Optimizing segmentation methods for feature detection and classification in MRI images

Jin Huang
huangjin@umich.edu
Reva Kulkarni
kulkarnr@umich.edu
Keagan Moo
kgmoo@umich.edu
Zhenjie Sun
zjsun@umich.edu
Julia Rosander
juliaros@umich.edu

April 27, 2023

Abstract

Magnetic resonance imaging (MRI) is one of the predominant non-invasive methods of producing representations of internal physiology. Though MRI is relatively safe and broadly standardized it is also widespread and a lot of images are produced that require expert assessment to be of diagnostic use. A common issue that both experts and their machine learning aides run into when processing large amounts of image data is that most of a given image will not be of interest. Most of the physiology will be, by definition, typical. Thus, the segmentation and classification of image segments can be foundational in increasing the efficiency and accuracy of image analysis. In this project we propose a broad comparison of image segmentation methods to be bench-marked on a known Kaggle data-set using Dice coefficient with the end goal of optimizing performance of feature classification using intersection over Dice Similarity metrics. The code is publicly available on github [1].

Keywords: Segmentation; Feature detection; MRI images;

[1] https://github.com/Jn-Huang/545_project_brain_segmentation

1 Introduction

Machine learning methods are proving useful in the field of medicine to analyze images such as magnetic resonance imaging (MRI). MRI is a medical imaging technique that produces anatomical scans or physiological representations. MRI provides detailed images of the internal structures of the body without using ionizing radiation, making it a safe and non-invasive diagnostic tool. Segmentation with accuracy measurement could provide physicians another metric or measurement for diagnosis. Cancer, tumors and neurodegenerative diseases such as Alzheimer's all utilize MRIs to further investigate disease severity. A machine learning method that could determine regions of the brain impacted would help clinical decision making. Segmentation methods in the past have had limited performance leading to inconsistent detection and lack of confidence by physicians. Sub-par performance and difficult implementation have made the medical field wary of implementing machine learning methods such as these.

In this project, we intend to use multiple machine learning methods, including Convolutional Neural Networks (CNNs) and U-Net, to predict unique features in images. We also test new models, such as SegmentAnything to determine whether newer architectures provide better accuracy and more accessible frameworks. These models will predict segmentation of specific regions of the brain based on disease or tumor depending on the utilized dataset. We hope to use segmentation tools to more accurately predict specific features of each MRI.

2 Proposed Methodology

2.1 Brain Tumor Segmentation Dataset

We are using the Brain Tumor Segmentation dataset in our study. The dataset [Buda et al., 2019] consists of preoperative imaging and genomic data from 110 patients with lower-grade gliomas from The Cancer Genome Atlas (TCGA) and The Cancer Imaging Archive (TCIA). Fluid-attenuated inversion recovery (FLAIR) is annotated and verified by board eligible radiologists. The dataset with manual segmentation is made public available on Kaggle[2].

2.2 Data Processing Pipeline

The large volumes of complex and multidimensional data produced by MRI scans require efficient and accurate data processing pipelines for optimal analysis and interpretation. Figure 1 shows our pipeline on data processing, which comes from [Buda et al., 2019] with modification on postprocessing.

2.2.1 Preprocessing

Preprocessing is an essential step in the MRI data processing pipeline to ensure data quality and reduce noise, artifacts, and other potential sources of error. Prior to analysis, the images were preprocessed using the following steps:

1. scaling them to a standard frame of reference
2. adjusting the window and level based on the image histogram to standardize tissue intensities across cases

[2] https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation

3. normalizing the entire data set using Z-scores
4. argumentation including scaling, rotating and horizontal flipping for training set only

2.2.2 Segmentation

Preprocessed data is then segmented by different algorithms, which will be discussed in the next section. Segmentation performed by human experts serves as the ground truth label.

2.2.3 Postprocessing

After segmentation, In order to increase the precision, we utilized a post-processing technique that eliminates incorrect identifications. We accomplished this by applying a connected components algorithm to segmentation mask for each patient. We only included connected tumor volume larger than 5 pixels in the final segmentation mask. This post-processing method improves the extraction of shape features since they are affected by singular incorrect pixel segmentation. This step is different from [Buda et al., 2019], in which they only choose the largest tumor volume as the final prediction. We found that keeping multiple areas improves accuracy.

