Deep Learning

Assignment 3

Sireejaa Uppal
(M23CSE023)

Introduction

The aim of this report is to compare two MobileNetV2-based image segmentation models trained on the ISIC 2016 dataset. Both models utilize similar architectures but are trained on datasets and with different configurations. The performance metrics including loss, accuracy, Intersection over Union (IoU), and Dice score are evaluated on both training and validation sets. Both the segmentation models use a MobileNet pre-trained on the ImageNet dataset as an encoder and I have designed a decoder that predicts segmented masks. In this report, we evaluate two implementations of MobileNetV2 decoders for skin lesion segmentation using the ISIC 2016 dataset. Both models were trained for 25 epochs and evaluated on a separate test set.

Network Architectures

- Both models utilize MobileNetV2 as the encoder and consist of a decoder followed by upsampling layers.
- The first model decoder's parameters are set to be trainable, whereas the second model's decoder parameters are frozen.
- The ISIC 2016 dataset was divided into training and testing sets, with corresponding masks provided for segmentation.
- Augmentation techniques were applied during training, including random rotation, horizontal flipping, color jitter, and random resized cropping.
- Both models were trained using the Adam optimizer with a learning rate of 0.001 and optimized using the binary cross-entropy loss function.
- Training and validation metrics such as loss, accuracy, Intersection over Union (IoU), and Dice score were monitored across epochs.

Model 1 (Trainable Decoder):

- Achieved a final training loss of 0.2896, with a training accuracy of 0.8722.
- Final validation loss was 0.2971, with a validation accuracy of 0.8696.
- Mean Intersection over Union (IoU) and Dice scores were 0.6954 and 0.7370, respectively.

Model 2 (Frozen Decoder):

- Converged to a slightly higher training loss of 0.3076 and a lower training accuracy of 0.8638.
- However, it yielded a slightly lower validation loss of 0.3347 and a validation accuracy of 0.8539.
- The mean Intersection over Union (IoU) and Dice scores were 0.6823 and 0.7176, respectively.

Both models show a similar trend in terms of loss and accuracy throughout training epochs. The model with a trainable decoder, exhibits slightly better performance metrics in terms of IoU and Dice scores on both training and validation sets. The frozen decoder in Model 2 might have limited the model's capacity to learn intricate features, resulting in slightly inferior segmentation performance compared to other. Overall, Model appears to be slightly superior in terms of segmentation accuracy, likely due to the flexibility of updating decoder parameters during training.

Results

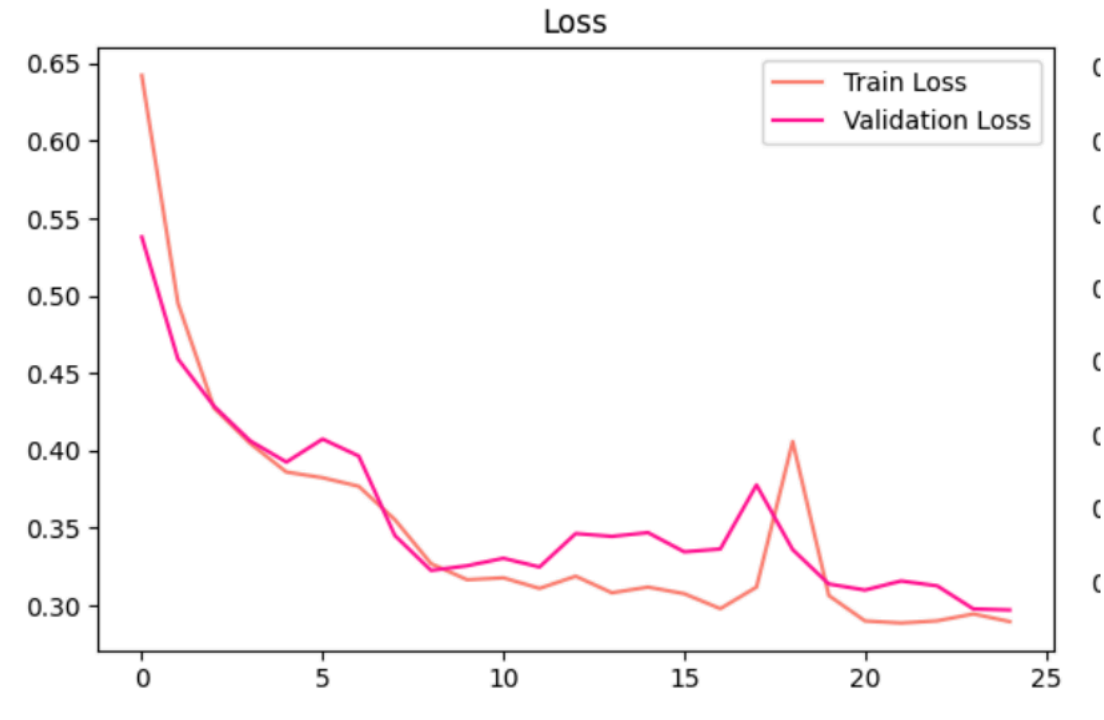
1. Architecture 1 :

