INDIAN INSTITUTE OF TECHNOLOGY JODHPUR

Deep Learning

CSL7590

Assignment 3

Submitted By:

Aman Kanshotia
M21MA201

Instructor:

Deepak Mishra

March 19,2024

Contents

List of Figures

1 Introduction about Dataset

The ISIC (International Skin Imaging Collaboration) 2016 dataset is a comprehensive collection of dermoscopic images that has aim for advancing research in the field of dermatology and computer vision. Dermoscopic imaging offers a detailed view of the skin, aiding in the diagnosis of various skin conditions, including melanoma and other forms of skin cancer.

This dataset, released as part of the ISIC Challenge 2016, comprises thousands of high-resolution dermoscopic images, each annotated with clinical metadata and ground truth labels provided by expert dermatologists. The dataset encompasses a wide range of skin lesions, including benign and malignant melanocytic lesions, as well as other non-melanocytic lesions, providing a rich resource for training and evaluating machine learning algorithms and deep learning models.

The ISIC 2016 dataset serves as a benchmark for the development and validation of automated systems for skin lesion classification, with the ultimate goal of improving early detection and diagnosis of skin cancer. Its availability has facilitated numerous studies and competitions, fostering collaboration among researchers and healthcare professionals worldwide to address challenges in dermatologic imaging and diagnosis.

The ISIC dataset includes:

- train: 900 training images
- train masks: Segmented masks for training images
- test: 379 test images
- test masks: Segmented masks for test images

1.1 Data Visualization

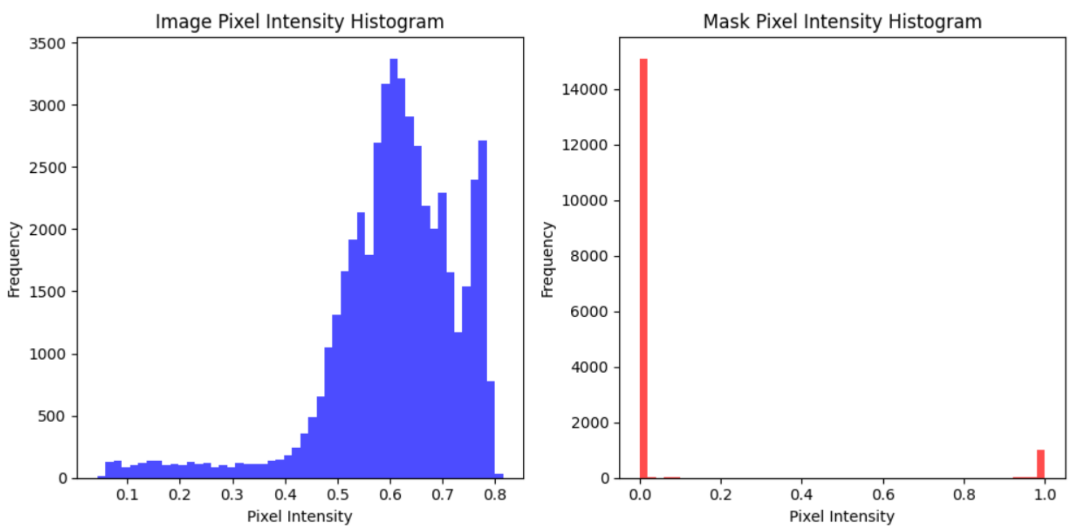

Figure 1: Pixel Intensity for Image and Mask

Image Pixel Intensity Histogram (Blue):

- The first histogram shows the frequency of different pixel intensities in an image.
- It has a bimodal distribution with peaks around 0.3 and 0.6 pixel intensity.

- The x-axis represents "Pixel Intensity" in the range from 0.0 to 0.8, and the y-axis represents "Frequency" in the range of 0 to 3500.

