Gastro-Intestinal Tract Segmentation Using Multi-Task Learning

Bryan Chia
Stanford University
Department of Statistics
bryancws@stanford.edu

Helen Gu
Stanford University
Department of Statistics
helengu@stanford.edu

Nicholas Lui
Stanford University
Department of Statistics
niclui@stanford.edu

Abstract

Our paper investigates methods to improve on the baseline methods of semantic segmentation in medical imaging. Building on the UNet architecture, we implement two baseline methods, a UNet trained with a ResNet50 backbone and a more parsimonious and streamlined UNet. Building on the better-performing streamlined UNet, we investigate using multi-task learning via supervised (regression) methods and self-supervised (contrastive learning) methods. We find that the contrastive learning method has some benefits in cases where the test distribution is signficantly different from the training distribution (i.e. the patient is not seen by the model during training time). Finally, we also investigate a method of improving on the UNet model by adding image metadata such as the position of the MRI scan cross-section, and the pixel height and width known as Feature-wise Linear Modulation (FiLM). We find that FiLM is beneficial when there is a slight overlap in the training and test distribution, in that the test distribution consist of future scans of patients previously trained on.

1. Introduction

Deep learning techniques can help in the segmentation task of tracing out the stomach and intestines. Building on the baseline UNet model, we investigate using multi-task learning and feature-wise linear modulation to improve performance on unseen test data in similar medical applications where patient data is scarce.

In 2019, approximately 2.5 million people were being treated for gastro-intestinal tract cancer worldwide using radiation therapy. Radiation oncologists attempt to deliver high doses of radiation using X-ray beams pointed to tumors for 10-15 minutes a day over 1-6 weeks. Prior to delivering high doses of radiation to treat gastro-intestinal tract cancer, radiology oncologists are required to spend roughly 1 hour per day manually tracing out the stomach and intestines in MRI scans in order to ensure that they direct x-ray beams in a manner to avoid those crucial organs.

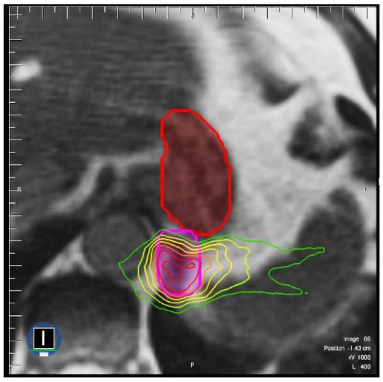


Figure 1. Example of a GI Tract with Stomach and Intestinal Mask

Fig. 1 shows an example input MRI scan, the tumor (thick pink line), which is close to the stomach (thick red line), is targeted with various does of radiation highlighted by the different colored thin lines (with a larger area representing a larger dosage). Since the position of the tumor and intestines varies day-to-day, tracing such scans is a time-intensive process that prolongs treatment from 15 minutes per day to about an hour, reducing the ability of radiologists to treat more patients, and can be difficult for the patients to tolerate.

Deep learning can help automate the segmentation process by segmenting the stomach and intestines to allow for faster treatment. In this project, we create a model to segment the stomach and intestines on MRI scans. As seen in Fig. 2, our inputs are cross sectional 2D images of an MRI scan from actual cancer patients who had 1-5 MRI scans on separate days during their radiation treatment. We start with a baseline UNet model to predict masks for each pixel, indicating whether the pixel constitutes a (i) stomach, (ii) small intestine, (iii) large intestine, or (iv) combination of the stomach and small/large intestine. Thus, we have a multi-label segmentation problem. [15]

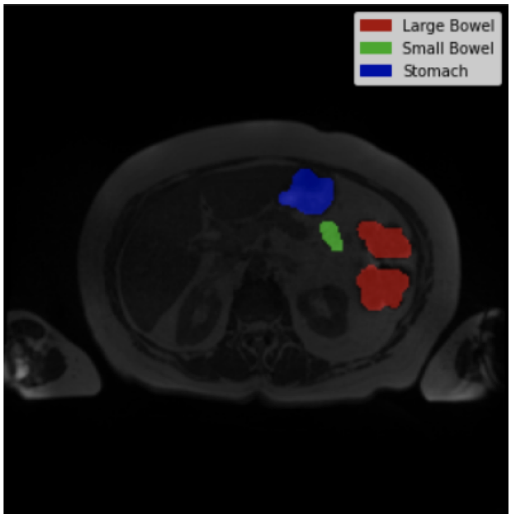


Figure 2. Example of an Input with Target Mask Overlay

2. Related Work

2.1. Semantic Segmentation in Medical Imaging

Our work builds on a wide literature of semantic segmentation in medical imaging, including brain tumor segmentation [13] and polyp detection in colons [12]. These segmentation models have attained extremely promising results, paving the way for their use in real-world medical applications. For instance, Tran et al. [24] develop a modified UNet to achieve a high dice score (>0.9) for six out of seven popular medical segementation tasks, demonstrating the high generalizability and real-world applicability of cutting-edge segmentation models.