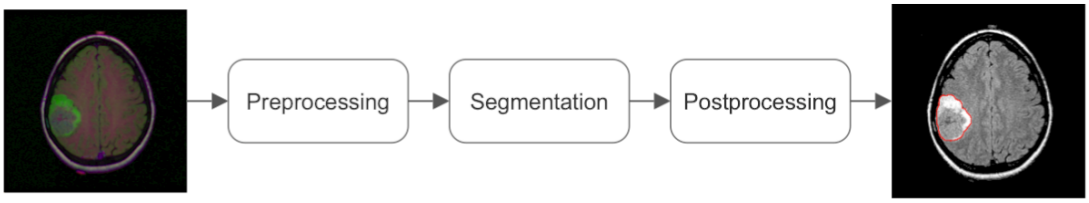

Figure 1: Data processing pipeline of MRI image

2.3 Metric

Dice similarity coefficient (DSC) is a measure of similarity between two sets of data, often used in medical image segmentation tasks.

The equation for Dice similarity defined on two sets A and B can be written as:

$$DiceSimilarity(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{1}$$

In this equation, A and B are two sets of elements (e.g., pixels or voxels), and $|A|$ and $|B|$ denote the cardinality (number of elements) of the sets. The intersection of the sets ($A \cap B$) represents the number of elements that are common to both sets.

The Dice similarity calculates the ratio of the overlap to the total number of elements in the sets. The Dice similarity is also commonly used in image segmentation tasks, where A and B represent the predicted and ground truth segmentation masks, respectively. The Dice similarity measures the similarity between the predicted and ground truth segmentation and can be used as a metric to evaluate the segmentation performance. The higher the value of the Dice similarity, the better the segmentation performance.

For the loss of function, we simply use Dice loss on two sets A and B,

$$DiceLoss(A, B) = 1 - \frac{2|A \cap B|}{|A| + |B|} \tag{2}$$

3 Related work

3.1

In the 2019 paper introducing this processing pipeline for the kaggle dataset. The authors use U-Net segmentation to identify tumor features to make tumor subtype predictions [Buda et al., 2019]. Our method follows a similar structure but aims to expand on the analysis by improving the image segmentation method by adding attention to the U-Net architecture and comparing it to more complex architecture that can incorporate 3D information such as MeshNet. This allows our work to be comparable to past methods on the same data set via Dice and IoU metrics while still supporting substantial experimentation with segmentation architecture. We have chosen this structure to improve interpretability. Previous methods classify whole images, even with great accuracy, are complicated with the difficulty of justifying these results in a clinical or biological context [Veeramuthu et al., 2022]. Image segmentation allows for important review of which parts of the MRI are informing the algorithm. Our method aims to improve both the accuracy of MRI classification algorithms and the intrepretability of the classifications once they are made.

3.2 Data Pipeline

Several studies have proposed and evaluated different pre-processing techniques, including:

a) Motion correction: Due to the sensitivity of MRI to motion, algorithms have been developed to correct for head movement during the acquisition process [Godenschweger et al., 2016]. These techniques use rigid body transformations to realign the MRI data to a common reference frame, thus minimizing the impact of motion artifacts.

b) Spatial normalization: To facilitate group analyses and comparisons, MRI data is often spatially normalized [Mangin et al., 2016]. This involves non-linear warping and resampling of the individual MRI data to match the template.

c) Intensity normalization: To account for variability in signal intensity across subjects and scanners, intensity normalization techniques have been proposed to standardize the overall signal intensity or contrast within the MRI data [Shah et al., 2011].

We adopt both normalization in our pipeline and defer motion correction for future work.

4 Experimental results

The three main models we trained and tested are the U-Net, Segformer, and SegmentAnything. We previously tested several baseline models such as a basic CNN and MeshNet (see A.1) to inform these key model comparisons. We chose these models because U-Net has historically been used for general segmentation, Segformer uses state-of-the-art vision transformers instead of simple convolutional layers, and SegmentAnything was recently released and would be easily accessible by physicians.