Mask Pixel Intensity Histogram (Red):

- The mask histogram is simple, with two recognizable red bars at the extremes on pixel intensity scale.
- It has one bar is at nearly 0 intensity, while the other one is at exactly 1 intensity.
- From here we can say the mask consists mainly of pixels that are either fully transparent or fully Dense.
- Axes are same as image histogram.

Sample Image and their corresponding masks

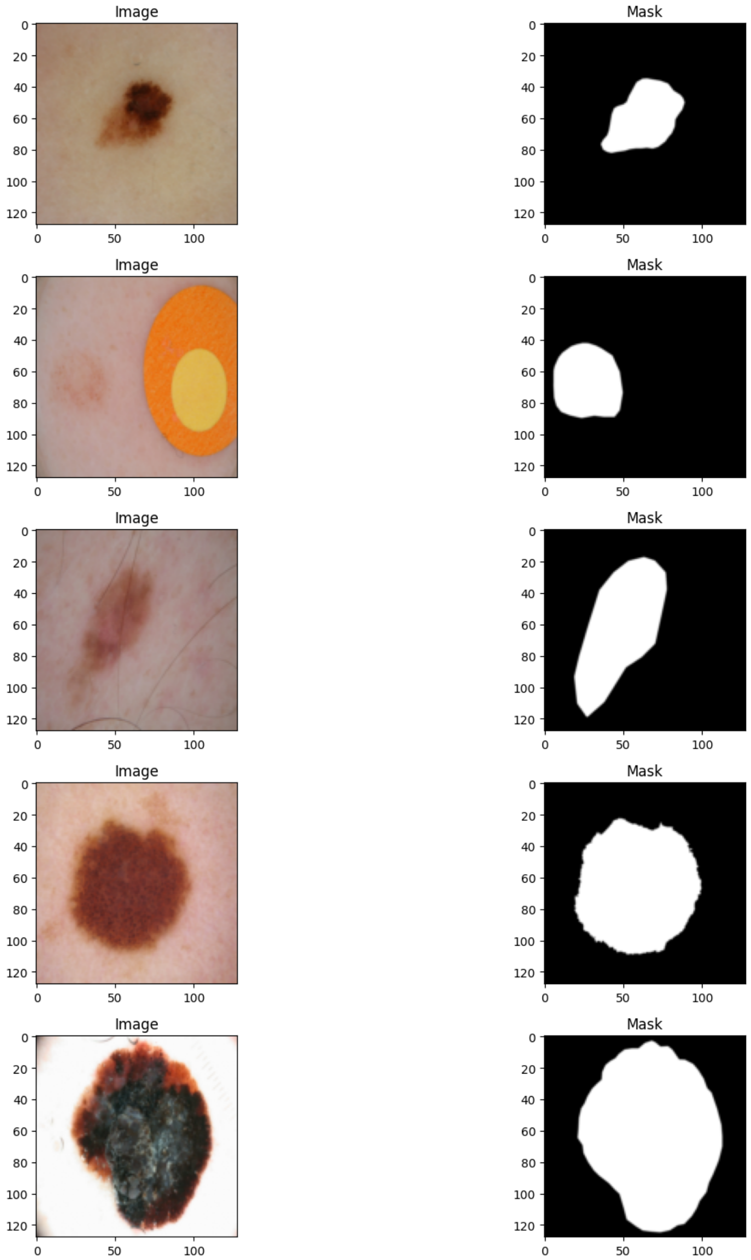

Figure 2: Sample Images and Masks

The above figure shows that the dermoscopic images and their masks with a gray-scale image.

2 Data Pre-Processing

We have ISIC2016 dataset where we have only images and their corresponding masks.So, we have to do preprocess our dataset to use it useful for our task and we can use for training and testing our model.In the series of preprocessing the dataset we perform the following methods.

- **Resizing:** we have resized the images and masks to a fixed size of pixels 128x128. This ensures that all the images and masks have same dimensions,which is often a necessary task for training our model.
- **Convert to Tensor:**The Resized images and masks are further converted to tensors to make the data compatible with our PyTorch framework.

