

A REPORT
On
ASSIGNMENT 3(Deep Learning)

Shripad pate(M23MAC007)



॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Data and Computational Sciences (DCS)
Department of Mathematics
Indian Institute of Technology, Jodhpur

March 2024

Introduction

The primary goal of this assignment is to develop a segmentation model for the ISIC 2016 dataset. The approach involves leveraging a MobileNet architecture pre-trained on the ImageNet dataset as the encoder,

extracting features from the input images. The encoder's pre-training on ImageNet allows it to effectively capture a wide range of visual features. However, instead of performing classification tasks as it traditionally does, the encoder's features will aid in segmenting skin lesion images in the ISIC 2016 dataset. To accomplish this, a custom decoder architecture is designed to predict segmented masks. The decoder takes the encoded features from the MobileNet encoder and processes them to generate pixel-wise predictions, delineating the boundaries of skin lesions within the images. This segmentation process is crucial for various medical applications, particularly dermatology, where accurate delineation of skin lesions can aid in diagnosis and treatment planning. By combining the pre-trained MobileNet encoder with a custom decoder, the model aims to achieve accurate segmentation results on the ISIC 2016 dataset, contributing to advancements in automated medical image analysis and diagnosis. This approach capitalizes on the strengths of transfer learning, utilizing knowledge gained from pre-training on ImageNet to enhance performance on the task-specific ISIC 2016 dataset, ultimately facilitating more efficient and accurate segmentation of skin lesions.

Data Analysis

- **Training Image:** The training set consists of 900 images. These images serve as the basis for training the segmentation model. Each image contains a skin lesion, and the goal is to train the model to accurately segment these lesions from the background.
- **Training Mask:** Alongside the training images, there are corresponding segmented masks. These masks provide pixel-level annotations indicating which parts of the image correspond to the skin lesion and which parts belong to the background. These annotations serve as ground truth data during training, guiding the model in learning to accurately predict segmentation masks.
- **Test Image:** The test set contains 379 images. These images are kept separate from the training set and are used to evaluate the performance of the segmentation model after it has been trained. The model has not seen these images during training and must generalize its segmentation capability to unseen data.
- **Test Mask:** Similar to the training set, the test set also includes segmented masks corresponding to the test images. These masks are used during evaluation to compare the model's predicted segmentation masks with the ground truth masks.

In our custom dataset creation, we tailor input and output images to dimensions of 128x128 pixels. With a batch size of 32 for both training and testing.

Model Architecture

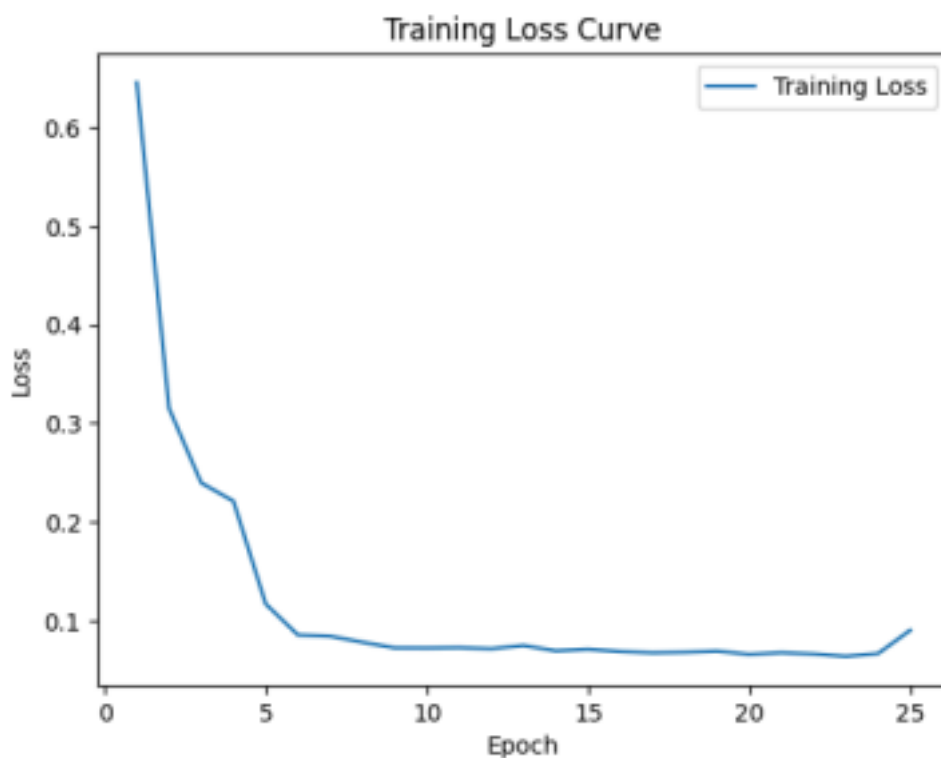
Encoder: Our segmentation model employs a pre-trained MobileNet as the encoder, originally trained on the ImageNet dataset for classification tasks. Despite MobileNet's classification-oriented design, we repurpose it for segmentation by utilizing it as a feature extractor. The MobileNet architecture comprises a total of 18 layers, and to adapt it for segmentation, we remove the top four layers, including the classification layer. By discarding these layers, we focus on extracting features relevant to segmentation tasks, prioritizing lower-level details and spatial information essential for accurate segmentation. The initial layers in MobileNet are adept at capturing fundamental visual features such as edges and textures,