However, while medical imaging in many parts of the body have been studied, there is a lack of robust research on the segmentation of gastro-intestinal tract scans. We aim to fill this gap by applying architectures and techniques from the available research to the problem of GI tract segmentation.

2.2. Multi-Task Learning

Multitask learning, in which a model learns to solve multiple tasks simultaneously, can improve efficiency and overall performance of the main task by improving generalization as the model learns shared representations through training on related tasks [5]. Novosel *et al.* [14] demonstrate performance boosts to semantic segmentation using a self-supervised multitask approach. They leverage depth prediction and colorization, both of which can be learned by the model with no additional annotation cost, as auxiliary tasks to segmentation.

Previous literature [10] has found weakly supervised multitask learning to be effective in segmentation of medical images containing lesions. Additionally, we find a number of papers that explore self-supervised learning using position prediction as an auxiliary task to improve segmentation [4] [25]. One potential task that can easily be trained without additional segmentation labels using our data is position regression, suggesting that multitask learning with position regression might prove promising in this GI tract segmentation task.

2.3. Contrastive Learning

Our contrastive learning methods are based on the seminal work by Chen *et al.* [8] that showed a simple framework for doing contrastive learning to learn pretrained representations of images. Linear classifiers trained on these self-supervised representations outperformed other methods for fewer labelled data. Chen *et al.* cited some important aspects of contrastive learning such as data augmentation operations, separating the learned representations and the contrastive loss through a learnable non-linear transformation, a larger batch size, and longer training. Using contrastive learning to learn pretrained representations have been shown to be effective in medical image segmentation with limited annotated data as it allows us to learn information beyond the annotated dataset. Chaitanya *et al.* shows that using local representations derived from pretraining on a contrastive learning task is useful for semantic segmentation, and in fact orthogonal of data augmentation methods that can also strengthen results when there is limited labelled data. [6]

Finally, Zeng *et al.* tweaks the conventional contrastive learning methods as it suggests that using just data augmentations to generate contrastive data pairs may lead to a large false negative rate for medical images. This is because consecutive cross-sectional slices look extremely similar to each other. It instead proposes using relative position difference rather than hard partition strategy to distinguish positive from negative examples when designing batches. [26]

2.4. Feature-wise Linear Modulation (FiLM)

Perez et al. [19] introduce feature-wise linear modulation (FiLM) layers which carries out a simple, feature-wise affine transformation on a neural network's intermediate features, conditioned on an arbitrary input. Lemay et al. [16] apply this concept to image segmentation by using an image's metadata to condition the network's intermediate features. They one-hot encode the metadata of spinal cord scans which includes information on the scan (e.g. vendor, acquisition parameters) and the patient (e.g. demographic characteristics). The metadata vector is then passed into a fully-connected networks to generate a set of scale and shift parameters. The scale and shift parameters are subsequently applied to the output of each convolutional layer in the UNet. By conditioning on metadata, they achieve a 5.1% improvement in the dice score.

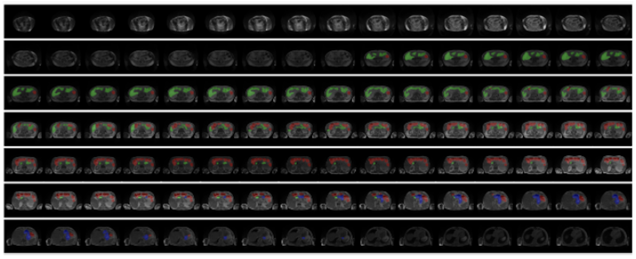


Figure 3. Selection of Scans for Case 134 On a Given Day

3. Dataset and Features

3.1. Raw Dataset

Our training (including validation) data consists of 85 different cases (patients), with roughly 3-5 days of scans per patient provided in 16-bit gray-scale PNG format. Each day has about 150 different scan slices, which is a 2D cross-section of the patient. Fig. 3 shows an example of just 1 case over 1 day, with the various scan slices. Slices have been colored based on the masks for the stomach, large, and small intestines. [23]

The unseen test set from Kaggle are roughly 50 different cases, some of which are later days for cases in the training data, and some of which are not in the training data. [2] A description of the testing metric will be described in Sec. 5.2.

3.2. Preprocessing

We decode the given segmentation masks using run-length decoding. For each scan, we obtain three segmentation masks corresponding to the three organs of interest. In each mask, a pixel takes on a value of 1 is the organ is present and 0 otherwise. We concatenate all three masks together to obtain a single mask tensor for each scan.

