Brain Tumor Segmentation using 3D U-Net

Sachin Salim (sachinks@umich.edu)
Shrikant Arvavasu (ashri@umich.edu)
Nowrin Mohamed (nowrin@umich.edu)

EECS 504: Computer Vision Fall 2022.
Instructor: Andrew Owens.

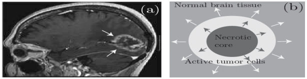
Dec 13, 2022

Why Brain Tumour Segmentation?

- The project focuses on a more prevalent type of brain tumor called Gliomas. Gliomas have different degrees of aggressiveness, variable prognosis and various heterogeneous histological sub-regions say,

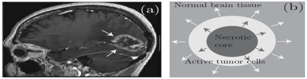
Why Brain Tumour Segmentation?

- The project focuses on a more prevalent type of brain tumor called Gliomas. Gliomas have different degrees of aggressiveness, variable prognosis and various heterogeneous histological sub-regions say,
    - Necrotic core (NCR)

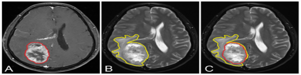


Why Brain Tumour Segmentation?

- The project focuses on a more prevalent type of brain tumor called Gliomas. Gliomas have different degrees of aggressiveness, variable prognosis and various heterogeneous histological sub-regions say,
  - Necrotic core (NCR)

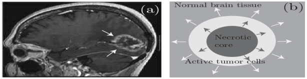


  - Peritumoral edema (ED)

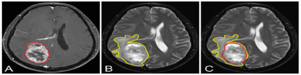


Why Brain Tumour Segmentation?

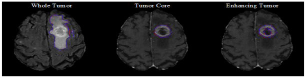
- The project focuses on a more prevalent type of brain tumor called Gliomas. Gliomas have different degrees of aggressiveness, variable prognosis and various heterogeneous histological sub-regions say,
  - Necrotic core (NCR)



  - Peritumoral edema (ED)

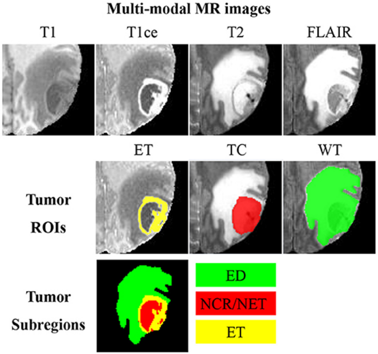


  - Enhancing tumor core (ET)



Problem Statement

- To develop a Brain Tumor Segmentation architecture to segment tumorous tissues from the healthy tissue and in turn from different sub-regions of glioma using the clinically acquired BraTS 2021 dataset.



Motivation behind this work

- This tumor varies greatly in shape, size and appearance which makes the diagnosis a slow, and an extremely challenging problem. Developing a reliable machine learning model that can accurately predict the genetics of cancer could significantly speed up the process and avoid the requirement of multiple invasive surgeries and therapies.

About the Dataset

- We signed up in synapse.org and downloaded **BraTS 2021 data**.

About the Dataset

- We signed up in synapse.org and downloaded **BraTS 2021 data**.
- Total number of cases - **1252** with a 70:20:10 ratio for Training, Validation and Testing respectively.

About the Dataset

- We signed up in synapse.org and downloaded **BraTS 2021 data**.
- Total number of cases - **1252** with a 70:20:10 ratio for Training, Validation and Testing respectively.
- Number of channels/modalities taken is four, say,

About the Dataset

- We signed up in synapse.org and downloaded **BraTS 2021 data**.
- Total number of cases - **1252** with a 70:20:10 ratio for Training, Validation and Testing respectively.
- Number of channels/modalities taken is four, say,
  - **Native (T1)**

About the Dataset

- We signed up in synapse.org and downloaded **BraTS 2021 data**.
- Total number of cases - **1252** with a 70:20:10 ratio for Training, Validation and Testing respectively.
- Number of channels/modalities taken is four, say,
    - **Native (T1)**
    - **Post-contrast T1-weighted (T1Gd)**

About the Dataset

- We signed up in synapse.org and downloaded **BraTS 2021 data**.
- Total number of cases - **1252** with a 70:20:10 ratio for Training, Validation and Testing respectively.
- Number of channels/modalities taken is four, say,
    - **Native (T1)**
    - **Post-contrast T1-weighted (T1Gd)**
    - **T2-weighted (T2)**, and,

About the Dataset

- We signed up in synapse.org and downloaded **BraTS 2021 data**.
- Total number of cases - **1252** with a 70:20:10 ratio for Training, Validation and Testing respectively.
- Number of channels/modalities taken is four, say,
  - **Native (T1)**
  - **Post-contrast T1-weighted (T1Gd)**
  - **T2-weighted (T2)**, and,
  - **T2 Fluid Attenuated Inversion Recovery (T2-FLAIR)**

Dataloading

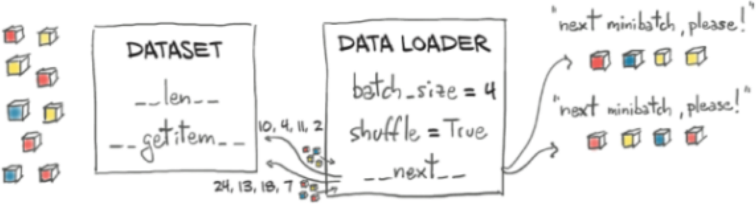
- A data sequence generator is created which enables multiprocessing and avoids loading all data at once to the memory.

