METHODOLOGY REPORT

VR-Based Brain Tumor Visualization System

Using BraTS2020 Dataset and Meta Quest 3

Team Members:

Sezer Ataş

Mete Serpil

Muhammed Yusuf Özcan

Alperen Berke Çetinkaya

Contents

Abstract

This report is intended to present a comprehensive framework for the development of a Virtual Reality (VR) based brain tumor visualization system. Advanced deep learning segmentation techniques are integrated with immersive VR technology to create an interactive 3D visualization platform for brain tumor analysis and surgical pre-planning. Multi-modal MRI data from the BraTS2020 dataset is processed, automated tumor segmentation is performed using the nnU-Net architecture, and the results are rendered in an immersive VR environment using the Meta Quest 3. This approach is aimed at enhancing medical education, treatment planning, and patient communication through the intuitive 3D visualization of complex neurological structures.

1.1 Background

Brain tumors represent one of the most challenging conditions in modern medicine, requiring precise diagnosis and treatment planning. Traditional 2D medical imaging techniques, while valuable, often fail to convey the complex three-dimensional spatial relationships between tumors and surrounding brain structures. The advent of Virtual Reality (VR) technology presents a transformative opportunity to visualize and interact with medical imaging data in ways that closely mirror the actual anatomical reality.

The BraTS (Brain Tumor Segmentation) Challenge has established itself as the premier benchmark for evaluating brain tumor segmentation algorithms. The BraTS2020 dataset provides multi-parametric MRI scans including T1-weighted, T1-weighted with contrast enhancement (T1ce), T2-weighted, and FLAIR sequences, along with expert-annotated tumor segmentation masks distinguishing between different tumor sub-regions.

1.2 Project Objectives

This project aims to develop an end-to-end pipeline that:

- Automatically segments brain tumors from multi-modal MRI scans using state-of-the-art deep learning techniques

- Converts medical imaging data into formats compatible with real-time 3D rendering engines

- Renders the composite brain model in an immersive VR environment using Meta Quest 3

- Provides intuitive interaction mechanisms for exploring and analyzing tumor characteristics in VR

1.3 Significance

The proposed system offers several significant advantages over traditional visualization methods:

Enhanced Spatial Understanding: VR enables clinicians to perceive tumor location, size, and relationship to critical brain structures in true 3D space, improving surgical planning and risk assessment.

Improved Patient Communication: Patients and families can better understand their condition through intuitive, immersive visualization rather than abstract 2D scans.Medical Education: Medical students and residents can explore realistic tumor cases in a risk-free, interactive environment.

Reproducibility and Automation: The automated segmentation pipeline ensures consistent, reproducible results across different cases and operators.

2. Part I: Brain Tumor Segmentation and Processing

2.1 Dataset: BraTS2020

The BraTS2020 dataset serves as the foundation for our tumor segmentation component. This dataset contains multi-institutional, multi-parametric MRI scans of glioblastoma and lower-grade glioma patients.

Dataset Characteristics:

• Imaging Modalities: Four MRI sequences per patient (T1, T1ce, T2, FLAIR)

• Volume Dimensions: Typically 240×240×155 voxels

• Voxel Resolution: $1mm^3$ isotropic resolution

• Annotation Labels: 0 (background), 1 (necrotic/non-enhancing tumor core), 2 (peritumoral edema), 4 (enhancing tumor)

• Pre-processing: Skull-stripped, co-registered to the same anatomical template, and interpolated to the same resolution

2.2 Tumor Segmentation: nnU-Net Architecture

The nnU-Net (no-new-Net) deep learning framework is employed for automated tumor segmentation. This architecture is widely recognized as the state-of-the-art standard in biomedical image segmentation, having consistently demonstrated superior performance and robustness across diverse datasets, including the BraTS benchmarks.

2.2.1 Architecture Overview

nnU-Net is a self-configuring semantic segmentation method that automatically adapts to new datasets. The architecture is based on the U-Net encoder-decoder structure with the following key components:

• Encoder Path: Progressive downsampling through convolutional blocks to capture semantic features at multiple scales

• Decoder Path: Symmetric upsampling path with skip connections to recover spatial resolution

• Skip Connections: Direct connections between encoder and decoder at matching resolutions to preserve fine-grained spatial information

• Deep Supervision: Auxiliary outputs at multiple decoder levels to improve gradient flow during training

2.2.2 Input Processing

The model processes all four MRI modalities simultaneously as a 4-channel input volume. Pre-processing steps include:

Data Preparation:

• Loading of all four MRI sequences (FLAIR, T1, T1ce, T2) from the BraTS validation set

• Intensity normalization: Z-score normalization per modality to standardize intensity distributions

• Resampling: Ensuring consistent voxel spacing across all modalities

• Padding/Cropping: Adjusting volumes to network-compatible dimensions

2.2.3 Inference and Output

The inference process is executed using the following command structure:

```
nnUNetv2_predict -i imageTsv/ -o predictions/ -d 500 -c 3d_fullres -f 0
```

where parameters specify input directory (-i), output directory (-o), dataset ID (-d), configuration (-c), and fold (-f).

