Delaram Fartoot

Image Processing Project

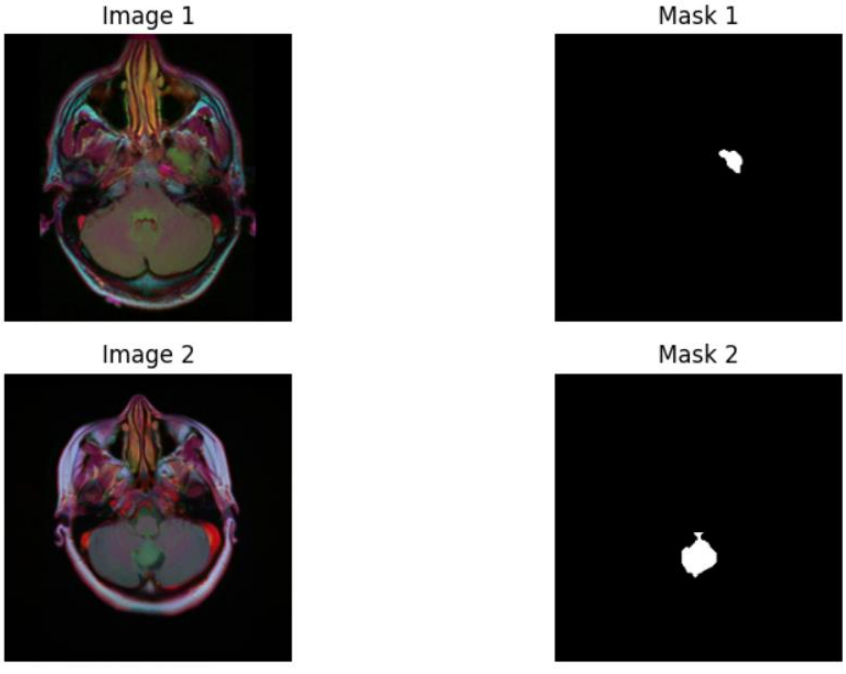
TransNetR for brain MRI segmentation

Introduction

In this project, we employed the TransNetR architecture, which is detailed in the paper titled "TransNetR: Transformer-based Residual Network for Polyp Segmentation with Multi-Center Out-of-Distribution Testing." Although the original design of TransNetR was intended for the segmentation of polyps in medical images, we adapted this architecture for the segmentation of brain MRI scans to detect tumors.

Dataset:

We utilized the LGG Segmentation Dataset, which consists of 3,929 brain MRI images and their corresponding masks. The majority of these images are non-tumorous scans, where the corresponding masks are plain black, indicating the absence of tumors. Since our primary focus was not on distinguishing between tumorous and non-tumorous scans, we chose to concentrate solely on the images that contained tumors. This decision allowed us to more effectively train our model on the segmentation of tumors, ensuring it could accurately learn the features and patterns specific to tumorous tissue. Out of the total 3,929 images, only 1,373 are tumorous, and these were the images we used for training and evaluating our model. Additionally, we visualized some of these scans and their corresponding masks to better understand the data and the segmentation task. This visualization helped us grasp how tumors are represented in the images and masks, which was crucial for developing and validating our model.



Architecture

WE employed the TransNetR architecture. Which is visualized as follows:

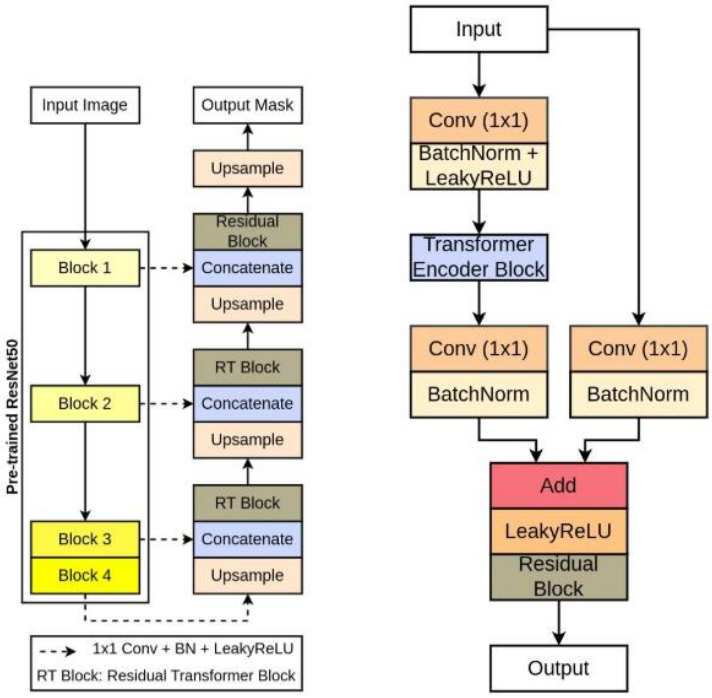


Figure 2: Block diagram of TransNetR along with the Residual Transformer (RT) block.

