PERSONAL REPORT

DATS 6303: Deep Learning

Liang Gao

2024-5-3

Contents

List of Figures

List of Tables

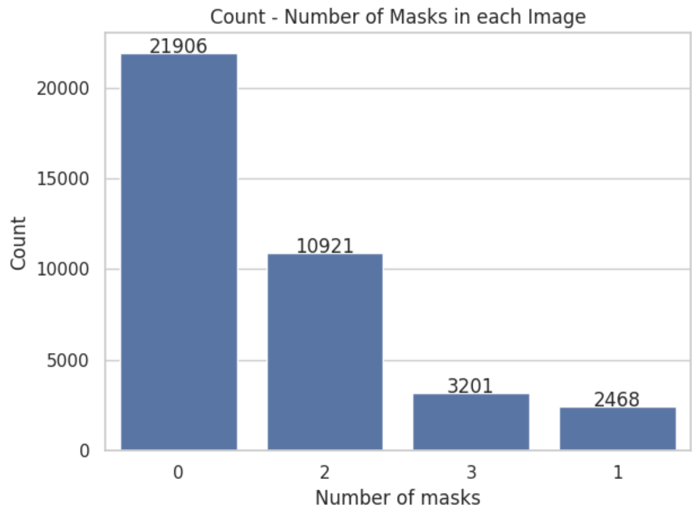

Figure 1: Number of Masks in each image

1 Introduction

The primary objective of this team project is to gain a comprehensive understanding of medical image segmentation and to delve into the functionality of convolutional neural networks (CNN) in addressing complex image-related challenges.

I am mainly responsible for the U-Net model. I started with data import, processing datasets, and model development. I prepared the presentation slides and most of the demo parts in Streamlit.

2 Personla work description

In the raw dataset, each 'id' represents a unique image slice. Each image slice has separate rows that classify segments of the image as 'large_bowel,' 'small_bowel,' and 'stomach.' The dataset structure has been reorganized such that each image slice is represented by a single row with three distinct columns dedicated to the three organ classes. Each column contains RLE-encoded pixel values initially listed in the segmentation columns of the dataset. We counted the number of masks(Figure 1) in each slice and deleted the images without any masks. Figure 2 illustrates the distribution of images containing segmentation masks across three anatomical regions. It presents the percentages of the images that feature segmentation masks for each specified organ, indicating that 36.59% of the images have masks for the large bowel, 29.1% for the small bowel, and 22.41% for the stomach.

The U-Net model architecture code is based on the Pytorch-UNet repository by Milesial (2020)[1], and the code for the training part references Saraki, (2023)[2]. We used the DICE coefficient as our metric. It is a statistical tool used to measure the similarity between two sets of data. Compare the pixel-wise agreement between a ground truth segmentation and a predicted segmentation. The equation to calculate the DICE coefficient is:

$$\text{Dice} = \frac{2 \times |X \cap Y|}{|X| + |Y|} \tag{1}$$

The corresponding loss function is the DICE loss, which directly considers the overlap between the predicted and true segmentation masks.

$$\text{Dice loss} = 1 - \text{Dice Coefficient} \tag{2}$$

For the training part, most techniques I used were the same as exams, such as adding data augmenta-

