# Application of Convolutional Neural Networks in Gastro-Intestinal MRI Image Segmentation

Mingyu Zhong, Ryan Donoghue, Rongzhi Zhang, Ruide Li, Gabriel Vieira
{mingyuz, rdonoghu, peterzrz, ruideli, gvieira}@umich.edu

April 2023

## 1 Abstract

Convolutional Neural networks for the purpose of image segmentation have a variety of applications within the medical field. The application our group explored was the identification and positioning of vital organs based on MRI images in patients undergoing radiation treatment for cancer of the GI tract. The automation of this process promises to both increase the efficacy and safety of radiation treatment as well as decrease total patient time in the hospital. The data is given in the form of MRI scans composed of around 150 16-bit gray-scale PNGs. The intended output is likewise a corresponding image mask outlining the stomach, small intestine, and large intestine. A multi-stage pipeline of convolutional neural networks was employed in order to handle the tasks of image classification and segmentation.

## 2 Introduction

Radiation therapy uses targeted radiation in an attempt to kill cancerous cells in a patients body. Over half of the five million diagnosed with cancer of the GI tract globally, each year are eligible for this line of treatment. The radiation is used for around fifteen minutes a day, every day for up to six weeks.

A major practical issue with radiation therapy is that it requires the mapping of the positions of the tumour, stomach, small intestine, and large intestine in order to minimise the exposure of vital organs to harmful radiation. The tumor as well as these organs can change postion daily meaning every step in the treatment process must be repeated at each administration. Currently many steps in this process are done manually by the administering doctors using Integrated Magnetic Resonance Imaging and Linear Accelerator Systems (MR-Linacs) in a painstaking process that can increase the treatment time from fifteen minutes to more than an hour each day.

Convolutional neural networks can automate this process and decrease the duration of regular visits for the patient as well as remove the possibility of human error during the outlining process. The success of the automation, therefore, would greatly decrease the discomfort of patients in a very stressful situation and could be potentially lifesaving in extreme circumstances.

## 3 Methods

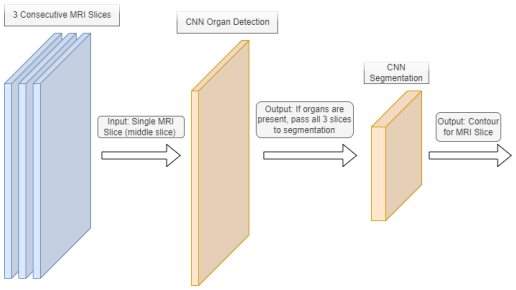
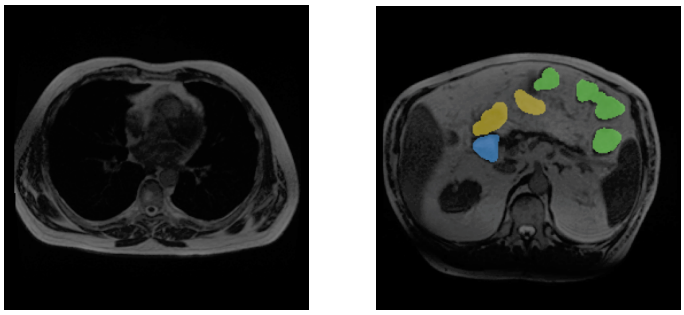



Figure 1: Implemented method pipeline

The problem of MRI image segmentation for radiation therapy was divided into 2 tasks as show in Figure 6: Detection of the gastrointestinal tract to determine the presence of desired features; and a data processing and segmentation stage to outline the stomach, large intestine, and small intestine. This modular structure allowed for the comparison of several machine learning algorithms within the same architecture. The preprocessing stage decreases the total time necessary to train the segmentation model.

### 3.1 Organ Detection



(a) Scan with No Stomach, small intestine, or tumor present

(b) Scan with organs present

Figure 2: MRI Scans from Case 123, Day 20

The data set is a series of MRI images for 85 cases each composed of 1-5 scans taken on different days of radiation treatment. Each scan is a ∼150 individual 16-bit grayscale PNG images as can be seen in Figure 2. These images do not always contain sections of the body where the stomach, large bowel, and small bowel are present (Figure 2a). There are additionally some cases and scans that do not contain any data and are just black images. Processing these images to try and find contours would dramatically increase training time and needlessly waste resources; therefore, a preprocessing stage was implemented using a Convolutional Neural Network (CNN) to prune the data set and remove superfluous images.