a. Output

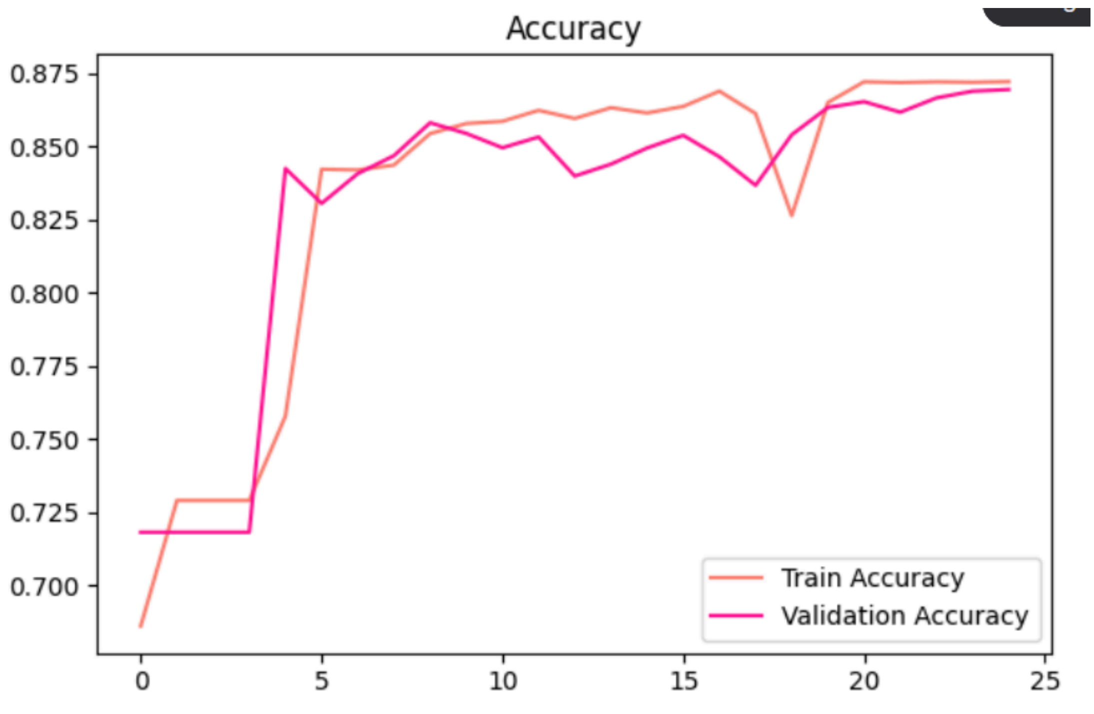
```
cuda
/opt/conda/lib/python3.10/site-packages/torchvision/models/_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be
removed in the future, please use 'weights' instead.
  warnings.warn(
/opt/conda/lib/python3.10/site-packages/torchvision/models/_utils.py:223: UserWarning: Arguments other than a weight enum or `None` for 'weights' are
deprecated since 0.13 and may be removed in the future. The current behavior is equivalent to passing `weights=MobileNet_V2_Weights.IMAGENET1K_V1`. Yo
u can also use `weights=MobileNet_V2_Weights.DEFAULT` to get the most up-to-date weights.
  warnings.warn(msg)
Downloading: "https://download.pytorch.org/models/mobilenet_v2-b0353104.pth" to /root/.cache/torch/hub/checkpoints/mobilenet_v2-b0353104.pth
100%|██████████| 13.6M/13.6M [00:00<00:00, 66.1MB/s]
Epoch 1/25, Train Loss: 0.6423, Train Acc: 0.6860, Val Loss: 0.5380, Val Acc: 0.7181, Train IoU: 0.4204, Val IoU: 0.4705, Train Dice: 0.4296, Val Dic
e: 0.4705
Epoch 2/25, Train Loss: 0.4953, Train Acc: 0.7290, Val Loss: 0.4592, Val Acc: 0.7181, Train IoU: 0.4796, Val IoU: 0.4705, Train Dice: 0.4796, Val Dic