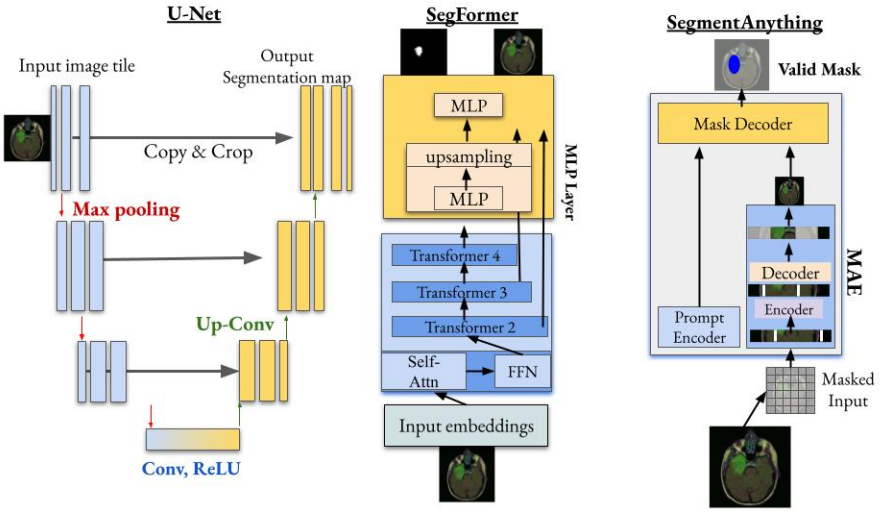

Figure 2: Segmentation Architecture Model Comparison with encoder (blue) and decoder (yellow) frameworks

4.1 U-Net

U-Net is first introduced to perform segmentation on biomedical image datasets [Ronneberger et al., 2015]. The name "U-Net" comes from its characteristic U-shaped architecture, which consists of a contracting path that captures context and a symmetric expanding path that enables precise localization. The contracting path is composed of several convolutional and max-pooling layers, while the expanding path consists of up-convolutional and concatenation layers. Figure 3 shows the structure of U-Net in [Buda et al., 2019], which we use in our study.

We trained a U-Net as the segmentation method in our data pipeline. we observe that U-Net report a average dice similarity coefficient of 91% on ten randomly chosen patients.

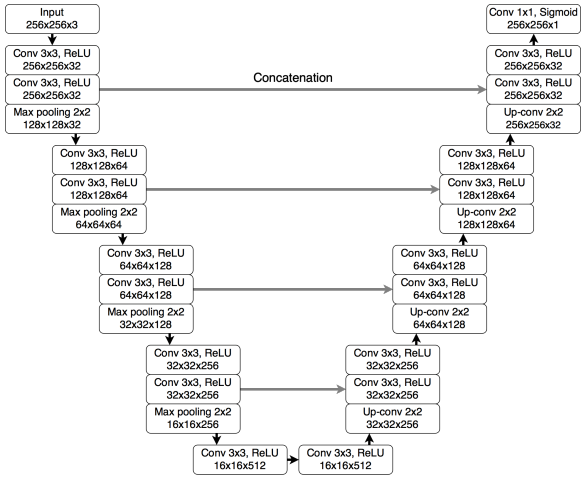

Figure 3: Structure of U-Net

4.2 Segformer

SegFormer Xie et al. [2021] is a novel segmentation architecture that combines transformers with a multi-scale design. Like U-net, SegFormer is also an encoder-decoder structure, but it uses a transformer-based backbone, specifically the Multi-scale Vision Longformer (MViT), as its encoder. The MViT backbone consists of multiple stages, each with self-attention layers and feedforward networks, which efficiently capture contextual information at different scales. The decoder in SegFormer consists of multiple parallel MLP-heads that are designed to work at different resolutions. Each MLP-head processes the feature maps from the corresponding stage of the MViT encoder and generates segmentation predictions. Finally, the predictions from all MLP-heads are upsampled and fused together to obtain the final segmentation output. This multi-scale, transformer-based architecture enables SegFormer to learn effective representations and achieve superior performance in a variety of segmentation tasks.

