Brain Tumor Segmentation

T Ram Srikumar | Varshith P Singh

Introduction

Automatic segmentation of MRI brain tumors based on deep learning can automatically extract more expressive and discriminative brain image features. Different deep learning techniques can be used to automate the segmentation process, saving time and effort compared to manual segmentation, which is tedious, time consuming and requires expert intervention from doctors and specialists. This project aims to use modern deep learning techniques such as U-NET to segment MRI scans in order to detect glioblastoma (the most common type of brain tumor) and segment the MRI scans. Misdiagnosis of a tumor can be dangerous or even fatal in some cases, as it allows the brain tumors to grow and affect brain functions. Therefore, this project is essential in order to prevent such consequences.

Dataset

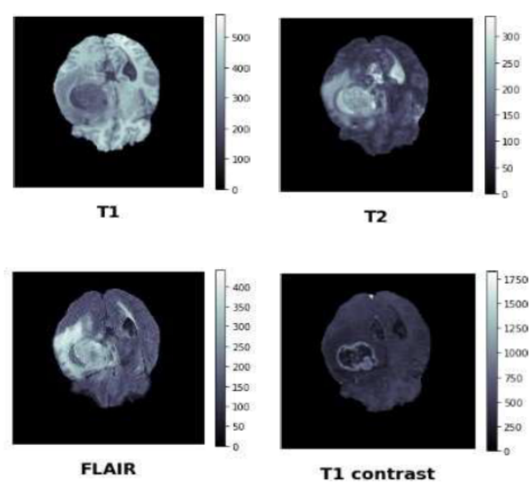
Brain Tumor Segmentation(BraTS2020) - [Brain Tumor Segmentation(BraTS2020) | Kaggle]

•Collection of medical imaging data designed for brain tumor segmentation.

•It includes MRI scans from various sources, covering different modalities:

- ➔ T1-weighted
- ➔ T2-weighted
- ➔ T1-weighted images with contrast enhancement (**T1CE**)
- ➔ Fluid-attenuated inversion recovery (**FLAIR**) images

•Size of all images is 240×240×155 pixels.



Base Model

3D U-NET:

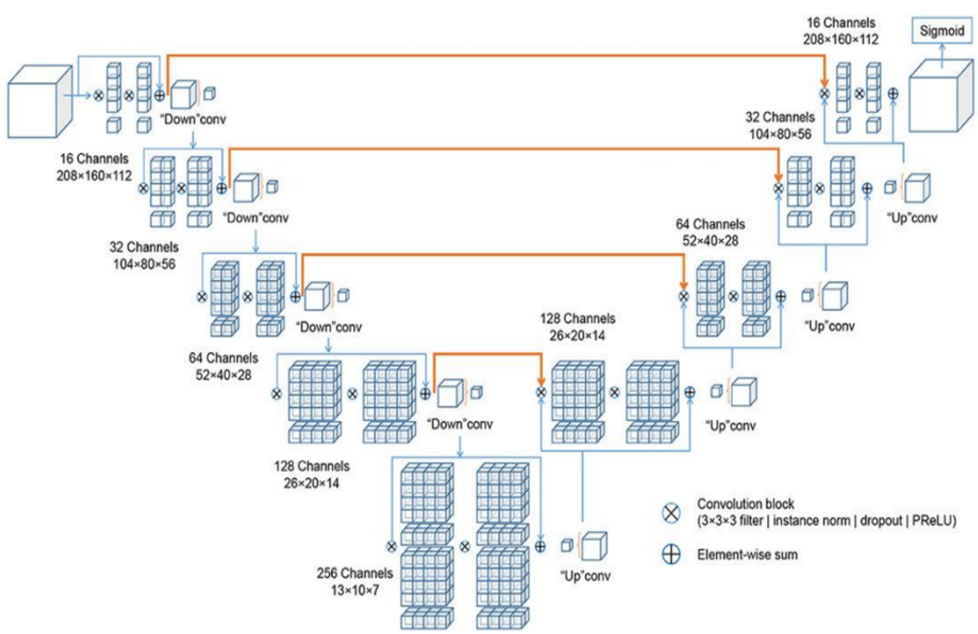
The 3D U-Net architecture is designed for 3D image segmentation tasks and is particularly valuable in medical imaging applications where precise delineation of anatomical structures or abnormalities is crucial.

It consists of:

- Encoding layer
  - ➔ Extracts features from input image
- Decoding layer
  - ➔ Upsamples the compressed feature maps
- Skip connections
  - ➔ Captures the fine-grained spatial details in the input image
- Residual connections
  - ➔ Copies feature maps from output of one block to input of next block
- Fully convolutional layer
  - ➔ Generates the final segmentation mask(same size as that of input image)

Here, we made use of FLAIR images only. We have used 100 images in total with a ratio of 70:30 for training & testing respectively, with reduced dimension of 64 x 64 x 64 pixels each.

