Few-Shot Dermoscopic Image Segmentation with a Frozen Self-Supervised Pretrained Encoder

Zixiao Xu (zxu129@jh.edu)

Johns Hopkins University

The Whiting School of Engineering

Abstract:

Accurate segmentation of dermoscopic images is essential for early skin cancer detection, but it is often limited by the high cost and scarcity of expert-labeled data. This study presents a practical few-shot segmentation framework that leverages a frozen self-supervised pretrained encoder to reduce data and computational requirements. Using the ISIC 2017 dataset, we pretrain a ViT-based encoder on a self-supervised image reconstruction task and reuse it in a segmentation pipeline with a lightweight decoder. We compare this approach with end-to-end training across various training data sizes. Results show that the frozen encoder models maintain better segmentation performance—particularly in low-data settings—and reduce training time by an average of 47.79%. These findings suggest that decoupling representation learning from task-specific fine-tuning offers a robust and efficient solution for deploying medical image segmentation in low-resource clinical environments.

Open-Source Code Repository Url: IsaacXu0808/DLMI-project

1 Introduction

Dermoscopy is a non-invasive imaging technique widely used for the early detection of skin cancer. Automated analysis of dermoscopic images typically involves two key tasks: global lesion classification (e.g., melanoma vs. benign) and pixel-level segmentation to delineate lesion boundaries. While deep learning has shown great promise in both tasks, its success remains heavily dependent on large-scale, expert-annotated datasets.

However, in medical imaging—particularly dermatology—annotating images is time-consuming, expensive, and requires expert knowledge [1, 2]. This leads to a scarcity of high-quality labeled data, especially for fine-grained tasks like lesion segmentation. At the same time, unlabeled dermoscopic images are abundant, and cloud-based pretraining is computationally feasible. Yet, in many real-world clinical deployments, computation and annotation budgets are limited.

To address this challenge, I propose a practical and efficient paradigm: pretrain a high-capacity encoder on unlabeled dermoscopic data using a self-supervised learning objective, freeze the encoder, and reuse it for downstream segmentation tasks in few-shot settings. This approach decouples the expensive model training phase from downstream deployment, enabling accurate and resource-efficient medical image analysis in label-scarce and compute-constrained environments.

This study demonstrates that, under extremely limited supervision, although the loss of performance is inevitable, freezing a self-supervised pretrained encoder—rather than performing end-to-end training—preserves more performance and generalization ability. Moreover, this strategy reduces training time by approximately 47,79%, making it promising for low-resource medical imaging scenarios.

2 Data Set

2.1 Eligibility and Inclusion Criteria

This study used the ISIC 2017 Challenge dataset [3] (Part 1: Lesion Segmentation), which contains dermoscopic images of skin lesions—including both benign and malignant cases—collected from international clinical centers. Released as part of a public challenge, the dataset includes only high-quality images with expert-annotated segmentation masks. Lesions without clear boundaries or missing annotations were excluded during dataset curation. All masks were manually traced or refined by clinicians to ensure reliable ground truth. Images were acquired under clinical conditions with various dermatoscopes, contributing to diverse visual appearances. The dataset, compiled before 2017, is publicly available through the ISIC Archive.

2.2 De-identification Methods

All images and metadata in the ISIC 2017 dataset are fully de-identified in compliance with privacy standards [3]. The ISIC organizers removed all personal identifiers, including embedded EXIF metadata, before release. Only non-identifiable clinical variables (e.g., age, sex, diagnosis) are included, and each image is labeled with a random ID (e.g., "ISIC_#######"). The dataset is released under a public domain (CC-0) license and contains no private information. As such, this study used only anonymized data and required no additional de-identification.

2.3 Handling missing data

The ISIC 2017 dataset [3] contains no missing data for the segmentation task—each image has a corresponding expert-annotated mask. We verified that all image-mask pairs were complete and readable. As a result, no imputation or data exclusion was required, and the study proceeded with a complete dataset.

2.4 Selection of data partitions

This project followed the official ISIC 2017 challenge splits [3], which divide the dataset into 2,000 training, 150 validation, and 600 test images, each with expert-provided segmentation masks. These partitions are disjoint at the image level, ensuring no overlap or information leakage between subsets. The training set was used to learn model parameters, the validation set to tune hyperparameters, and the test set—whose masks were withheld during training—was used for final performance evaluation. By adhering to the original challenge configuration, we ensure consistency with prior studies. The dataset is publicly available on the ISIC 2017 Challenge website, and test masks were released after the challenge for benchmarking.

2.5 Definition and Rationale of Ground Truth (GT) Reference

This study used the ISIC 2017 [3] Challenge Part 1 dataset as the ground truth for lesion segmentation. Each dermoscopic image includes an expert-annotated binary mask created by dermatologists. As a benchmark from an international challenge curated by ISIC, it provides a widely accepted, publicly available, and reproducible standard for evaluating segmentation performance.