We applied these preprocessing steps to both the training and testing datasets using transformations that we defined in the dataset initialization. Our main aim to do these preprocessing steps is to prepare the data in a suitable format.

3 Model Architecture

In this assignment i use the Convolutional neural network to do the task for segmentation. The Following layers i used in my model architecture:

- **ConvTranspose2d Layers:** I used these layers for up-sampling or transpose convolution because these layers increase the spatial resolution of the feature maps, allowing the network to generate a larger output. I have taken 4 layers with different kernelsizes, strides, and padding. The 4th Convo layer produces the final output that captures the spatial information.
- **Dropout Layers:**This layer is a regularization technique that is used to prevent the model to be overfit. It randomly sets a fraction of input units to zero during training, which helps in preventing the network from relying too much on specific features and improves generalization.We have taken 0.5 probability in it.these layers are applied after each convolutional layer in my model.
- **Interpolation:** After applying the final convolution layer this interpolation is applied for resizing the output to the desired size 128x128 pixels. This is a necessary step in the decoder because the convolutional layers may change the spatial dimensions, and this interpolation gives us a satisfaction that the output matches the expected size for further processing or comparison with ground truth masks that means without it we will got dimension errors.

4 Model 1

In the first model we used a pretrained model named Mobilenet that was trained on ImageNet dataset.Here we freezed the features that were gotten while training this pretrained model and in this model we used them as they were. For calculating the loss we have use the Binary Cross-Entropy Loss (BCEwithlogitsloss) and Nestrove Adam optimizer with 0.001 learning rate to optimize the parameters and trained the model on Gpu.Also we have calculated the following scores for our model:

- **IOU Score** It measures the overlapping between the predicted segmentation mask and the ground truth mask. It's calculated as the ratio of the intersection of the predicted and ground truth masks to their union. It is between the range of 0 to 1. This score is sensitive to both the false positives and false negatives in the segmentation task.
- **Dice Score** It is an another matric used for quantifing the similarity between the true masks and predicted masks. we calculate it as the ratio of twice the intersection of the predicted and true masks to the sum of their areas. It also range between 0 to 1.

4.1 Results

We have trained this model for 25 epochs and calculate the losses for training and testing. we also calculate the iou score and dice score for test dataset.After training of the model we visualize some of the images with their both masks.The Following results are after 25 epochs.

- **Training and testing Losses**
  Training Loss : 0.4830
  Testing Loss : 0.3972
- **Loss Curve**