We first process each image as a grayscale image with only one channel dimension, and normalize the image using Min-Max normalization. We then replicate each channel three times and concatenate them so it becomes a grayscale image in the RGB three dimensional space.

Each image-mask pair is then resized to 224 x 224 pixels so that they can be uniformly processed by the DataLoader. Some masks are resized from larger shapes, and the bilinear interpolation method is used to determine the labels for each pixel.

Our dataset is then split into a 70/20/10 split for training, validation, and testing. We decided to split the dataset on two levels to mimic the final test dataset from the Kaggle challenge. The first split is done on the patient level, so that the model is not able to learn certain patient-specific features from training and apply that during validation. The second split is done on a patient-day level as the Kaggle test set also sometimes includes future MRI scans of patients whose MRI scans are seen during training.

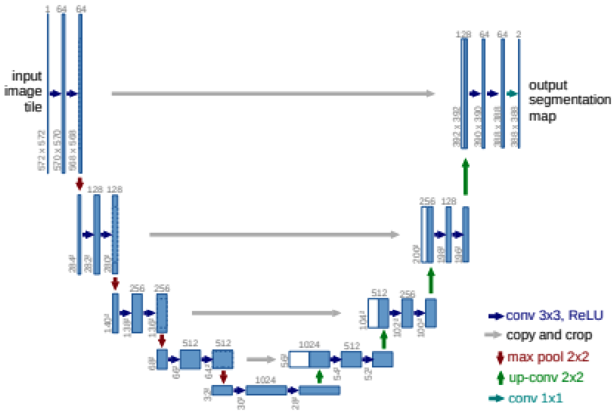


Figure 4. U-Net Architecture from Original Paper

4. Methods

4.1. Model Architecture

We first implement the UNet architecture for semantic segmentation in order to establish a baseline performance for our novel methods.

UNet is a semantic segmentation model that has 5 to 6 downsampling layers and the same amount of upsampling layers. The downsampling layers are used to encode information about each of the pixels much like a CNN. The upsampling layers are then used as a decoder to decode the class (in our case, presence of a class as we have a multi-label problem), much like an autoencoder. While most semantic segmentation models are based on autoencoders, the distinguishing feature about the UNet model is that it uses the same number of upsampling and downsampling layers because each upsampling layer also copies over cropped encodings from the downsampling layers by concatenating them to the decoded information from previous upsampling layers as seen in Fig. 4. [22]

Furthermore, the UNet was first developed as a method for biomedical image segmentation, and thus has close relations to our implementation.

We first implemented our own manual version of UNet so that we could tweak the architecture to allow for multi-task learning. We created two versions of the UNet. The initial version (*"Big UNet"*) follows the model by Ronneberger *et al.* very closely in terms of model dimensions, and uses a pre-trained ResNet50 on Imagenet as the encoder. We referenced an example implementation from an online Github example implementation to build this [18]. Due to memory limitations, we had to adapt the original ResNet50 by removing the last 2048 channel layer, so our last encoder layer instead outputs a 1024 channel encoding.

However, this corresponds with the original UNet architecture, where the final encoder layer has 1024 channels. [22] We tried various experiments which implemented freezing the first 4 layers of ResNet50, and without and found that there was little performance discrepancy between the two.

We also implemented a simplified version of the above UNet (*"Small UNet"*), inspired by an online Kaggle example implementation [3]. The simplified model has just two convolutional layers (akin to one residual block in the ResNet), and no skip connections. The Big UNet instead stacks multiple residual blocks (with much higher dimensions) per downsampling layer, as well as skip connections. Having fewer convolutional layers and smaller dimensions per downsampling layer reduces the parameters, and thus the need for residual connections as it is easier to optimize. The issue with using our simplified encoder though, is that it is not pretrained on any image tasks and therefore may take a longer time to converge to optimal values, although we found that this was not generally true.

4.1.1 Segmentation Loss Function

Although our ultimate goal as mentioned in Sec. 5.2 was to increase the dice score, we combined the dice loss with the binary cross-entropy loss for each class, using the method proposed by Rajput, who showed that combining the binary cross-entropy loss and dice loss generally leads to a higher dice score. [20] Weighting the binary cross-entropy loss with the dice loss allows the model to focus on not just one class in semantic segmentation (since dice loss balances between precision and recall). This allowed our model to consider within-class label imbalance, as we had a lot more negative than positive labels for each class.

Our segmentation loss for a cross-sectional slice i, pixel j, and label-type l, which is a weighted average of a binary cross-entropy loss and dice loss as given in Eq. (1) .

$$\mathcal{L}_{seg}(S_{il}, \hat{S_i l}) = 0.6 * \sum_{j=1}^{J} S_{ijl} log(\hat{S}_{ijl}) + 0.4 * (1 - \frac{2|S_{il} \cap \hat{S_i l}|}{|S_{il}| + |\hat{S_i l}|}) \tag{1}$$

4.2. Multi-Task Learning

After training the aforementioned baseline image segmentation models, our goal is to create what we believe is a novel approach towards semantic segmentation. Traditionally, training high-performing segmentation models that generalize well requires an abundance of labeled data; however, labeling GI scans is laborious and time-consuming. Thus, we aim to explore whether multitask learning approaches that require few additional segmentation labels can improve model performance and generalizability.