2.6 Source of GT Annotations

The ground truth segmentation masks in the ISIC 2017 [3] dataset were manually created by experienced dermatologists and dermoscopy experts affiliated with the International Skin Imaging Collaboration (ISIC). Each annotator was a board-certified clinician or trained expert in skin lesion analysis, qualified to accurately delineate lesion boundaries in dermoscopic images. Annotations were produced using standardized protocols to ensure

that the entire lesion area—excluding surrounding healthy skin or artifacts—was consistently captured. In cases of uncertainty, annotations were reviewed or corrected by senior experts to maintain consistency. All masks were included as part of the official ISIC 2017 Challenge dataset release, ensuring that the annotations are both clinically reliable and suitable as a reference standard for benchmarking segmentation algorithms.

2.7 Annotation tools

Segmentation masks in the ISIC 2017 [3] dataset were manually created using specialized annotation tools. Common methods included freehand polygon drawing, interactive flood-fill with manual refinement, and semi-automated segmentation followed by expert correction. Masks were saved as binary images aligned with the input dimensions. Most annotations were created using ISIC's web-based or equivalent medical imaging tools. These expert-generated masks provide consistent, high-quality lesion boundaries suitable for use as ground truth.

2.8 Annotation Variability and Quality Control

While some inter-rater variability exists among experts, ISIC 2017 masks were manually reviewed to ensure consistency. In prior ISIC studies, expert agreement reached an average Jaccard index of ~0.78, reflecting high concordance. Discrepancies were resolved through expert review and consensus, helping to ensure reliable and standardized ground truth.

2.9 Preprocessing Steps

On top of the original ISIC2017 dataset [3], all the 2750 images and corresponding masks were resized to 224 by 224.

3 Models

3.1 Model Architectures

3.1.1 Image Encoder

This project selected a ViT [4]-based image encoder (VE). The components are as follows:

- **Input**: The model takes a batch of RGB images of shape [B, 3, 224, 224], where B is the batch size.
- **Patch Embedding**: A 16×16 convolution (Conv2d) divides the images into non-overlapping patches and projects them into 768-dimensional vectors. Output shape: [B, 768, 14, 14].
- **Flatten & Transpose**: The spatial dimensions are flattened and transposed to form a sequence of 196 tokens (one per patch). Output shape: [B, 196, 768].

- **[CLS] Token & Positional Embedding:** A learnable [CLS] token is prepended to the sequence, making it 197 tokens long. A learned positional embedding is added to retain spatial order. Output shape: [B, 197, 768].
- **Transformer Encoder**: The sequence passes through 12 Transformer blocks, each with self-attention and MLP layers. Output shape: [B, 197, 768].
- **Layer Normalization**: Applied to the final output of the Transformer. Output shape: [B, 197, 768].
- **Output Separation**: CLS token: Extracted as [B, 768] for classification tasks.
- **Patch tokens**: Remaining 196 tokens reshaped into a feature map. Final shape: [B, 768, 14, 14]. Note that the [CLS] token is not trained in the self-supervised image reconstruction pipeline.

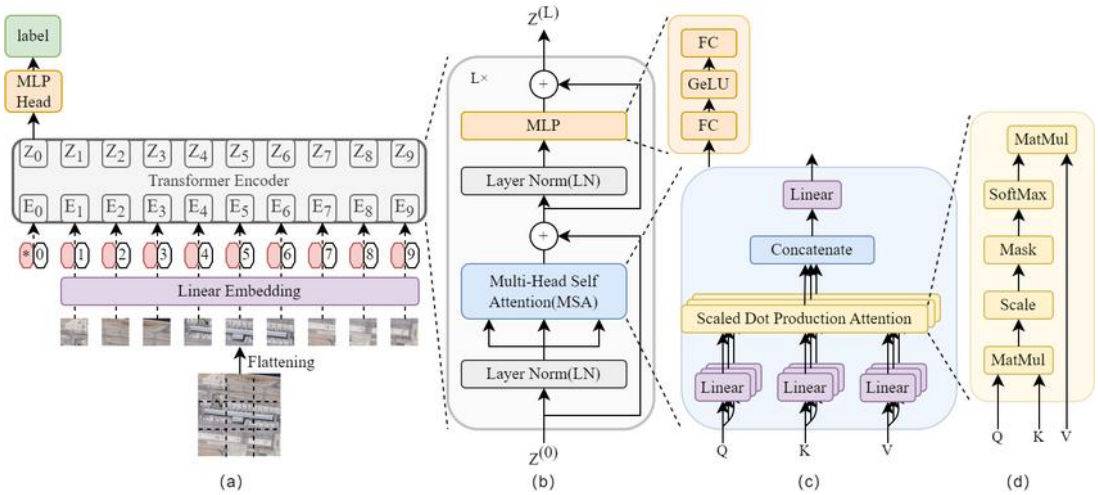

Figure 1. The standard architecture of ViT.