Output Characteristics:

• Format: 3D segmentation masks in NIfTI format

• Labels: Multi-class segmentation distinguishing necrotic core (Label 1), edema (Label 2), and enhancing tumor (Label 4)

• Dimensions: Matches input dimensions (240×240×155 voxels)

• Preservation: Maintains tumor morphology and regional structure as captured in the original patient scan

2.3 Brain Tissue Detection in Prototype

To integrate the segmented tumor into a standardized visualization environment, a brain prototype (dummy brain model) is utilized. This prototype serves as the anatomical reference into which patient-specific tumor data is integrated.

2.3.1 Prototype Characteristics

The brain prototype is a standardized anatomical model representing average brain structure. Key properties include:

• Anatomical Accuracy: Based on averaged MRI data from subjects

• Standard Space: Aligned to a common coordinate system for consistent spatial referencing

• Tissue Segmentation: Pre-segmented into gray matter, white matter, and cerebrospinal fluid regions

• File Format: NIfTI (.nii) format for compatibility with medical imaging tools

2.3.2 Brain Tissue Extraction Method

We employ Otsu thresholding combined with morphological operations to extract brain tissue from the prototype. This classical computer vision approach provides robust tissue detection without requiring deep learning models.

Algorithm Steps:

1. Image Loading: Load the brain prototype in NIfTI format

2. Intensity Normalization: Standardize intensity values to a common range

3. Otsu Thresholding: Automatically determine optimal threshold to separate brain tissue from background based on intensity histogram bimodality

4. Morphological Operations: Apply erosion and dilation to remove noise and fill small gaps

5. Connected Component Analysis: Identify and retain the largest connected component as brain tissue

6. Binary Mask Generation: Create a binary mask delineating brain tissue regions

2.4 Spatial Alignment and Integration

A critical step in creating realistic visualizations is accurately aligning the patient-specific tumor mask with the brain prototype. This process ensures that the tumor appears in anatomically plausible locations within the prototype brain.

2.4.1 Spatial Registration Methodology

We implement a rigid transformation approach that preserves tumor morphology while repositioning it within the prototype brain. This method balances anatomical accuracy with computational efficiency.

Method: Center-of-Mass Alignment

The alignment process is based on computing and matching the centers of mass (CoM) of both the tumor and the brain prototype:

Step 1 - Compute Brain Center:

```
brain_center = center_of_mass(brain_tissue)
```

This calculates the geometric center of the brain prototype by averaging the coordinates of all brain tissue voxels.

Step 2 - Compute Tumor Center:

```
tumor_center = center_of_mass(tumor_mask)
```

This determines the geometric center of the segmented tumor volume.

Step 3 - Calculate Shift Vector:

```
shift_vector = brain_center - tumor_center
```

The shift vector represents the translation needed to move the tumor from its original position to the center of the prototype brain.

Step 4 - Apply Rigid Transformation:

```
aligned_mask = shift(tumor_mask, shift_vector)
```

The tumor mask is translated using the shift vector, maintaining its original shape and internal structure.

2.4.2 Rationale for Rigid Transformation

The choice of rigid transformation (translation only, no rotation or scaling) is justified by several factors:

Morphology Preservation: Rigid transformations preserve the exact shape and size of the tumor as extracted from the patient scan, maintaining clinical accuracy.

Visualization Priority: For educational and patient communication purposes, preserving tumor morphology is more important than achieving perfect anatomical registration.

Trade-offs: While more sophisticated deformable registration methods (such as Advanced Normalization Tools - ANTs, or SimpleElastix) would provide better anatomical fit, they come with significantly higher computational costs and risk distorting tumor characteristics. These advanced methods would be more appropriate for surgical planning applications where precise anatomical correspondence is critical.

2.5 Positioning Tumor to Brain Tissue

After spatial alignment, the tumor mask is constrained to appear only within brain tissue regions of the prototype. This step ensures anatomical plausibility by preventing tumor voxels from appearing in non-brain areas.

Implementation:

```
mask_constrained = mask_aligned * brain_mask
```

This element-wise multiplication ensures that only voxels that are both part of the tumor and within brain tissue are retained in the final mask.

2.6 Intensity Enhancement for Visualization

To maximize tumor visibility in volume rendering, we apply region-specific intensity enhancement. This technique mimics the appearance of contrast-enhanced MRI where active tumor regions accumulate gadolinium and appear bright.