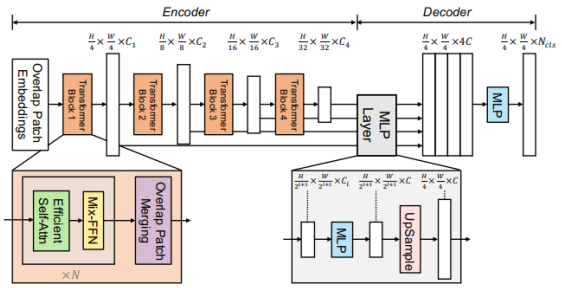

Figure 4: Encoder-decoder Architecture of SegFormer

In our study, we conduct experiments with both pretrained and randomly initialized SegFormers to compare their performance. The pretrained SegFormer is a model that has been previously trained on the ADE20k dataset, a large-scale dataset for semantic segmentation that contains diverse scenes from everyday environments. This allows the model to learn a wide variety of features and structures, which can be beneficial when transferred to other segmentation tasks. On the other hand, the randomly initialized SegFormer starts with weights that are set randomly before the training process. This version of SegFormer does not have any prior knowledge from previous training and must learn all features from the ground up during the training on the target task. As expected, the performance of the pretrained SegFormer surpasses that of the randomly initialized model in our tests. As illustrated in Figure 5, the dice loss for the pretrained variant on the test dataset is substantially lower, confirming its superior efficacy.

4.3 Segment Anything Model (SAM)

The SAM (Segment Anything Model) by Meta AI is a novel image segmentation model that can transfer zero-shot to new image distributions and tasks, requiring point(s) or mask(s) as prompts for segmentation. In Figure 5, we experimented with center prompts and tumor-based prompts. While center prompts only capture the overall brain outline, providing no assistance in tumor detection, tumor-based prompts offer improved tumor capture. However, the latter may still produce inaccurate segmentation results (refer to the final figure in row 1).

Fine-tuning the SAM model for specific tumor segmentation tasks is a promising approach to utilize

its pre-trained capabilities. However, as SAM has not yet released model details, we defer this to future work. We also recognize SAM's potential for enabling real-time segmentation in hospitals, as demonstrated on their website[3].

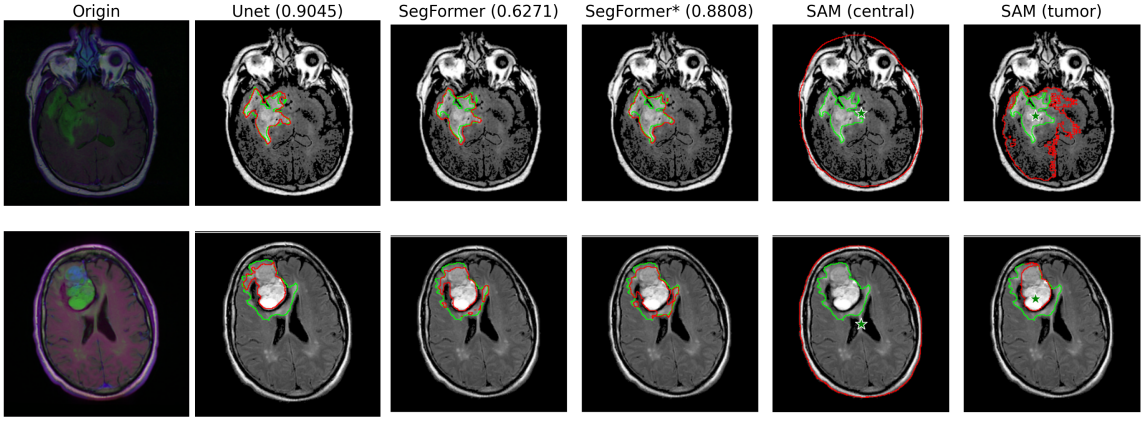

Figure 5: Mask comparison between ground truth (green) and prediction (red) [DSC].
*:fine-tuned SegFormer on ADE20k; SAM result is disregarded b/c of poor performance

5 Conclusion

Through extensive research and experimentation we have gained a comprehensive understanding of the state-of-the-art techniques for brain MRI segmentation. We designed and trained specifically the U-Net, Segformer, and SAM on small datasets to evaluate their performance and identify effective techniques. From our experiments, we see that U-Net achieves the highest accuracy (dice similarity) while SAM achieves the lowest. While this may suggest more reliable continued use of the U-Net for brain MRI segmentation, there are several steps we can take below to further improve both accuracy and physician accessibility.

