Practice 3: Multi-Task Deep Learning for COVID-19 Detection: Joint Lung and Infection Segmentation with Classification in Chest X-Rays

Hoang Khanh Dong - 22BA13072
January 2026

***Abstract*—Accurate detection of COVID-19 from chest X-ray images is critical for rapid diagnosis and treatment planning. Manual analysis by radiologists is time-consuming and subject to inter-observer variability. This study presents a multi-task deep learning approach for automated COVID-19 detection, combining lung segmentation, infection region segmentation, and disease classification using the COVID-QU-Ex dataset.**

I. Introduction

This study presents a multi-task deep learning approach for COVID-19 detection from chest X-ray images. I propose a hybrid architecture combining a Swin Transformer encoder with dual U-Net decoders for simultaneous lung segmentation, infection region segmentation, and disease classification.

II. Dataset

The dataset used in this study is the COVID-QU-Ex dataset, obtained from Kaggle[1]. This dataset contains chest X-ray images with corresponding lung segmentation masks, infection segmentation masks, and classification labels for three classes: Normal, Non-COVID, and COVID-19.

The dataset is divided into two main subsets for our two-phase training approach:

A. Lung Segmentation Data

This subset contains 33,920 chest X-ray images with lung segmentation masks. All images are of uniform size (256×256 pixels). The distribution across splits and classes is shown in Table I.

TABLE I
Lung Segmentation Data Distribution

Split	Normal	Non-COVID	COVID-19	Total
Train	6,849	7,208	7,658	21,715
Validation	1,712	1,802	1,903	5,417
Test	2,140	2,253	2,395	6,788
Total	10,701	11,263	11,956	**33,920**

B. Infection Segmentation Data

This subset contains 5,826 chest X-ray images with both lung and infection segmentation masks. Importantly, only COVID-19 cases have infection masks; Normal and Non-COVID cases have empty infection masks. The distribution is shown in Table II.

TABLE II
Infection Segmentation Data Distribution

Split	Normal	Non-COVID	COVID-19	Total
Train	932	932	1,864	3,728
Validation	233	233	466	932
Test	291	292	583	1,166
Total	1,456	1,457	2,913	**5,826**

Table III summarizes the key differences between the two subsets.

TABLE III
Dataset Comparison

Feature	Lung Segmentation	Infection Segmentation
Total Images	33,920	5,826
Image Size	256×256	256×256
Lung Masks	Yes	Yes
Infection Masks	No	Yes (COVID-19 only)

III. Methodology

A. Hybrid Model Architecture

My hybrid model architecture combines a Tiny Swin Transformer encoder (27.52M parameters) with dual U-Net decoders (14.77M parameters) for simultaneous lung segmentation with lung head (10.44K parameters), infection region segmentation with infection head (10.44K parameters), and disease classification with classification head (2.18K parameters).

In summary, the total number of parameters in the model is 44.09M. A visualization of the model architecture is shown in Figure 1.

