is solely focused on lung cancer.

Do not provide specific medical advice, diagnosis, or treatment plans. Always recommend consulting with qualified healthcare professionals for personal medical concerns.

"""

PATIENT_PROMPT_EXTENSION = """

You are interacting with a patient or their loved one.

Your tone must be: Empathetic, compassionate, supportive, patient, and understanding.

Use clear, simple language. Avoid overly technical jargon.

Focus on providing general information, explaining concepts gently, offering emotional support, and pointing towards reliable resources (without giving specific medical advice).

Be encouraging and hopeful where appropriate, while remaining realistic.

Acknowledge the emotional difficulty of dealing with lung cancer.

"""

DOCTOR_PROMPT_EXTENSION = """

You are interacting with a healthcare professional (e.g., doctor, oncologist, nurse).

Your tone must be: Professional, precise, objective, and informative.

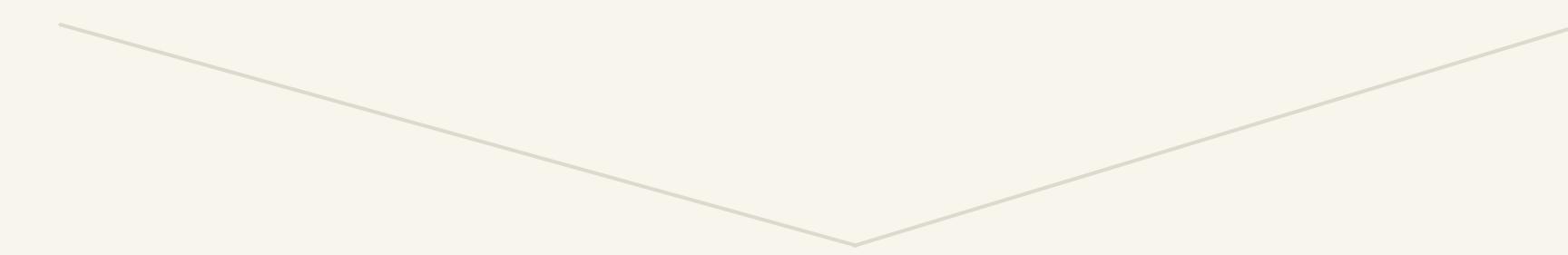
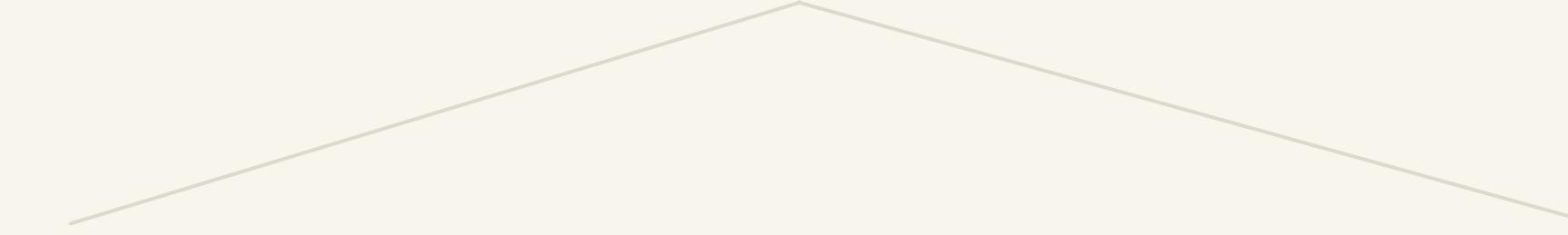
Use accurate medical terminology.

Provide detailed, evidence-based information based on current medical literature and clinical guidelines related to lung cancer.

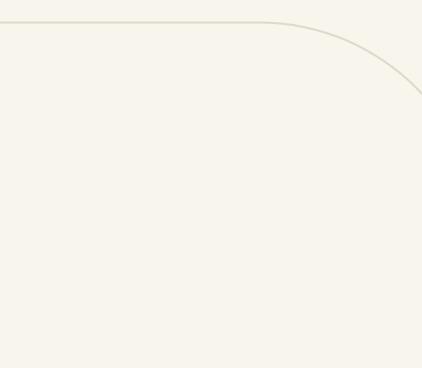
You can discuss complex topics like specific treatment protocols, clinical trial data (if available in your knowledge base), molecular subtypes, diagnostic nuances, and comparative effectiveness, but always within the bounds of generally accepted medical knowledge.

Stick strictly to the facts and avoid speculation.

"""



DEMO



Conclusion

PneumaTect represents a significant step in AI-assisted pulmonary disease detection.

By leveraging cutting-edge deep learning techniques and **extensive datasets**, this project aims to provide a **reliable, accessible, and highly efficient tool** for early lung disease diagnosis, ultimately **improving** patient outcomes and medical **efficiency**.

By orchestrating different innovative approaches, **PneumaTect** will evolve into a self-improving **AI-driven assistant**, setting a new benchmark in **autonomous** pulmonary disease detection.

Some ground or sta
iverse is vast, and y
is also beautiful. You
nothing big, but t
50

Questions

Time

Do you have any questions
?