A CNN was used due to its high initial classification accuracy and superior recall. These metrics are elaborated on in Section 5. The base model CNN (3-Layer) had a significantly higher classification recall compared to the SVM methods we tested. Optimizing recall over accuracy allowed the method to maximize the amount of images that are passed to subsequent stages. It was more important that the model never miss an image that does contain useful information than it was for it to never pass in unusable slices. This led to our decision to use the 2-layer convolutional Neural Network (CNN) which had the highest recall overall.

#### 3.1.1 Architecture

The CNN consists of six layers with two layers used for data size reduction and shape manipulation as can be seen in Figure 3. The initial two layers are used to reduce the data size via a 2D convolution and max pooling algorithms. Without these layers, this initial image classification problem quickly becomes intractable as Many of the images we are processing are 266 x 266 pixels or larger giving us a dimensionality of 70,756. An input vector of this size for a neural network would necessitate a large amount of resources to train and subsequently use which is not feasible both in development and application. Additionally, a final max pooling and flattening layer is used to further reduce our dimensionality and turn our 2D vector into a 1D vector that can act as our input for the final fully connected (dense) layers. These were used to finally classify an image as having the desired organs present. The images that contain the desired organs are subsequently passed to the second stage of our pipeline.

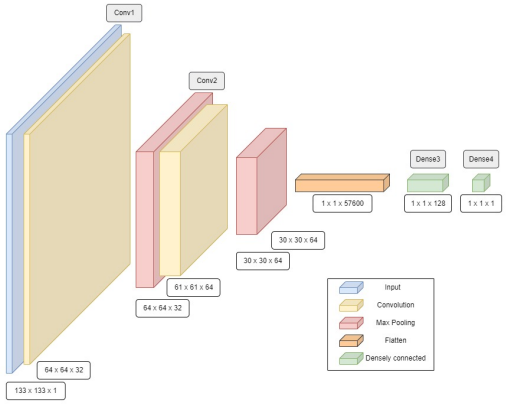



Figure 3: CNN Architecture for organ detection. Sizes denote output size. Names taken from Kera.

### 3.2 Segmentation

In the previous task, MRI slices that do not contain the target organs are filtered out. In the following section a Fully Convolutional Neural Network is used to segment an image into a binary mask.

#### 3.2.1 Overview

To achieve this goal, the model will need to output an arbitrary size set of pixels that represent the organ in the slice image. This is difficult to accomplish by using traditional CNN network that outputs a one-hot vector of classes. Therefore, A U-net model structure was adopted that utilizes an encoder and a decoder using CNN layers as can be seen in 5. The encoder stage down-samples the input image into a number of hidden features. Following, this the decoder gradually restores the dimensions back to the size of the original image. This method outputs a mask

with the same dimension as the input image. The value of each point on the mask represents the probability that its corresponding pixel is a part of the target organ in the input slice. A threshold value is used to turn these probabilities to a dichotomous (0, 1) mask as the final segmentation output.

### 3.2.2 2.5-Dimensionality

This naive U-net model will take a 2D MRI slice as input and output its corresponding 2D mask. However, this is not sufficient in the context of this problem. Our data set contains slices of MRI scans for each patient, date pair. The internal connections between these slices can be exploited to give the model an understanding of the spatial relationship of consecutive organ slices.

A simple solution to solve this issue is to change the input and output from 2D to 3D. Instead of passing in each individual valid slice as input, the model is given information from all valid slices in the scan. With this change, the parameters in the model will be able to learn the "position" of each slice in the entire scan and return better results based on the additional information encoded in the surrounding slices.

While the 3D U-net does resolve the previous problem, it requires a larger number of parameters to fit and additional computational power to train and implement. This is problematic as the data set would only contain $\sim$ 400 training examples. Given the large number of parameters and general model complexity, it is not feasible to train a 3D U-net on this small of a sample size. Instead, a 2.5D data processing method that exploits the inter-slice relationship while still preserving a relatively small parameter size in the model was used. For each individual slice, it is sufficient to use the prior and posterior slices to encode the inter-slice positional information. These three slices were compressed into one input image with three channels. While this does increase the number of training parameters slightly it also allows the model to learn inter-slice relationships and maintains the size of the training set allowing the model to still be trained effectively.