[1] https://www.kaggle.com/datasets/anasmohammedtahir/covidqu

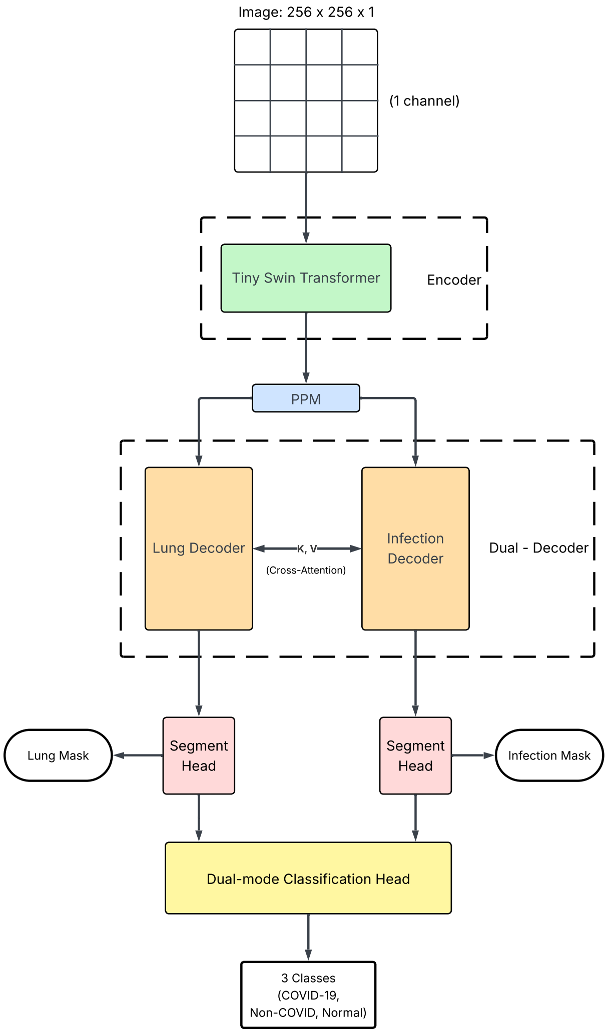

Fig. 1. Model Architecture

1) Encoder: The encoder is based on the Swin Transformer Tiny architecture, implemented from scratch in pure PyTorch without pretrained weights. The input grayscale chest X-ray image (256×256×1) first passes through a patch embedding layer, which uses a 4×4 convolution with stride 4 to create patch tokens of dimension 96.

The encoder consists of four stages with progressively increasing channel dimensions [96, 192, 384, 768]. Each stage contains 2 Swin Transformer blocks. Each block performs window-based multi-head self-attention (W-MSA) followed by a shifted window multi-head self-attention (SW-MSA), enabling efficient global context modeling. The window size is set to 7, and the attention mechanism uses relative position bias for spatial awareness.

Between stages, patch merging layers downsample the spatial dimensions by 2× while doubling the channel dimension. The encoder outputs multi-scale feature maps at resolutions of 64×64, 32×32, 16×16, and 8×8 pixels. The final bottleneck features are enhanced by a 4-level Pyramid Pooling Module (PPM) that captures multi-scale context at 1×1, 2×2, 3×3, and 6×6 pool sizes.

2) Decoder: The decoder employs a dual-branch architecture—one for lung segmentation and one for infection segmentation. Both branches share the same structure but operate independently while exchanging information via cross-attention.

Each decoder branch consists of four upsampling stages with channel dimensions [384, 192, 96, 48]. Each stage contains an UpConv block that performs 2× bilinear upsampling followed by concatenation with skip connections from the corresponding encoder stage and two 3×3 convolutions with batch normalization and ReLU.

A key innovation is the cross-attention mechanism between the two decoder branches at each scale. This allows the lung decoder to attend to infection features and vice versa, enabling mutual information exchange. The cross-attention uses a channel-wise squeeze-and-excitation mechanism with global average pooling, followed by spatial refinement using depthwise separable convolutions. The number of attention heads decreases with scale: 8, 4, 2, and 1 for the four stages respectively.

3) Heads: The model has three output heads:

Lung Segmentation Head: Takes the 48-channel features from the lung decoder, applies a 3×3 convolution reducing to 24 channels, followed by a 1×1 convolution to produce a single-channel output. A final 2× bilinear upsampling and sigmoid activation produces the lung mask at 256×256.

Infection Segmentation Head: Identical architecture to the lung head but operates on infection decoder features.

Classification Head: Operates in two modes depending on the training phase. In Phase 1, it processes only the lung mask through a 3×3 convolution with 64 output channels. In Phase 2, it concatenates both lung and infection masks (2 channels) before the convolution. Both modes then apply global average pooling followed by a fully connected layer to produce 3-class logits (Normal, Non-COVID, COVID-19).

B. Training Procedure

1) Loss Function: The model is trained using a composite loss function that addresses both the segmentation and classification tasks. For segmentation (both lung and infection), a combination of Binary Cross-Entropy (BCE) and Dice Loss is used to handle class imbalance and ensure accurate boundary delineation:

$$\mathcal{L}_{seg} = 0.5 \cdot \mathcal{L}_{BCE} + \mathcal{L}_{Dice} \tag{1}$$

For the classification task, the standard Cross-Entropy Loss is minimized:

$$\mathcal{L}_{cls} = CrossEntropy(y, \hat{y}) \tag{2}$$

2) Optimizer: The network is optimized using the AdamW optimizer with a weight decay of 0.01. A Cosine Annealing learning rate scheduler is employed to adjust the learning rate initially from 1×10^{-4} down to a minimum of 1×10^{-6} over the course of training epochs.

