Problem Statement

This project aims to improve the accuracy of the Universeg model for segmenting instruments within medical images.

Datasets

The datasets used are KVASIR-SEG and KVASIR-INSTRUMENT.

Both KVASIR-SEG and KVASIR-INSTRUMENT are datasets of images and their corresponding segmentation masks. The images are 2D jpeg format of varying dimensions. Each segmentation mask has the same size as its corresponding image, however the segmentation masks were black and white only png files.

In KVASIR-SEG the segmentation task was to segment polyps from within gastrointestinal images, whereas in KVASIR-INSTRUMENT the segmentation task was to create a mask for medical instruments in the images (like endoscopes).

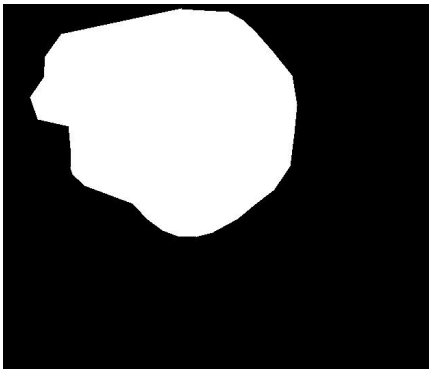

Figure1

Figure 2

Figure 1 is a segmentation mask for figure 2. Both are from KVASIR-SEG.

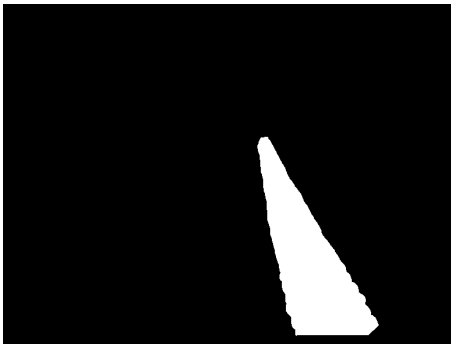

Figure 3

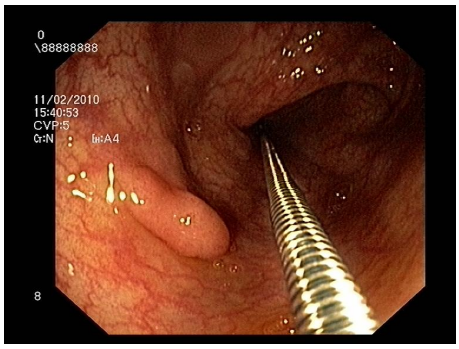

Figure 4

Figure 3 is the segmentation mask for figure 4 in KVASIR-INSTRUMENT.

Implementation Overview

In our project, we utilized Universeg, a convolutional neural network utilizing cross blocks in order to provide zero-training single forward-pass predictions on unseen medical image segmentation datasets.

Universeg has been pre-trained on around 50 medical image segmentation datasets. These datasets were from a diverse background of tasks. However, none of them involved instrument segmentation. Since Universeg is supposed to be a universal image segmentation model, we decided to test Universeg on KVASIR-INSTRUMENT, and found that it performed with low dice scores of around 0.4. Thus, we decided to fine tune Universeg on KVASIR-INSTRUMENT.

We ran SGD on Universeg with KVASIR-INSTRUMENT. We used the pre-trained Universeg weights rather than random ones, in order to fine tune Universeg to Kvasir-instrument and hopefully instrument segmentation. We measured the average training accuracy per epoch, and randomly sampled a support set for each prediction per batch, such that all data points in the support set were outside of the batch.

The resultant dice score was still low even after the fine tuning. Perhaps the amount of training data in KVASIR-INSTRUMENT was too low. Next steps could involve augmenting the data within KVASIR-INSTRUMENT, or utilizing techniques such as early stopping or differential privacy to prevent overfitting.

Lastly, we saved the resultant weights of our model in a persistent file.

Evaluations

We evaluated the average test accuracy of Universeg with just pretrained weights on KVASIR-SEG. We sampled a test set of around 10% of KVASIR-SEG and got the following accuracy: 0.37 dice score. This can be seen in the file Kvasir-seg_inference_testing.ipynb. This is the notebook that contains all of the initial inference testing on KVASIR-SEG.