Dataloading

- A data sequence generator is created which enables multiprocessing and avoids loading all data at once to the memory.
- The chosen batch size is five.

Dataloading

- A data sequence generator is created which enables multiprocessing and avoids loading all data at once to the memory.
- The chosen batch size is five.
-



Preprocessing

1. **Patch extraction**

Preprocessing

1. **Patch extraction**
   - The patch size is chosen as 64 * 64 * 64. Patch size was decided after considering average tumor size, and the computation power available to us.

Preprocessing

1. **Patch extraction**
   - The patch size is chosen as 64 * 64 * 64. Patch size was decided after considering average tumor size, and the computation power available to us.
   - The smallest cuboid (with margin) containing all foreground (tumor) voxels is found.

Preprocessing

1. **Patch extraction**
   - The patch size is chosen as 64 * 64 * 64. Patch size was decided after considering average tumor size, and the computation power available to us.
   - The smallest cuboid (with margin) containing all foreground (tumor) voxels is found.
   - In each dimension, if it's bigger than the expected size, it's cropped, else it's padded to the intended patch size.

Preprocessing

1. **Patch extraction**
   - The patch size is chosen as 64 * 64 * 64. Patch size was decided after considering average tumor size, and the computation power available to us.
   - The smallest cuboid (with margin) containing all foreground (tumor) voxels is found.
   - In each dimension, if it's bigger than the expected size, it's cropped, else it's padded to the intended patch size.
2. **Data Reshaping**

Preprocessing

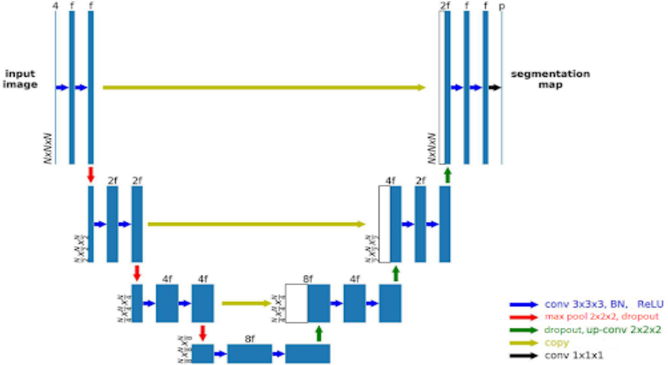
1. **Patch extraction**
   - The patch size is chosen as 64 * 64 * 64. Patch size was decided after considering average tumor size, and the computation power available to us.
   - The smallest cuboid (with margin) containing all foreground (tumor) voxels is found.
   - In each dimension, if it's bigger than the expected size, it's cropped, else it's padded to the intended patch size.
2. **Data Reshaping**
   - All four modalities of the MRI volume are concatenated together along the dimensions(B, H, W, D, Channels).

Preprocessing

1. **Patch extraction**
   - The patch size is chosen as 64 * 64 * 64. Patch size was decided after considering average tumor size, and the computation power available to us.
   - The smallest cuboid (with margin) containing all foreground (tumor) voxels is found.
   - In each dimension, if it's bigger than the expected size, it's cropped, else it's padded to the intended patch size.

2. **Data Reshaping**
   - All four modalities of the MRI volume are concatenated together along the dimensions(B, H, W, D, Channels).
   - Segmentation data is encoded to one-hot tensors (B, H, W, D, Classes).

Implemented three 3D U-Net Models



M: Encoding/decoding layer size

N : Input size

f: Filter size

Variations to the 3D U-Net Model

- **Model 1: (M = 3, N = 64, f = 64) [25.7M params]**

```
Total params: 25,679,044
Trainable params: 25,673,412
Non-trainable params: 5,632
```

Variations to the 3D U-Net Model

- **Model 1: (M = 3, N = 64, f = 64) [25.7M params]**

```
Total params: 25,679,044
Trainable params: 25,673,412
Non-trainable params: 5,632
```

- **Model 2: (M = 3, N = 96, f = 48) [14.4M params]**

```
Total params: 14,448,532
Trainable params: 14,444,308
Non-trainable params: 4,224
```

Variations to the 3D U-Net Model

- **Model 1: (M = 3, N = 64, f = 64) [25.7M params]**

```
Total params: 25,679,044
Trainable params: 25,673,412
Non-trainable params: 5,632
```

- **Model 2: (M = 3, N = 96, f = 48) [14.4M params]**

```
Total params: 14,448,532
Trainable params: 14,444,308
Non-trainable params: 4,224
```

- **Model 3: (M = 4, N = 96, f = 24) [14.6M params]**

```
Total params: 14,570,764
Trainable params: 14,566,348
Non-trainable params: 4,416
```

Training

- Training requires stochastic gradient-based optimization to minimize the cost function with respect to its parameters. We adopted the **adaptive moment estimator (Adam)** to estimate the parameters. The parameters are set as: learning rate $= 10^{-4}$ and the number of epochs $= 30$.