2.6.1 Multi-Region Enhancement Strategy

Different tumor sub-regions are enhanced with varying intensities to reflect their biological characteristics and improve visual discrimination:

Region	Original Intensity	Enhanced Intensity	Contrast Ratio
Brain Tissue	$\mu \pm \sigma$	$\mu \pm \sigma$	1.0x (baseline)
Necrotic Core (Label 1)	-	$0.1 \times \mu$	0.1x (very dark)
Edema (Label 2)	-	$1.3 \times \mu$	1.3x (slightly bright)
Enhancing Tumor (Label 4)	-	$4.0 \times \max$	4.0x (ultra bright)

2.6.2 Justification

The extreme contrast (4x maximum intensity for enhancing tumor) ensures that tumor regions are clearly distinguishable in volume rendering, mimicking the appearance of contrast-enhanced MRI where gadolinium accumulation creates bright signals in active tumor tissue. This enhancement is critical for effective visualization in VR, where depth perception and spatial relationships must be immediately apparent.

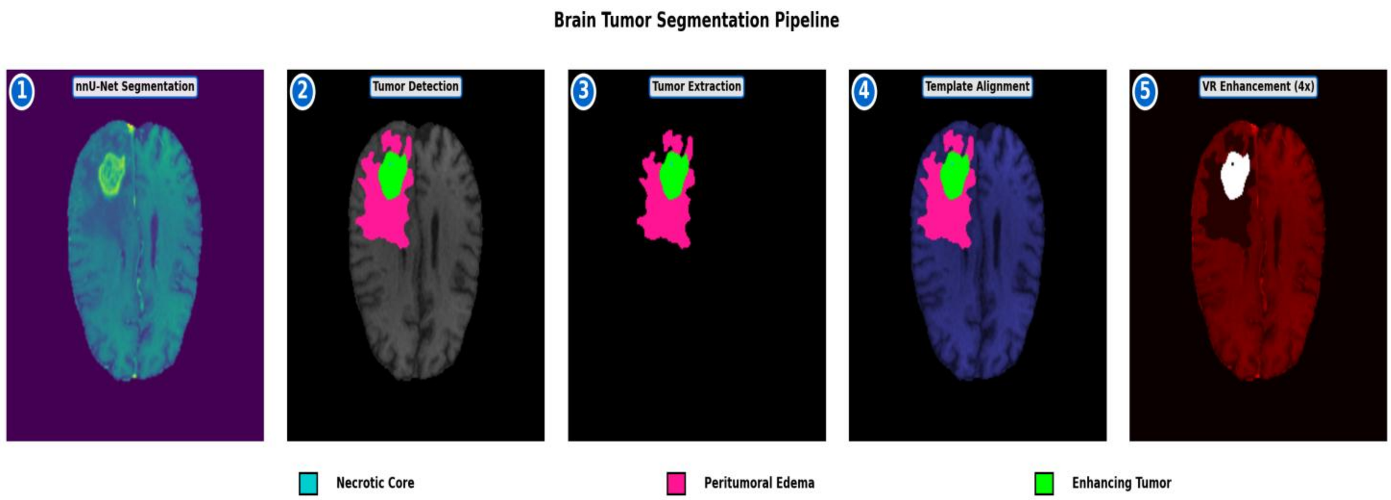
2.7 Multi-Scale Boundary Blending

To create smooth, realistic transitions between tumor and brain tissue, we implement multi-scale Gaussian blending. This approach prevents sharp, artificial-looking boundaries that would detract from visualization quality.

2.7.1 Rationale

This multi-scale approach prevents sharp artificial boundaries while mimicking realistic tumor infiltration patterns. Glioblastoma cells diffusely invade surrounding tissue rather than forming discrete borders, and our blending algorithm recreates this biological characteristic. The varying scales capture both fine-detail infiltration (small σ) and broader transition zones (large σ), producing visually convincing and biologically plausible results.

2.7.2 Figure Example (Segmentation Presentation)



3. Part II: VR Integration and Visualization

The second major component of the project involves converting the processed medical imaging data into a format suitable for real-time 3D rendering and implementing an immersive VR visualization system using Unity and Meta Quest 3.

3.1 Data Format Conversion

Medical imaging data from Part I exists in NIfTI format (.nii), which is not directly compatible with real-time 3D engines. We must convert this volumetric data into formats that Unity can efficiently load and render.

3.1.1 RAW Volume Format

RAW format represents volumetric data as a continuous stream of voxel values without compression or metadata. This simple format is ideal for direct GPU upload and volume rendering.