In particular, we believe that the baseline model can be improved by incorporating additional contextual information from the images. Specifically, the baseline model considers each image independently of others and fails to consider that some images come from the same patient, the same slice, or the same point in time– all pieces of information that may help improve the model's segmentation accuracy. Thus, we devise a number of tasks that aim to capture this information that can be treated as auxiliary to the main segmentation task.

Fig. 6 shows two proposed architectures to adapt UNet for multi-task learning. The UNet Model will be trained on the main task of multi-label segmentation, and an auxiliary task. We formulated two versions of the auxiliary task, one being self-supervised learning, and the other a supervised learning task. The UNet model will share encoder weights between the main and auxiliary tasks but have task-specific decoder weights and architectures for the two tasks. The loss function of the main and auxiliary tasks will be combined in a weighted loss function. We experiment with two different types of weighted loss functions, a constant weight loss function (Eq. (2)) and an training-time adaptive loss function where the auxiliary task weight decays over time (Eq. (3)).

$$\sum_{ijl} \mathcal{L}_(S_{ijl}, \hat{S_i jl}) = \sum_{ijl} \mathcal{L}_{seg} + c \sum_{i} \mathcal{L}_{aux} \tag{2}$$

$$\sum_{ijl} \mathcal{L}_(S_{ijl}, \hat{S_i jl}) = \sum_{ijl} \mathcal{L}_{seg} + \frac{c}{t} \sum_{i} \mathcal{L}_{aux} \tag{3}$$

We designed two variations of our auxiliary training tasks in such a way that induces the model to focus not just on individual images, but also on contextual information such as slice position and relationships between images to other MRI scans for a given patient and time. We hypothesize that this allows the model to perform better on unseen test patients by enabling the model to learn better high-level representations of scan features, improving the model's ability to to recognize the stomach or intestinal structures. Furthermore, if we have more unlabelled data than labelled data, this model will allow us to circumvent the issue of label sparsity.

4.2.1 Auxiliary Task: Position Learning

The first auxiliary task attempts to learn the position of an cross-sectional scan (for most scans, this ranged from 1-144 or 1-80), based on the encoder outputs. Position information is relatively straightforward to extract from the data using image names, and previous work suggests that incorporating position in medical imaging segmentation tasks can help improve performance [4] [25]. Decoder outputs were passed into a mean squared error loss function for regression, given by Eq. (4) for each cross-sectional slice i:

$$\mathcal{L}_{class}(S_i, \hat{S}_i) = |S_i - \hat{S}_i|^2 \tag{4}$$

4.2.2 Auxiliary Task: Contrastive Learning

The second auxiliary task attempts to classify an image passed in with the reference image as either being from the same or different cluster. We considered a cluster to be similar if it came from the same patient, same time, and were adjacent cross-sectional slices of the MRI scan.

We define the negative slices in the batch as all other possible cross-sectional slices, except that we place a restriction that each batch should not contain more than one patient's scans to impose a restriction that negative MRI scans that are too close in cross-sectional proximity to the positive MRI scan could lead to false negatives and confuse the model. We also decided against using data augmentation methods for our baseline model per the discussion from Zeng *et al.*, that suggests that excessive data augmentation could confuse the model in medical image segmentation applications where there is very fine-grained dissimilarities between positive and negative examples. [26]

We then use the methods developed in SimCLR to perform contrastive learning. We pass in a batch of 32 scans at a time or 16 pairs, requiring the model to learn which of the other 16 images is its adjacent image. We perform two variations of this - one with and one without data augmentation included. We recognize that 32 scans in a batch is smaller than what is typically used for contrastive learning applications, but were unable to increase the batch size due to memory constraints.

The classification task loss function for SimCLR is the InfoNCE loss function, given by Eq. (5) for each scan i:

$$\mathcal{L}_{class}(S_i) = -log \frac{exp(S_i, Si^+)}{exp(S_i, Si^+) + \sum_{j=1}^{15} exp(S_i, Si^-)} \tag{5}$$

4.3. Feature-wise Linear Modulation (FiLM)

Finally, we explore the use of image metadata as a conditioning vector for intermediate outputs in the neural network. For each scan, we create a metadata vector which comprises the slice position, the case number, the day number, the pixel height and width, and the original image height and width. The metadata vector is passed into two different three-layer FCNs to generate a 48x1 scale vector and a 48x1 shift vector respectively. The scale and shift parameters are applied to the output of the first convolutional layer in the Small UNet, allowing the model to incorporate image metadata from an early stage.