which are crucial for delineating object boundaries in segmentation. By leveraging MobileNet as the encoder, we exploit its pre-trained weights and the knowledge it gained from the extensive ImageNet dataset, enabling our model to learn rich and meaningful representations of input images.

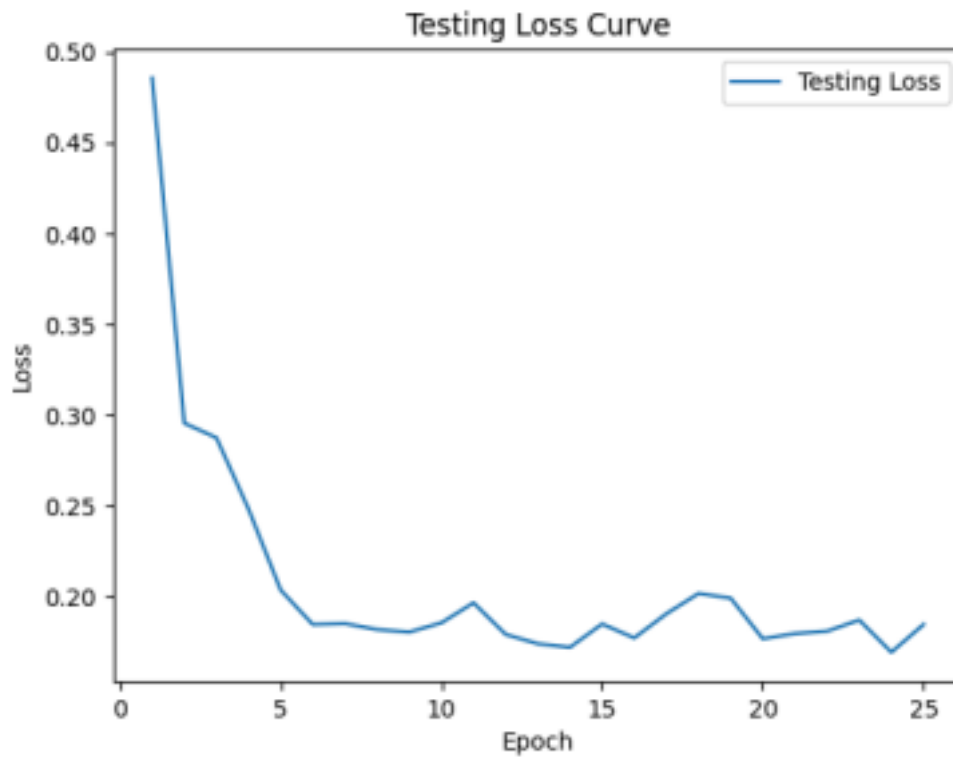
Decoder:In the decoder of our segmentation model, we employ a combination of 2D convolution and 2D transposed convolution layers for upsampling. ReLU activation functions are applied throughout the decoder to introduce non-linearity and enhance feature extraction. The output of the decoder is a grayscale image, serving as the segmentation mask. This mask delineates the boundaries and regions of objects within the input image, providing a clear visual representation of the segmented areas. By utilizing these techniques, our decoder effectively reconstructs detailed segmentation masks, enabling precise identification and localization of objects in the input images.