Conversion Process:

1. Load NIfTI Volume: Read the composite brain-with-tumor volume generated in Part I using nibabel library

2. Normalize Intensity: Scale voxel values to 0-255 range (8-bit unsigned integers) for efficient storage and rendering

3. Extract Voxel Array: Obtain the raw numpy array representing the 3D volume

4. Flatten and Export: Serialize the 3D array to a 1D byte stream and write to .raw file

3.2 Unity Integration

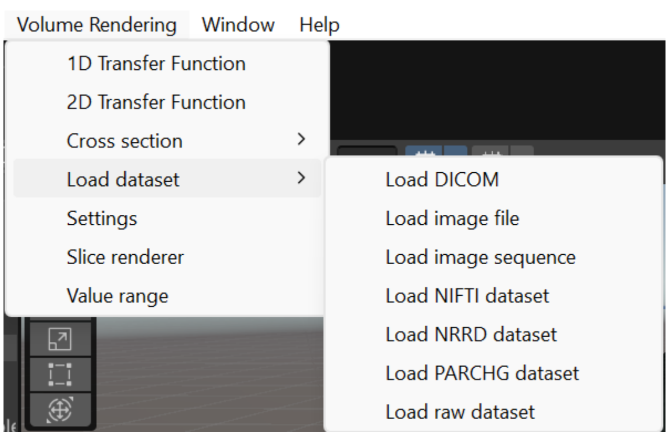
3.2.1 UnityVolumeRendering Plugin

Unity integration for this project is accomplished using the UnityVolumeRendering open-source plugin developed by Matias Lavik (mlavik1). This comprehensive volume rendering solution is specifically designed for medical imaging applications and provides robust support for multiple medical imaging formats including NIfTI (.nii), DICOM, NRRD, and RAW formats.

The UnityVolumeRendering plugin represents a mature, production-ready solution with over 520 stars on GitHub and active maintenance. It implements GPU-accelerated ray marching shaders optimized for real-time volume rendering, making it ideal for AR applications where performance is critical. The plugin has been successfully deployed in various medical visualization projects and educational applications, demonstrating its reliability and effectiveness.

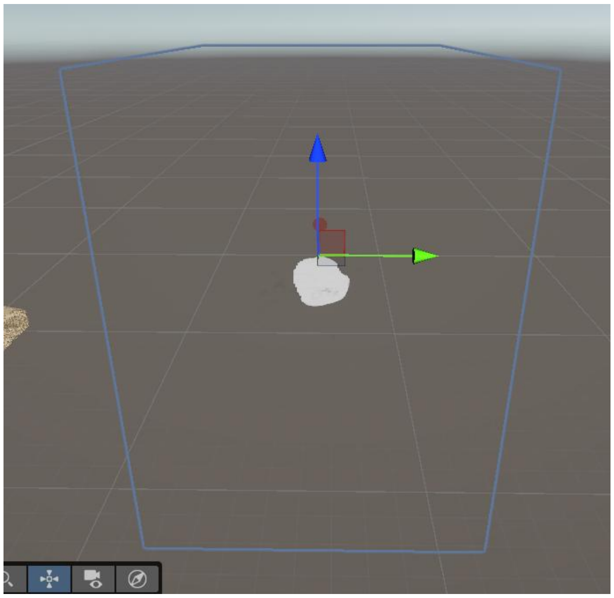
Key Features and Capabilities:

Native NIfTI Support: The plugin includes built-in importers for NIfTI-1 format (.nii and .nii.gz), which is the output format from our processing pipeline. This eliminates the need for manual format conversion and allows direct loading of the composite brain volumes generated in Part I.

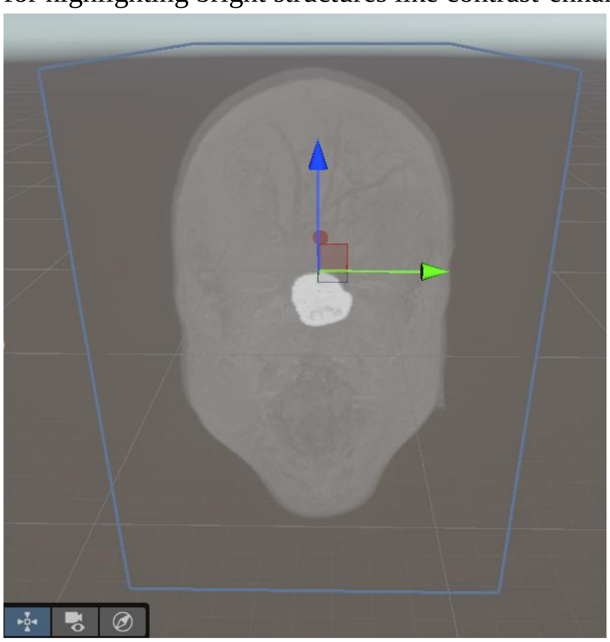


Multiple Rendering Modes: The plugin supports three primary rendering techniques that can be toggled in real-time:

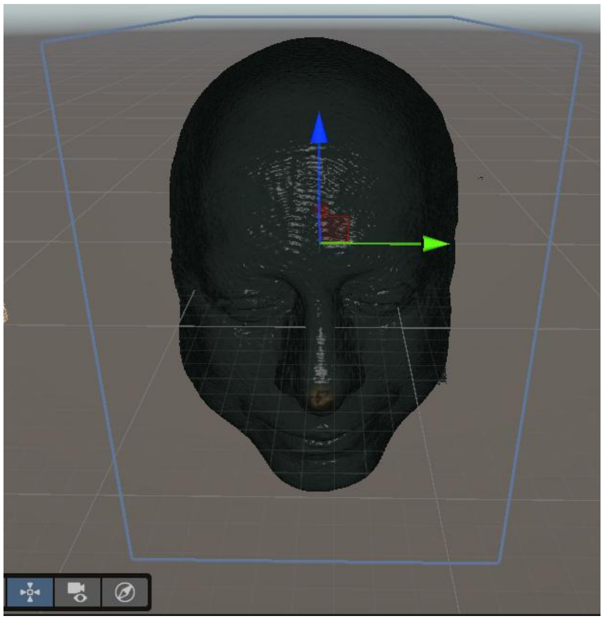
- Direct Volume Rendering (DVR): Uses 1D or 2D transfer functions to map voxel intensity to color and opacity, enabling detailed visualization of different tissue types and tumor regions.



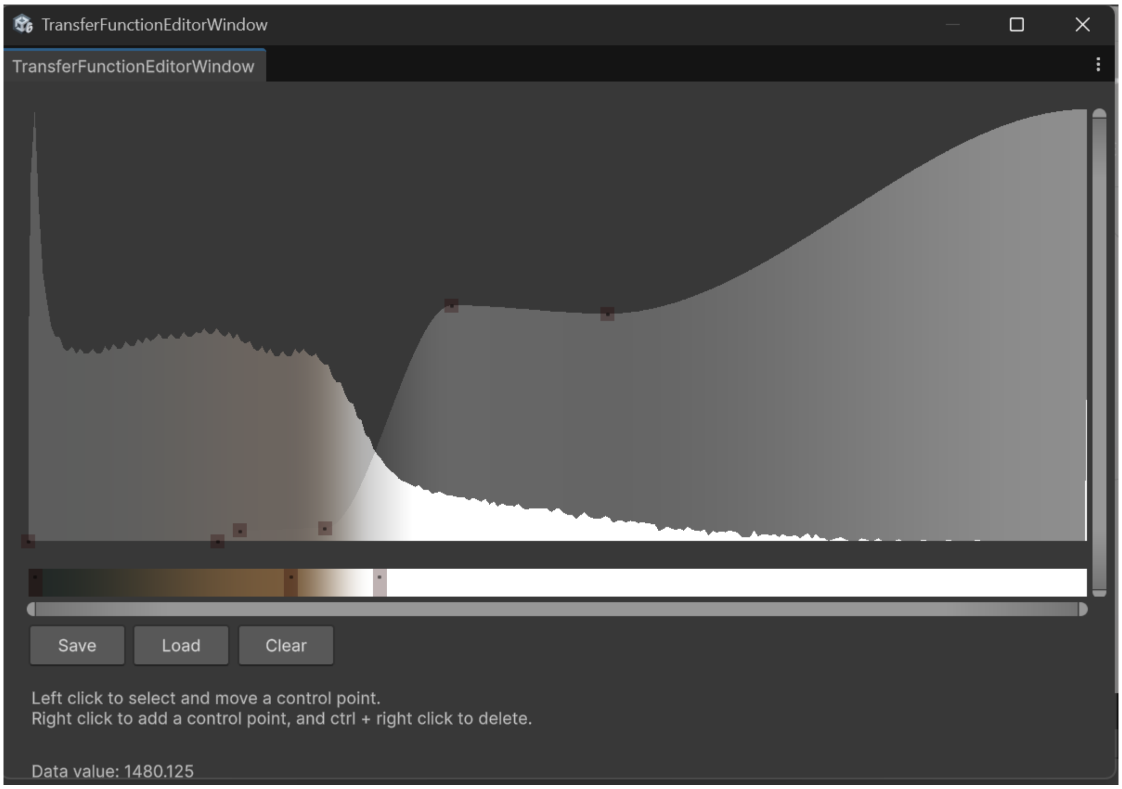
- Maximum Intensity Projection (MIP): Projects the maximum intensity along each ray, useful for highlighting bright structures like contrast-enhanced tumors.



- Isosurface Rendering: Renders surfaces at specific intensity thresholds, similar to traditional mesh-based rendering but computed directly from volume data.



Transfer Function Editor: Interactive editors for both 1D and 2D transfer functions allow real-time adjustment of visualization parameters. The 1D transfer function maps intensity to color and opacity using an intuitive curve editor, while the 2D transfer function additionally considers gradient magnitude to distinguish surfaces from homogeneous regions. This enables fine-tuned control over how tumor sub-regions appear in the visualization.



VR Compatibility: The plugin is fully compatible with Unity's XR framework and has been tested with various VR headset Meta Quest 3. For Quest 3 deployment, the plugin requires setting the stereo rendering mode to 'multi-pass' in Unity's XR settings to ensure proper stereoscopic rendering.

