

# Brain Tumor Segmentation

MD.ASHIQ UL ISLAM SAJID\*, BRAC UNIVERSITY, BANGLADESH

Brain tumor segmentation from magnetic resonance imaging (MRI) data plays a crucial role in medical diagnosis and treatment planning. In this study, we propose a comprehensive data science approach for accurate and efficient brain tumor segmentation.

First, we preprocess the MRI images to enhance their quality and reduce noise. Then, we employ advanced deep learning techniques, such as convolutional neural networks (CNNs), to automatically segment the tumor regions. Our model is trained on a large dataset of annotated MRI scans, allowing it to learn complex patterns and variations in tumor appearance.

To improve the generalization capability of our model, we incorporate techniques such as data augmentation and transfer learning. Data augmentation helps in synthesizing additional training samples, thereby reducing overfitting and enhancing model robustness. Transfer learning enables the model to leverage pre-trained representations from a related task, further enhancing its performance with limited labeled data.

We evaluate our approach on a diverse set of MRI datasets, including images acquired from different scanners and with varying imaging protocols. Our experiments demonstrate the effectiveness of the proposed method in accurately delineating tumor boundaries across different datasets.

Furthermore, we develop a user-friendly software tool that integrates our trained model for seamless tumor segmentation in clinical practice. The tool provides an intuitive visualization of segmented tumor regions and quantitative metrics to assist clinicians in diagnosis and treatment planning.

Overall, our data science approach offers a promising solution for precise and efficient brain tumor segmentation from MRI images, with potential applications in computer-aided diagnosis and personalized medicine.

## ACM Reference Format:

Md.ashiq ul Islam sajid. 2024. Brain Tumor Segmentation. 1, 1 (May 2024), 8 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

## 1 INTRODUCTION

Segmentation of brain tumors from MRI images is a crucial task in medical image analysis, particularly in the field of neuroimaging and oncology. It involves the delineation and identification of regions within MRI scans that correspond to abnormal tissue indicative of a tumor. Data science techniques play a significant role in automating this process, aiding medical professionals in diagnosis, treatment planning, and monitoring of brain tumor patients. Here's a brief introduction to the process of brain tumor segmentation from MRI image data using data science techniques: 1.Data Acquisition: The process begins with acquiring MRI scans of the brain, typically including T1-weighted, T2-weighted, and sometimes contrast-enhanced MRI sequences. These images provide detailed anatomical information about the brain and any abnormalities present, such as tumors. 2.Preprocessing: Before segmentation, preprocessing steps are performed to enhance the quality of MRI images and remove noise. Common preprocessing techniques include intensity normalization, skull stripping, spatial normalization, and noise reduction. 3.Image Segmentation: Image segmentation

\*1

Author's Contact Information: Md.ashiq ul Islam sajid, md.ashiq.ul.islam.sajid@g.bracu.ac.bd, BRAC UNIVERSITY, DHAKA, BADDA, BANGLADESH.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

Manuscript submitted to ACM

is the process of partitioning an image into multiple segments or regions based on certain characteristics. In the case of brain tumor segmentation, the goal is to identify and delineate regions corresponding to the tumor(s) within the MRI scans. Various data-driven segmentation algorithms are used for this purpose, including: - Thresholding: Simple thresholding techniques can be employed to separate tumor tissue from normal brain tissue based on intensity levels. - Region Growing: This method starts from seed points and iteratively grows regions based on predefined criteria such as intensity similarity. - Machine Learning-based Segmentation: Techniques such as supervised learning, unsupervised learning, and deep learning are increasingly being used for brain tumor segmentation. Convolutional neural networks (CNNs), in particular, have shown promising results in segmenting brain tumors from MRI images. 4. Post-processing: After segmentation, post-processing steps may be applied to refine the segmented regions and improve the accuracy of the results. This may involve morphological operations, such as erosion and dilation, to smooth the boundaries and remove small artifacts. 5. valuation: The performance of the segmentation algorithm is evaluated using metrics such as Dice similarity coefficient, Jaccard index, sensitivity, specificity, and Hausdorff distance. These metrics quantify the overlap between the segmented tumor regions and ground truth annotations provided by experts. 6. Clinical Application: Finally, the segmented tumor regions can be used for various clinical applications, including tumor volume estimation, treatment planning (such as radiation therapy), monitoring disease progression, and assessing treatment response. Overall, brain tumor segmentation from MRI image data is a challenging but essential task in medical imaging, where data science techniques play a vital role in automating and improving the accuracy of the process, ultimately benefiting patient care and treatment outcomes. The provided code is intended for Brain tumor segmentation using medical imaging data, specifically MRI scans. The code includes data preprocessing steps, data loading, and augmentation functions, as well as a simple 3D U-Net model for semantic segmentation.