e: 0.4705
Epoch 3/25, Train Loss: 0.4275, Train Acc: 0.7290, Val Loss: 0.4287, Val Acc: 0.7181, Train IoU: 0.4796, Val IoU: 0.4705, Train Dice: 0.4796, Val Dic
e: 0.4705
Epoch 4/25, Train Loss: 0.4046, Train Acc: 0.7290, Val Loss: 0.4063, Val Acc: 0.7181, Train IoU: 0.4796, Val IoU: 0.4705, Train Dice: 0.4796, Val Dic
e: 0.4705
Epoch 5/25, Train Loss: 0.3862, Train Acc: 0.7578, Val Loss: 0.3926, Val Acc: 0.8425, Train IoU: 0.5169, Val IoU: 0.6523, Train Dice: 0.5325, Val Dic
e: 0.7003
Epoch 6/25, Train Loss: 0.3825, Train Acc: 0.8423, Val Loss: 0.4075, Val Acc: 0.8306, Train IoU: 0.6394, Val IoU: 0.6393, Train Dice: 0.6823, Val Dic

Epoch 23/25, Train Loss: 0.2901, Train Acc: 0.8722, Val Loss: 0.3127, Val Acc: 0.8667, Train IoU: 0.6929, Val IoU: 0.6873, Train Dice: 0.7355, Val Dic
e: 0.7337
Epoch 24/25, Train Loss: 0.2946, Train Acc: 0.8720, Val Loss: 0.2977, Val Acc: 0.8690, Train IoU: 0.6931, Val IoU: 0.6970, Train Dice: 0.7367, Val Dic
e: 0.7377
Epoch 25/25, Train Loss: 0.2896, Train Acc: 0.8722, Val Loss: 0.2971, Val Acc: 0.8696, Train IoU: 0.6954, Val IoU: 0.6934, Train Dice: 0.7372, Val Dic
e: 0.7370
```