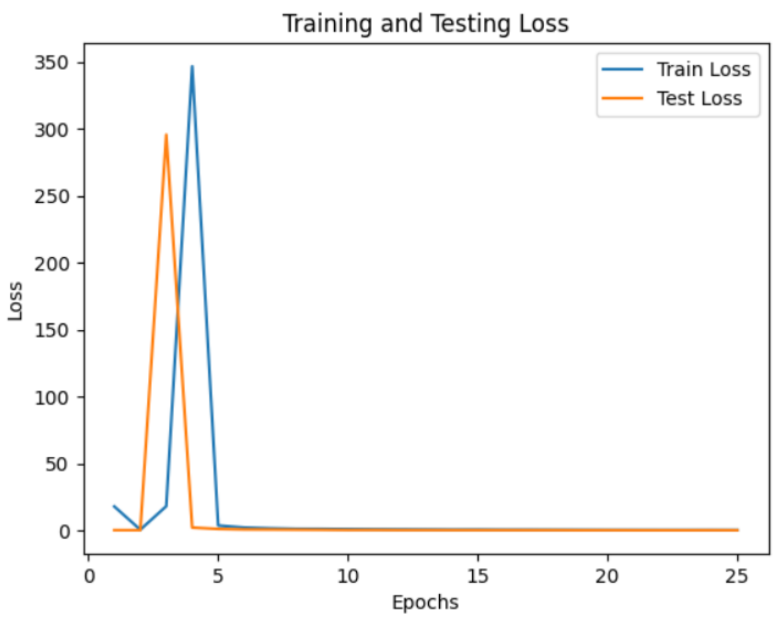

Figure 3: Training and Testing loss per Epochs

- **IOU and Dice Scores:**
  Test Iou score : 0.2985
  Test Dice Score : 0.7522

- **Good segmentations**

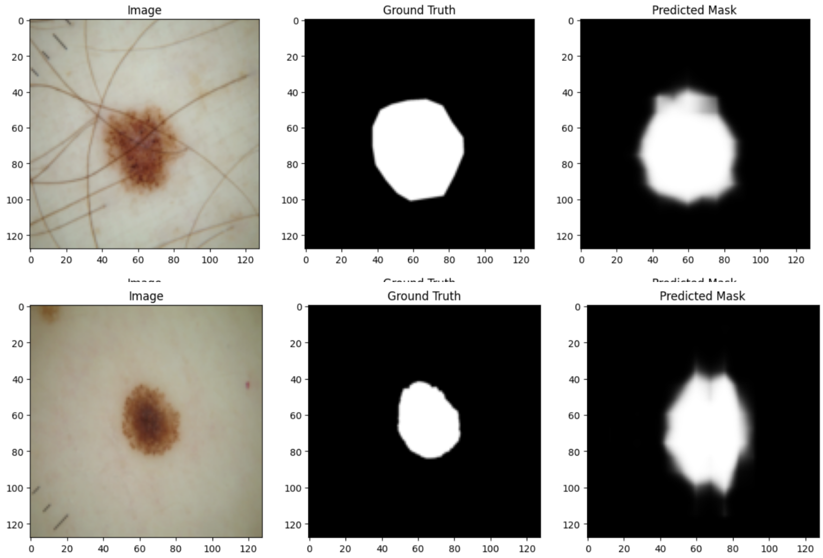

Figure 4: Good Results on test data

- **Bad Segmentations**

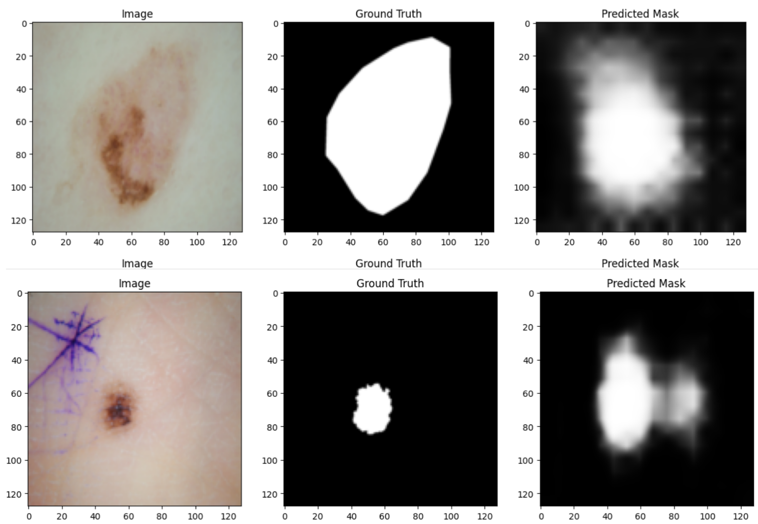

Figure 5: Bad Results on test data

4.2 Observations

In this model we have used the freezed weights from the pretrained model as they were means we don't do back propagation through the encoder layers. We freezed the weights and use them for our task.We observed the following things in this model:

- **Train Loss:** The Train Loss at the initial stage of training we have a high loss and at 4th epoch we have the maximum loss in the training of this model but after 4th epochs we have seen that the lossis decreasing over time, which indicates that the model is learning from the training data but some misleads are their.We can say from here that the to train this model is a tricky things because the things that i observed are, may be different with different architectures.