Model Architecture



Base Models Results

Evaluation Metric Used :

Dice Similarity Coefficient(**DSC)** - a versatile, widely adopted metric in image segmentation that provides a quantitative measure of how well a segmentation aligns with the ground truth, helps to assess the accuracy of the segmentation.

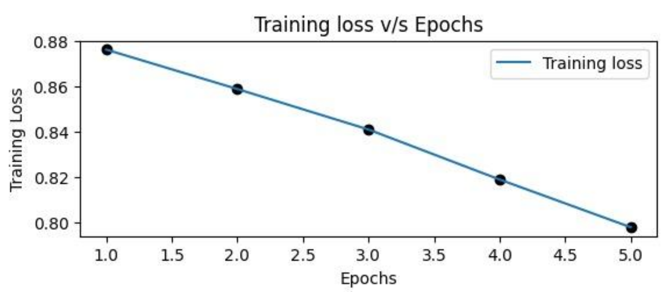
Results	*Training data*	*Test data*
Recent performance(2021)	***0.87***	***0.83***
Our model's performance	***0.93***	***0.93***

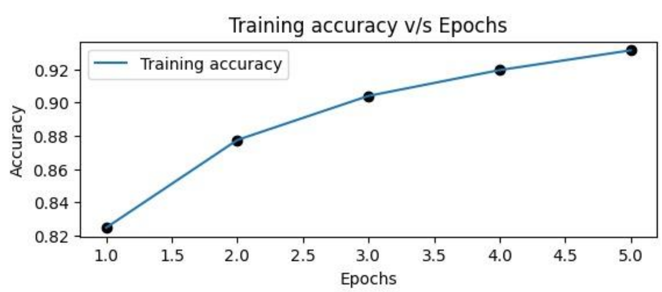
```
hist = model.fit(x_train,y_train,batch_size=8,epochs=5,validation_data=(x_test,y_test))
```

```
Epoch 1/5
9/9 [==============================] - 45s 5s/step - loss: 0.8763 - accuracy: 0.8248 - val_loss: 0.8823 - val_accuracy: 0.8560
Epoch 2/5
9/9 [==============================] - 43s 5s/step - loss: 0.8588 - accuracy: 0.8775 - val_loss: 0.8677 - val_accuracy: 0.9070
Epoch 3/5
9/9 [==============================] - 43s 5s/step - loss: 0.8409 - accuracy: 0.9040 - val_loss: 0.8497 - val_accuracy: 0.9217
Epoch 4/5
9/9 [==============================] - 43s 5s/step - loss: 0.8188 - accuracy: 0.9195 - val_loss: 0.8291 - val_accuracy: 0.9283
Epoch 5/5
9/9 [==============================] - 43s 5s/step - loss: 0.7977 - accuracy: 0.9314 - val_loss: 0.8091 - val_accuracy: 0.9294
```

```
print('Loss obtained :',round(prediction[0],2),'\nAccuracy obtained :',round(prediction[1],2))
```

```
Loss obtained : 0.81
Accuracy obtained : 0.93
```





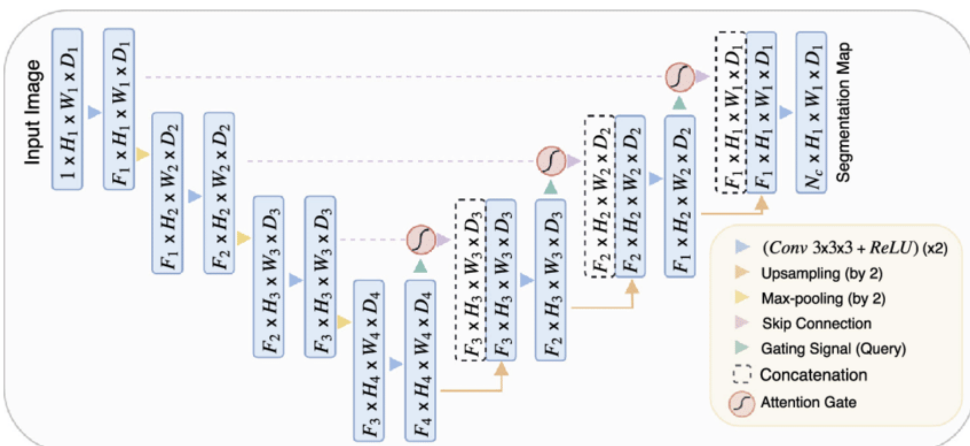

```
<Figure size 640x480 with 0 Axes>
```

Refinements

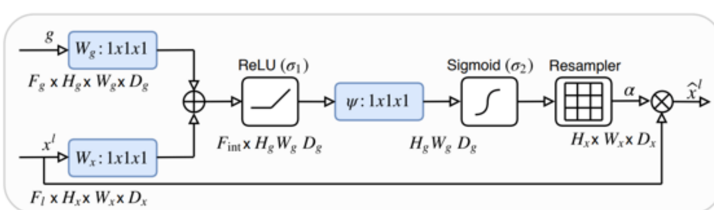
Refinements that we incorporated into the base model to achieve improvements in overall performance and efficiency :

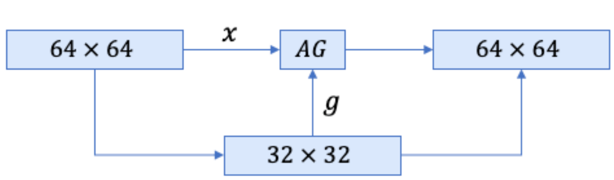
- We have incorporated a self-attention mechanism, which is designed to improve the modeling of long-range dependencies & contextual information within the images in order to enhance the segmentation performance of the model.
- It helps to bridge the semantic gap between distant regions in the image.
- It helps the network to generalize better from the training data to unseen images.