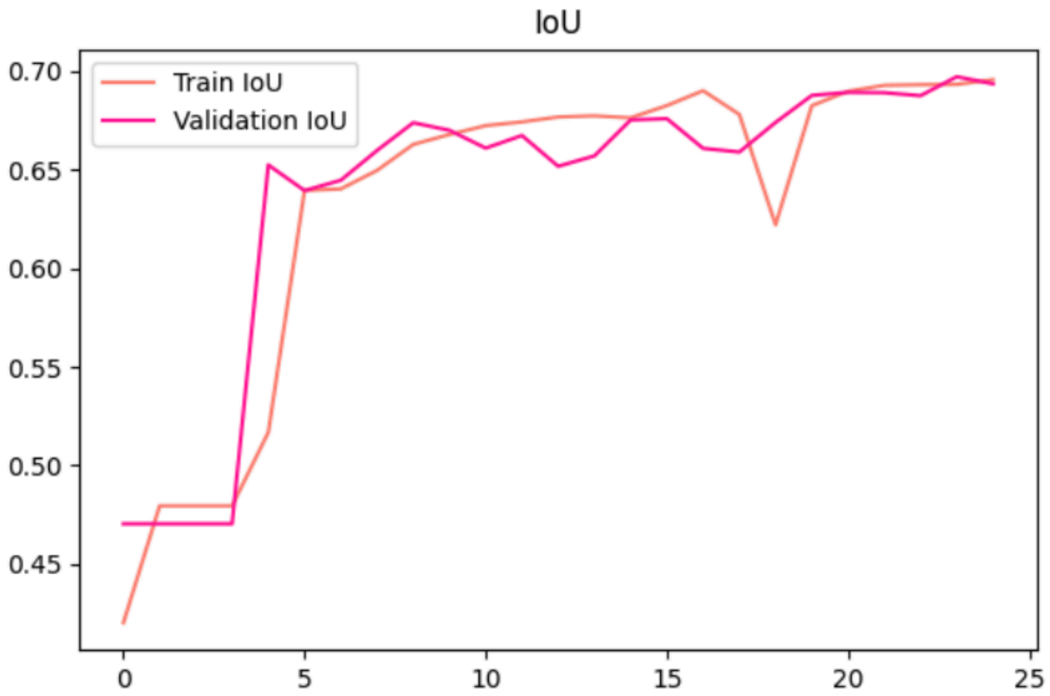
b. Loss in all 25 Epochs



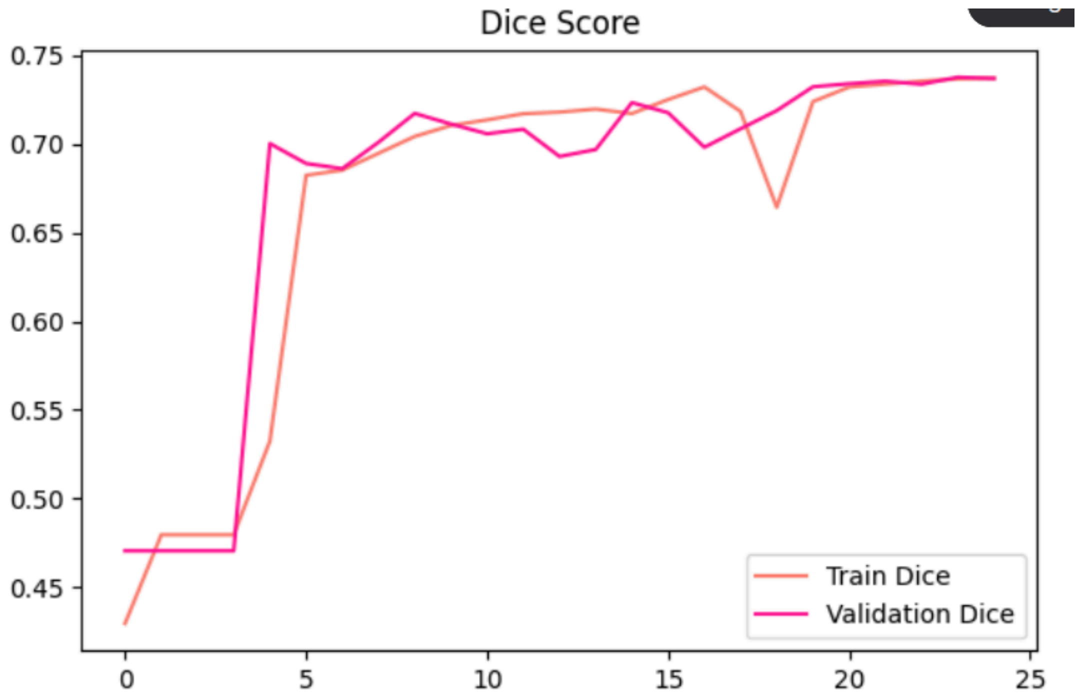
c. Accuracy in Training and Validation