As observed in the figure, TransNetR is an encoder-decoder network that begins with a pre-trained ResNet50 as the encoder. We implemented ResNet50 using the source code provided by PyTorch. The input image is fed into the pre-trained encoder, which extracts four different intermediate feature maps. These intermediate feature maps are then passed through a 1 × 1 convolution layer without padding and with a stride of 1. This is followed by batch normalization and a LeakyReLU activation function with a negative slope of 0.1. The purpose of the 1 × 1 convolution layer is to reduce the number of output feature channels, thereby decreasing the number of parameters.

Following the encoder, the decoder network consists of three decoder blocks. The reduced feature map from the encoder is fed into the first decoder block, where it undergoes bilinear upsampling. This upsampling layer increases the spatial dimensions of the feature maps by a factor of two. The upsampled feature map is then concatenated with the next reduced feature map that has the same spatial dimensions. This concatenation creates a shortcut connection from the pre-trained encoder to the decoder block, facilitating better information flow between the encoder and decoder. These shortcut connections help recover feature maps that might be lost due to the depth of the network.

The concatenated feature maps are then passed through the proposed Residual Transformer (RT) block. The RT block, visualized on the right side of the image, first reshapes the feature maps into patches and then processes them through transformer layers. These layers include multi-head self-attention, which enhances feature representation learning. The output from the first decoder block is subsequently passed to the second and then to the final decoder block.

In the final decoder block, the Residual Transformer block is replaced with a simple residual block to reduce the number of trainable parameters. The output from the final decoder is subjected to another bilinear upsampling layer, further increasing the spatial dimensions of the feature maps by a factor of two. Finally, the upsampled feature map is passed through a 1 × 1 convolution layer with a sigmoid activation function, producing the final segmented output.

Training

First, as mentioned earlier, we only selected the tumorous MRIs. We then divided our dataset into a training set and a test set with an 80/20 split. Subsequently, we further divided our training set into training and validation sets using the same ratio. Next, we introduced our dataloader for training. The dataloader performs transformations on the input, such as normalizing and changing the dimensions of the images, which are applied in the getitem function of the class.

With the data prepared, we moved on to training the model. We set the number of epochs to 500, but included early stopping with a patience of 50 to prevent overfitting. This means that whenever our F1 score did not improve, we got one step closer to triggering early stopping. The F1 score is a good metric for this task because the dataset is imbalanced, with background pixels outnumbering the foreground (tumor) pixels. We used the Adam optimizer and Binary Cross-Entropy Dice (BC Dice) as the loss function. During each epoch, we calculated and printed the loss, F1 score, Jaccard coefficient (IoU), recall, and precision for both the training and validation sets. Ultimately, we met the early stopping conditions. The resulting metrics for the training and validation sets were quite close, suggesting there wasn't any overfitting. The metrics from the last epoch were as follows:

```
Epoch: 90 | Epoch Time: 0m 33s
        Train Loss: 0.0927 - Jaccard: 0.8112 - F1: 0.8802 - Recall: 0.8753 - Precision: 0.9180
         Val. Loss: 0.1144 - Jaccard: 0.7556 - F1: 0.8333 - Recall: 0.8299 - Precision: 0.8934

Epoch: 91 | Epoch Time: 0m 33s
        Train Loss: 0.0891 - Jaccard: 0.8142 - F1: 0.8830 - Recall: 0.8757 - Precision: 0.9223
         Val. Loss: 0.1153 - Jaccard: 0.7521 - F1: 0.8306 - Recall: 0.8256 - Precision: 0.8947

Epoch: 92 | Epoch Time: 0m 33s
        Train Loss: 0.0884 - Jaccard: 0.8175 - F1: 0.8860 - Recall: 0.8801 - Precision: 0.9196
         Val. Loss: 0.1151 - Jaccard: 0.7520 - F1: 0.8298 - Recall: 0.8243 - Precision: 0.8958

Epoch: 93 | Epoch Time: 0m 33s
        Train Loss: 0.0880 - Jaccard: 0.8180 - F1: 0.8876 - Recall: 0.8826 - Precision: 0.9184
         Val. Loss: 0.1144 - Jaccard: 0.7547 - F1: 0.8329 - Recall: 0.8273 - Precision: 0.8952

Early stopping: validation loss stops improving from last 50 continously.
```