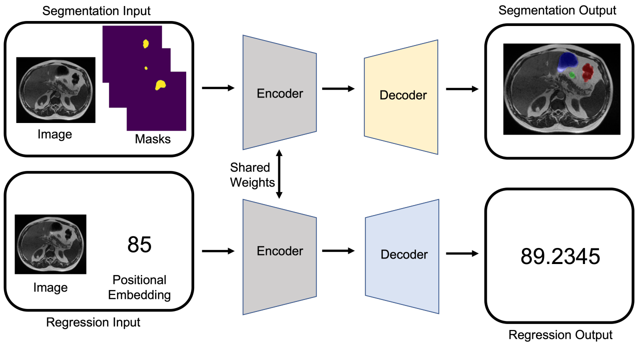


Figure 5. Multi-Task UNet with Position Prediction

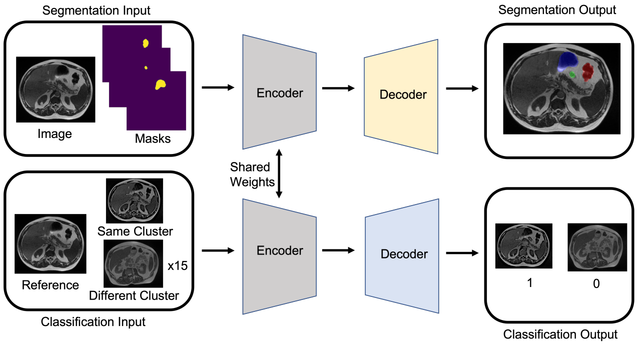


Figure 6. Multi-Task UNet with Contrastive Learning

5. Experiments

5.1. Experiments

In terms of hyperparameters, we use an Adam Optimizer and Cosine Annealing. Pytorch Lightning's Auto Learning-Rate Finder is used to choose an initial learning rate (a small run is done to select an optimal intial learning rate). Because we eventually wanted to incorporate SimCLR, the largest possible batch size of 32 was chosen. While SimCLR typically uses a larger batch size, we were limited by memory constraints. Max epochs was set at 10 with early stopping incorporated. Our experiments are developed on top of the Stanford ML group's starter code base [21].

5.2. Evaluation Metric

The metric for evaluation is the Dice coefficient, which is a commonly used statistic to compare the pixel-wise agreement between a predicted segmentation mask and the ground truth. [1] The formula is given by Eq. (6). For instance, for a given scan's segmentation mask for the stomach, we compute the pixel-wise agreement between our predictions and the binary ground truth labels. The dice coefficient is defined for the image to be 0 when both the predicted and actual segmentation mask is empty.

Model	Auxiliary Task	Overall	Large Bowel	Small Bowel	Stomach
Small UNet	-	0.8323	0.7982	0.8046	**0.8941**
Big UNet	-	0.8033	0.7705	0.7770	0.86238
Small UNet (Multitask)	Position Learning	0.6991	0.6294	0.6959	0.7721
Small UNet (Multitask)	Contrastive Learning	0.8249	0.7916	0.8037	0.8795
Small UNet (FiLM)	-	**0.8345**	**0.7985**	**0.8114**	0.8936

Table 1. Validation Dice Coefficient Results (Patient-Day Split)

Model	Auxiliary Task	Overall	Large Bowel	Small Bowel	Stomach
Small UNet	-	**0.8235**	0.7782	0.8089	**0.8833**
Big UNet	-	0.7847	0.7483	0.7721	0.8337
Small UNet (Multitask)	Position Learning	0.5806	0.4745	0.4728	0.7946
Small UNet (Multitask)	Contrastive Learning	0.8234	**0.7864**	**0.8166**	0.8670
Small UNet (FiLM)	-	0.8172	0.7799	0.8037	0.8680

Table 2. Validation Dice Coefficient Results (Patient Split)

$$\text{Dice Coefficient} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{6}$$

5.3. Results

Overall, we find that the methods we explored yield little to no improvement over the baseline Small UNet on the validation data (Tab. 2). FiLM has the best overall performance by a very slim margin on the patient-day split, whereas small UNet has the best performance on patient split. Further, we find that the performance of smallUNet degrades only very slightly when generalizing to the patient split, suggesting that it has learned generalizable representations allow it to perform relatively well even on unseen cases.