Advanced Rendering Features:

- Lighting and Shadows: Optional gradient-based lighting calculations provide depth cues and enhance 3D perception. Shadow volumes can be enabled for more realistic rendering, though at higher computational cost.
- Cubic Interpolation: Hardware-accelerated trilinear or tricubic interpolation of the 3D volume texture ensures smooth rendering without visible voxel artifacts.
- Early Ray Termination: Optimization technique that stops ray marching when accumulated opacity exceeds a threshold, significantly improving performance for opaque structures.
- Depth Writing: Proper depth buffer integration allows volume-rendered objects to correctly occlude and be occluded by other 3D geometry in the scene.

Data Pipeline Integration:

Integration Strategy to mitigate the computational overhead associated with parsing raw medical data on a mobile standalone headset, the proposed pipeline decouples data processing from visualization. All data acquisition and manipulation tasks are offloaded to a high-performance PC environment.

PC-Based Preprocessing (Host Environment) Raw NIfTI volumes are imported and processed exclusively within a desktop environment (via the Unity Editor or a dedicated companion application). This stage facilitates comprehensive preprocessing, including the modulation of transfer functions, volumetric cropping, and the refinement of segmentation masks. By leveraging the superior processing capabilities of a desktop workstation, this approach ensures the efficient handling of complex data conversion and scene configuration tasks prior to deployment.

Meta Quest 3 Deployment and Visualization Following optimization on the host PC, the finalized 3D assets are transferred to the Meta Quest 3 hardware. The Head-Mounted Display (HMD) application is architected strictly for the rendering and interactive visualization of these pre-processed volumes. Consequently, the device is relieved of resource-intensive runtime file I/O and NIfTI parsing operations, thereby guaranteeing high-fidelity visualization and stable frame rates during the clinical review process.

3.2.2 Transfer Function Design

Transfer functions map volumetric data intensities to visual properties (color and opacity). Proper transfer function design is critical for effective visualization of different tissue types and tumor regions.

3.2.3 Scene Setup

Unity Scene Components:

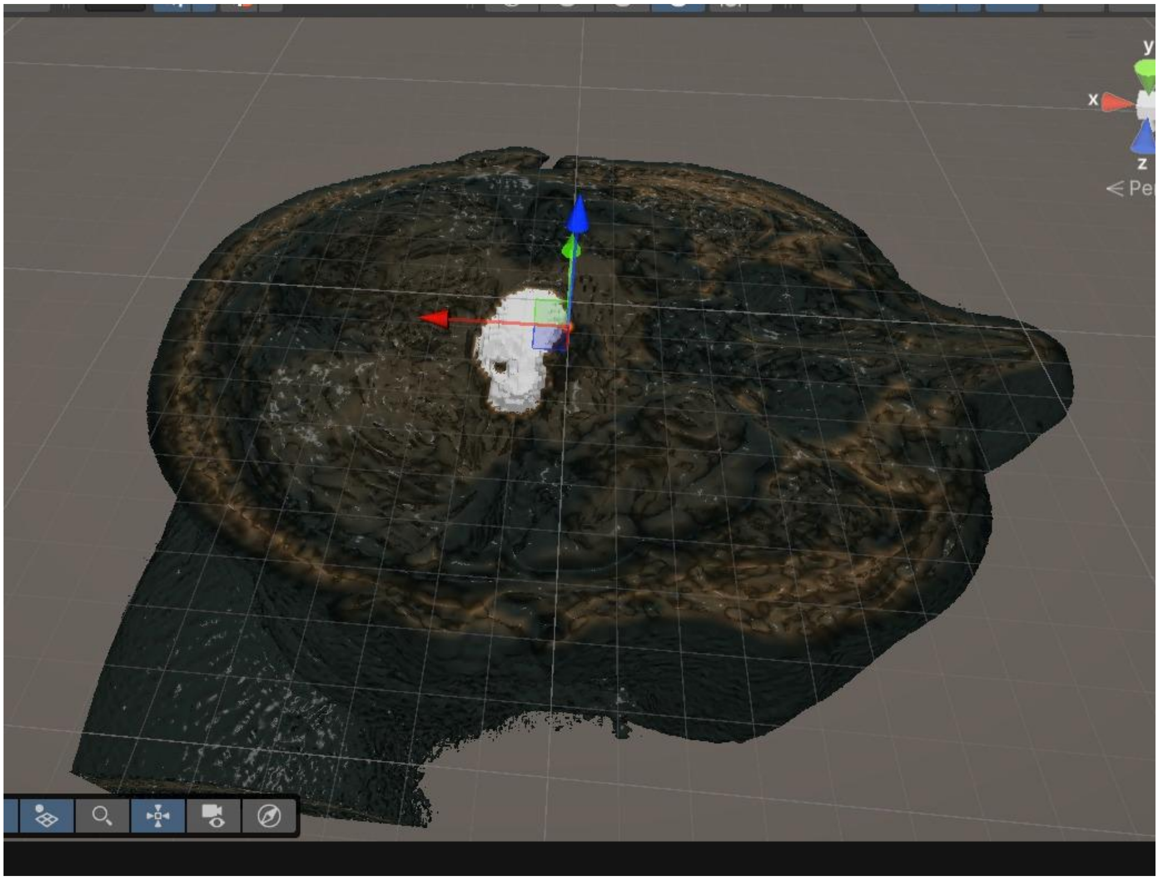
1. VR Camera Rig: XR Interaction Toolkit camera rig configured for Meta Quest 3