On top of ViT, this project removed the MLP head. Instead, a layer normalization is applied to the output of transformer encoders. Then the output of shape [B, 197, 768] is split into the [CLS] token of shape [B, 1, 768], and the feature map of shape [B, 196, 768]. Eventually, the feature map is reshaped into an image-like tensor of shape [B, 768, 14, 14] and fed to the decoders for downstream tasks. The number of parameters is 85,798,656 (86M). Using float32 for parameters, the memory consumption is 327.30 MB (forward pass not included).

3.1.2 Reconstruction Decoder

The Reconstruction Decoder (RD) is a lightweight convolutional decoder with ReLu activations and up-sampling layers, designed to reconstruct images from high-level feature maps output by a transformer-based encoder. It is a classic example of a decoder architecture in an encoder-decoder framework, commonly used in autoencoders, self-supervised learning, and image-to-image translation tasks. The number of parameters is 5,089,923 and the memory consumption is 19.42 MB (forward pass not included). The details are as follows:

Tabel 1. Layers of Reconstruction Decoder

Layer	Output Shape	Kernel Size	Scale Factor
Conv2d-1	[B, 512, 14, 14]	3	
ReLU-1	[B, 512, 14, 14]		
Upsample-1	[B, 512, 28, 28]		2
Conv2d-2	[B, 256, 28, 28]	3	
ReLU-2	[B, 256, 28, 28]		
Upsample-2	[B, 256, 56, 56]		2
Conv2d-3	[B, 128, 56, 56]	3	
ReLU-3	[B, 128, 56, 56]		
Upsample-3	[B, 128, 112, 112]		2
Conv2d-4	[B, 64, 112, 112]	3	
ReLU-4	[B, 64, 112, 112]		
Upsample-4	[B, 64, 224, 224]		2
Conv2d-5	[B, 3, 224, 224]	3	
Sigmoid	[B, 3, 224, 224]		

Note that the layers in table 1 are connected sequentially, without residual blocks or skip connections.

3.1.3 Segmentation Decoder

The Segmentation Decoder (SD) is a lightweight convolutional decoder with ReLu activations, batch normalization layers and up-sampling layers, designed to compact feature maps from VE into high-resolution, per-pixel predictions of binary labels (binary segmentation masks). The number of parameters is 2,158,017 (2M) and the memory consumption is 8.23 MB (forward pass not included). The details are as follows:

Table 2. Layers of Segmentation Decoder

Layer	Output Shape	Kernel Size	Scale Factor
Conv2d-1	[-1, 256, 14, 14]	3	
ReLU-1	[-1, 256, 14, 14]		
BatchNorm2d-1	[-1, 256, 28, 28]		
Upsample-1	[-1, 256, 28, 28]		2
Conv2d-2	[-1, 128, 28, 28]	3	
ReLU-2	[-1, 128, 28, 28]		
BatchNorm2d-2	[-1, 128, 28, 28]		
Upsample-2	[-1, 128, 56, 56]		2
Conv2d-3	[-1, 64, 56, 56]	3	
ReLU-3	[-1, 64, 56, 56]		
BatchNorm2d-3	[-1, 64, 56, 56]		
Upsample-3	[-1, 64, 112, 112]		2
Conv2d-4	[-1, 32, 112, 112]	3	

Table 2. Layers of Segmentation Decoder (Continued)

Layer	Output Shape	Kernel Size	Scale Factor
ReLU-4	[-1, 32, 112, 112]		
BatchNorm2d-4	[-1, 32, 112, 112]		
Upsample-4	[-1, 32, 224, 224]		2
Conv2d-5	[-1, 1, 224, 224]	3	
Sigmoid	[-1, 1, 224, 224]		

Note that the layers in table 2 are connected sequentially, without residual blocks or skip connections.

3.2 Software libraries, Frameworks, and Packages

- This project utilized Python libraries *PyTorch* and *torchvision* to build data processing pipelines, data loaders, deep learning models and model training pipelines and evaluation pipelines.
- Python library *matplotlib* was used for visualization.
- CUDA [5] was utilized for GPU acceleration.

3.3 Initialization of Model Parameters

- **VE**: *ViT_B_16_Weights.DEFAULT* from *PyTorch*, which were trained from scratch by using a modified version of DeIT's training recipe [6].
- **Frozen VE**: VE is then loaded with pretrained parameters and frozen for efficient image segmentation training, where these parameters were pretrained on the ISIC2017 training dataset with the self-supervised image reconstruction task.
- **SD** and **RD**: Default initialization for convolutional layers from *Pytorch*, which utilizes Kaiming's initializer [7].

3.4 Details of the Training Approach

3.4.1 Self-supervised Dermoscopic Image Reconstruction

- The self-supervised dermoscopic image reconstruction model is composed of VE and RD and pretrained end-to-end (the parameters are θ). The pretrained VE (VE*) is saved as an image encoder for later steps.
- The ground truth of this task is the resized training images, and the input is each original training image (x) masked by a pixel-level Gaussian noise with zero-mean and a standard deviation of 0.1 (ϵ).
- The loss function of this task is MSE loss and the optimizer is Adam optimizer [8]. The hyperparameters include: batch size = 40, learning rate = 2e-5, number of epochs = 32.