Model	Auxiliary Task	Overall	Large Bowel	Small Bowel	Stomach
Small UNet	-	0.8242	0.7903	0.8058	0.8763
Small UNet (Multitask)	Contrastive Learning	0.8095	0.7748	0.7944	0.8594
Small UNet (FiLM)	-	**0.8314**	**0.7957**	**0.8114**	**0.8872**

Table 3. Test Dice Coefficient Results (Patient-Day Split)

Model	Auxiliary Task	Overall	Large Bowel	Small Bowel	Stomach
Small UNet	-	0.7992	0.7507	0.7876	0.8593
Small UNet (Multitask)	Contrastive Learning	**0.8093**	**0.7763**	**0.7948**	0.8568
Small UNet (FiLM)	-	0.8026	0.7628	0.7846	**0.8605**

Table 4. Test Dice Coefficient Results (Patient Split)

We evaluate the three best performing models (Small UNet, multitask with contrastive learning, and FiLM) on the test set. On the test set, we find that FiLM and multitask with contrastive learning yield marginal improvements over the baseline Small UNet. These slight improvements may be due to noise, but they may also suggest that Small UNet overfits to the validation set, whereas the noisy auxiliary tasks that the multitask and FiLM models learn provide some regularization.

5.4. Discussions and Error Analysis

5.4.1 Resnet50 (Big UNet)

Surprisingly, we find that Small UNet outperforms Big UNet across the board, despite the more expressive capacity of Big UNet. We believe that this may be due to undertraining of Big UNet, particularly for patient split results where Big UNet achieves a substantially lower Dice coefficient than Small UNet. Examining the validation loss in Fig. 7, it appears that while validation loss converges for Small UNet, loss for Big UNet may still decrease further with a larger number of training epochs. As we find Small UNet produces decent results and converges more quickly than Big UNet, we build upon Small UNet in our experiments exploring various architectures.

5.4.2 Multi-Task Learning: Regression

We find that training position prediction through linear regression as an auxiliary task for segmentation degrades performance, both on the patient-day and patient split datasets. We speculate that incentivizing the model to learn to predict slice position may cause it to focus disproportionately on global features of the image such as overall slice size, shape, and orientation, rather than on the more local regions

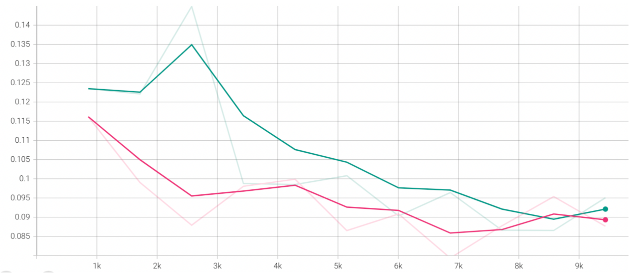


Figure 7. Small UNet (pink) vs. BigUNet (green) Validation Loss on Patient Split

that capture information helpful for segmentation. Further, slice index prediction is likely to be a difficult and noisy task, so it is perhaps unsurprising that it does not help improve performance on the segmentation task.

5.4.3 Multi-Task Learning: Contrastive Learning

Adding contrastive learning as an auxiliary task to semantic segmentation improves on the performance of the Small UNet very slightly as seen in Tab. 4, although this is only the case when the model has not seen the patient in the test set. The benefits of adding contrastive learning is reduced when the model has seen the patient on a different day in the past.

It should be noted that it is important to weight the loss of the auxiliary task at a much smaller amount as supposed to the main task the performance of the main task could be degraded as the gradient signals from the additive loss function confuses the weights for the main decoder. Our experiments found that using an equal weighting scheme in the loss function, as supposed to a 1:9 weighting scheme for the auxiliary and main task can degrade performance by up to 2%. This is also the trade-off that we saw in multi-task learning, that while it may help the encoder learn richer representations of the image, the combined loss can result in impure gradient signals to the main task decoder. By imposing a constant weighting scheme, our loss function creates a natural annealing of the auxiliary task weight as the SimCLR loss decreases over time to 0 as seen in Fig. 8. However, seeing that adding contrastive learning as the auxiliary task did not provide the added benefits like we hoped, perhaps using more sophisticated weighting schemes such as random weighting or adaptive auxiliary losses using gradient similarity might be helpful. We discuss these approaches in Sec. 6.

Breaking things down by class, we note that contrastive learning does help slightly by providing about a 1-2% gain for classes that are more difficult to learn such as the large bowel and small bowel in Tab. 4. However, this could also be because we note a trend that adding contrastive learning tends to cause the model to over-predict smaller classes.