We evaluated the average test accuracy of Universeg with just pretrained weights on KVASIR-INSTRUMENT as well. We sampled a test set of around 10% of KVASIR-INSTRUMENT and got the following accuracy: 0.15 dice score. This can be seen in the file Kvasir_instruments_prediction.ipynb. This is the notebook that contains all of the initial inference testing on KVASIR-INSTRUMENT.

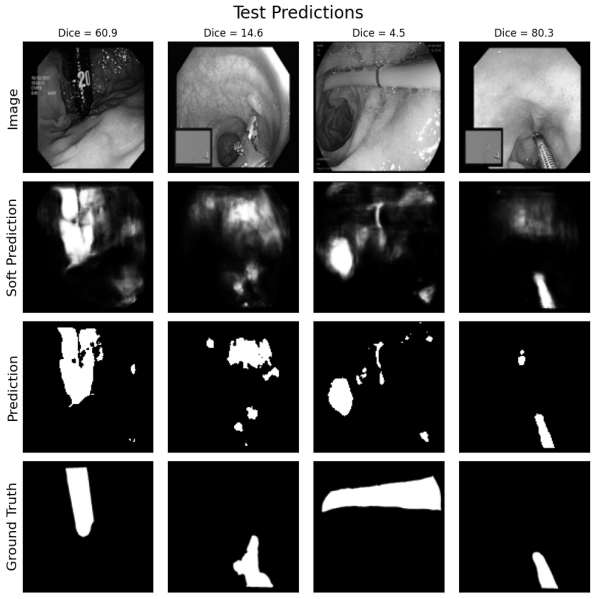

Image 1

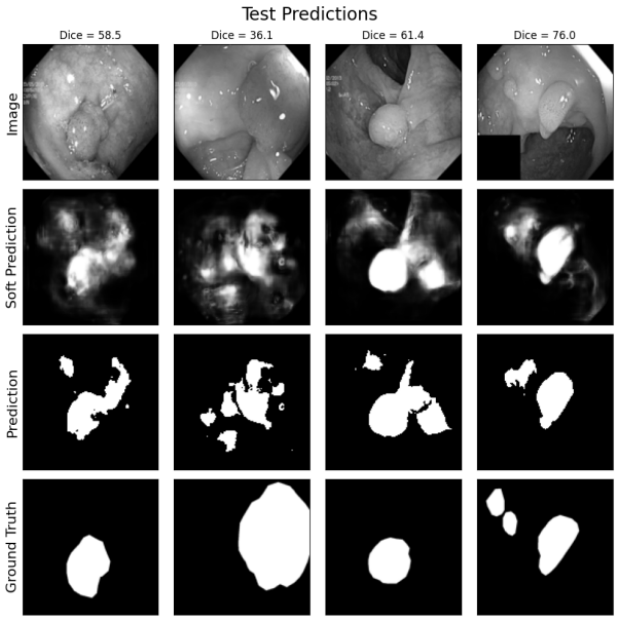

Image 2

Image 2 contains cursory visualizations and assessments of the accuracy of Universeg pre-trained on some images from the unseen dataset Kvasir-Seg. Likewise,image 1 contains the same visualizations and accuracy on images from Kvasir-Instument.