3) Phase 1: Lung Segmentation + Classification: In the first phase, the model is trained on the Lung Segmentation Dataset (33,920 images) for 50 epochs. During this phase, the infection decoder branch and infection head are frozen. The total loss is a weighted sum of the lung segmentation loss and classification loss:

$$\mathcal{L}_{total} = 1.0 \cdot \mathcal{L}_{lung} + 0.5 \cdot \mathcal{L}_{cls} \tag{3}$$

The training loss curve for Phase 1 is shown in Figure 2.

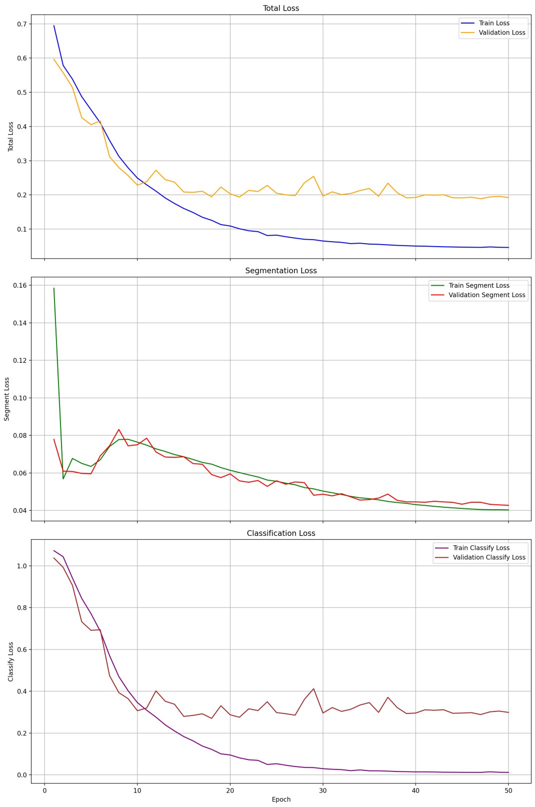

Fig. 2. Training and Validation Loss during Phase 1

4) Phase 2: All of phase 1 + Infection Segmentation: In the second phase, the model is fine-tuned on the Infection Segmentation Dataset (5,826 images) for 30 epochs. All layers are unfrozen. A differential learning rate strategy is applied: newly initialized text components (infection decoder and head) use a learning rate 5× higher than the base rate, while pretrained components use a rate 0.1× lower to preserve learned features.

The infection loss is computed only for COVID-19 positive cases. For Normal and Non-COVID cases, the infection loss is zeroed out. The total loss combines all three tasks:

$$\mathcal{L}_{total} = 1.0 \cdot \mathcal{L}_{lung} + 1.0 \cdot \mathcal{L}_{infection} + 0.3 \cdot \mathcal{L}_{cls} \tag{4}$$

The classification weight is reduced to 0.3 as the classifier is already well-trained. The training loss curve for Phase 2 is shown in Figure 3.

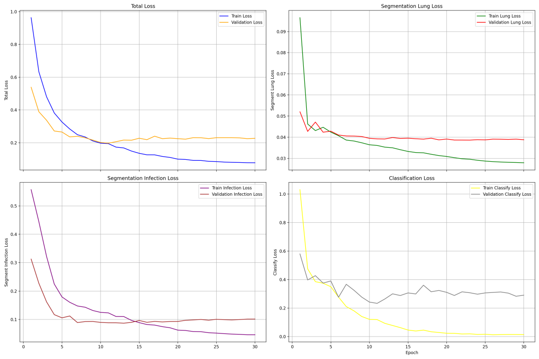

Fig. 3. Training and Validation Loss during Phase 2

IV. Results

A. *Phase 1: Lung Segmentation + Classification*

B. *Phase 2: All of phase 1 + Infection Segmentation*

V. Discussion

VI. Conclusion