GI-Tract Image Segmentation

Simar Atwal, Sruti Dondapati,
Srikar Chunduri, Kenny Wang

Introduction

- Oncologists manually outline healthy organs in MRI/CT scans of the GI tract in a slow, labor-intensive process for targeted radiation treatment
- Kaggle Challenge: Use 2.4 GB of CT scans to train a model to identify different organs using segmentation

- What we've done:
  - Process the data
  - Implement a U-Net
  - Implement a Trans U-Net
  - Analyze the results

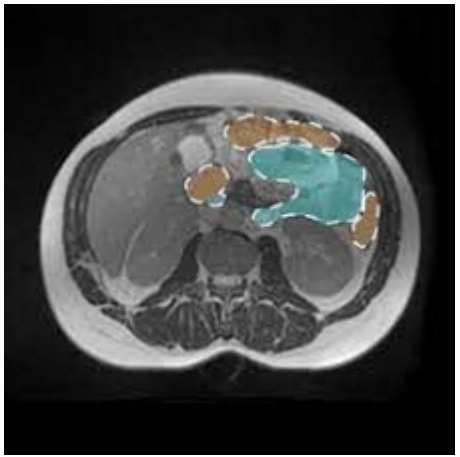

Figure 1: Example CT scan

Methodology: Data Processing

- We needed to process a large set of raw CT slices + RLE masks and convert them into more filtered inputs for our model training
- Decoding the RLEs and build a three channel mask for the three different organs being analyzed, stomach, small bowel, and large bowel
- Applying transformations using albumentations we can resize, normalize, then ToTensor so images and masks can stay together
    - Normalizing helps match pretrained encoder expectations and stabilize training for faster convergence and more consistency in activations
    - ToTensor handels the normalized data which is in (H, W, C) to float32 (C, H, W) tensor for the model
- Keeping them in their own channels allows us to keep the data from cross contamination with each other before we pass it to the training model

Methodology: U-Net

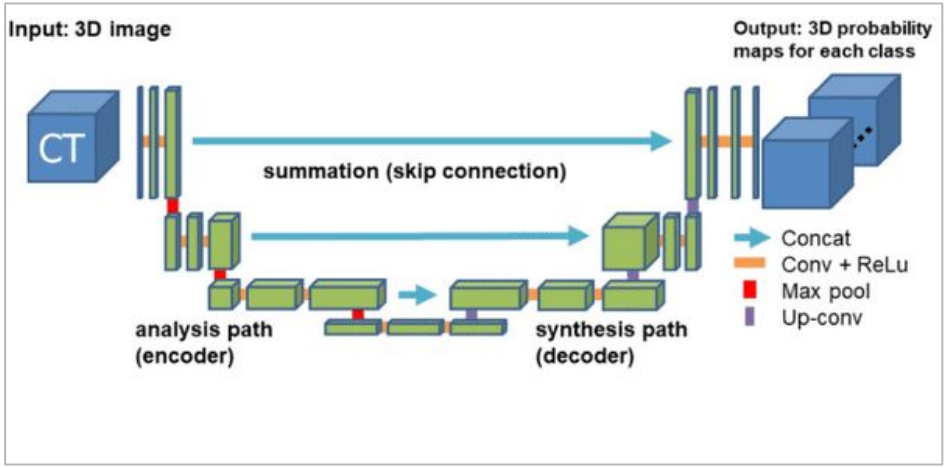

Figure 2: UNet Example Architecture

U-Net is basically a symmetric encoder–decoder network designed specifically for biomedical image segmentation. It takes in a CT scan and outputs a pixel-wise mask showing where each organ is

- **UNet = CNN + skip connections**, skip connections restore the high resolution details
- Encoder: 4 convolutional blocks + max pooling
- Decoder: Upsampling that reverses the encoders downsampling
- Bottleneck: Deep Convolutional feature extractor