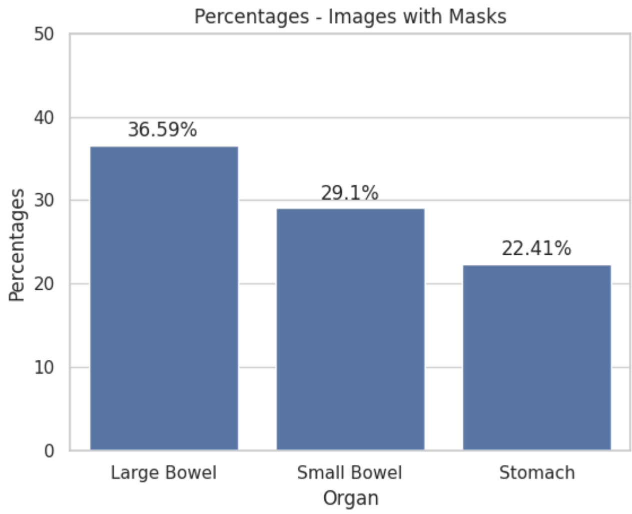

Figure 2: Percentage: Image with Masks

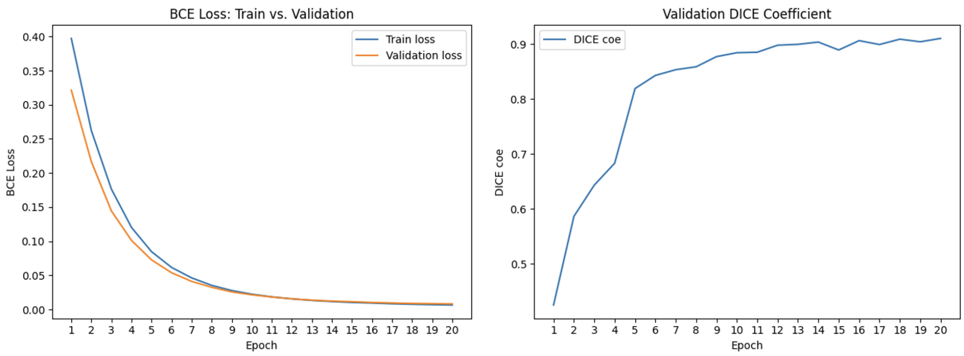

Figure 3: Results: BCE loss as criterion

tion for training data and changing image size and number of channels. I tried both 1(grey) and 3(RGB) as my input channel. There was not much difference in the performance, so I finally set my input channel as 1. We have three classes/masks, so the output channel is three. Both input and output are images in the torch tensor format. For the training criterion, I tried both DICE loss and BCEWithLogitsLoss (used in the Exam) and compared the results. I plot the training loss versus validation loss against the number of epochs and the Dice coefficient of the validation dataset. Figure 3 is the result with BCE-WithLogitsLoss as the criterion, figure 4 is the result of using Dice los, and figure 5 is the result with a combination of both loss functions. For the loss plot, the curve with BCEWithLogitsLoss is a little bit smoother than the other two. There is not much difference in the results plot. I simply want to compare the results of two different loss functions: one that I used during the exam and another that is new to me. The result of the test dataset used the model with Dice loss as the criterion.

Test Result I calculated the Dice coefficient separately for three classes. The results are in table 1.

Classes	Large bowel	Small bowel	Stomach
Dice coe	0.90	0.88	0.93

Table 1: Dice coe for three classes in test dataset

The ground truth segmentation is in figure 6, and the predicted segmentation is in figure 7.