3.4.2 Supervised (Few-shot) Dermoscopic Image Segmentation

- The dermoscopic image segmentation model is composed of VE and SD.
- In the end-to-end training, the VE is initialized with techniques in section 3.3. In the training with frozen and pretrained VE, VE is loaded with parameters of VE* and frozen. SD is always initialized with Kaiming's initializer [7].
- The ground truths of this task are the resized binary segmentation masks, and the inputs are the resized images augmented with random horizontal and vertical flips.
- The loss function is BCE loss, and the optimizer is SGD optimizer [9]. The hyperparameters include learning rate = 2e-4, momentum = 0.9, weight decay = 1e-4. To simulate the lack of labelled data, different data sizes from 2000 to 16 (number of training images **Randomly** selected from the training data) are used. The number of epochs, validation images and batch sizes change accordingly (see table 3).
- The optimal models under each data size are named E-{data size} for models trained end-to-end, and F-{data size} for models trained with frozen pretrained VE.

3.5 Method of Selecting the Optimal Model

This project selects models with minimal validation loss during the training. For example, in the following plot of the losses during the training of the reconstruction model, the model is considered optimized after the epoch at dashed vertical line.

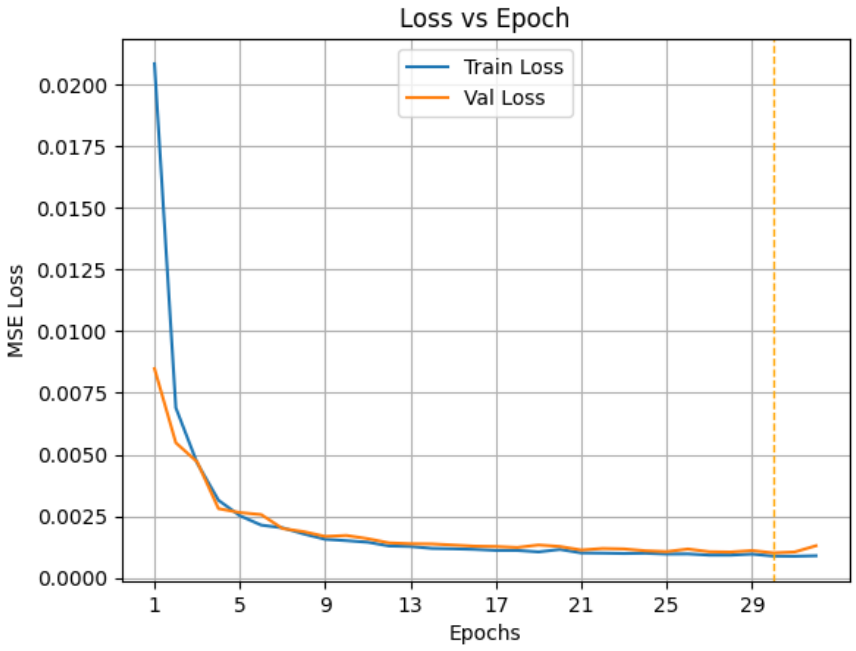

Figure 2. Loss Curves of Reconstruction Model Training

For the loss curves of all segmentation model trainings, see appendix.

4 Evaluation and Results

4.1 Reconstruction

This project does not include metrics for reconstruction models since it focuses on segmentation. From fig. 2 and the following visualizations, we can see that the model has converged and performed well. Although the details are blurred in the output, encoded information about the images should be sufficient for image segmentation.

