Kristóf Mátyás Pásztor (KGU90S)
Marcell Balázs Tóth (FQP4X1)

Brain Tumor Segmentation

Partners in Prediction

Kristóf Pásztor (KGU90S), Marcell Balázs Tóth (FQP4X1)

Introduction

This project addresses the challenge of brain tumor segmentation using advanced deep learning techniques. Specifically, we utilize the BraTS 2017 dataset [1], which contains diverse MRI scans of patients with brain tumors. The primary objective is to develop a model capable of segmenting various tumor regions in brain MRI images, thereby supporting diagnosis and treatment planning.

Our approach involves preprocessing the dataset, training a segmentation model, and evaluating its performance. Throughout the development process, we drew inspiration from established methodologies, including Andriy Myronenko's work [2], insights from Zeeshan Latif [3], and innovations highlighted in a paper on the UNETR architecture [4].

Additionally, we developed a user-friendly Gradio [5] interface, allowing users to easily interact with our model. Through this interface, users can upload MRI scans, view segmentation outputs, and explore different slices of the segmentation results. This makes the model accessible not only to researchers but also to clinicians and other stakeholders.

Methods of training

In this project, we used a combination of established deep learning architectures, tools, and techniques to train our brain tumor segmentation models. Below, we outline the key methods used during the training process.

Architectures Used

We utilized two deep learning architectures often used in medical image segmentation:

- **SegResNet** [6]: A residual network specifically designed for segmentation tasks, offering improved feature extraction and deep learning capability.
- **UNet** [7]: A widely used architecture for biomedical image segmentation, known for its encoder-decoder structure that enables precise localization and context preservation.

Dataset Preparation

The BraTS 2017 dataset was used as the foundation for training our models. The dataset was split into three subsets:

- **Training Set**: Used to train the models and update their weights through backpropagation.
- **Validation Set**: Used to monitor the performance during training and to guide hyperparameter adjustments.
- **Test Set**: Reserved for final evaluation to measure the models' generalization performance.

This split ensured that the models were evaluated on data they had not seen during training, preventing overfitting and providing an accurate assessment of their performance.

Hyperparameter Optimization

We leveraged **Weights & Biases** (**WandB**) [8] for hyperparameter tuning. WandB helped us track and experiment with various hyperparameters, such as learning rate, batch size, and optimizer configurations. However, due to limited computational resources, we were unable to run extensive experiments or train the models for many epochs.

Framework

Training was conducted using **PyTorch Lightning** [9], a high-level interface for PyTorch that simplifies the training loop and enhances reproducibility. This framework enabled us to manage model training more efficiently by automating common tasks such as checkpointing and logging, while maintaining flexibility for custom implementations.

Training Process

The training process involved the following steps:

- **Data Preprocessing**: Normalizing and augmenting the MRI images to improve model generalization and performance.
- **Model Training**: Iteratively training the models using the training set, optimizing the loss function to improve segmentation accuracy.
- **Validation**: Monitoring performance metrics on the validation set during each epoch to prevent overfitting and guide adjustments to training parameters.
- **Testing**: Evaluating the trained models on the test set to measure their accuracy in segmenting tumor regions.

Challenges Encountered

While we successfully implemented the above methods, limited resources posed challenges, particularly in running longer training sessions with multiple hyperparameter configurations. Despite this, our approach yielded a functional segmentation model.

Evaluation

This section dives into the evaluation results for our brain tumor segmentation model. The primary focus was on assessing the model's ability to accurately segment different brain tissue types, with an emphasis on tumor detection. Our evaluation criteria include both individual performance metrics for each tissue type and overall metrics that summarize model effectiveness:

- **Dice Similarity Coefficient (DSC):** Measures the overlap between predicted and actual segmentations for each tissue type. The results of our solution are presented in Figure 1.

Tissue Type	Dice Score
Background	99.83%
Edema	71.61%
Non-Enhancing Tumor	53.90%
Enhancing Tumor	71.59%

Figure 1: DSC Scores

- **Mean Dice Score**: Aggregates the Dice scores across all classes to provide an overall performance metric. We managed to get a Mean DSC of **74.23%**.
- **Recall**: Focuses on how well the model identifies tumor regions, ensuring no critical regions are missed. The results we have achieved can be seen in Figure 2.

Tissue Type	Recall
Edema	77.10%
Non-Enhancing Tumor	59.72%
Enhancing Tumor	75.26%

Figure 2: Recall Scores

- **Precision**: Examines how many of the predicted tumor pixels are correct. Our precision results are shown in Figure 3.

Tissue Type	Precision
Edema	75.52%
Non-Enhancing Tumor	65.84%
Enhancing Tumor	85.28%

Figure 3: Precision Scores

- **Weighted Recall**: Assigns higher importance to critical tumor regions, especially enhancing tumor areas. Adding a weight of **0.2** to Edema, **0.3** to Non-Enhancing Tumor and **0.5** to Enhancing Tumor, we got to a weighted recall value of **70.97%**.
- **Confusion Matrix**: Visualizes the model's performance across all tissue types to understand misclassification trends. The confusion matrix produced by our model can be seen in Figure 4.

	Background	**Edema**	**Non-Enhancing Tumor**	**Enhancing Tumor**
Background	121814237	167561	7932	5402
Edema	177595	800384	49879	10218
Non-Enhancing	32854	73533	203543	30883
Enhancing	22348	18414	47790	269395

Figure 4: Confusion Matrix

Conclusion

Through this project, we have learned a lot about building machine learning solutions, from preparing data to training and testing models. Working with the BraTS 2017 dataset and using deep learning techniques has helped us understand more about brain tumor segmentation and the challenges involved.