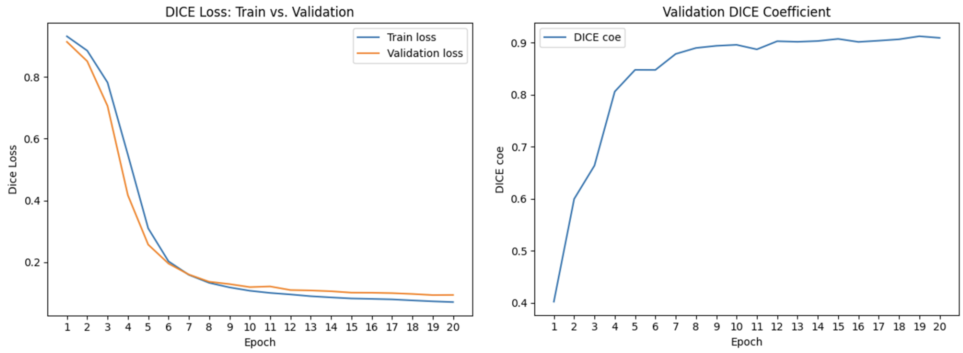

Figure 4: Results: Dice loss as criterion

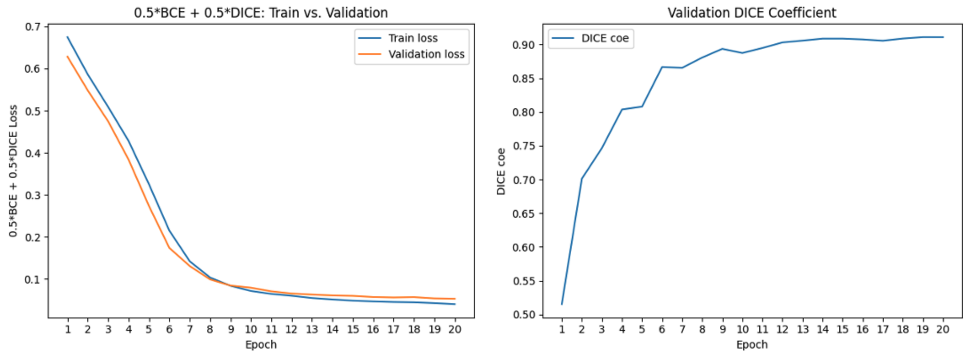

Figure 5: Results: Combination of Dice and BCE loss as criterion

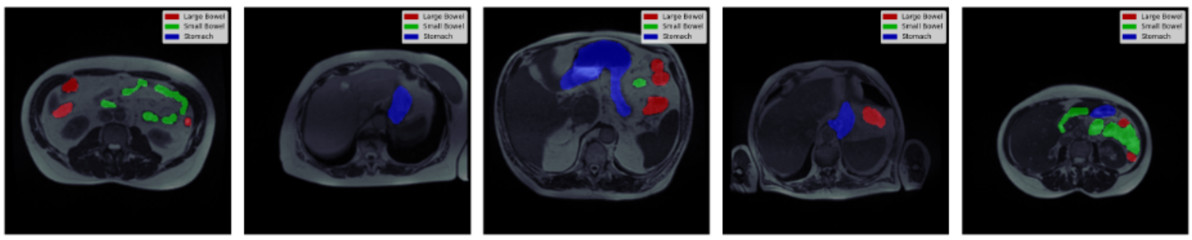

Figure 6: Ground Truth segmentation

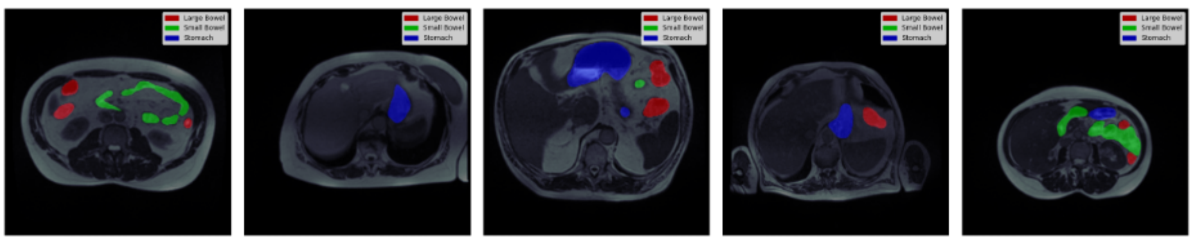

Figure 7: Prediction segmentation

3 Conclusions

Three important points about this model I learned through this project: first, the Double Conv layer enables it to build on the already extracted features to capture even more detailed information. Second, no dense layers in this model architecture, which reduces parameters, making U-Net less prone to over-fitting (no over-fitting in my training) and computationally efficient. Additionally, the spatial relationships and the location information within the image are preserved, which is crucial for accurate segmentation where the precise localization of objects within an image is necessary. Third, in the expansive path, concatenation with the contracting path allows it to use a higher-resolution feature map and capture more information. I think these are very useful and practical for my future work when constructing model architecture, especially in the computer vision domain.

Most core codes(about 50 to 60%) for this project are from the internet. This is my first time finishing a deep learning project from dataset processing to modeling. Although most work referenced others, I still learned a lot. For example, I gained more understanding of the data loader(how to deal with image data) and how it works with batch, and also model construction from graph to code. Those are very useful for my future study and work.

References

[1] Milesial, "Pytorch-unet," https://github.com/milesial/Pytorch-UNet, 2020.

[2] S. E. Saraki, "UW-Madison GI Tract Image Segmentation Train EDA," 2023, accessed: 2024-05-03. [Online]. Available: https://www.kaggle.com/code/sabahesaraki/uwmgi-train-eda