### 3.2.3 Architecture

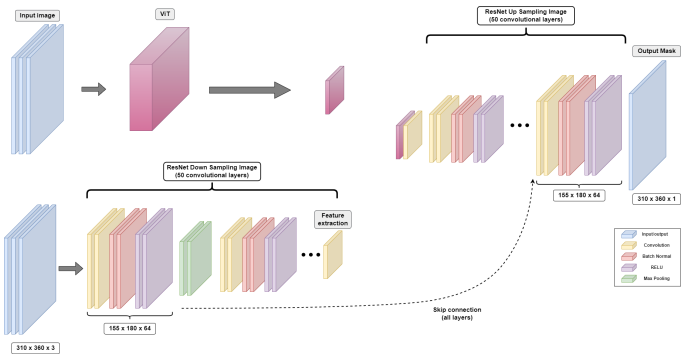



Figure 4: ResNet-50 with ViT for image segmentation.

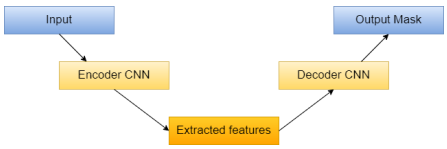



Figure 5: ResNet-50 with ViT for image segmentation.

The final model consisted the ResNet-50 CNN coupled with a ViT as can be seen in Figure 4. The ResNet-50 model was chosen due to the availability of smaller models such as ResNet-34 and larger models like ResNet-101. This allowed for easy transition to smaller or larger models should any under or over fitting be experienced during training. The ResNet architecture also utilizes skip connections as part of the decoding step. These connections allow for each layer of the decoder to include both global feature extraction from the previous transpose convolution and local feature extraction from the corresponding encoder layer.

Additionally a ViT is used on the input image and stacked with the final feature extraction from the ResNet-50. A ViT is a form of transformer that allows attention to be focused on specific sections of an image in order to facilitate feature extraction. They operate similar to traditional attention based models; however, they are geared specifically toward pixel relationships lending to the term Vision Transformer. This transformer provides additional information to the decoding layers. During testing it was discovered this additional information provided slightly better testing results compared to ResNet-50 or ViT individually. While the additional complexity does increase the training time, as now both the ViT and ResNet-50 need to be trained, implementation did not experience any significant hindrances to performance. Since both individual approaches were under development already for compari-

son purposes, the combined model was chosen due to its higher performance.

#### 3.2.4 Loss Functions

A combined loss function was used in determining the performance of the model. The individual losses used were the focal loss and Tversky loss. Both of these are modification of the standard loss functions, cross-entropy and dice loss respectively. Focal loss allows the model to adjust the focus of its training to prioritize harder to segment images. It outputs the pixel to pixel difference between the true and predicted masks. Additionally Not every image being input has all three organs to segment leading to there being more training examples for the large intestine than there are for the stomach. Tversky loss allows this class imbalance to be addressed. This loss function is a modification of dice loss and as such aims to maximize or minimize the amount of overlapping shapes between the true and predicated mask. Tversky loss however is known to have difficulty with harder to classify images, which led to the use of the combined loss function. The equations for all three loss functions can be seen in Equations 1-3 where $TP$ refers to the number of True positives, $FP$ refers to the number of False Positives, $FN$ refers to the number of False Negatives, and $bce$ refers to the cross entropy loss. $\alpha$ and $\gamma$ are tuning parameters for the Focal Loss funciton.

$$\mathcal{L} = 0.33 * FL + 0.67 * TL \tag{1}$$

$$FL = mean(\alpha(1 - e^{-bce})^{\gamma} * bce) \tag{2}$$

$$TL = \frac{TP}{TP + \alpha FP + \beta FN} \tag{3}$$

#### 3.2.5 Training

The ResNet-50 model was trained in two stages due to the size of the model. First the down sampling layers were frozen while the model trained the up sampling stages. The down sampling layers were capable of reasonable feature extraction to begin with and could be held as they were in order to train the up sampling layers. In the second stage the entire model was unfrozen and trained. Since the up sampling stage was already trained, little work needed to be done to adjust any of the weights. The down sampling however was designed for RGB images not the three consecutive gray scale images used as input. As a result, majority of the training in this stage was for the down sampling layers.