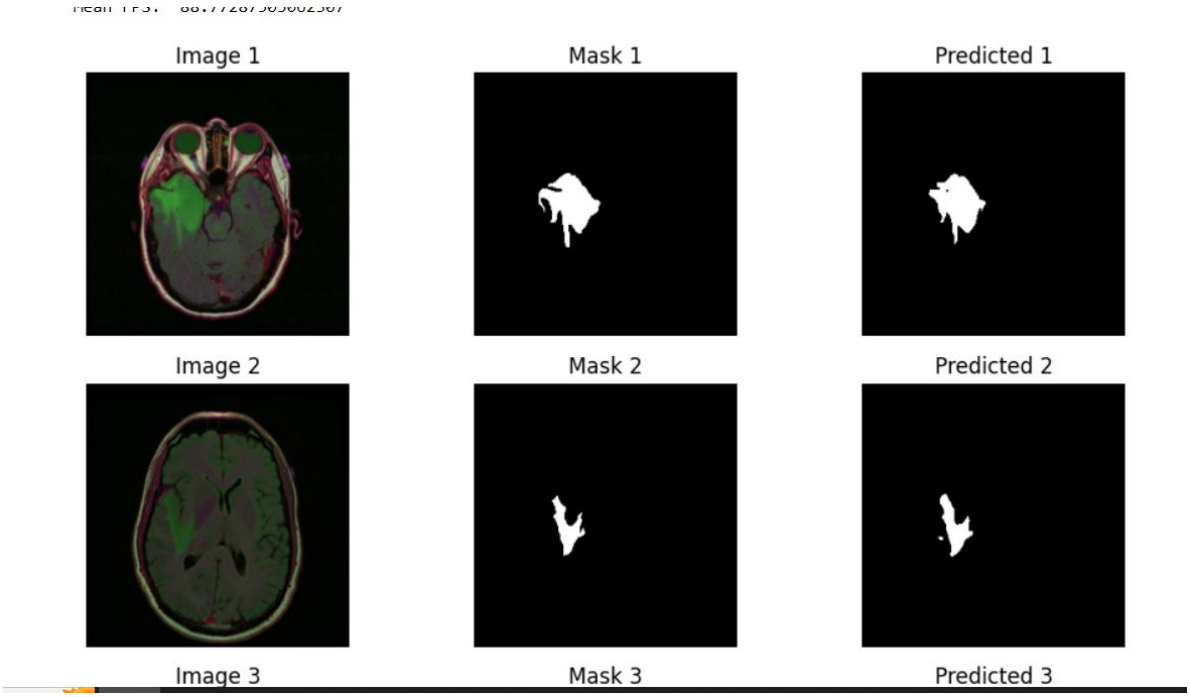
Evaluating

Our final step is the evaluation of our model. This involves assessing the performance of our trained model on the test set, which was held out during the training phase to provide an unbiased estimate of the model's performance. During the evaluation, we use the test set to generate predictions, which are then compared to the true labels to calculate various performance metrics.

Jaccard (IOU)	F1 score	Recall	Precision	Accuracy	F2 score	HD
0.7700	0.8511	0.8622	0.8846	0.9952	0.8546	2.8647

The close alignment of these metrics with those observed during the training and validation phases indicates that our model performs consistently and effectively on new data. This suggests that the model has successfully learned to generalize from the training data to new, unseen MRI scans, providing reliable segmentation of brain tumors.

Following are some figures illustrating several images, their corresponding masks, and the predicted masks.



Comparing with other papers:

Tumor detection is crucial for human health, and extensive research has been conducted in this area. In comparison to other studies on brain MRI segmentation for tumors, our code has proven to be highly effective. With appropriate autotuning and further development, it can match the performance of state-of-the-art models. As is typical in segmentation projects, we evaluate our code against others using the IoU (Jaccard coefficient) metric.

Model	IOU
OM-Net + CGAp	87%
U-Net + more filters + data augmentation + dice-loss	85%
3D CNN + CRF	85.0%
AFN-6	84%
TransNetR (ours)	77%

Conclusion

In conclusion, our project successfully highlighted the capabilities of the TransNetR architecture to achieve accurate and reliable brain tumor segmentation in MRI images. The comprehensive evaluation and consistent performance metrics underscore the model's potential for real-world medical applications, offering a valuable tool for aiding in the diagnosis and treatment planning of brain tumors. This work not only demonstrates the efficacy of transformer-based approaches in medical imaging but also sets the stage for further improvements and adaptations in related domains.