One of the highlights of this project is the Gradio interface we created. It allows users to easily view and interact with the segmentation results, making our work more accessible and practical.

We are proud of what we have achieved with this project. It has been a great learning experience, and we are excited about the final product.

If you wish to read more about our project, visit its GitHub repository page.

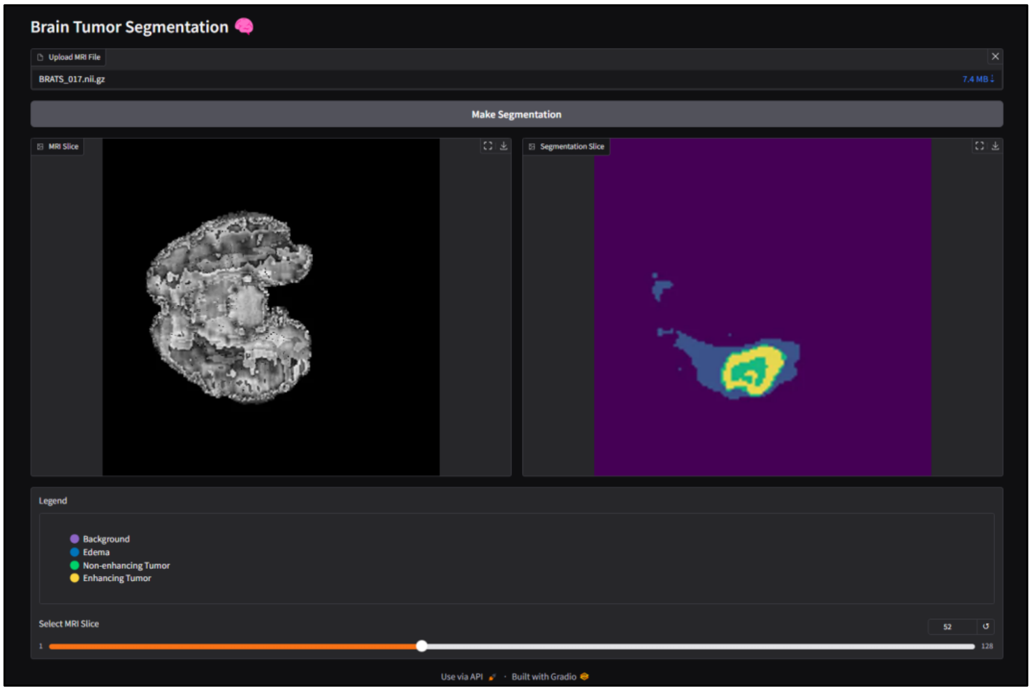
User Interface

The Gradio interface (see Figure 5) allows users to perform brain tumor segmentation using MRI scans. Users can upload a medical imaging file (nii.gz) by selecting the **Upload MRI File** option. Once uploaded, the segmentation process can be initiated by clicking the **Make Segmentation** button.

The interface displays two side-by-side panels:

- The **MRI Slice** panel shows the raw MRI image slice.
- The **Segmentation Slice** panel visualizes the predicted segmentation with color-coded labels indicating different regions:
  - **Purple**: Background
  - **Cyan**: Edema
  - **Green**: Non-enhancing Tumor
  - **Yellow**: Enhancing Tumor

A slider at the bottom allows users to navigate through different slices of the MRI scan. This tool is intuitive and provides a clear visual comparison between the original MRI data and the segmented regions, making it ideal for medical imaging analysis.



Upload MRI File: BRATS_017.nii.gz 7.4 MB

Select MRI Slice: 52

Figure 5: User interface

LLM Usage

We have made use of ChatGPT [10] on multiple occasions. It helped us in writing the documentation of the project and in some cases with debugging our solutions.

Bibliography

[1] Perelman School of Medicine, "Multimodal Brain Tumor Segmentation Challenge 2017," [Online]. Available: https://www.med.upenn.edu/sbia/brats2017/data.html. [Accessed 5 december 2024].

[2] A. Myronenko, "3D MRI brain tumor segmentation using," 2018. [Online]. Available: https://arxiv.org/pdf/1810.11654.

[3] Z. Latif, "Brain Tumor Segmentation using U-Net," 2024. [Online]. Available: https://www.kaggle.com/code/zeeshanlatif/brain-tumor-segmentation-using-u-net#6.-DataGenerator.

[4] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth and D. Xu, "UNETR: Transformers for 3D Medical Image Segmentation," 2021. [Online]. Available: https://arxiv.org/pdf/2103.10504.

[5] Open Source, "Gradio," [Online]. Available: https://github.com/gradio-app/gradio. [Accessed 5 december 2024].

[6] MONAI, "Source code for monai.networks.nets.segresnet," MONAI, [Online]. Available: https://docs.monai.io/en/0.3.0/_modules/monai/networks/nets/segresnet.html. [Accessed 5 December 2024].

[7] R. Olaf, F. Philipp and B. Thomas, "U-Net: Convolutional Networks for Biomedical," 18 May 2015. [Online]. Available: https://arxiv.org/pdf/1505.04597. [Accessed 5 December 2024].

[8] Weights & Biases, "Weights & Biases," [Online]. Available: https://wandb.ai/site/. [Accessed 5 December 2024].

[9] Open Source, "Pytorch Lightning," [Online]. Available: https://lightning.ai/docs/pytorch/stable/. [Accessed 5 December 2024].

[10] OpenAI, "ChatGPT," OpenAI, [Online]. Available: https://chatgpt.com/. [Accessed 5 december 2024].