## 4 Related Works

One of the most relevant related works for is the Deep Learning MBIR (DL-MBIR) algorithm mentioned in ***2.5D Deep Learning For CT Image Reconstruction Using a MULTI-GPU Implementation***. The DL-MBIR method is trained to produce reconstructions that approximate true MBIR images using a 16-layer residual convolutional neural network implemented on multiple GPUs using Google TensorFlow. The core of this algorithm is the conversion from 3D to 2.5D and 2D similar to the second pipeline stage. DL-MIBR aids in understanding the methods and subsequent advantages of transformation. The structured convolution kernel of the 2.5D model it builds is 2D, and the kernel size is 3x3. MBIR only needs to output one slice after inputting 3 adjacent FBP reconstruction slices. With a sliding window of three input slices moving in the z-direction, the entire 3D output can still be produced. In practice, even if three input slices are used as input the computation time is not significantly affected. This can not only retain the advantages of the 3D model but also greatly save time and cost.

Although GANs can produce clearer and more realistic samples in the fields of unsupervised learning and semi-supervised learning, training GANs needs to reach a Nash equilibrium, which is difficult to achieve with gradient descent. In addition, GANs have problems such as unstable training and gradient disappearance. Therefore, comparatively, our group prefered the usage of CNN in the field of image processing. It supports the sharing of convolution kernels, which is efficient and easily implemented when processing high-dimensional data into desired target features.

Our approach is somewhat similar to the DL-MIBR algorithm mentioned above. Our model was constructed on the basis of CNN and utilized the conversion a 3D composed of 2D slices into a 2.5D image. In addition, a number of adjustments were made to better fit the data set. In addition to the U-net structure, an identifier was implemented in step 1 to distinguish between valid and invalid slices using a CNN. This avoids redundant subsequent image preprocessing and prevents us from feeding minimally useful data into the U-net model in subsequent steps.

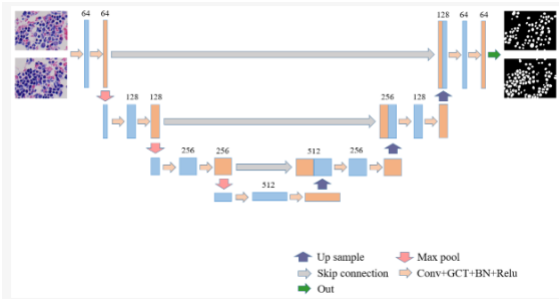



Figure 6: The structure of our proposed model of others' work

For image classification tasks, architecture using Vision Transformers (ViT) are starting to be used. In these circumstances, the ViT model is usually utilized as a standalone architecture without any further processing. In Vision Transformers, the input image is split into fixed-size patches, which are then linearly embedded and processed using the self-attention mechanism built into the transformer architecture. An example implementation with U-net transformation can be found here: ***GCT-UNET: U-Net Image Segmentation Model for a Small Sample of Adherent Bone Marrow Cells Based on a Gated Channel Transform Module***. the paper discusses how a U-net structure similar to UNET is used for image segmentation.

The ViT model is employed as an encoder in a segmentation task, something that not typically done with ViT alone. The U-Net, a well-liked segmentation model with an encoder-decoder structure, served as an inspiration for this architecture. The following are the primary distinctions between the architecture employed here and a conventional U-Net.

1. Encoder: A pre-trained ViT model is used as the encoder in place of the U-Net's set of convolutional layers. As a result, the model may make use of the sophisticated features that the ViT learned during pre-training.

2. Decoder: Similar to the decoder in U-Net, this component upsamples the feature maps to their original input size using a sequence of transpose convolution layers. The primary distinction is that this model does not make use of the skip connections seen in the conventional U-Net.

3. Adaptation layers: The feature maps between the decoder's various layers are resized using extra adaptive average pooling layers. This is required because the size of the feature maps generated by the ViT encoder differs from the feature maps in a conventional U-Net.

In conclusion, our model uses the ViT coupled with the standard U-net convolutional layers as an encoder for feature extraction. This allows for the inclusion important elements of the U-Net design, such as skip connections, particularly in the decoder portion. The performance of the model in semantic segmentation tasks may be enhanced by combining ViT and a decoder inspired by U-Nets.