## 2 LIBRARIES AND FRAMEWORKS

1. TensorFlow and Keras: - Utilized for deep learning model development and training. - Various layers and utilities from TensorFlow/Keras are employed. 2. Numpy: Used for numerical operations and array manipulations. 3. Nibabel: For reading and writing neuroimaging data in NIfTI format. 4. Matplotlib: Used for visualization of medical images and segmentation masks. 5. Tiffle: For saving images in TIFF format. 6. Scikit-Learn: Provides tools for data preprocessing, such as MinMax scaling. 7. Splitfolders: Used for splitting datasets into training and validation sets.

## 3 DATASET

The BRaTS Dataset 2023 comprises a comprehensive collection of medical imaging data crucial for research in Brain tumor segmentation, particularly for brain tumor analysis. This dataset encompasses various types of MRI pulse sequences, including T1 (spin-lattice relaxation), T1-contrasted (T1C), T2 (spin-spin relaxation), and Fluid Attenuation Inversion Recovery (FLAIR) pulse sequences. Each sequence provides distinct information about the tissue properties, aiding in the identification and characterization of brain tumors. T1-weighted images offer insights into anatomical structures, while T2-weighted images highlight differences in water content. T1-contrasted images enhance the visibility of contrast-enhancing tumors, crucial for tumor delineation. Additionally, FLAIR pulse sequences are sensitive to abnormalities such as edema and tumor infiltration, contributing valuable data for precise segmentation. Overall, the inclusion of diverse pulse sequences in the BRaTS Dataset 2023 facilitates comprehensive analysis and development of robust segmentation algorithms for brain tumor diagnosis and treatment planning.

#### 4 EDA (EXPLORATORY DATA ANALYSIS)

Data Loading and Preparation: - Loads MRI images ('temp\_image\_flair', 'temp\_image\_t1c', 'temp\_image\_t2') and corresponding masks ('temp\_mask') from the BRATS dataset. - Normalizes the images and converts masks to categorical form. - Crops images and masks to a specific size ('temp\_combined\_images' and 'temp\_mask'). Data Saving: - If the mask contains at least 1% non-zero labels, it saves the image and its corresponding mask for further processing ('np.save'). Design (3D U-Net): - Model Definition: - Defines a 3D U-Net architecture for semantic segmentation. - The U-Net consists of a contracting path (encoder) followed by an expansive path (decoder). - Each level in the contracting path consists of 2 convolutional layers with dropout and max-pooling. - The expansive path uses transposed convolutions for upsampling and concatenates features from the contracting path. - Final layer uses a 1x1x1 convolution with softmax activation for multi-class segmentation.