Training

- Training requires stochastic gradient-based optimization to minimize the cost function with respect to its parameters. We adopted the **adaptive moment estimator (Adam)** to estimate the parameters. The parameters are set as: learning rate $= 10^{-4}$ and the number of epochs $= 30$.
- Dice loss is separately computed for each class and combined using weights with respect to class size.

$$DL = 1 - DSC$$

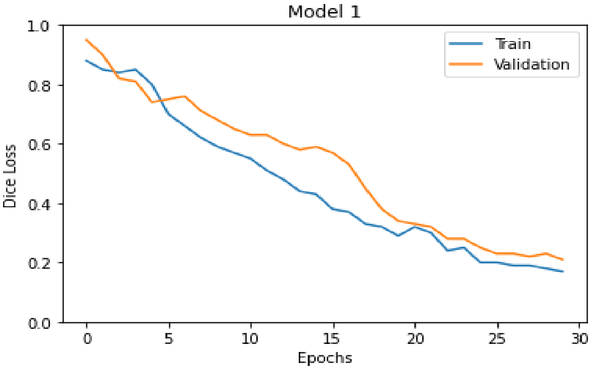
Training

- Training requires stochastic gradient-based optimization to minimize the cost function with respect to its parameters. We adopted the **adaptive moment estimator (Adam)** to estimate the parameters. The parameters are set as: learning rate $= 10^{-4}$ and the number of epochs $= 30$.
- Dice loss is separately computed for each class and combined using weights with respect to class size.

$$DL = 1 - DSC$$

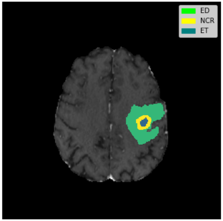
- We are saving the model after each epoch if the model is improving and accordingly the learning rate is also reduced if loss converges.

Training plots

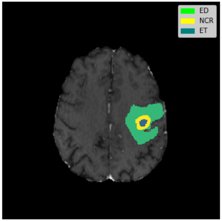


Training Results

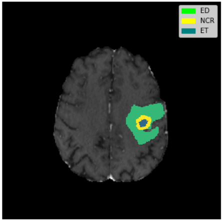
Model 1



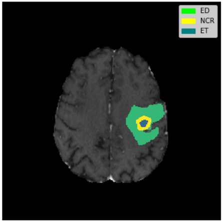
Model 2



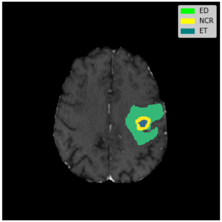
Model 3



Ensemble



Ground Truth



Post Processing

- Our initial attempt to down-scale the input image to NxNxN and up-scaling the segmented results didn't provide desired results.

Post Processing

- Our initial attempt to down-scale the input image to NxNxN and up-scaling the segmented results didn't provide desired results.
- Hence, we used an approach where we extract the patches in a sliding window and feed to the model, and finally the results are stitched together.

Post Processing

- Our initial attempt to down-scale the input image to NxNxN and up-scaling the segmented results didn't provide desired results.
- Hence, we used an approach where we extract the patches in a sliding window and feed to the model, and finally the results are stitched together.
- Binary closing is done on the result to fill the holes.

Post Processing

- Our initial attempt to down-scale the input image to NxNxN and up-scaling the segmented results didn't provide desired results.
- Hence, we used an approach where we extract the patches in a sliding window and feed to the model, and finally the results are stitched together.
- Binary closing is done on the result to fill the holes.
- Segmentation classes are combined to obtain "Enhancing tumor" (ET), the "Tumor core" (TC=ET+NCR), and the "Whole tumor" (WT=TC+ED).

Evaluation Metrics

- **Dice Similarity Coefficient**

$$DSC = \frac{2TP + \epsilon}{FP + 2TP + FN + \epsilon}$$

Evaluation Metrics

- **Dice Similarity Coefficient**

$$DSC = \frac{2TP + \epsilon}{FP + 2TP + FN + \epsilon}$$

- **90% Hausdorff distance**

$$h(A, B) = \max_{a \in A}\{ \min_{b \in B}\{ d(a, b) \} \}$$

$$H(A, B) = \min\{ h(a, b) \ , \ h(b, a)\}$$

Tabulated Evaluation Metric

Model #	Dice Score			90% Hausdorff Distance		
	WT	TC	ET	WT	TC	ET
1	0.756	0.761	0.711	4.56	6.98	5.03
2	**0.812**	0.756	0.702	4.02	8.22	4.78
3	0.804	0.732	0.724	4.33	7.64	4.56
Ensemble	0.805	**0.769**	**0.735**	**3.84**	**6.72**	**4.23**

Future Directions

- 3D U-Net with attention layers added can potentially yield better results.

Future Directions

- 3D U-Net with attention layers added can potentially yield better results.
- With access to more computational resources, the depth of the model can be increased or more filters can be added.


THANK YOU!