# 5 Experimental Result

## 5.1 Stage 1: Classification of the Presence of Interested Organ

| Classification Networks | SVM (Sigmoid) | SVM (RBF) | SVM (Poly) | CNN 2-L | CNN 3-L | CNN 4-L |
|---|---|---|---|---|---|---|
| T = 0 & P = 0 | 11647 | 14876 | 14856 | 14494 | 14471 | 14320 |
| T = 0 & P = 1 | 3288 | 59 | 79 | 441 | 464 | 615 |
| T = 1 & P = 0 | 3350 | 2672 | 2673 | 715 | 805 | 745 |
| T = 1 & P = 1 | 964 | 1642 | 1641 | 3599 | 3509 | 3569 |
| Accuracy | 0.65515 | 0.85812 | 0.54391 | 0.93994 | 0.93407 | 0.92934 |
| F1-Score | 0.22507 | 0.54596 | 0.85703 | 0.86162 | 0.84686 | 0.83996 |
| Recall | 0.22345 | 0.38062 | 0.38038 | 0.83426 | 0.81339 | 0.82730 |

Table 1: Training results on training set for different classification models

| Final Classification Network | Threshold = 0.5 | Threshold = 0.0009 |
|---|---|---|
| T = 0 & P = 0 | 7250 | 4687 |
| T = 0 & P = 1 | 120 | 2683 |
| T = 1 & P = 0 | 237 | 7 |
| T = 1 & P = 1 | 2788 | 3018 |
| Accuracy | 0.9657 | 0.7412 |
| F1-Score | 0.9398 | 0.9398 |
| Recall | 0.9217 | 0.9977 |

Table 2: Training results on testing set for final classification model with different threshold

| Segmentation Networks | AUC | Accuracy | F1-scores | Recall | Specificity | Dice Loss | BCE Loss |
|---|---|---|---|---|---|---|---|
| 2D ResNet | 0.7486 | 0.9948 | 0.6432 | 0.4977 | 0.9995 | 0.3491 | 0.0417 |
| 2.5D ResNet Without Skip | 0.7758 | 0.9952 | 0.6854 | 0.5522 | 0.9994 | 0.3095 | 0.0353 |
| 2.5D ResNet Model | 0.7880 | 0.9955 | 0.7124 | 0.5764 | 0.9996 | 0.2845 | 0.0349 |
| ViT Model | 0.5004 | 0.9903 | 0.0019 | 0.0010 | 0.9997 | 0.6589 | 0.2156 |
| ResNet & ViT Model | 0.7416 | 0.9943 | 0.6207 | 0.4839 | 0.9993 | 0.2476 | 0.0366 |

Table 3: Training results on testing set for different segmentation models

### 5.2 Stage 2: Mask Segmentation of Interested Organs

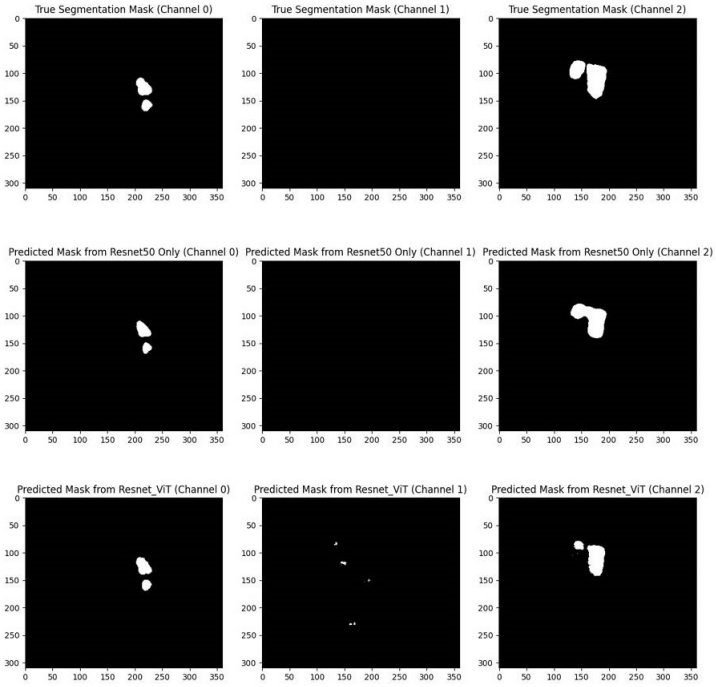



Figure 7: Example segmentation results from our model

## 6 Conclusion

From our experimental data, we could see some satisfactory results from our proposed model and its advantages over the other baseline models we tried.

Ideally the pipeline would be compared to existing architectures capable of handling the provided MRI imaging data. However, there currently exist no well known open source solutions for automatic gastrointestinal MRI image segmentation for varying image sizes. While some industry solutions exist for MRI image segmentation, they are a paid software that our group was not in the position to experiment with. Additionally, many of these solutions require doctors to still provide a rough outline of the desired organ. Unfortunately, the only other solutions to this problem are submissions into a kaggle competition, which our

group does not have access to.

