Efficient Finetuning of a Visual State Space Model (VMamba)

for Multiclass Medical Image Segmentation

An Implementation of ActiveFT with a VM-UNet on the ACDC Dataset

Presented by:

Chamakuri Sowjanya
EE25MT013
M.Tech EECE (CSPML)

Instructor:

Dr. Vandhana Bharthi,
Assistant Professor,
IIT Dharwad

Introduction

The General Problem

Deep learning models for medical imaging require vast amounts of high-quality, pixel-level annotations.

The Annotation Bottleneck

This manual annotation process is a major bottleneck. It is extremely expensive and requires thousands of hours from highly trained medical experts.

Our Proposed Solution

Instead of random selection, we will implement **ActiveFT**, a 2023 algorithm that intelligently selects a small, data-efficient subset for training .

Core Technologies

We will combine this selection method with a new, powerful 2024 architecture, **VMamba**, to build a highly efficient segmentation pipeline .

Problem Formulation

Project Hypothesis

We can achieve high segmentation accuracy by training on only a **fraction** of labeled data, if we select that fraction intelligently using **ActiveFT**.

Formal Task (ACDC Dataset)

- **Given:** A pool of 1,841 2D cardiac slices with 4 classes (BG, RV, Myo, LV).
- **Objective:** Select a 20% subset ($\mathcal{D}_{20\%}$) and train a model (f_θ) that minimizes the segmentation loss.

Mathematical Objective

Minimize a hybrid loss on the selected subset:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_{20\%}} \left[\mathcal{L}_{CE}(f_\theta(x), y) + \mathcal{L}_{Dice}(f_\theta(x), y)\right]$$

Proposed Approach

Core Objectives

1. Implement the **'VMamba-Small'** ('BackboneVSSM') as a pre-trained feature extractor.
2. Apply the **ActiveFT (CVPR 2023)** algorithm to intelligently select a 20% data subset.
3. Design and train a **VM-UNet** with a pre-trained encoder and skip connections.
4. Evaluate the final multiclass segmentation performance (RV, Myo, LV) on a held-out test set.

The ACDC Dataset: Data & Preprocessing

Raw Data

100 cardiac MRI volumes (`.nii.gz`), each with segmentation masks.

Preprocessing

- Converted 3D volumes to 2D slices.
- Removed slices with empty masks (label $= 0$ everywhere).

Final Dataset

1,841 labeled 2D slices (4 classes):
0: BG, 1: RV, 2: Myo, 3: LV.

Feature Extraction: VMamba-S Backbone

Goal

Get high-quality "embedded tokens" for all 1,841 slices.

Model: VMamba-S[s2l15]

- **Architecture:** `depths=[2, 2, 15, 2],dims=[96, 192, 384, 768]`
- **Checkpoint:** `vssm_small_...pth` (ImageNet pre-trained).

Process

- A single forward pass (inference) was performed over all slices.
- We took the output from the final stage (before the classification head) and applied global average pooling.
- **Output:** 'features.npy' with shape '(1841, 768)'.

Methodology: The VMamba Architecture

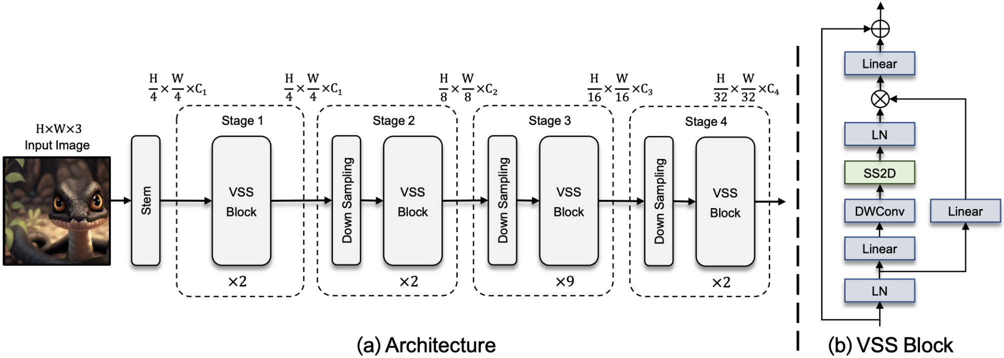


Figure: The VMamba hierarchical architecture (a) and the core VSS Block (b)[1]