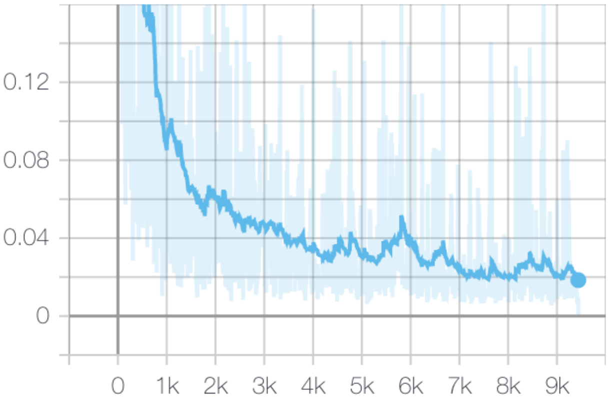
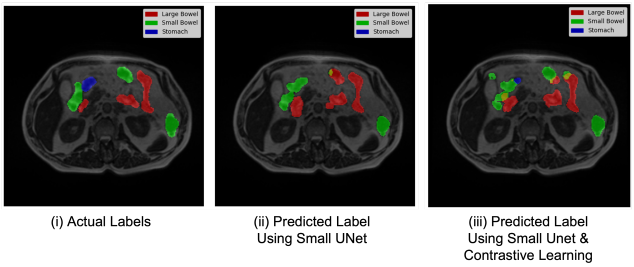


Figure 8. SimCLR Training Loss

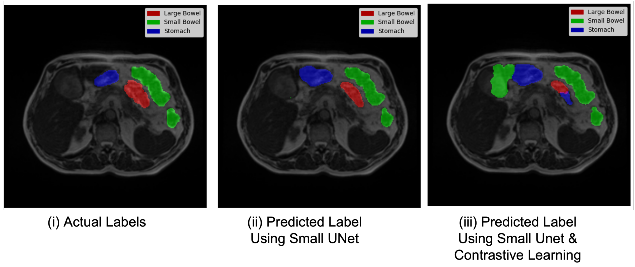


(i) Actual Labels

(ii) Predicted Label Using Small UNet

(iii) Predicted Label Using Small Unet & Contrastive Learning

Figure 9. Example 1: Comparison of Actual Labels and Predicted Labels using *Small UNet* and *Small UNet with Contrastive Learning*



(i) Actual Labels

(ii) Predicted Label Using Small UNet

(iii) Predicted Label Using Small Unet & Contrastive Learning

Figure 10. Example 2: Comparison of Actual Labels and Predicted Labels using *Small UNet* and *Small UNet with Contrastive Learning*

The Small UNet with contrastive learning has a higher false positive:false negative ratio for the large and small bowel classes as supposed to the Small UNet. This could help, as seen in Fig. 9, but also hurt performance, as seen in Fig. 10.

5.4.4 Feature-wise Linear Modulation

The small UNet with feature-wise linear modulation does not produce a significant improvement in performance over the small UNet baseline. This is unsurprising. The metadata we used comprises information on slice position, case number, day number, slice height/width, pixel height/width. With the exception of slice position, it is not clear how the remaining metadata will contribute to better segmentation performance. This is contrast to Lemay et al. [16] who use metadata on the patient's demographic characteristics and disease type. The information they use has a clear mapping to better segmentation (for instance, a patient's demographic characteristics provides information on the anatomical structure of scans). Given that our metadata is not very relevant to segmentation, the benefits of feature-wise linear modulation is counteracted by the additional noise introduced, leading to a lack of significant performance gains.

6. Conclusion and Future Work

In this paper, we investigated ways to improve the performance of segmentation models on gastro-intestinal tract scans. We explore multi-task learning, with slice position prediction and contrastive learning as auxiliary tasks. We also explore feature-wise linear modulation to improve model performance by conditioning intermediate outputs on the image metadata. While these techniques individually did not produce a convincing improvement over the UNet baseline, we do see a slight improvement in test set performance for certain classes. We consider future directions that we can take to refine the techniques in the paper:

6.1. Adaptive Weight Loss Functions

Lin *et al*. suggests a simple weighting scheme for auxiliary tasks that is easy to implement, and evaluated against many image datasets show that it is effective when compared to state-of-the-art strategies as it has a higher probability to escape local minima than using just fixed weights. [17] Another weighting scheme from Du *et al*. adapts the auxiliary losses using gradient similarity between auxiliary and main task to allow the model detect and only use the auxiliary loss when it is helpful to the main loss. This might have allowed us to use more noisy contrastive learning methods such as data augmentation as well. [11]

6.2. MoCoV2

Given our limitations in batch size, future work could also include a contrastive learning method that does better in small batch sizes. MoCoV2 is a method that keeps a running queue of negative samples to reduce reliance on large batch sizes for performance, and has been shown to outpeform SimCLR using a large batch size. The paper also showed that correct data augmentation is important, which would be an interesting area of future work. [9]

6.3. Different architectures

We can also explore the use of different segmentation architectures, such as DeepLab [7]. DeepLab is a CNN developed and open-sourced by Google that relies heavily on atrous convolutions to achieve better segmentation performance.

6.4. Ensembling

While the techniques explored do not individually lead to a boost in model performance, we recognize that the models trained have learnt different feature representations. By averaging predictions over models trained with different techniques, we might be able to achieve a better overall performance by averaging out random noise patterns.