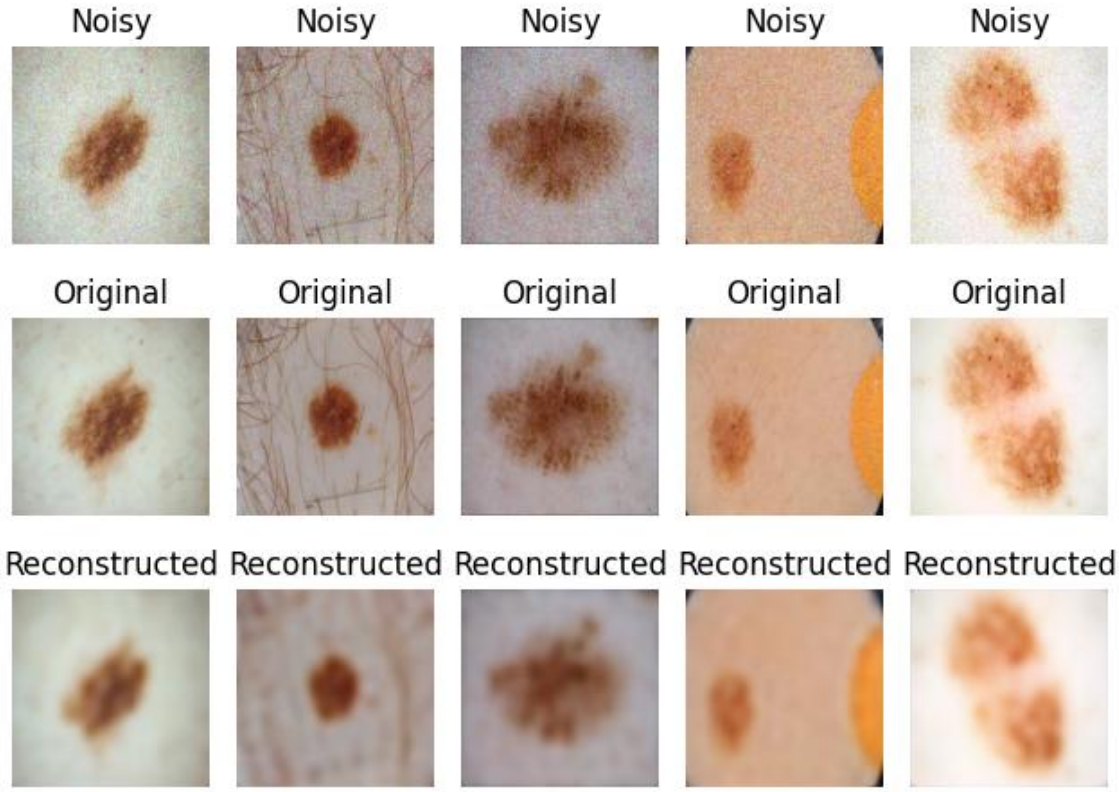

Figure 3. Reconstruction Model Outputs

4.2 Segmentation

4.2.1 Time Consumption of Traing

To demonstrate the efficiency of the proposed training pipeline using frozen pretrained encoder, this project measured the training times of models. The training runs on a 64-bit Windows computer with Intel(R) Core i7-9700K CPU and Nvidia RTX4070 Ti Super GPU.

The following table shows the data sizes and the corresponding numbers of epochs and batch sizes, the models' names, their training time consumption and relative times saved. The threshold for pixel-level binary classification is 0.5.

Table 3. Time Consumption of Model Training

Data Size	Val Size	#Epochs	Batch Size	Model	Time (sec)	Time Saved
2000	150	100	50	E-2000	1638.6	55.29%
				F-2000	**732.7**	
160	80	200	10	E-160	347.6	48.73%
				F-160	**178.2**	
80	40	300	10	E-80	293.6	48.23%
				F-80	**151.8**	
40	20	500	10	E-40	260.4	43.01%
				F-40	**148.4**	
20	10	800	10	E-20	275.0	44.91%
				F-20	**151.5**	
10	5	1200	5	E-10	175.6	46.53%
				F-10	**93.9**	

From table 3, we can see that the proposed training pipeline using frozen pretrained encoder saves on average 47.79% time consumption of training compared to training both the encoder and the decoder end-to-end.

4.2.2 Performance Metrics

To evaluate the performance of segmentation models, this project includes three main metrics from ISIC2017 [3]: *Jaccard Index (IoU)*, *Dice Score* and *Sensitivity*. While this paper provides the results of all three metrics, this project chooses Jaccard Index as the primary evaluation metrics of performance since the participants of the challenge were ranked and awards granted based solely on the Jaccard index [3].

$$IoU = \frac{TP}{TP + FP + FN + \varepsilon} \qquad Dice\ Score = \frac{2 \times TP}{2 \times TP + FP + FN + \varepsilon} \qquad Sensitivity = \frac{TP}{TP + FN + \varepsilon}$$

Note that a small factor ε =1e-6 is added to denominators to prevent divided-by-zero error.

To ensure a fair comparison, all performance metrics are obtained from the test dataset, as different models were exposed to different training and validation data during training. The metrics are averaged over the 600 test images.