Methodology: Trans U-Net

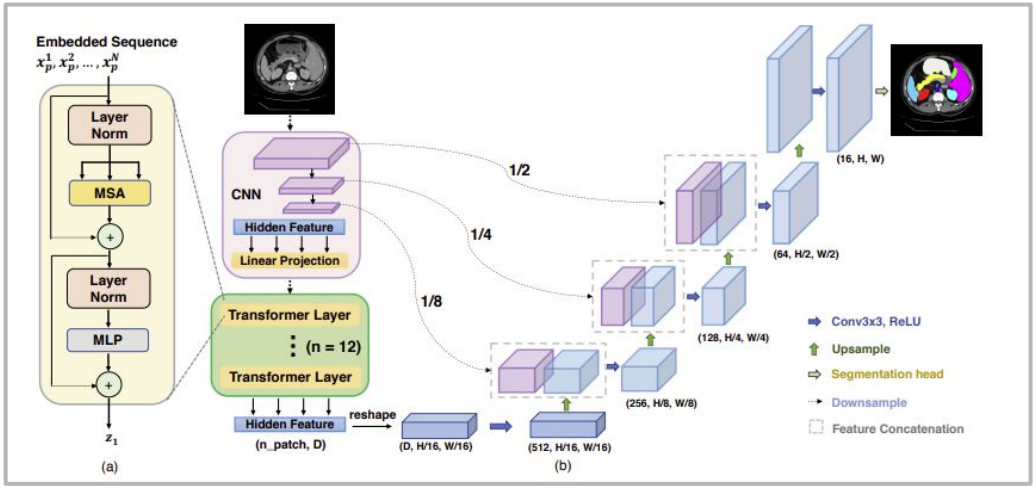

Figure 3: Trans U-Net Example architecture

- **Loss Function:** DiceLoss
- **Optimizer:** AdamW

- **Encoder:** 4 conv blocks
- **Transformer bottleneck:** 12 Transformer Encoder Blocks
- **Decoder:** Mirrors encoder

- **Trans U-Net = U-Net + Transformers**
- Encoder: Simplified CNN --> transformers
  - Capture long-range dependencies through self-attention
- Decoder: transformer output -> CNN decoder

Methodology: Trans U-Net Cont.

- Pitfall: Small dataset leads to rapid overfitting
- Solutions:
  - Pre-Trained Weights → finetune on dataset
  - Try different loss functions
  - Lower Learning Rate
  - Stronger data albumentations
    - Blur, rotations, etc.

Evaluations

- Evaluation Metric: Dice Coefficient
  - The Dice Coefficient measures how much of the predicted segmentation overlaps with the ground truth mask
  - It ranges from 0 to 1: 0 meaning there is a weak overlap, 1 meaning there is a perfect overlap
  - The dice coefficient was a better evaluation metric because it disregards relying on traditional pixel accuracy which would be misleading and less accurate
- Loss Function: Dice Loss
  - Dice Loss = 1 - Dice Coefficient
    - Allowed the model to focus on improving the shape & boundary of the organs rather than being influenced by background pixels during training

Visualizations

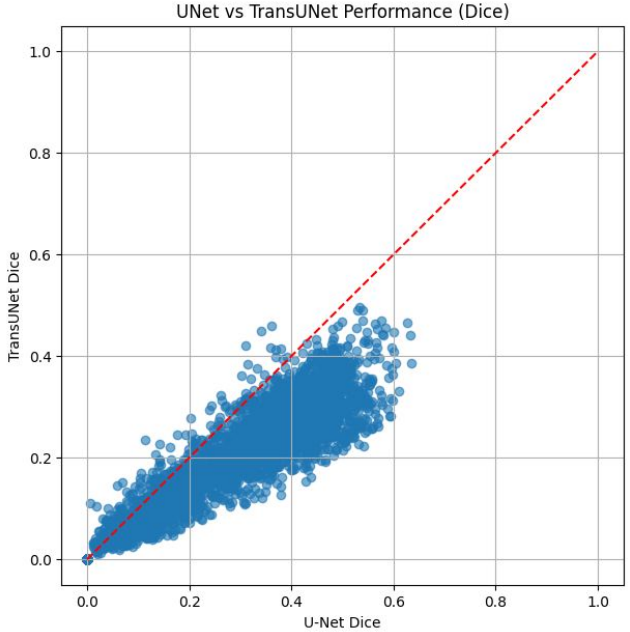

Figure 4: U-Net vs Trans U-Net Dice Comparison (Sruti Dondapati)