Methodology: Architecture Explained

Hierarchical Architecture (Fig. 3a)

- The model is built in 4 hierarchical stages, as seen in the diagram.
- Our `VMamba-S` config uses `depths=[2, 2, 15, 2]`:
  - **Stage 1:** 2 VSS Blocks
  - **Stage 2:** 2 VSS Blocks
  - **Stage 3:** **15** VSS Blocks
  - **Stage 4:** 2 VSS Blocks

The VSS Block (Fig. 3d)

- This is the core building block of the entire model.
- It contains the crucial **2D Selective Scan (SS2D)** module.
- The SS2D module is what gives VMamba its power, replacing the self-attention mechanism from Transformers.

Algorithm: ActiveFT (CVPR 2023)

Goal

Select a small subset that is both **representative** and **diverse**.

Optimization

Select k centers $\{\theta_i\}_{i=1}^{k}$ by minimizing:

$$L = L_{emd} + \lambda L_{div}$$

$$L_{emd} = \text{Earth Mover's Distance (Sinkhorn)}, \quad L_{div} = -\log(\text{mean}\|\theta_i - \theta_j\|)$$

Outcome

Selected 20% subset $\rightarrow$ 368 most informative slices.

VM-UNet (Final Model)

Encoder

VMamba-S pretrained backbone for strong feature extraction.

Decoder

- ConvTranspose2d upsampling layers.
- Skip connections ($e_i \rightarrow d_i$) restore spatial details.

Training Setup & Results (ActiveFT 20% Subset)

Setup Summary

Data Split (20% ActiveFT):

Split	Slices	% of Total
Train	282	15.3
Val	49	2.7
Test	184	10.0

Training

Details:

- Optimizer: AdamW (LR 3×10^{-4})
- Loss: CE + Dice
- Scheduler: Cosine Annealing
- Batch Size: 2

Training Progress (4 Epochs)

Epoch	Train loss	Val loss
1	0.9527	0.4580
2	0.3398	0.2853
3	0.2256	0.2116
4	**0.1622**	**0.2024**

- Smooth convergence over epochs.
- Best model: **Epoch 4 (Val = 0.2024)**.
- Training time: $\sim$51 min/epoch.

Training Results (ActiveFT 20% Subset)

```
Epoch 1/4 [VAL]: 100%|██████████| 25/25 [00:35<00:00,  1.43s/it]
Epoch 1/4 | Train: 0.9527 | Val: 0.4580 | LR: 2.56e-04 | Time: 3125.4s
  BEST MODEL SAVED (val_loss=0.4580)
Epoch 2/4 [TRAIN]: 100%|██████████| 141/141 [51:13<00:00, 21.80s/it]
Epoch 2/4 [VAL]: 100%|██████████| 25/25 [00:33<00:00,  1.35s/it]
Epoch 2/4 | Train: 0.3398 | Val: 0.2853 | LR: 1.50e-04 | Time: 3107.5s
  BEST MODEL SAVED (val_loss=0.2853)
Epoch 3/4 [TRAIN]: 100%|██████████| 141/141 [51:09<00:00, 21.77s/it]
Epoch 3/4 [VAL]: 100%|██████████| 25/25 [00:34<00:00,  1.36s/it]
Epoch 3/4 | Train: 0.2256 | Val: 0.2116 | LR: 4.39e-05 | Time: 3103.2s
  BEST MODEL SAVED (val_loss=0.2116)
Epoch 4/4 [TRAIN]: 100%|██████████| 141/141 [51:09<00:00, 21.77s/it]
Epoch 4/4 [VAL]: 100%|██████████| 25/25 [00:34<00:00,  1.39s/it]
Epoch 4/4 | Train: 0.1622 | Val: 0.2024 | LR: 0.00e+00 | Time: 3104.8s
  BEST MODEL SAVED (val_loss=0.2024)

Training finished! Best val loss: 0.2024
```