First_model: Freezing encoder weight

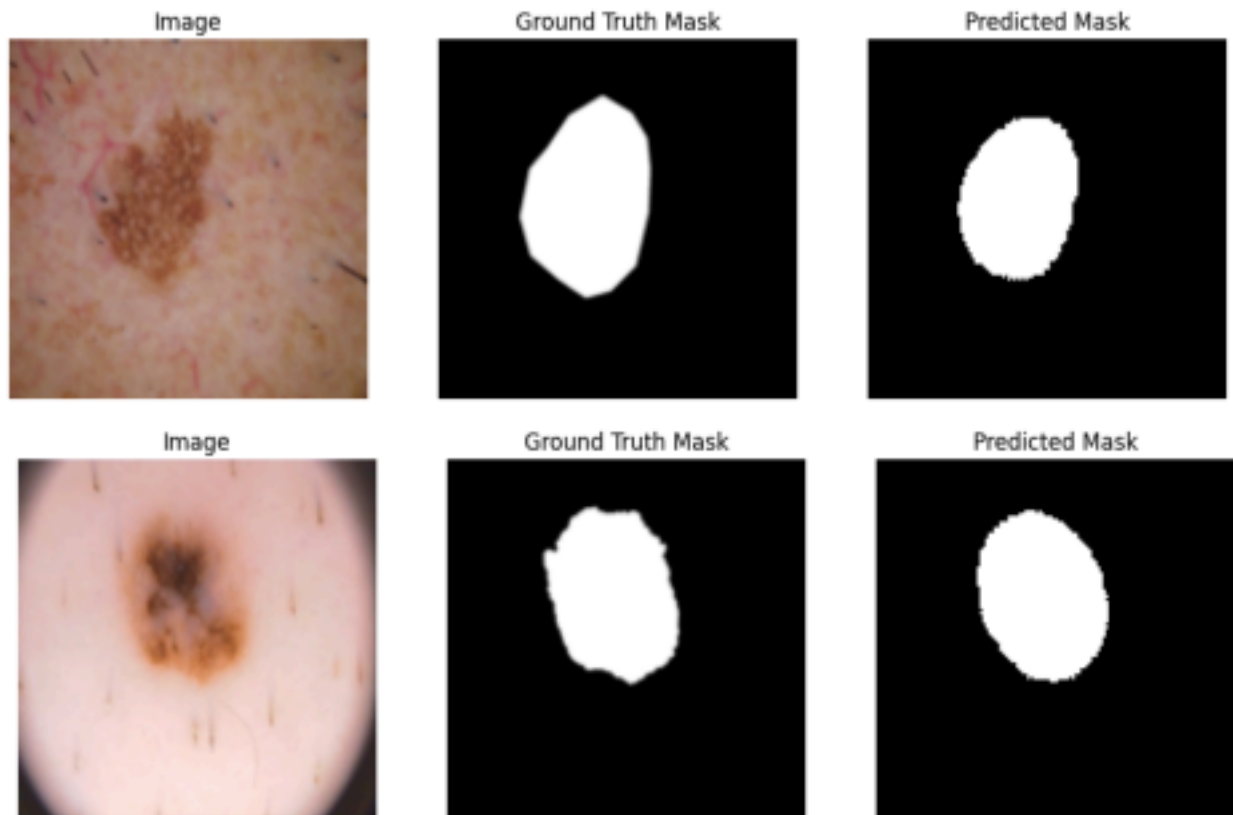
In our initial model configuration, binary cross-entropy is the loss function optimized by the Adam optimizer with a learning rate of 0.001 over 25 epochs. With a final training loss of 0.09, our model demonstrates rapid convergence, owing to Adam's adaptive learning rate mechanism and momentum optimization.