Attention U-Net



Attention block





Refinement Results

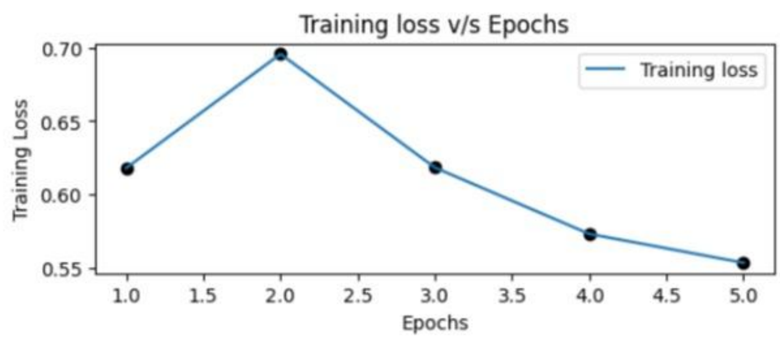
Results	*Training data*	*Test data*
Recent performance(2021)	***0.87***	***0.83***
3D - UNet (Base Model)	***0.93***	***0.93***
Attention 3D - UNet (Refinement)	***0.95***	***0.98***

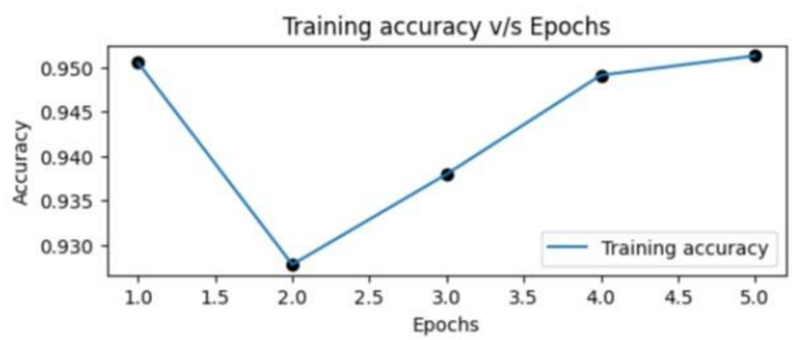
```
hist_2 = model.fit(x_train,y_train,batch_size=4,epochs=5,validation_data=(x_test,y_test))
```

```
Epoch 1/5
18/18 [==============================] - 44s 2s/step - loss: 0.6181 - accuracy: 0.9505 - val_loss: 0.9566 - val_accuracy: 0.9781
Epoch 2/5
18/18 [==============================] - 42s 2s/step - loss: 0.6956 - accuracy: 0.9278 - val_loss: 0.8106 - val_accuracy: 0.8686
Epoch 3/5
18/18 [==============================] - 48s 3s/step - loss: 0.6184 - accuracy: 0.9380 - val_loss: 0.5080 - val_accuracy: 0.9670
Epoch 4/5
18/18 [==============================] - 42s 2s/step - loss: 0.5734 - accuracy: 0.9491 - val_loss: 0.8072 - val_accuracy: 0.9781
Epoch 5/5
18/18 [==============================] - 42s 2s/step - loss: 0.5538 - accuracy: 0.9513 - val_loss: 0.6209 - val_accuracy: 0.9357
```

```
print('Loss obtained after refinement:',round(prediction_2[0],2),'\nAccuracy obtained after refinement:',round(prediction_2[1],2))
```

```
Loss obtained after refinement: 0.62
Accuracy obtained after refinement: 0.98
```





Inference

From the results given above we can get a clear picture that the Attention 3D U-Net model outperforms the vanilla U-Net model. Even though the evaluation is done on a significantly smaller sample of the complete dataset, we can expect the model to perform on a similar basis if it is being evaluated on the complete dataset.

References

1. GuangRui Liu & YiLi Zhao(2022) - A survey of deep learning methods for MRI brain tumor image segmentation
2. Hrishikesh Lamdade, Arjun Pansare, Gaurav Parulekar & Jignesh Sisodia(2023) - Brain Tumor Segmentation, Grade of Tumor and Survival Duration Prediction using Deep Learning
3. Xian Zhang; Ziyuan Feng; Tianchi Zhong; Sicheng Shen; Ruolin Zhang; Lijie Zhou(2021) - DRA U-Net: An Attention based U-Net Framework for 3D Medical Image Segmentation