4.2.2.1 *Mean Jaccard Index (IoU) Results*

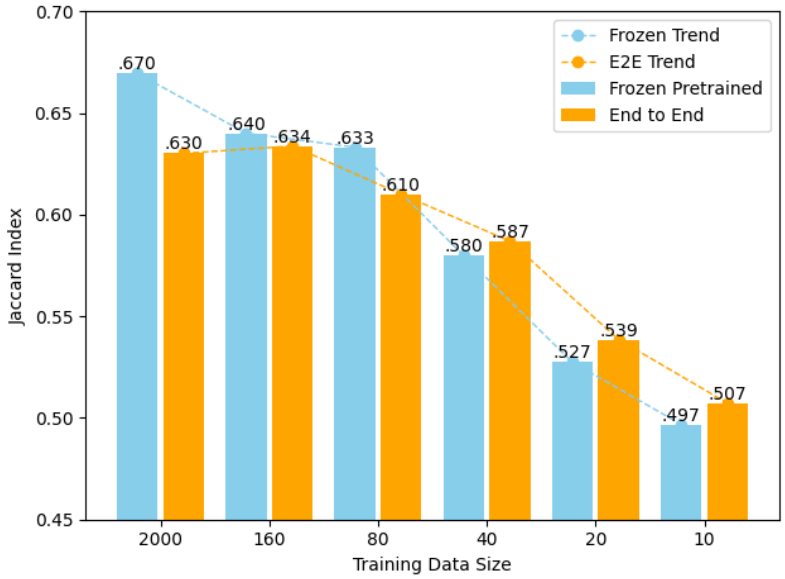

Figure 4. Mean Jaccard Index (IoU) Results

Figure 4 indicates that while the performance of both series of models decreased significantly with the decrease in training data size, the performance of models with frozen pretrained encoders decreased slower as the training data size decreased, compared to models trained end-to-end. Moreover, F-{40, 20, 10} outperformed E-{40, 20, 10} when the training data size falls below 40 (60 labeled images including validation images).

4.2.2.2 *Mean Dice Score Results*

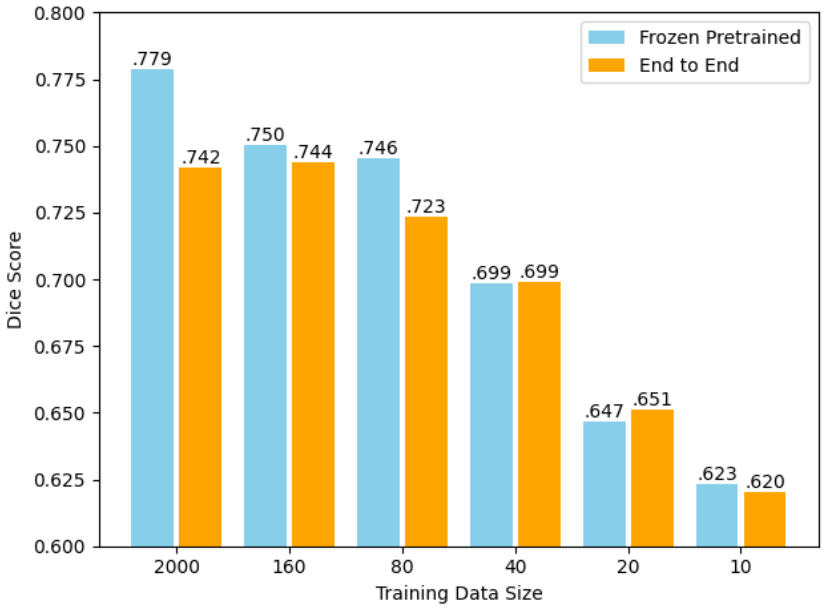

Figure 5. Mean Dice Score Results

The Dice Scores show patterns that are similar the Jaccard Indices.

4.2.2.3 Mean Sensitivity Results

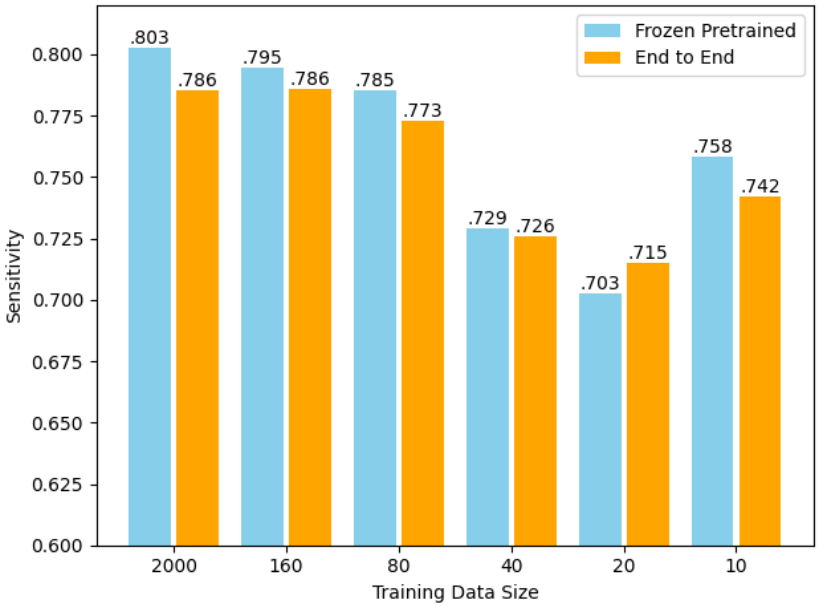

Figure 6. Mean Sensitivity Results

The increase in Sensitivity and decrease in other metrics indicates when the training data size falls below 10, the models start to collapse and indiscriminately specify more and more pixels as true class, resulting in a large TP and small FN. However, FP is also large.