Figure: Training and Validation Loss

Testing Results (Evaluation on 184 Slices)

Evaluation Summary

- **Dataset:** 184 test samples from `splits_20p/test.csv`
- **Runtime:** ∼3.3 minutes (PyTorch fallback mode)

Quantitative Metrics (Mean over RV, Myo, LV)

Metric	Value	Interpretation
Dice (mDice)	**0.7515**	Excellent overlap accuracy
IoU (mIoU)	0.6677	Strong region agreement
Sensitivity	0.7621	Correctly detected heart pixels
Specificity	0.9985	Near-perfect background suppression
HD95	4.2983	Accurate boundary localization (≈4.3 px)
Pixel Accuracy	0.9928	Overall excellent pixel-level accuracy

Testing Results

```
  warnings.warn(
Testing: 100%|██████████| 184/184 [03:17<00:00,  1.07s/it]
--- TEST SET RESULTS (Mean over RV, Myo, LV) ---
dice_mean            : 0.7515
iou_mean             : 0.6677
sensitivity_mean     : 0.7621
specificity_mean     : 0.9985
hd95_mean            : 4.2983
pixel_acc            : 0.9928
```

Figure: Testing results

Testing Results(ActiveFT 5% Subset)

The best model (from Epoch 10, val_loss=0.2697) was evaluated on the independent test set (184 slices).

Final Metrics (Mean over RV, Myo, LV)

Metric	Value	Interpretation
Dice (mDice)	**0.7314**	A **good score**, proving the model is learning accurately.
IoU (mIoU)	**0.6375**	Solid region agreement.
HD95	**6.5433**	**Good boundary precision** (~6.5 pixels).
Sensitivity	0.7559	Successfully finds ~76% of true heart pixels.
Specificity	0.9976	Near-perfect background suppression.
SSIM	**0.9856**	**Excellent perceptual similarity.** .
Pixel Accuracy	0.9903	High (dominated by background).

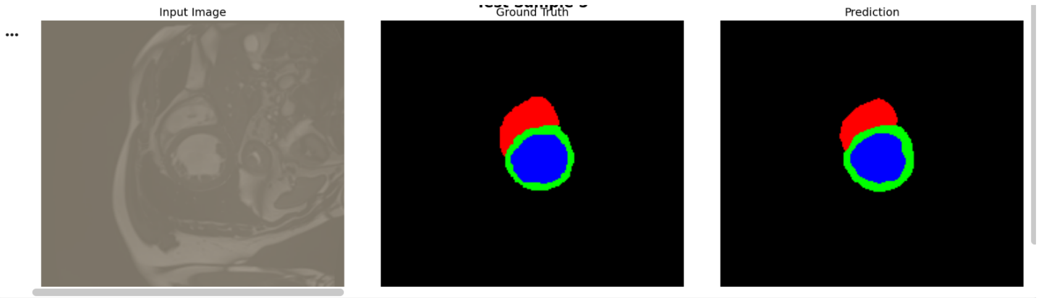
Testing Results K=5%

```
Testing: 100%|██████████| 184/184 [03:11<00:00,  1.04s/it]
--- TEST SET RESULTS (Mean over RV, Myo, LV) ---
dice_mean            : 0.7314
iou_mean             : 0.6375
sensitivity_mean     : 0.7559
specificity_mean     : 0.9976
hd95_mean            : 6.5433
ssim_mean            : 0.9856
pixel_acc            : 0.9903

Results saved to /content/results_summary.json
EVALUATION COMPLETE
```