2. Volume Object: GameObject containing the loaded brain volume with volume renderer component

3. Lighting: Directional light for surface illumination of UI elements and spatial reference

4. Interaction Components: Ray interactors for hand controllers enabling object manipulation

3.2.4 Figure Example (Scene Image)



3.3 Meta Quest 3 VR Implementation

Meta Quest 3 is a standalone VR headset offering high-resolution displays, inside-out tracking, and powerful mobile processing. Its standalone nature eliminates the need for external computers or sensors, making it ideal for portable medical visualization applications.

3.3.1 Hardware Specifications (Meta Quest 3)

• Display: Dual LCD panels, 2064×2208 pixels per eye, 120Hz refresh rate

• Processor: Qualcomm Snapdragon XR2 Gen 2

• RAM: 8GB LPDDR5

• Tracking: Inside-out 6DOF tracking with 4 wide-angle cameras

• Controllers: Touch Plus controllers with haptic feedback

• Field of View: 110° horizontal, 96° vertical

3.3.2 Development Environment Setup

Required Software and SDKs:

1. Unity 6000.0.6 LTS or later (Long Term Support version for stability)

2. Meta XR SDK: Official Unity package providing Quest-specific features

3. XR Interaction Toolkit: Unity's cross-platform VR framework

4. Android Build Support: Required for deploying to Quest's Android-based OS

5. Meta Quest Developer Hub: For wireless deployment and debugging

Configuration Steps:

1. Install Unity with Android build support

2. Import Meta XR Tools package from Unity Asset Store

3. Configure project settings: Set platform to Android, set minimum API level to Android 10

4. Configure Oculus as active XR runtime

5. Enable developer mode on Quest 3 headset

6. Connect headset via USB or WiFi for deployment

3.3.3 Interaction Design

VR interaction mechanisms must be intuitive and comfortable for extended use. We implement the following interaction paradigms:

Navigation and Manipulation:

• Object Grabbing: Direct hand or controller interaction to grab, rotate, and scale the brain volume

• Scaling: Pinch gesture or button-based scaling to zoom in/out on regions of interest

Visualization Controls:

• Volumetric Slicing: Dynamic cross-sectioning mechanism to visualize internal anatomical structures and pathologies.

• Measurement Tools: Distance measurement between points for quantitative analysis

4. System Architecture

4.1 Overall Pipeline Architecture

The complete system consists of four main processing stages that transform raw MRI data into an immersive VR experience:

Stage 1: Tumor Extraction (nnU-Net)

• Input: BraTS Patient MRI (4 modalities: FLAIR, T1, T1ce, T2)

• Process: nnU-Net 3D Segmentation

• Output: Tumor mask with labels (0=background, 1=necrotic core, 2=edema, 4=enhancing tumor)

Stage 2: Template Preparation

• Input: Brain template (dummy.nii)

• Process: Brain tissue detection using Otsu thresholding and morphological operations

• Output: Binary brain mask delineating tissue regions

Stage 3: Integration

• Inputs: Tumor mask from Stage 1, Brain mask from Stage 2

• Sub-processes:

  - Resize tumor mask to match template dimensions

  - Align tumor center to brain center (CoM-based rigid registration)

  - Constrain tumor to brain tissue regions

  - Perform multi-scale Gaussian boundary blending

• Output: Composite brain volume with integrated synthetic tumor

Stage 4: Output Generation

• Input: Composite volume from Stage 3

• Processes:

  - Save as NIfTI format for medical imaging applications

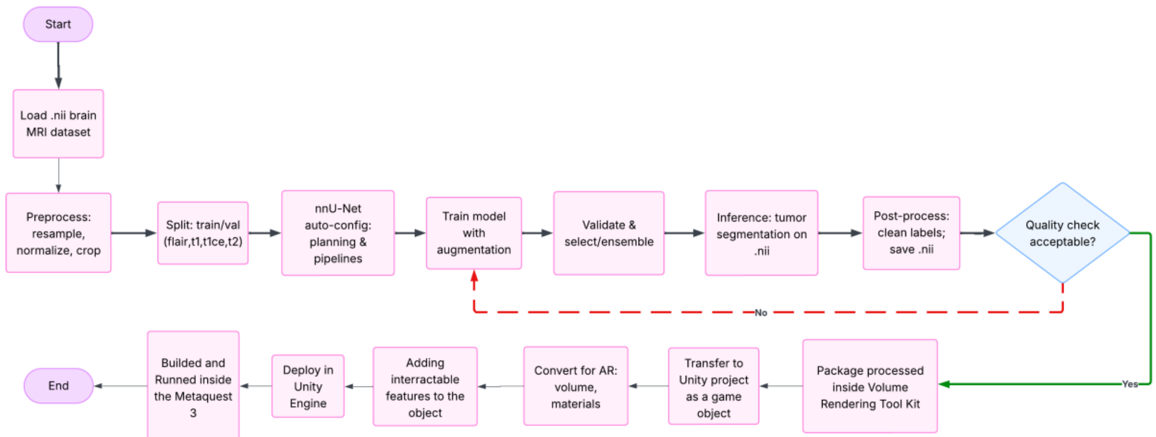
  - Convert to RAW format for Unity import