During the evaluation of the test data, we attain a mean Dice score of 0.8786, reflecting the accuracy of segmentation by measuring the overlap between predicted and ground truth masks. Additionally, achieving a mean Intersection over Union (IoU) score of 0.7974, further validates the effectiveness of our model in accurately delineating object boundaries. The average test loss of 0.1846 indicates the model's ability to generalize well to unseen data while maintaining low error. These metrics collectively highlight the robustness and efficacy of our model in performing accurate segmentation tasks, showcasing its potential for various applications in image analysis and computer vision tasks.



Visualizing test dataset examples with input images, corresponding ground truth masks, and predicted masks showcases the segmentation model's performance.



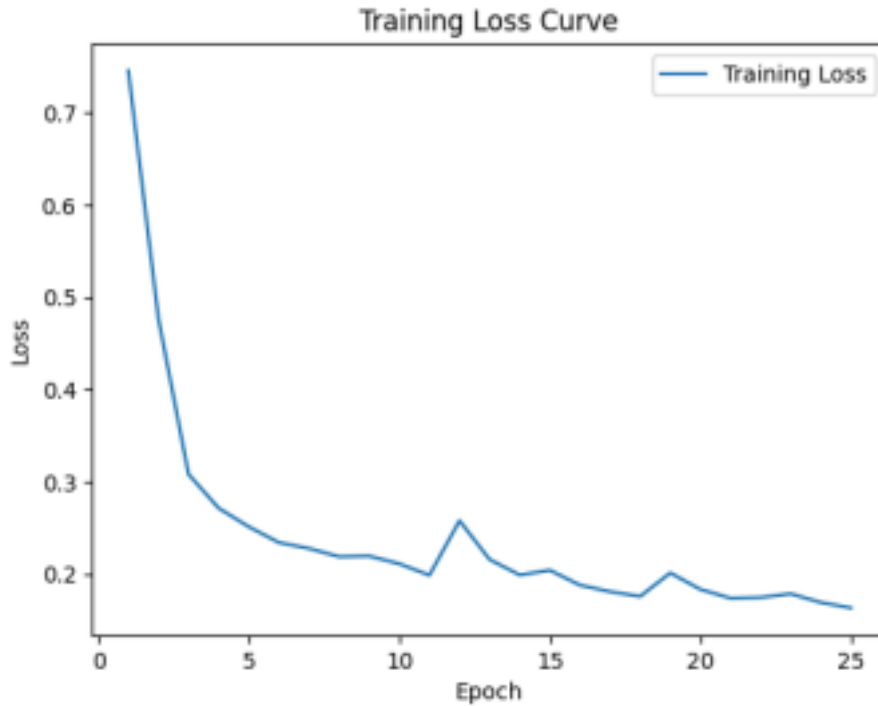


\

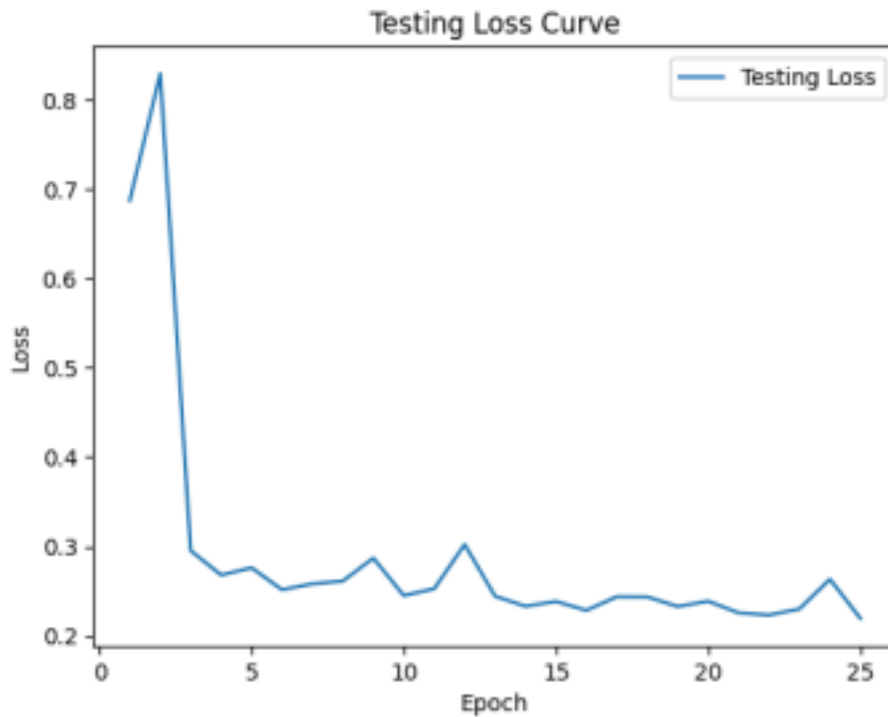


Second_model: Fine tuning encoder weight

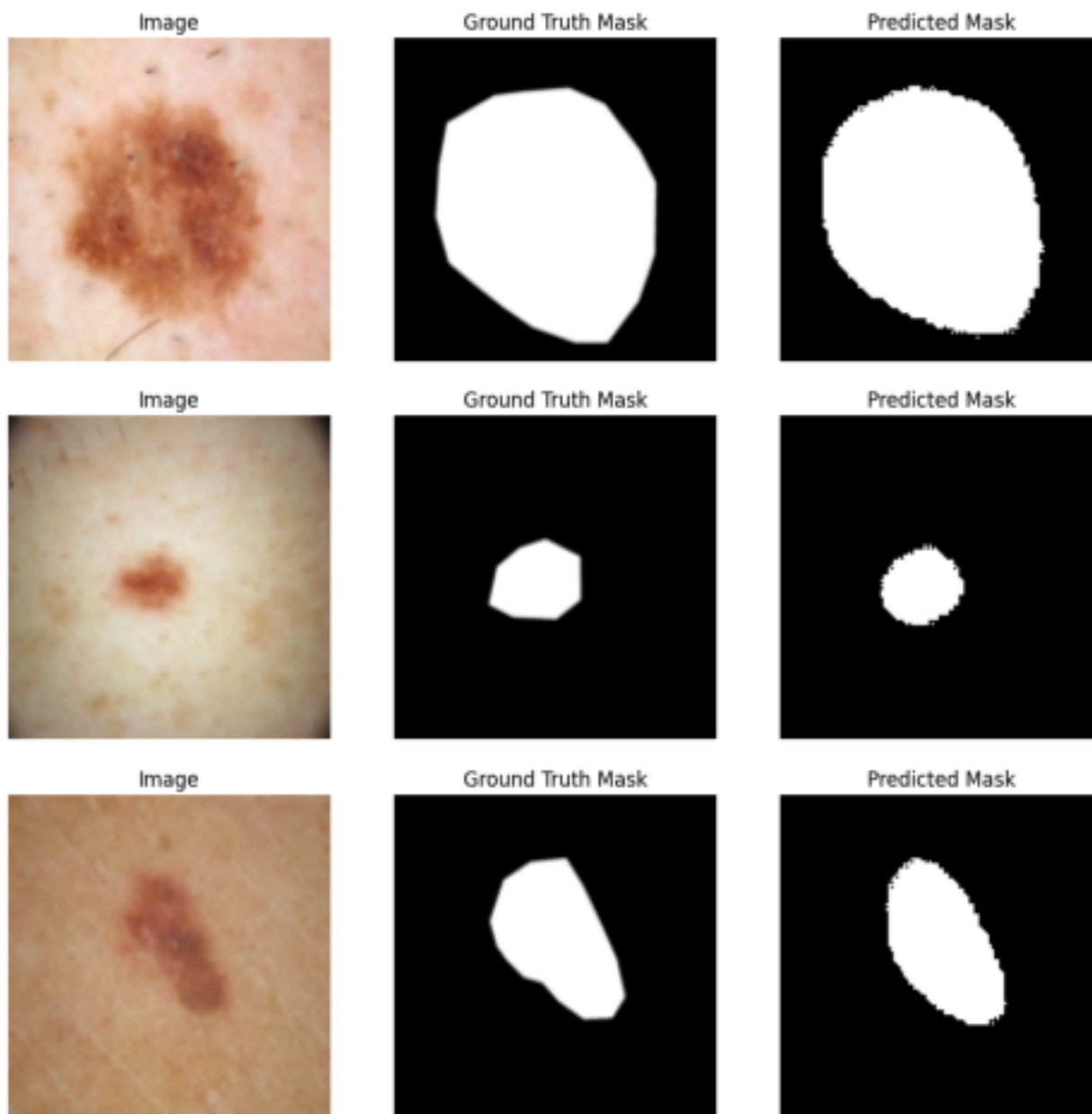
In this model, we fine-tune the encoder-decoder architecture by jointly training the decoder along with the encoder's weights. Employing binary cross-entropy loss and Adam optimizer with a learning rate of 0.001, we refine the model's understanding of data. In the final epoch, we achieve a loss of 0.16, indicating effective convergence. This fine-tuning process enhances the model's ability to capture intricate features and nuances in the data, leading to improved segmentation performance.