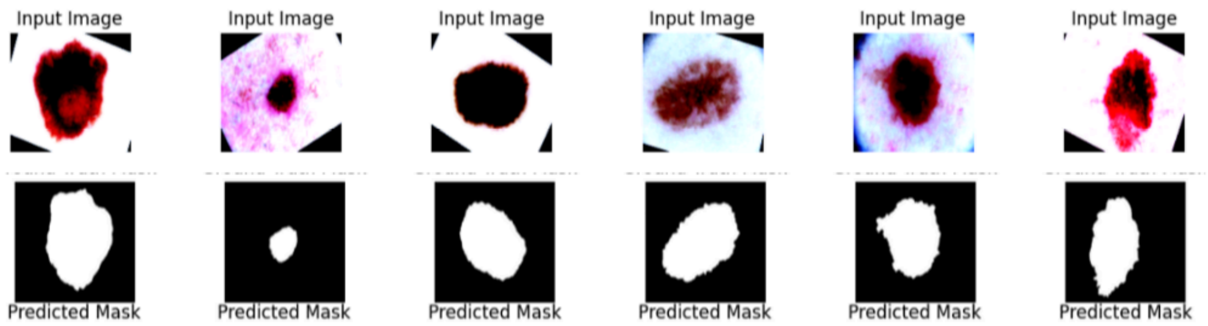
d. IoU trend



e. Dice Score trend in all epochs



f. Final output images for a few samples



2.Architecture 2:

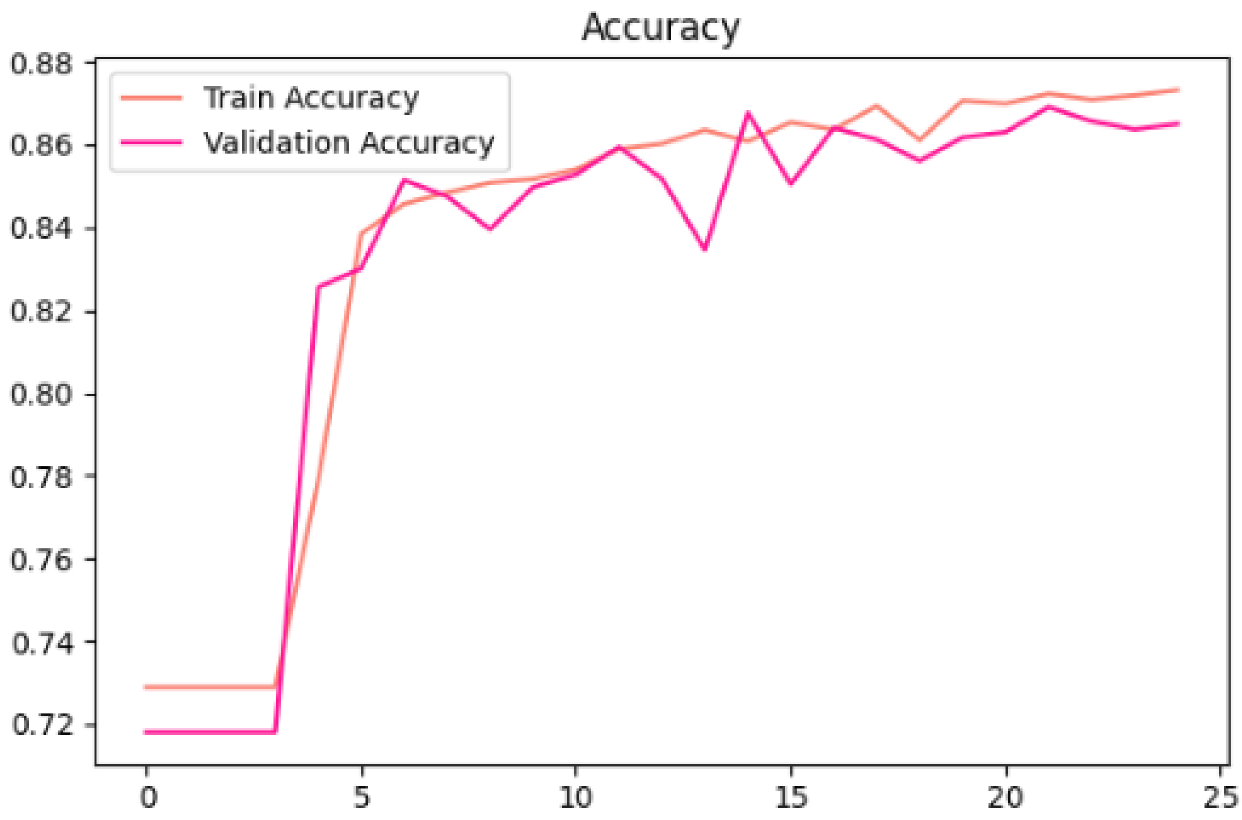
a. Output

```
  warnings.warn(msg)
Downloading: "https://download.pytorch.org/models/mobilenet_v2-b0353104.pth" to /root/.cache/tor
ch/hub/checkpoints/mobilenet_v2-b0353104.pth
100%|██████████| 13.6M/13.6M [00:00<00:00, 66.9MB/s]
Epoch 1/25, Train Loss: 0.6244, Train Acc: 0.7290, Val Loss: 0.5358, Val Acc: 0.7181, Train IoU:
0.4794, Val IoU: 0.4705, Train Dice: 0.4794, Val Dice: 0.4705
Epoch 2/25, Train Loss: 0.4786, Train Acc: 0.7290, Val Loss: 0.4446, Val Acc: 0.7181, Train IoU:
0.4796, Val IoU: 0.4705, Train Dice: 0.4796, Val Dice: 0.4705
Epoch 3/25, Train Loss: 0.4397, Train Acc: 0.7290, Val Loss: 0.4174, Val Acc: 0.7181, Train IoU:
0.4796, Val IoU: 0.4705, Train Dice: 0.4796, Val Dice: 0.4705
```