### 6.1 Classification

Table 2 shows the performance comparison for the models of stage 1. For the row names, T refers to the true label, p refers to the predicted label, 1 indicates the presence of organ, and 0 indicates the absence of organ. The columns are SVM models with different kernels and CNN models with different number of hidden layers.

In the overall pipeline, the Stage 1 classification for the organs of interest is built as a preliminary screening to eliminate some inputs that do not contain any organs of interest. It serves as a preprocessing tool of the data set to prevent some noisy and irrelevant scans from being fed to the segmentation network; yet, at the same time, we would want the model to not prune away some useful scans from the data set. Therefore, when analyzing the performance, we prioritize on the maximization of the recall while trying to achieve a reasonably small amount of false positives (T = 0 & P = 1). We could see that the CNN 2-L model performs the best in accomplishing this goal with a recall score of 0.83426 and only 441 false positives out of the total 19249 slices (around 2.29%). This is already a reasonable and acceptable metric for our Stage 1 preprocessing considering the fact that the original data set contains around 59% of input slices without any target organs.

Furthermore, to prioritize on the maximization of recall value, the threshhold value could be modified to increase our tolerance of low prediction scores. The decrease of the threshold is viewed as a decrease of the confidence interval of the model, where images are considered to be "organless" only when it is highly confident. From Table 2, we could see that when the threshhold is decreased to a value of 0.0009, the recall value rises to around 0.9977 with only 7 false negatives. While the accuracy score also drops as a result of this decrease in threshold, this tradeoff is still acceptable since this model is designed to only provide a quick and relatively accurate preliminary screening to prevent medical experts from wasting time on irrelavant slices in a MRI scan.

### 6.2 Segmentation

Apart from the ResNet & ViT Model proposed, several other segmentation models were implemented for comparisons.

To observe the effectiveness of the 2.5D approach, a 2D resnet model was used which takes only one slice as input. From Table 2, we could observe that with respect to every metric tested, the naive 2D model is unable to compete with the remaining models (which all use the 2.5D method). In fact, after training the 2D resnet for 50 epochs, the resulting segmentation mask contains only a few, meaningless pixels. This result was surprising considering our implementation utilized a pre-trained Resnet-50 model as our basis, which is known for good feature extractions and semantic segmentation. A possible explanation for this extremely poor performance might be that the task of segmenting gastro-intestinal organs, unlike other larger organs such as the brain and the heart, is extremely complicated as they appear as small ares sporadic spread out on the MRI scan. As such, it showcases the advantages of 2.5D inputs to encode spatial relationships between adjacent slides, which performs much better from the resulting data.

In addition to the use of 2.5D input, the model also exploits the use of ViT to reflect certain spatial information between organs on the same slide. In many output segmentations from the 2.5D ResNet Model, the mask tends to group together into a large, unified clumps instead of separating into different small parts (as shown in Figure 7, Channel 2). We expected the ViT to encode this spatial information and help the model learn this separation through the self-attention layers in the ViT. As can be seen in Figure 7, the final proposed ResNet & ViT model does accomplish this goal and achieve a lower Dice Loss, which is the most crucial metric in determining the effectiveness of a segmentation model. However, with the incorporation of ViT, the model suffers from a notable increase in variance, which results in the "noisy" white pixels in Channel 1. While we try to downsize our ViT model to decrease potential overfitting in an attempt to address this issue, it does not effectively solve this problem. A possible guess is that with a significant amount of new parameters in the Vit, we will have to train on more data to mitigate the increase in variance.

Aside from some of our success with our proposed model, we encounter some major failures when trying to implement a ViT-only segmentation model. After many epochs of training, the training loss does not seem to converge. A possible reason is that the ViT-only model might need to connect multiple ViT's together to function, while

a single Vit already exhausts all our GPU computation power with its parameters inside.

# 7 Future Work

Overall, the predicted segmentation masks are quite well aligned with the true segmentation masks. It is still possible to improve the performance of the segmentation for the masks with the small regions since the predicted mask is smaller than the true mask as the area of the true mask decreases. The methods for improvement are stated in an increasing workload order.

1. Adapting 2.5D Input Data - The input data includes the spatial information of a given slice with a 2.5D design which includes the layer before and the layer after the current slice. The segmentation results may be improved by incorporating more spatial information by adding consecutive layers of the given slice as input to the network and adjusting the input layer of the segmentation network to the corresponding sizes.