- **Test Loss:** Here also we have seen that the Loss values are fluctuating till 5th epochs but generally show a decreasing trend across epochs. This suggests that the model is also improving on the test set, although the fluctuations could indicate variability in the test data or model's performance.

- **Test IoU:** The Test IoU (Intersection over Union) is fluctuating as the loss fluctuates but it remains constant at 0.2985 for all epochs after 5th. This is unusual as we would expect some variation and improvement over time. It might indicate an issue with the testing process.

- **Test Dice:** The Test Dice coefficient is increasing across all the epochs.But a little bit fluctuations are there but over all it is nearly 0.7 score.

Therefore, we can say that our model predicts some anomalies because of frozen features.Now lets see what will happen if we don't freeze the features.

5 Model 2 Results

we have used the same model architecture as model 1.This model is also trained for 25 epochs and calculate the losses for training and testing. we also calculate the iou score and dice score for test dataset.After training of the model we visualize some of the images with their both masks.The Following results are after 25 epochs.

- **Training and testing Losses**
  Training Loss : 0.0927
  Testing Loss : 0.1578

- **Loss Curve**

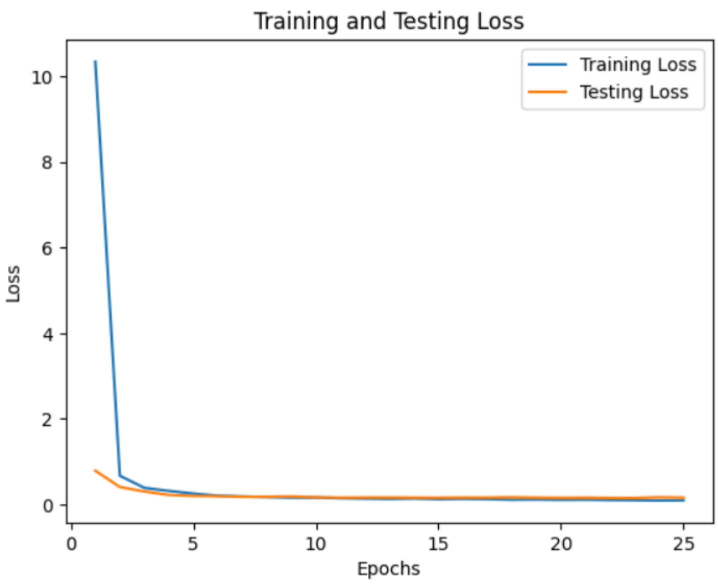

Figure 6: Training and Testing loss per Epochs

- **Iou and Dice Scores:**
  Test Iou score : 0.2985
  Test Dice Score : 0.8620

- **Good segmentations**

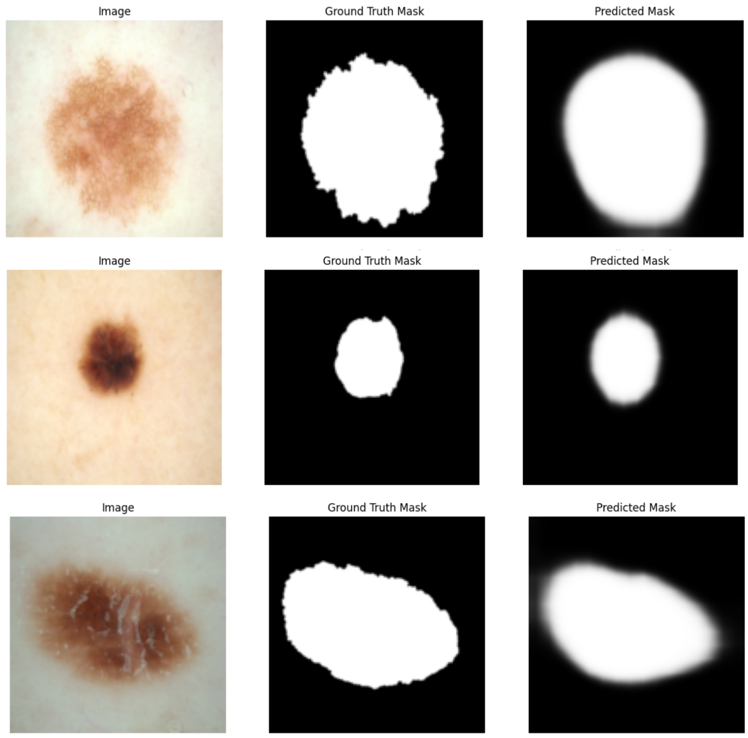

Figure 7: Good Results on test data

- **Bad Segmentations**

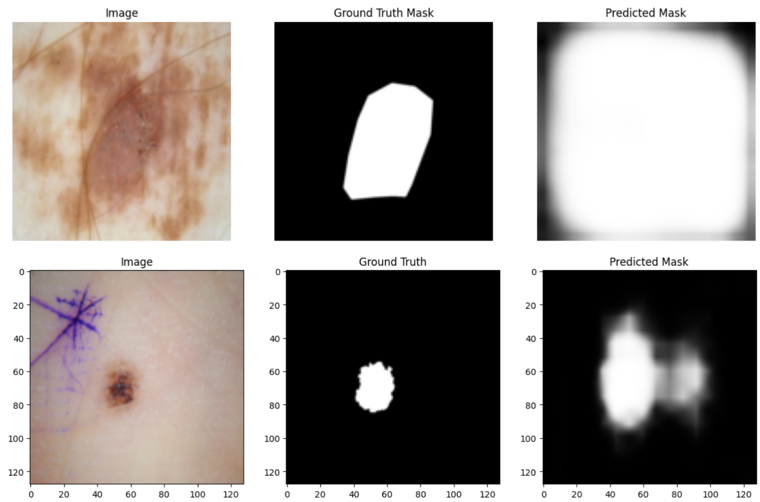

Figure 8: Bad Results on test data

5.1 Observations

In this second model we do back-propagation through all the encoder layers and get the features and do the segmentation. **So,the main difference between these models is that here we didn't freeze the weights but there we did.**We observed the following things in this model:

- **Train Loss:**The Train Loss is decreasing across all the epochs, which indicates that the model is learning from the training data. This is a positive indication as it suggests the model is improving its performance on the training set.
- **Test Loss:** The Test Loss values is also decreasing across all the epochs, which indicates that the model is learning from the training data and doing correct predictions.
- **Test IoU:** The Test IoU (Intersection over Union) is fluctuating in the initial epochs but it remains constant at 0.2985 for all epochs.
- **Test Dice:** The Test Dice coefficient is increasing across all the epochs. We get a good dice score for this model which is 0.8620 that means we are doing 86.20 % correct predictions.

Therefore, we can say that this model is doing correct predictions and we are getting good results in term of good segmentation masks.

6 Comparison between Models

In this assignment our main objective is to train a segmentation model using a MobileNet pre-trained on the ImageNet dataset as an encoder and a custom decoder that predicts segmented masks. Therefore, we make these two models one without fine-tuning and other one with fine-tuning.The main difference between these models is that how much the pre-trained model's weights are adjusted during training on our segmentation task.We have observed that model 1 is doing many kind of mistakes that model 2 doesn't e.g. Anomalies in loss calculations,Fluctuations in iou and dice scores. I am not saying that model 2 is giving all the correct predictions but it is doing well in the compression of model1.The Predicated masks by model 2 is almost similar to the ground truth masks.We know that the model predictions depends on the architecture we used for the task.Here we have used Convolutional Neural Network(CNN).Moreover, we can say that a model with fine-tuning is giving better results than without fine-tuning.

References

1.**Google Colab:** Click here
2. PyTorch Documentation
3. Article on Image Segmentation
4. Iou and Dice Score Article