5 Discussion and Conclusion

5.1 Metrics Analysis

This study evaluated segmentation performance under extreme data scarcity using three key metrics: Jaccard Index (IoU), Dice Score, and Sensitivity. Across all metrics, performance consistently declined as training data size decreased. However, models using a frozen self-supervised pretrained encoder (F-series) showed greater robustness to limited supervision than those trained end-to-end (E-series).

5.1.1 Jaccard Index (IoU)

As shown in Figure 4, Jaccard Index dropped with decreasing training data, but the F-series models exhibited a slower decline. Notably, when the number of training images dropped below 40, F-{40, 20, 10} models outperformed their end-to-end counterparts by at most 2%, indicating that the pretrained encoder captured generalizable features that reduced the dependence on large, labeled datasets. As the primary performance metric, the Jaccard Index reflects overlap between predicted and ground truth masks. Its slower decline in the F-series models suggests that frozen self-supervised encoders preserve spatial feature quality under limited supervision.

5.1.2 Dice Score

The Dice Score trends in Figure 5 mirrored the IoU patterns, supporting the conclusion that frozen pretrained encoders preserve segmentation accuracy better under few-shot conditions. This reinforces the practical value of self-supervised pretraining in medical imaging where labels are scarce.

5.1.3 Sensitivity

Figure 6 shows that sensitivity increased when training data became extremely limited (e.g., 10 or fewer samples). This suggests the models began to overpredict positive regions, classifying many pixels as part of the lesion to minimize false negatives. Although sensitivity rose, this came at the expense of precision, as false positives also increased—indicating model collapse in ultra-low data regimes.

5.1.4 Time Efficiency

Aside from accuracy, the F-series also demonstrated significant training efficiency, saving an average of 47.79% training time compared to full end-to-end models. This makes the frozen encoder approach particularly attractive for low-resource clinical settings where both data and computation resources are limited.

5.2 Failure Cases and Analysis

This section discusses outputs of models of two cases. In the following visualizations, the red masks indicate ground true segmentation masks and the green ones are the predicted segmentation masks with threshold = 0.5.

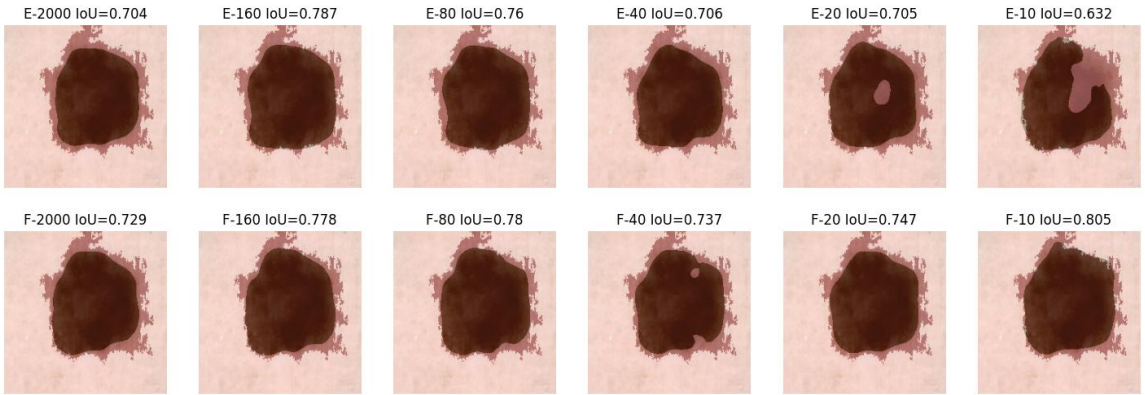

Figure 7. Case Study 1

In fig. 7, as the training data decreases, the performance of E-series models keeps decreasing while F-series models maintain consistent performance. The empty holes in the output of E-10 and E-20 indicate that without encoded information from the pretrained encoder, the models are not generalized for unseen data.