2. Adjusting Loss Function - The hyperparameters of the loss function could be further tuned with a cross-validation approach. The hyperparameters of the customized loss function include the importance weight of focal loss to Tversky loss and the parameters in focal loss and Tversky loss to penalize or pay more attention to a specific class or difficulty of samples. Additionally, if the result after training with the optimal hyperparameters of the combined focal loss and Tversky loss is still not ideal, we can try to incorporate new loss functions such as Jaccard Index and Hausdorff Distance to address the issue. Note that the smaller mask would have a greater error which is a size problem, Jaccard Index (Intersection over Union) is particularly helpful when dealing with vary-sized segmentation. Also, the current model can correctly identify all regions but cannot yield the exact area of the region, so Hausdorff Distance may help by optimizing the difference between the boundary of the predicted mask and the true mask.

3. Increasing Model Complexity - The pre-trained model used as the backbone for feature extraction may not be complex enough to identify the minor details in the image to yield a good result for the smaller mask. Thus, we can use a more complex pre-trained network than the current one. For example, we can replace ResNet50 with ResNet110 which has 110 convolution layers instead of 50, and replace vit_base_patch16_224 with vit_large_patch16_384 which uses 24 vision transformer units instead of 12 with the capacity to handle larger image with size $384 \times 384$ instead of $224 \times 224$.

4. Adding Additional Segmentation Model to Pipeline - Similar to the idea in step 1 that adding more information may be helpful, the current model may perform better if it has some information about the potential area that it can pay more attention to. Thus, training another independent segmentation model like adding a Stage 1.5 model to produce a mask that identifies the regions of organ and background which could provide the final Stage 2 segmentation model with the correct regions and may simplify the segmentation problem into a classification problem.

# 8 Author Contributions

All co-authors were equally involved in writing this report. All co-authors equally contributed to this project. Rongzhi Zhang researched on the image segmentation methods and proposed the structure of the U-net pipeline. Mingyu Zhong helped 2.5D data set construction and preprocessing, formulated the customized training loss, and built the classification model in stage 1 and the segmentation model in stage 2. Ruide Li did the research on our relevant methods(CNN) and analyzed the similarities and differences between them, implement the Vision Transformer method, and discuss what we are different than others used from the U-net architecture. Ryan Donoghue helped with the High level pipeline construction, researched image reduction techniques, and worked on additional comparative architecture that was unused in this report. Gabriel Vieira contributed to synthesizing the abstract, introduction, and aided in the testing and construction of the ResNet and ViT architecture.

# 9 Reference

Entangled Decision Forests and Their Application for Semantic Segmentation of CT Images. In: Székely, G., Hahn, H.K. (eds) Information Processing in Medical Imaging. IPMI 2011. Lecture Notes in Computer Science, vol 6801.

Springer, Berlin, Heidelberg.
Montillo, A., Shotton, J., Winn, J., Iglesias, J.E., Metaxas, D., Criminisi, A. (2011).
https://doi.org/10.1007/978-3-642-22092-0_16

Semi-supervised semantic segmentation of prostate and organs-at-risk on 3D pelvic CT images
Zhuangzhuang Zhang et al 2021 Biomed. Phys. Eng. Express 7 065023. (2021)
https://iopscience.iop.org/article/10.1088/2057-1976/ac26e8/meta

Characterization of digital medical images utilizing support vector machines.
BMC Med Inform Decis Mak 4, 4 (2004). Maglogiannis, I.G., Zafiropoulos, E.P.
https://doi.org/10.1186/1472-6947-4-4

2.5D Deep Learning For CT Image Reconstruction Using a MULTI-GPU Implementation
Amirkoushyar Ziabari, Dong Hye Ye, Somesh Srivastava, Ken D. Sauer Jean-Baptiste Thibault, Charles A.Bouman (2018)
https://engineering.purdue.edu/ bouman/publications/orig-pdf/2018-Asilomar.pdf

GCT-UNET: U-Net Image Segmentation Model for a Small Sample of Adherent Bone Marrow Cells Based on a Gated Channel Transform Module
Jing Qin,Tong Liu, Zumin Wang, Lu Liu, Hui Fang (2022)
https://www.mdpi.com/2079-9292/11/22/3755

**Data set:**
https://www.kaggle.com/competitions/uw-madison-gi-tract-image-segmentation/overview