- Red diagonal line: areas where both models performed equally
- Points above diagonal: better performance of Trans U-Net model
- Points below diagonal: better performance of U-Net model

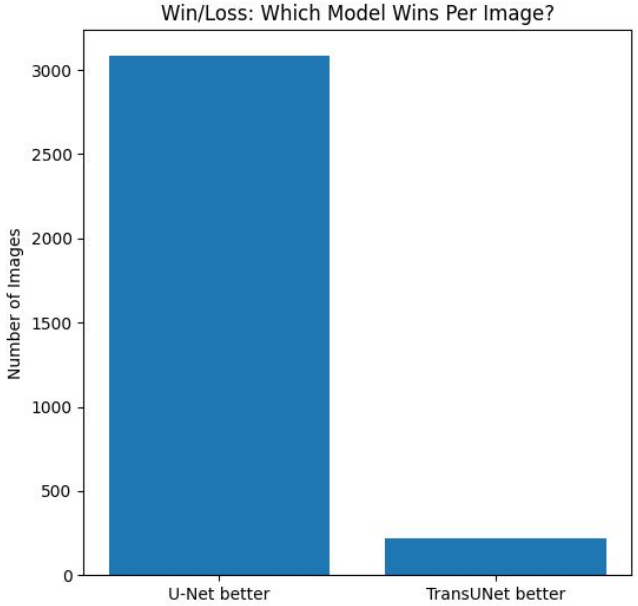

Figure 5: Model Image Performance (Sruti Dondapati)

- The number of images the models performed better on
  - U-Net performed better on ~3000 images
  - Trans U-Net performed better on ~300 images

Results/Analysis

- Overall based on the visualizations & evaluations, the U-Net model performed better than the Trans U-Net model in accurately segmenting the organs
  - Why U-Net outperformed Trans U-Net:
    - Stability with small datasets
      - Trans U-Net requires larger data to train its transformers better
      - Limited training data/examples could cause the model to quickly overfit
    - Training Stability
      - U-Net is easier train since it is less sensitive to hyperparameters and converges more consistently while Trans U-Net can become unstable during training

Future Work

- More filtering of the raw data
  - Preserve the aspect ratio instead of stretching the image into a square
  - Sample some empty-mask slices instead of getting rid of all of them
  - Split by patient case rather than random rows to avoid leakage
  - Add neighboring slices as extra input channels to give the model context
- Larger dataset
  - Look for larger dataset if possible
  - Kaggle limited

Citations

UW Madison, Kaggle. “UW-Madison GI Tract Image Segmentation.” *Kaggle*, 2022, www.kaggle.com/competitions/uw-madison-gi-tract-image-segmentation.

Figure 1. Lee, S.L., Yadav, P., Li, Y., Meudt, J.J., Strang, J., Hebel, D., Alfson, A., Olson, S.J., Kruser, T.R., Smilowitz, J.B., Borchert, K., Loritz, B., Gharzai, L., Karimpour, S., Bayouth, J., Bassetti, M.F., 2024. Dataset for gastrointestinal tract segmentation on serial MRIs for abdominal tumor radiotherapy. Data in Brief 57, 111159. https://doi.org/10.1016/j.dib.2024.111159. UW-Madison GI Tract Image Segmentation . https://kaggle.com/competitions/uw-madison-gi-tract-image-segmentation, 2022. Kaggle.

Figure 2. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509 (2019)

Figure 3. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

Thank you for listening!