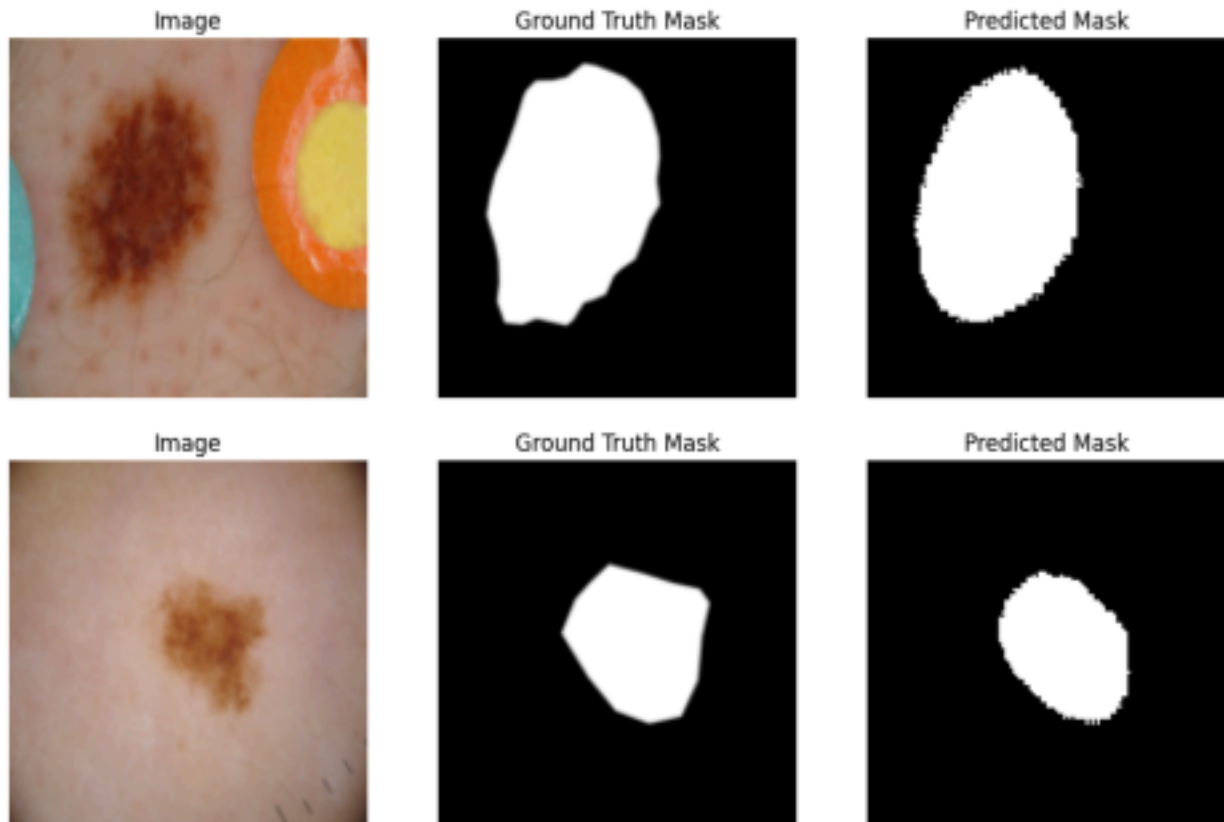


Post-training evaluation reveals promising segmentation performance metrics: a mean Dice score of 0.88 and mean Intersection over Union (IoU) score of 0.8 on the test dataset. The average loss across 25 epochs for testing stands at 0.2195. These metrics signify robust segmentation accuracy and model generalization, demonstrating the efficacy of the trained model in accurately delineating object boundaries and capturing semantic information within the test images.