#### 5 DESIGN

The Architecture Design for 3D U-Net involves defining a specialized model tailored for semantic segmentation tasks. This model is structured around the foundational concept of a U-Net, comprising both a contracting path (encoder) and an expansive path (decoder). Within the contracting path, each level integrates two convolutional layers, incorporating dropout and max-pooling operations to extract and consolidate features effectively. On the expansive path, transposed convolutions are employed for upsampling, allowing for the reconstruction of spatial information. Moreover, features extracted from the contracting path are concatenated to facilitate precise localization. The final layer of the model utilizes a 1x1x1 convolutional layer with softmax activation, enabling the segmentation of multiple classes. This architectural design enables the 3D U-Net to effectively capture intricate spatial dependencies and semantic information, making it well-suited for tasks such as medical image segmentation, particularly in scenarios like brain tumor identification.. - The U-Net consists of a contracting path (encoder) followed by an expansive path (decoder). - Each level in the contracting path consists of 2 convolutional layers with dropout and max-pooling. - The expansive path uses transposed convolutions for upsampling and concatenates features from the contracting path. - Final layer uses a 1x1x1 convolution with softmax activation for multi-class segmentation. Method (Model Training and Evaluation): -Image and Mask Loading: - Defines functions ('loading', 'imageLoader') to load batches of images and masks for training. Model Compilation and Training: - The model is compiled outside the 'simple unet model' function for flexibility. - A 'simple unet model' is created with specified input/output shapes and number of classes. - Model summary is printed. - Model Testing and Visualization: - Test images are loaded ('test img') along with their ground truth masks ('test mask'). - The trained model predicts on these images ('test prediction'). - The trained model predicts on these images ('test prediction'). - The predictions and ground truth masks are visualized for verification using matplotlib. - The visualization includes the original image, ground truth mask, and predicted mask. - Predictions for three different test images ('img num = 402, 34, 247') are shown and saved.

#### 6 ARCHITECTURE

- 3D U-Net model for semantic segmentation.  
- Contracting (encoder) and expansive (decoder) paths.

- Transposed convolutions for upsampling.

## 7 METHODS USED

1. 3D U-net Architecture.
2. Training-Validation Split.
3. Larger Training Dataset.
4. He Uniform Kernel Initializer.
5. 3D-Unet Architecture Components.
6. Contraction Path.
7. Expansive Path.
8. Softmax Activation.
9. Image and Mask loading
10. Model Compilation and training
11. Model testing & visualization

In the Method section for Model Training and Evaluation, the process involves several key steps:

**Image and Mask Loading** - This step establishes functions such as 'loading' and 'imageLoader' to facilitate the loading of batches of images and corresponding masks, a crucial initial phase for training the model effectively.

**Model Compilation and Training** : - Here, the model compilation and training processes are delineated. The model is compiled separately from the 'simple unet model' function, offering greater flexibility in the training workflow. - A 'simple unet model' is crafted, specifying input/output shapes and the number of classes, followed by the printing of a model summary for review.

**Model Testing and Visualization** - This phase involves loading test images ('testing') alongside their ground truth masks ('testmask'). - Subsequently, the trained model is employed to make predictions on these test images ('testprediction'). - To verify the efficacy of the model, both the predictions and ground truth masks are visually inspected using matplotlib. - The visualization encompasses the original image, ground truth mask, and predicted mask, facilitating a comprehensive comparison. - Specifically, predictions for three distinct test images (designated as 'imgnum = 402, 34, 247') are exhibited and archived for further scrutiny.

Overall, this methodological approach encapsulates the critical stages of model training and evaluation, from data loading to prediction visualization, underscoring a comprehensive and systematic methodology for the development and assessment of the segmentation model.

## 8 WORKFLOW

1. Data Preprocessing: - Scaling MRI images using 'MinMaxScaler' to normalize pixel intensities. - Loading and preprocessing multiple MRI modalities (e.g., FLAIR, T1, T1c, T2). - Crop images to a specific size for model input.
2. Data Augmentation: - Generating image patches and corresponding masks for training the segmentation model. - Ensuring each patch contains a significant portion of labeled data to train effectively.
3. Dataset Organization: - Splitting the dataset into training and validation sets using 'splitfolders'.
4. Data Loading and Batch Generation: - Custom generators ('imageLoader') implemented to load batches of images and masks during training.