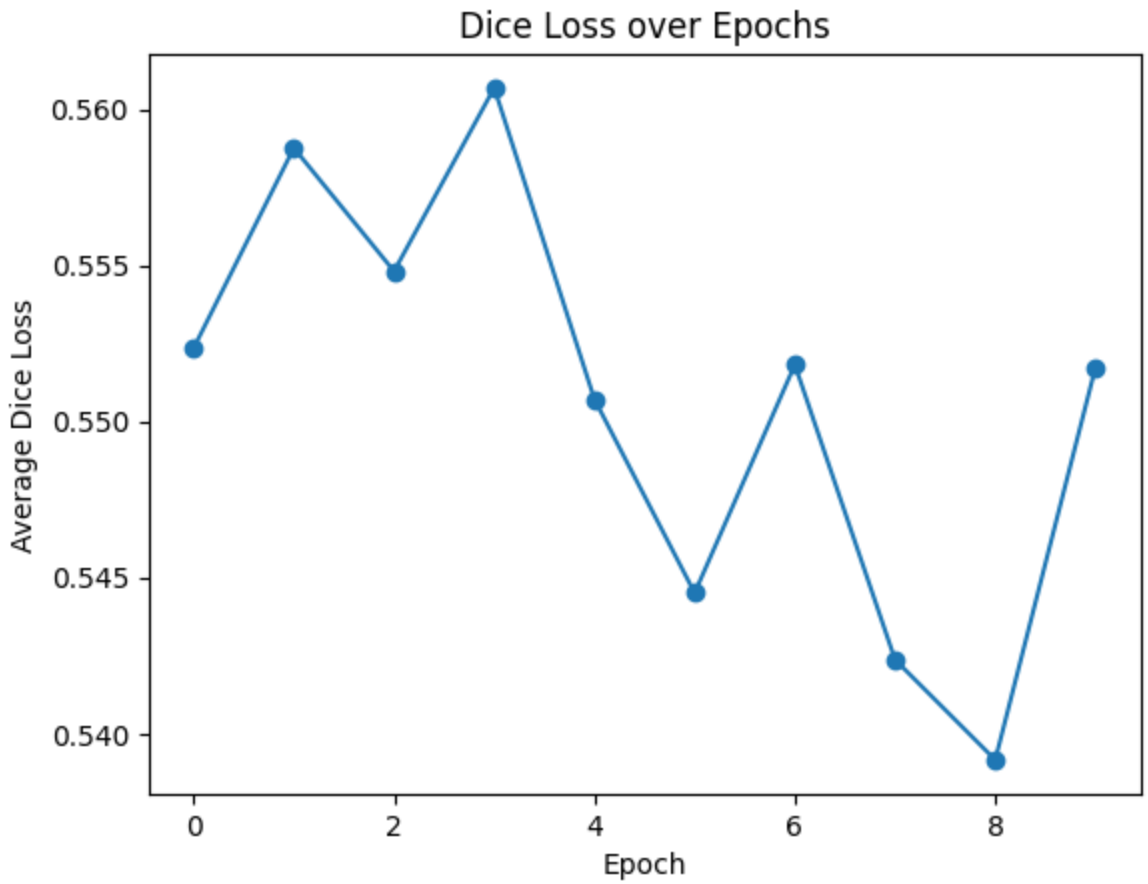

Figure 5.

Figure 5 contains the training accuracy using dice loss over the training set. It is the average dice loss per epoch. The final training dice loss was 0.552.

Lastly, we tested our fin-tuned model on a testing set from Kvasir-instrument to see if the fine tuning improved the accuracy of the pretrained Universeg weights on Kvasir-instrument.

The support set is taken from the first 32 image-label pairs from each batch (of size 64) sampled randomly from the test set. The support set is the same for each batch, it is the first 32 image-label pairs, and the rest of the 32 images on the batch are each input for predictions with the support set. This allows us to have a consistent support set for each batch and observe the accuracy with the support set over those 32 images. The randomization helps as well since, if the support set was chosen poorly, it can vary in the next run, possibly improving the accuracy.

The average test dice score was: 37

The notebook used to test can be found: https://colab.research.google.com/drive/1hFotU_zPkclLqhGKzFJs6orbsUBR3sug#scrollTo=jONsIKA9UOAE

Group Contributions

Since we were only a group of size 2, we largely worked on things together. However, some tasks were performed independently. Aman worked independently on the testing of the accuracy of pre-trained Universeg on Kvasir-seg. Konrad worked independently on the testing of the accuracy of pre-trained Universeg on Kvasir-instrument. Aman managed the github repository, and Konrad wrote the documentation. All other tasks were shared, and collaborative.

Repository

https://github.com/asyed03/CSC490-Tumor-Detection-Project/blob/main/evaluations/Kvasir_Instruments_Inference_Testing.ipynb

The fine tuned weights of Universeg can be found https://github.com/asyed03/CSC490-Tumor-Detection-Project/blob/main/trained_model_weights.pth. All evaluations can be found in the evaluations folder. The notebook used to fine tune our model can be found here: https://github.com/asyed03/CSC490-Tumor-Detection-Project/blob/main/Kvasir_Instrument_Fine_tuning.ipynb

References

- https://datasets.simula.no/kvasir-seg/
- https://arxiv.org/pdf/2304.06131.pdf
- https://datasets.simula.no/kvasir-instrument/