Moving forward, given the recent development and ease-of-use of SAM, one step is to fine-tune this model on specifically brain MRI data once a public API has been released. Additionally, we began to implement a simple modified version of the Segformer using a SwinT (shifted windows) backbone instead of ViT (see A.1.4), as this allows less computation time and potentially higher performance. Continuing to test and develop this SwinT model would thus also be beneficial.

Effective segmentation methods within the field of medicine could be greatly utilized. Recently, surgeons have implemented machine learning methods to plan surgically for tumor removal and other operations. Segmentation methods can portray clean borders of tumors in the brain and even help classify type. Using segmentation methods would also provide another measurement to help clinical decision making. Improving the accuracy of models will increase the confidence of medical practitioners and physicians in machine learning applications.

[3] https://segment-anything.com

6 Author Contributions

J.H. established a structured pipeline for handling data and carried out a test using Unet and SAM. R.K. implemented CNN and custom SwinT model, and wrote conclusion. K.M. condensed the related literature and composed the abstract. J.R. tested MeshNet and created the introduction. Z.S. implemented Unet with attention and the Segformer. All co-authors equally contributed to the project.

Name	Abstract	Intro	Method	Review	CNN	Unet	SAM	MeshNet	SwinT	Segformer	Conclusion
J.H.			✓			✓	✓				
R.K.					✓				✓		
K.M.	✓			✓							
J.R.		✓						✓			
Z.S.										✓	✓

Table 1: Contribution Table

References

Mateusz Buda, Ashirbani Saha, and Maciej A. Mazurowski. Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. *Computers in Biology and Medicine*, 109:218–225, 2019. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2019.05.002. URL https://www.sciencedirect.com/science/article/pii/S0010482519301520.

A. Veeramuthu, S. Meenakshi, G. Mathivanan, Ketan Kotecha, Jatinderkumar R. Saini, V. Vijayakumar, and V. Subramaniyaswamy. Mri brain tumor image classification using a combined feature and image-based classifier. *Frontiers in Psychology*, 13, 2022. doi: 10.3389/fpsyg.2022.848784. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2022.848784/full.

F. Godenschweger, U. Kägebein, D. Stucht, U. Yarach, A. Sciarra, R. Yakupov, F. Lüsebrink, P. Schulze, and O. Speck. Motion correction in mri of the brain. *Physical Medicine and Biology*, 2016. doi: 10.1088/0031-9155/61/5/R32. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4930872/.

L.-F. Mangin, J. Lebenberg, S. Lefranc, N. Labra, G. Auzias, M. Labit, M. Guevara, H. Mohlberg, P. Roca, P. Guevara, J. Dubois, F. Leroy, G. Dehaene-Lambertz, A. Cachia, T. Dickscheid, O. Coulon, C. Poupon, D. Rivière, K. Amunts, and Z.Y. Sun. Spatial normalization of brain images and beyond. *Medical Image Analysis*, 2016. doi: 10.1016/j.media.2016.06.008. URL https://www.sciencedirect.com/science/article/pii/S1361841516300858?via%3Dihub.

Mohak Shah, Yiming Xiao, Nagesh Subbanna, Simon Francis, Douglas L. Arnold, D. Louis Collins, and Tal Arbel. Evaluating intensity normalization on mris of human brain with multiple sclerosis. *Medical Image Analysis*, 2011. doi: 10.1016/j.media.2010.12.003. URL https://www.sciencedirect.com/science/article/pii/S1361841510001337?via%3Dihub.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021.

Haoxuan You Yutong Feng, Yifan Feng. Meshnet: Mesh neural network for 3d shape representation. *Association for Computing Machinery*, 2018. doi: 10.3389/fpsyg.2022.848784. URL https://arxiv.org/pdf/1811.11424v1.pdf.