Visualizing test dataset examples with input images, corresponding ground truth masks, and predicted masks showcases the segmentation model's performance.





Comparison of both model:

In the two models described, we explore different approaches to utilizing the encoder in a segmentation task: freezing its weights in the first model and fine-tuning them along with the decoder in the second model. These approaches leverage the encoder's ability to extract meaningful features from the input images, which is crucial for accurate segmentation. Let us delve deeper into the rationale behind each approach and analyze their performance based on metrics such as Intersection over Union (IoU) and Dice score.

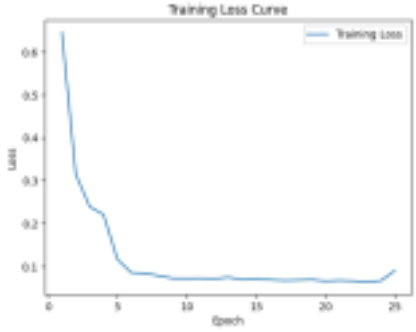



In the first model, we freeze the weights of the encoder. Freezing the encoder's weights means that during training, the parameters of the encoder remain fixed and unchanged. This approach is often employed when using pre-trained models, such as MobileNet, to prevent the model from forgetting the valuable representations learned during pre-training. By keeping the encoder frozen, we ensure that the features extracted by the encoder remain consistent and retain the knowledge gained from its prior training on the ImageNet dataset. This approach helps maintain stability during training and prevents overfitting, especially when dealing with limited data, like in medical imaging tasks.

However, while freezing the encoder's weights provides stability and prevents overfitting, it may limit the model's ability to adapt to the specific characteristics of the segmentation task. The features learned by the encoder from the ImageNet dataset might not be optimal for the segmentation task, as the visual patterns and semantic meanings in medical images, such as those in the ISIC dataset, can differ significantly from those in natural images. As a result, the model's performance may not reach its full potential, and it may need help to capture the intricacies of the data specific to the segmentation task.

To address this limitation, we employ fine-tuning in the second model, where we allow the weights of the

encoder to be updated along with those of the decoder during training. Fine-tuning enables the model to adapt its learned representations to suit the segmentation task's requirements better. By fine-tuning the encoder, the model can adjust its feature extraction process to focus on the relevant semantic features in the ISIC dataset, thus improving its ability to capture the nuances and details necessary for accurate segmentation.

The fine-tuning process allows the model to learn more task-specific features from the ISIC dataset while benefiting from the general features learned during pre-training on ImageNet. This adaptability leads to a more refined feature representation better aligned with the segmentation task's requirements. Consequently, the model becomes more adept at capturing the image data's semantic meaning and spatial relationships, resulting in improved segmentation performance.

	First_Model (freezing weight)	Second_model (fine tuning)
Train loss (last epoch)	0.09	0.16
		
Test loss (last epoch)	0.18	0.21
		
Avg Test loss	0.18	0.21
Mean Dice score	0.87	0.88
Mean Iou score	0.79	0.80

The table shows that while there's a noticeable difference in the performance of the two models, it's not very significant. This difference can be explained by the encoder's strengths when it's trained on a large dataset like ImageNet. Pre-training on ImageNet gives the encoder a good understanding of basic visual

elements like edges, textures, and shapes, which are crucial for tasks like segmentation. By removing certain layers from the encoder, we help it focus more on these basic features rather than complex ones. This ensures that the encoder captures the important visual details needed for segmentation.

When we freeze the encoder's weights during training, the model can use these pre-learned features without the risk of losing them or overfitting. This approach lets the model benefit from the general features learned from ImageNet while still adapting to the specific segmentation task. As a result, the model with the frozen encoder performs well, although slightly less so than the fine-tuned one. Still, the frozen encoder model effectively uses the basic features learned during pre-training, showing its effectiveness in segmentation tasks.