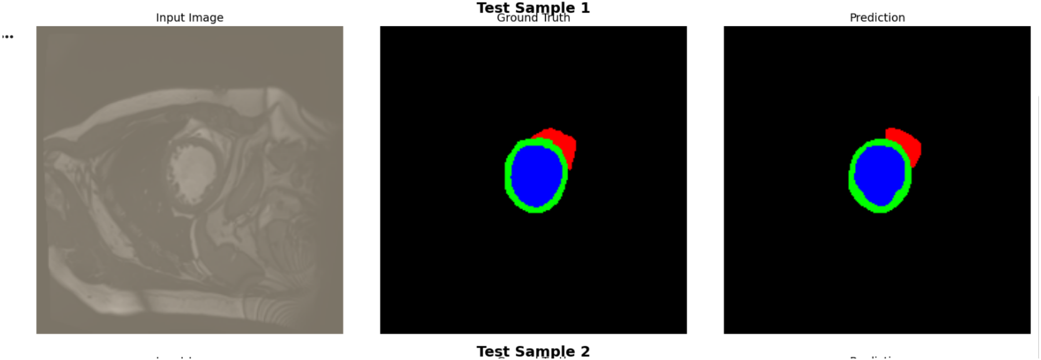
Figure: Test results

Qualitative Results K=20%



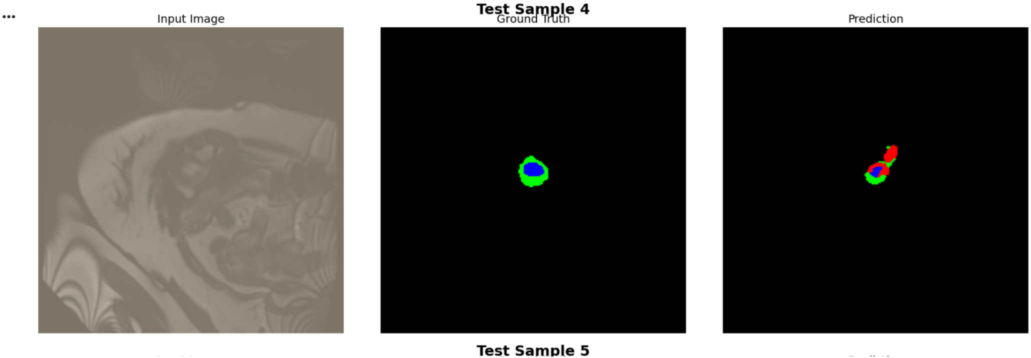
From left to right: Input Image, Ground Truth Mask, Model Prediction.

Qualitative Results K=5%



From left to right: Input Image, Ground Truth Mask, Model Prediction.

Qualitative Results K=5%



From left to right: Input Image, Ground Truth Mask, Model Prediction.

Project Novelty & Contributions

1. Novel Synthesis of SOTA Methods

- This project is a novel investigation combining two SOTA methods:
  - **VMamba (2024):** A Visual State Space Model.
  - **ActiveFT (2023):** A data-efficient selection algorithm.
- We successfully replaced the original paper's ViT backbone with the more modern VMamba.

2. Novel Architecture (The "PERFECT VM-UNet")

- We designed and implemented a custom `VM_UNet` architecture.
- **Key Feature:** It integrates a **pre-trained VMamba-S encoder** with a CNN decoder, and correctly implements **skip connections** (`torch.cat`) for precise segmentation.

Conclusion & Future Work

Conclusion

- ActiveFT reduced data requirement to 20%.
- VMamba-UNet achieved strong multiclass segmentation performance.

Future Work

- Requires CUDA-enabled GPU to enable fast VMamba kernels.
- Run the full experiment for different selection percentages (e.g., 2%, 5%, 10%) to analyze the performance-to-data-cost trade-off.
- Train all models for more epochs (e.g., 50-100) to achieve their optimal performance.

References I

Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, J. Jiao, and Y. Liu, "Vmamba: Visual state space model," 2024.

Thank You

Efficient Finetuning of a Visual State Space Model (VMamba)

November 17, 2025