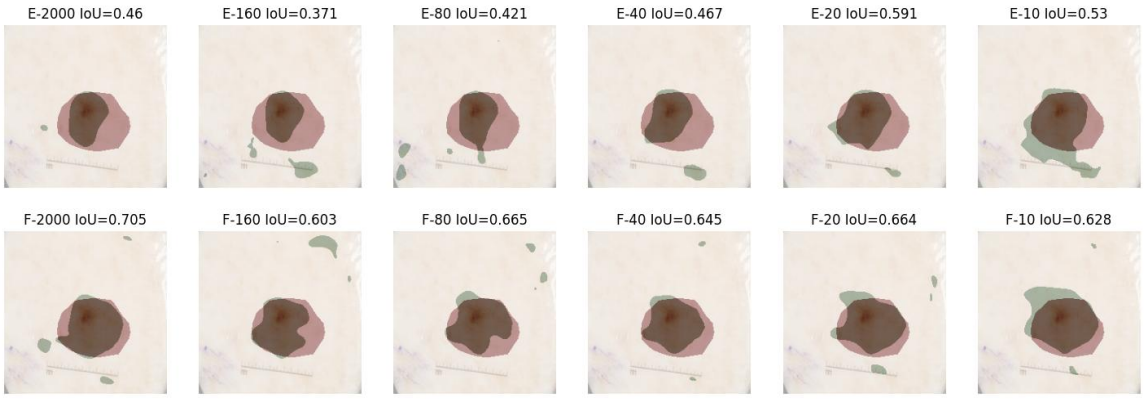

Figure 8. Case Study 2

The visual comparison reveals three key patterns: (1) the widely-spread green false-positive regions suggest that the models—especially end-to-end ones—are sensitive to color rather than lesion structure, often misclassifying similarly colored regions, (2) as training data decreases, E-series models predict increasingly large lesion areas, leading to more false positives and indicating model collapse, particularly visible in E-10, (3) in contrast, F-series models maintain more stable and localized predictions, even under extreme data scarcity, demonstrating better true positive retention and generalization from as early as F-2000.

5.3 Study Limitations

- **Performance Limitation**: This project focuses on the efficiency and effectiveness of the proposed training pipeline, not delivering a dermoscopic image segmentation method with competitive performance that is comparable to SOTAs.
- **Investigation of Model Architectures**: This project utilizes uniform encoder-decoder model architectures and not investigates into the effectiveness of different low-level model architectures.
- **Investigation of Data Augmentation**: Although data augmentation is an effective technique to enrich the training data especially for few-shot learning, it is not emphasized in this project, which only uses random horizontal and vertical flips instead of comparing various augmentation methods.
- **Statistical Uncertainty**: Since the validation data size falls under 20 for some cases, the non-inclusive data distribution of the validation data may fail the optimal model selection mechanism-choose the model with the minimum validation loss.
- **Generalizability**: This project validates the efficiency and effectiveness of the proposed training pipeline on a dermoscopic image segmentation dataset, but it may not be generalizable to other downstream tasks or medical imaging modalities.

5.4 Implications for Practice

The proposed frozen self-supervised pretrained encoder offers a practical and efficient solution for medical image segmentation in low-resource clinical settings. Its ability to maintain performance under few-shot conditions makes it especially valuable in dermatology, where annotated data is scarce and acquiring expert labels is costly. This approach is well-suited for use in semi-automated lesion analysis tools, mobile dermoscopy applications, and clinical decision support systems, enabling faster deployment with reduced training cost and minimal labeling effort. By decoupling representation learning from task-specific finetuning, this method helps bridge the gap between data-rich research environments and real-world clinical deployment.

5.5 Conclusion

This project presents a practical few-shot segmentation framework for dermoscopic images, leveraging a frozen self-supervised pretrained encoder. Through extensive experiments on the ISIC 2017 dataset, we demonstrate that this approach not only improves segmentation robustness under limited supervision but also significantly reduces training time by nearly 48% on average. Compared to end-to-end training, the frozen encoder pipeline yields better generalization and preserves segmentation accuracy, especially when training data is extremely scarce.

These findings highlight the effectiveness of decoupling representation learning from downstream training in medical imaging. By making model training more data- and compute-efficient, this work provides a promising direction for deploying deep learning systems in real-world clinical environments where annotation resources are limited.

References

[1] Ge Y, Guo Y, Das S, et al. *Few-shot learning for medical text: A review of advances, trends, and opportunities[J]*. Journal of Biomedical Informatics, 2023, 144: 104458.

[2] Irvin J, Rajpurkar P, Ko M, et al. *Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison[C]//Proceedings of the AAAI conference on artificial intelligence.* 2019, 33(01): 590-597.

[3] Gutman, David; Codella, Noel C. F.; Celebi, Emre; Helba, Brian; Marchetti, Michael; Mishra, Nabin; Halpern, Allan. *"Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC)"*. eprint arXiv:1605.01397. 2016.

[4] Dosovitskiy A, Beyer L, Kolesnikov A, et al. *An image is worth 16x16 words: Transformers for image recognition at scale*[J]. arXiv preprint arXiv:2010.11929, 2020.

[5] Kirk D. *NVIDIA CUDA software and GPU parallel computing architecture*[C]//ISMM. 2007, 7: 103-104.

[6] Touvron H, Cord M, Douze M, et al. *Training data-efficient image transformers & distillation through attention*[C]//International conference on machine learning. PMLR, 2021: 10347-10357.

[7] He K, Zhang X, Ren S, et al. *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1026-1034.

[8] Kingma D P. *Adam: A method for stochastic optimization*[J]. arXiv preprint arXiv:1412.6980, 2014.

[9] Ruder S. *An overview of gradient descent optimization algorithms*[J]. arXiv preprint arXiv:1609.04747, 2016.

Appendix-Loss vs Epoch Plots

Vertical dashed lines indicate the epoch where the optimal models are obtained.