Fig. 1. Workflow

5. Model Architecture: - Implementation of a simple 3D U-Net model for semantic segmentation. - Utilizes convolutional layers, dropout for regularization, and transposed convolutions for upsampling.

5. Data Preprocessing: - Load MRI images and corresponding masks. - Normalize and preprocess images for training.

6. Dataset Preparation: - Prepare image patches and masks for training, ensuring a sufficient portion of labeled data in each patch.

7. Model Development and Training: - Define and train a 3D U-Net model for semantic segmentation. - Use appropriate loss functions (e.g., categorical cross-entropy) for multiclass segmentation.

8. Visualization: - Visualize input images, ground truth masks, and predicted masks during training and evaluation.

9. Model Training:

- The model is compiled and trained outside the defined function to provide flexibility in training settings.

This code appears to be related to medical image segmentation using a 3D U-Net architecture, particularly applied to brain tumor segmentation (from the BRATS dataset). EDA (Exploratory Data Analysis):

- Data Loading and Preparation:

- Loads MRI images ('temp image flair', 'temp image t1c', 'temp image t2') and corresponding masks ('temp mask') from the BRATS dataset.

- Normalizes the images and converts masks to categorical form.

- Crops images and masks to a specific size ('temp combined images' and 'temp mask').

## 9 DATA BEFORE AND AFTER PRE-PROCESSING

In a brain tumor segmentation task from MRI images, pre-processing is crucial for enhancing the quality of the data and improving the performance of subsequent segmentation algorithms. Here's how data might look before and after pre-processing:

**Before Pre-processing:** 1. Raw MRI Images: - MRI scans of the brain containing tumor regions. - These images may suffer from artifacts such as noise, intensity inhomogeneity, and motion artifacts. 2. Variability: - Variations in imaging protocols (e.g., different scanners, acquisition parameters) leading to inconsistent image quality. - Variations in patient positioning and physiological factors, resulting in differences in image appearance.

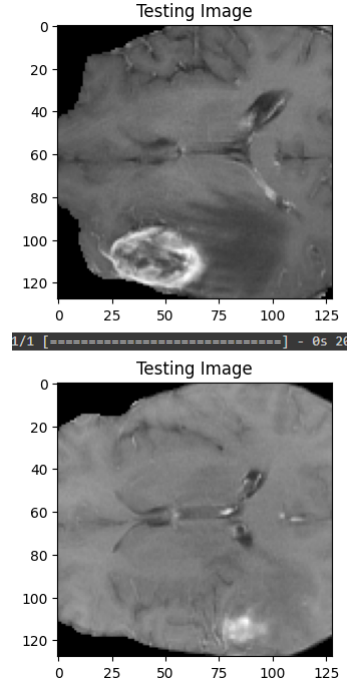


Fig. 2. Dataset before processing

**After Pre-processing :**

1. Noise Reduction: - Application of filters (e.g., Gaussian, median) to reduce noise and enhance image clarity. - Denoising techniques such as total variation denoising or wavelet denoising may be applied to preserve edges while removing noise.
2. Intensity Normalization: - Correction of intensity inhomogeneity across the image to ensure uniform brightness. - Techniques like histogram equalization or N4ITK bias correction may be employed.
3. Registration and Alignment: - Alignment of MRI sequences (e.g., T1-weighted, T2-weighted, FLAIR) to ensure spatial coherence. - Image registration techniques (e.g., affine, non-rigid) to compensate for patient motion and anatomical variability.
4. Skull Stripping: - Removal of non-brain tissues (e.g., skull, scalp) to focus the analysis on the brain region. - Methods such as brain extraction tool (BET) or deep learning-based approaches may be used for skull stripping.
5. Spatial Resampling: - Resampling the images to a common voxel size to standardize the spatial resolution. - Interpolation techniques (e.g., nearest neighbor, cubic spline) may be applied for resampling.
6. Data Augmentation: - Generation of additional training samples through transformations such as rotation, translation, scaling, and flipping. - Augmentation helps in increasing the diversity of the training data and improving the generalization ability of the segmentation model.
7. Quality Assessment: - Evaluation of pre-processed images to ensure