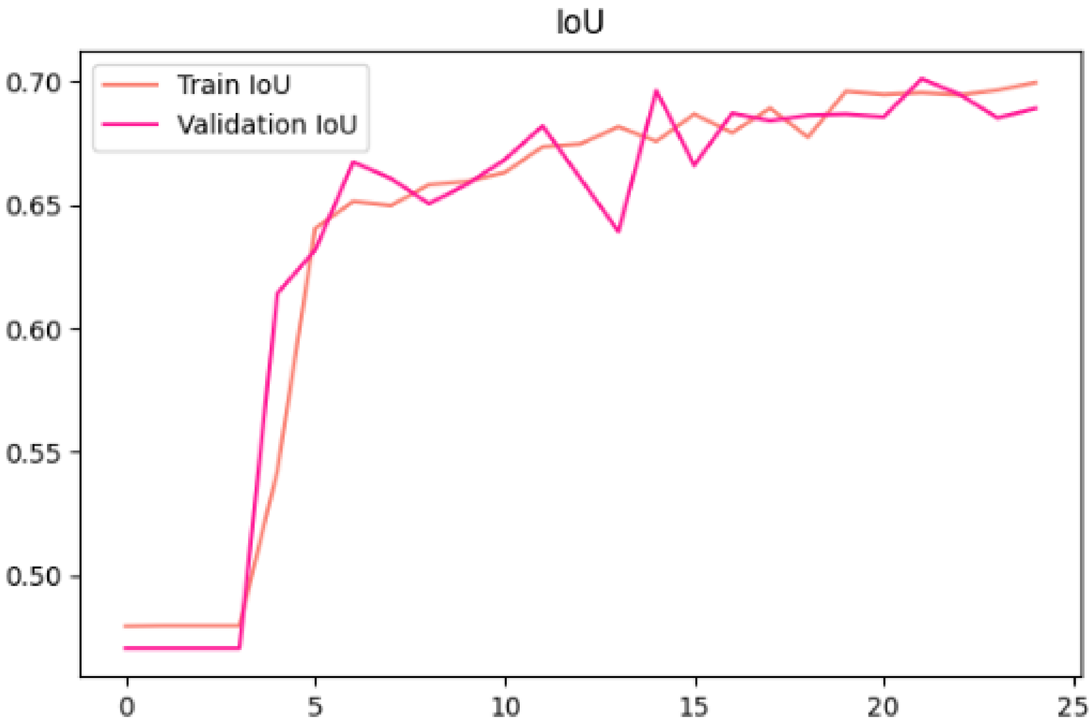
```
U: 0.6955, Val IoU: 0.7012, Train Dice: 0.7379, Val Dice: 0.7451
Epoch 23/25, Train Loss: 0.2935, Train Acc: 0.8708, Val Loss: 0.3108, Val Acc: 0.8656, Train Io
U: 0.6946, Val IoU: 0.6948, Train Dice: 0.7362, Val Dice: 0.7378
Epoch 24/25, Train Loss: 0.2872, Train Acc: 0.8718, Val Loss: 0.3161, Val Acc: 0.8637, Train Io
U: 0.6966, Val IoU: 0.6853, Train Dice: 0.7389, Val Dice: 0.7252
Epoch 25/25, Train Loss: 0.2860, Train Acc: 0.8731, Val Loss: 0.3101, Val Acc: 0.8649, Train Io
U: 0.6997, Val IoU: 0.6891, Train Dice: 0.7407, Val Dice: 0.7335
```