A Appendix

A.1 Additional Experiments

A.1.1 CNN

As a baseline for advanced segmentation, we first implemented and ran a basic CNN with only the convolutional layers, adding the required amount of padding in each layer to maintain the same image size. This trains all parameters (kernel weights and biases) to produce the segmentation mask we want as output. Historically, this was one of the first deep learning attempts of segmentation, and U-Net was eventually developed by modifications to this (by adding upsampling layers, and adding back downsampling results within upsampling layers). We used the Keras package for this, and modified a user's existing implementation [4] on the Brain MRI Kaggle dataset described above. The input images (and masks) are 256 x 256 x 3 (RGB channels). We use 4 convolution layers, with 64, 128, 64, and 1 filter each since these are both standard counts and used by the existing implementation on Kaggle. These filters all are size 3x3, and as specified, we set padding as 'same' in the model layers so our output features maps are the same 2d size as inputs. Below are the training and validation losses for 5 epochs, and one example of the results after epoch 1. The image shows the original MRI scan, expected mask, predicted mask, and original overlaid with the predicted mask. We can see that the expected region in the predicted mask is very slightly lighter in color, which shows up in the overlaid images on the far right, but it is not very accurate. Training just 5 epochs took over 4 hours, and the loss is comparatively higher than papers discussed for U-net which demonstrates the relative inefficiency of this method.

epoch	Training Loss	Validation Loss
1	0.3314	0.5383
2	0.1780	0.3628
3	0.1379	0.2448
4	0.1158	0.2064
5	0.1007	0.1716

Table 2: CNN: Loss each epoch

Figure 6: CNN: Sample segmentation from MRI

A.1.2 U-Net with attention

The Attention U-Net is a modified version of the original U-Net architecture, which was also designed for image segmentation. The key difference between the two is the addition of attention blocks in the Attention U-Net, which allow the network to weight the features from the encoder to the decoder selectively.

[4] https://www.kaggle.com/code/krepkiioreshek/brain-mri-segmentation#Simple-NN

In the experiment, we used dice loss as our objective function for training the Attention U-Net neural network. After training the network, we evaluate it with the intersection over union (IOU), which measures the degree of overlap between the predicted segmentation mask and the ground truth mask. The final accuracy of the network on the test dataset was 89%, indicating that the network was able to accurately segment 89% of the pixels in the test images compared to ground truth.

A.1.3 MeshNet

After implementing other methods surrounding U-Net, we explored other methods applied to more complex datasets. We discovered a model applied to a dataset that included 3-dimensional MRI scans. MeshNet was applied to the dataset to accurately segment regions of the images [Yutong Feng, 2018]. This model utilizes mesh representations to better understand 3D shape representation directly from this data type. U-Net was used in conjunction to better represent each MRI image slice and to provide a performance comparison. Each input included an MRI split into a z-stack of images through the transverse plane. The model predicted borders of brain regions and layers utilizing the z-stack of images. The model also predicted cerebellum border compared to the internal structures of the brain. This model format could be applied to our dataset or to get a better understanding of segmented brain regions in the future.

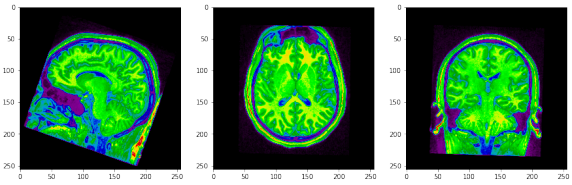

Figure 7: MRI segmentation of whole brain utilizing MeshNet

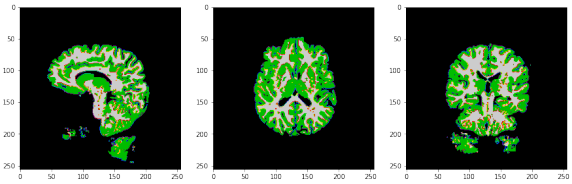

Figure 8: MRI segmentation of cerebellum using MeshNet and U-Net

A.1.4 SwinT

In parallel with implementing the Segformer, we attempted to create our own custom model using a Swin Transformer (SwinT) backbone (encoder) with several Convolutional Upsampling layers (decoder). This is similar to the Segformer, but uses a shifted windows approach instead of a traditional Vision Transformer (ViT) backbone. This means that each SwinT block still has the attention mechanism, but in two steps. The first step is running multi-head attention on local windows around each image patch (instead of using all patches as keys/values). The second step is shifting these windows so that each patch is still informed globally, not just locally. Our model has 3 such transformer blocks with patch-merging in between as the encoder, and the output is a set of feature maps. The decoder in our model is made up of convolution upsampling layers as mentioned, and outputs the final segmentation mask. We attempted to create this model using a pretrained SwinT backbone from HuggingFace and PyTorch decoder, but our loss values seemed to be negative and thus require

some debugging. While we did not have time to fix and test this, there is much potential to use a SwinT backbone for future MRI segmentation since this is more computationally efficient than a ViT.