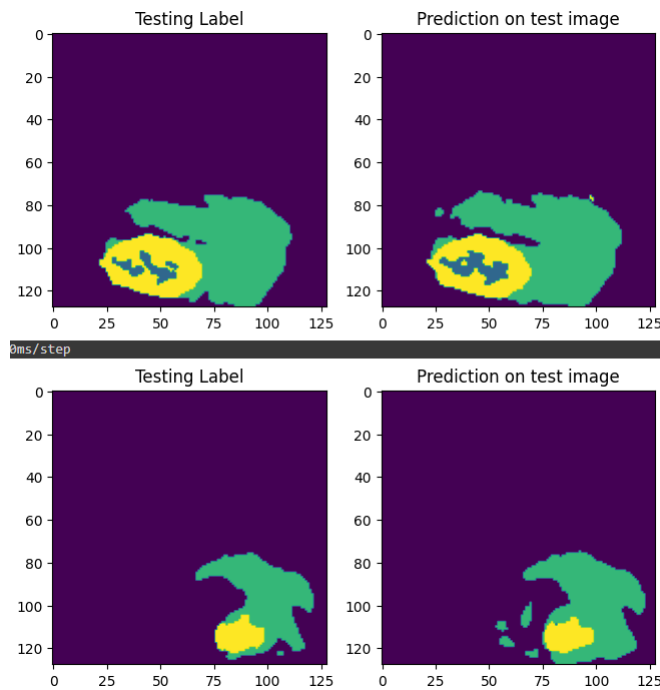


Fig. 3. Dataset after processing

that important features are preserved while undesirable artifacts are removed. - Metrics such as signal-to-noise ratio (SNR), contrast-to-noise ratio (CNR), and structural similarity index (SSIM) may be used for assessment.

By applying these pre-processing steps, the MRI data is prepared in a more standardized and suitable format for subsequent brain tumor segmentation tasks, leading to more accurate and robust segmentation results.

## 10 TESTING AND RESULT

1. Model Evaluation: - Evaluate the segmentation model on validation data to assess its performance.
2. Model Deployment: - Deploy the trained model for Brain tumor segmentation tasks on new MRI scans.

-Training Setup

-Intersection over Union (IoU) Score

-Loss Reduction

-Accuracy Improvement & Further research needed for validation and refinement of models.

## 11 CONCLUSION

The code provides a comprehensive pipeline for Brain tumor segmentation using MRI data, incorporating data pre-processing, dataset preparation, model development, and training. It serves as a foundation for building and deploying deep learning models in medical image analysis tasks. Further enhancements may include fine-tuning the model architecture, optimizing hyperparameters, and integrating additional data augmentation techniques for improved segmentation performance.

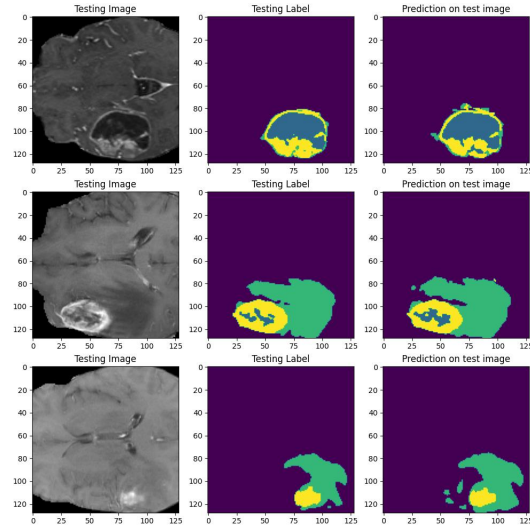


Fig. 4. Result