b. Loss in all 25 Epochs

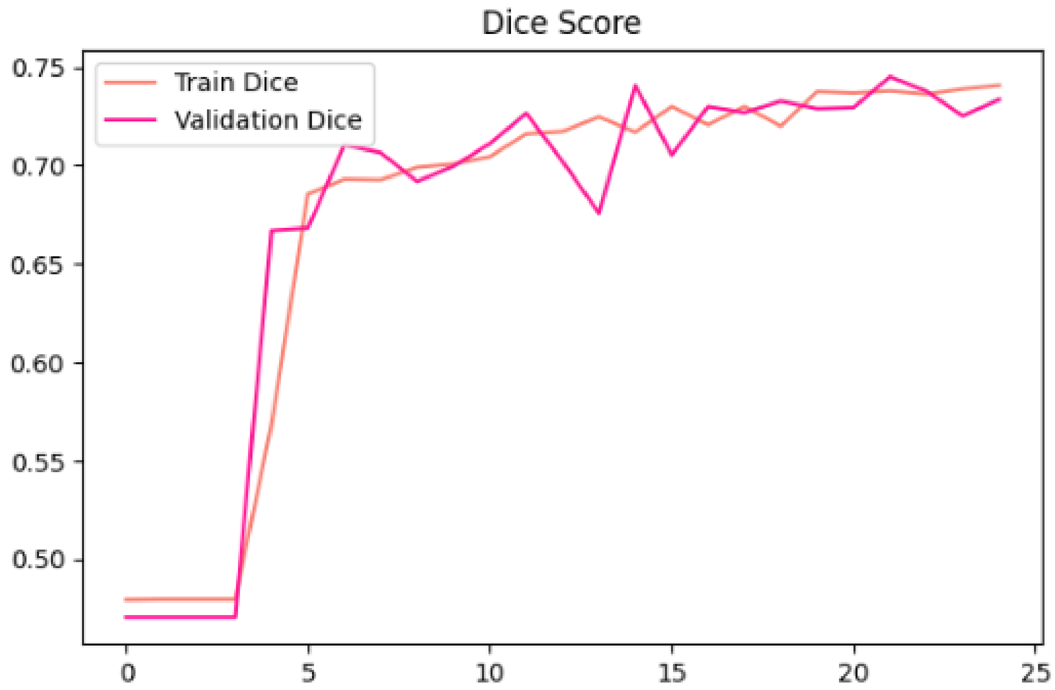
c. Accuracy in Training and Validation



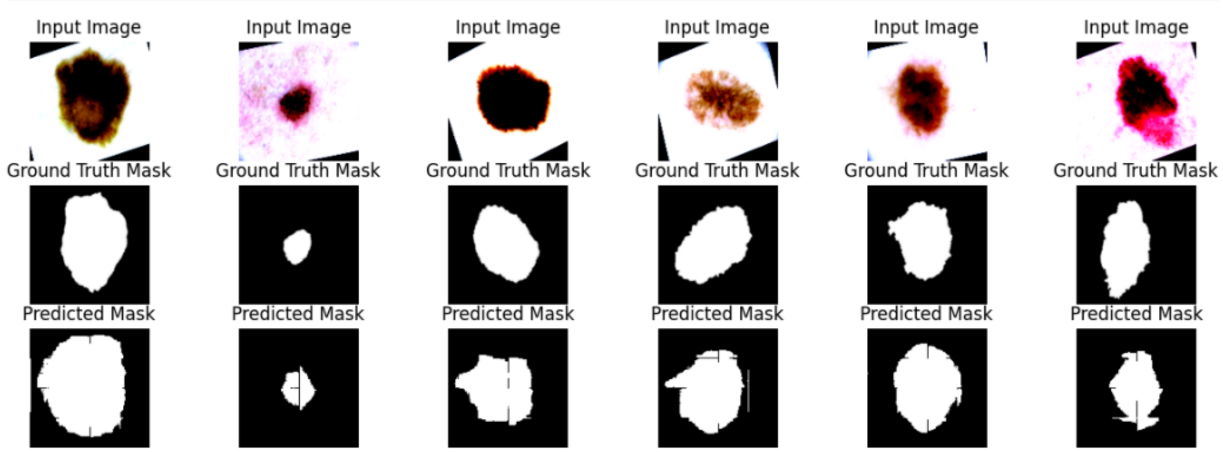
d. IoU trend



e. Dice Score trend in all epochs



f. Final output images for a few samples



Comparison and Conclusion

Model 2, which fine-tunes the entire architecture, performs better due to its ability to adapt the encoder to the specific segmentation task. By updating the encoder weights, the model can learn task-specific features that may not be adequately captured by the pre-trained weights from ImageNetV1. This adaptation enables the model to better capture fine details and nuances in the segmentation task's images, leading to more accurate masks. Additionally, fine-tuning allows for better generalization and optimization of model parameters, potentially resulting in improved performance across diverse datasets. Despite requiring more computational resources and training time, the benefits in accuracy and generalization justify this approach, especially for tasks where precise segmentation is crucial.

- In skin lesion segmentation tasks using the ISIC 2016 dataset, employing a MobileNetV2 decoder with trainable parameters yields slightly better segmentation performance compared to a frozen decoder.
- However, the differences in performance between the two models are marginal, suggesting that the choice between trainable and frozen decoders may not significantly impact segmentation results.

Report and Kaggle Links:

M23CSE023_DL_Assignment 3

https://www.kaggle.com/code/myrakapoor/m23cse023-two

https://www.kaggle.com/code/siree16/m23cse023-one