7. Contributions

All authors contributed to all parts of the project. Bryan primarily worked on the Large UNet and contrastive learning, Helen primarily worked on the Small UNet and the regression auxiliary task, while Nicholas primarily worked on FiLM and data processing.

We also acknowledge the work of Ammar Alhajali, and Kevin Lu for their example implementations of UNet.

References

[1] Sørensen–dice coefficient, Apr 2022. 5

[2] Uw-madison gi tract image segmentation leaderboard, May 2022. 3

[3] Ammar Alhaj Ali. Uwmgi: Unet pytorch [train] with eda. https://www.kaggle.com/code/ammarnassanalhajali/uwmgi-unet-pytorch-train-with-eda, May 2022. 4

[4] Wenjia Bai, Chen Chen, Giacomo Tarroni, Jinming Duan, Florian Guitton, Steffen E. Petersen, Yike Guo, Paul M. Matthews, and Daniel Rueckert. Self-supervised learning for cardiac mr image segmentation by anatomical position prediction. In Dinggang Shen, Tianming Liu, Terry M. Peters, Lawrence H. Staib, Caroline Essert, Sean Zhou, Pew-Thian Yap, and Ali Khan, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 541–549, Cham, 2019. Springer International Publishing. 2, 5

[5] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. 2

[6] Krishna Chaitanya, Ertunc Erdil, Neerav Karani, and Ender Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations. *CoRR*, abs/2006.10511, 2020. 2

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 8

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020. 2

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020. 8

[10] Tianshu Chu, Xinmeng Li, Huy V. Vo, Ronald M. Summers, and Elena Sizikova. Improving weakly supervised lesion segmentation using multi-task learning. In Mattias Heinrich, Qi Dou, Marleen de Bruijne, Jan Lellmann, Alexander Schläfer, and Floris Ernst, editors, *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning*, volume 143 of *Proceedings of Machine Learning Research*, pages 60–73. PMLR, 07–09 Jul 2021. 2

[11] Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity, 2018. 8

[12] Nguyen Thanh Duc, Nguyen Thi Oanh, Nguyen Thi Thuy, Tran Minh Triet, and Dinh Viet Sang. Colonformer: An efficient transformer based method for colon polyp segmentation. *arXiv preprint arXiv:2205.08473*, 2022. 2

[13] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017. 2

[14] Prashanth Viswanath Jelena Novosel and Bruno Arsenali. Boosting semantic segmentation with multi-task self-supervised learning for autonomous driving applications. *NeurIPS*, 2019. 2

[15] Sangjune Laurence Lee et al. Uw-madison gi tract image segmentation. https://www.kaggle.com/competitions/uw-madison-gi-tract-image-segmentation/. 1

[16] Andreanne Lemay, Charley Gros, Zhizheng Zhuo, Jie Zhang, Yunyun Duan, Julien Cohen-Adad, and Yaou Liu. Automatic multiclass intramedullary spinal cord tumor segmentation on mri with deep learning. *NeuroImage: Clinical*, 31:102766, 2021. 2, 8

[17] Baijiong Lin, Feiyang Ye, and Yu Zhang. A closer look at loss weighting in multi-task learning. *CoRR*, abs/2111.10603, 2021. 8

[18] Kevin Lu. Kevinlu1211/pytorch-unet-resnet-50-encoder. https://github.com/kevinlu1211/pytorch-unet-resnet-50-encoder. 3

[19] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[20] Vishal Rajput. Robustness of different loss functions and their impact on networks learning capability. *CoRR*, abs/2110.08322, 2021. 4

[21] Stanford ML Group. Stanford ml group starter code base. https://github.com/stanfordmlgroup/starter. 5

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 3, 4

[23] Darien Schettler. Uw-madison gi tract image segmentation - eda. 3

[24] Song-Toan Tran, Ching-Hwa Cheng, Thanh-Tuan Nguyen, Minh-Hai Le, and Don-Gey Liu. Tmd-unet: Triple-unet with multi-scale input features and dense skip connection for medical image segmentation. In *Healthcare*, volume 9, page 54. Multidisciplinary Digital Publishing Institute, 2021. 2

[25] Dewen Zeng, Yawen Wu, Xinrong Hu, Xiaowei Xu, Haiyun Yuan, Meiping Huang, Jian Zhuang, Jingtong Hu, and Yiyu Shi. Positional contrastive learning for volumetric medical image segmentation. In Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 221–230, Cham, 2021. Springer International Publishing. 2, 5

[26] Dewen Zeng, Yawen Wu, Xinrong Hu, Xiaowei Xu, Haiyun Yuan, Meiping Huang, Jian Zhuang, Jingtong Hu, and Yiyu Shi. Positional contrastive learning for volumetric medical image segmentation. *CoRR*, abs/2106.09157, 2021. 2, 5