  - Generate preview images (2D slices)

- Export quality metrics

• Outputs: .nii file, .raw file, metadata.txt, preview images

4.1.1 Pipeline Flowchart



- **Start:** The workflow begins here.
- **Load .nii brain MRI dataset:** The system loads the raw medical brain scans in the NIfTI (.nii) file format.
- **Preprocess:** The images are standardized by resampling, normalizing pixel intensity, and cropping.
- **Split:** The data is divided into training and validation sets using four MRI modalities (FLAIR, T1, T1ce, T2).
- **nnU-Net auto-config:** The nnU-Net framework automatically configures the network architecture and planning based on the dataset.
- **Train model with augmentation:** The AI model is trained using data augmentation (rotating/flipping images) to improve learning.
- **Validate & select/ensemble:** The model is tested for accuracy, and the best version (or a combination of models) is selected.
- **Inference:** The trained AI performs the actual tumor segmentation (prediction) on the .nii files.
- **Post-process:** The output labels are cleaned up to remove noise, and the final segmentation is saved as a .nii file.
- **Quality check acceptable? (Diamond):** If the segmentation is good, proceed to visualization (Green line); if bad, go back and retrain (Red line).

- **Package processed inside Volume Rendering Tool Kit:** The final MRI data is packaged into a format compatible with the volumetric rendering tools.
- **Transfer to Unity project as a game object:** The packaged data is imported into the Unity software as a manipulatable 3D object.
- **Convert for AR:** The object's volume and surface materials are adjusted to look correct in an Augmented Reality environment.
- **Adding interactable features to the object:** Features are added to allow the user to grab, move, or rotate the brain model in 3D space.
- **Deploy in Unity Engine:** The Unity project is prepared and configured for deployment to the hardware.
- **Builded and Runned inside the Metaquest 3:** The application is compiled ("built") and launched on the Meta Quest 3 headset.
- **End:** The process is complete.

4.2 Technology Stack

Processing Pipeline (Python):

• Python 3.9-3.11

• PyTorch 1.12+ (deep learning framework)

• nnU-Net V2 (segmentation framework)

• nibabel (NIfTI file I/O)

• scikit-image (computer vision algorithms)

VR Application (Unity):

• Unity 6000.0 LTS

• C# scripting

• Meta XR Tools

• Meta XR SDK

• Unity Volume Rendering Plugin (mlavik1/UnityVolumeRendering)

Hardware:

• Processing Workstation: NVIDIA GPU (RTX 5060 or higher recommended), 24GB+ RAM

• VR Headset: Meta Quest 3

5. Conclusion

5.1 Summary of Methodology

This methodology report has presented a comprehensive framework for developing a VR-based brain tumor visualization system that bridges advanced medical imaging processing with immersive technology. The proposed system integrates two major components: automated tumor segmentation using state-of-the-art deep learning (Part I) and immersive VR visualization using Meta Quest 3 (Part II).

Part I establishes a robust processing pipeline that transforms multi-modal MRI data from the BraTS2020 dataset into high-quality tumor segmentations using nnU-Net architecture. The segmented tumors are then integrated into a standardized brain template through spatial alignment, intensity enhancement, and multi-scale boundary blending, creating anatomically plausible synthetic volumes suitable for visualization.

Part II details the conversion of medical imaging data to VR-compatible formats and the implementation of an immersive visualization environment in Unity. The system leverages Meta Quest 3's standalone VR capabilities to provide intuitive, interactive 3D exploration of brain tumor anatomy without requiring expensive workstations or tethered hardware.

5.2 Key Innovations

The proposed system introduces several innovative aspects:

1. End-to-End Automation: Complete pipeline from raw MRI scans to VR-ready volumes without manual intervention

2. Ultra-Visible Enhancement: Novel intensity modulation scheme (4x contrast) specifically designed for volume rendering clarity

3. Multi-Scale Blending: Biologically-inspired boundary smoothing that mimics realistic tumor infiltration patterns

4. Standalone VR Deployment: First implementation targeting standalone mobile VR (Meta Quest 3) for medical volume visualization

5. Immersive Pre-operative Assessment: Designed to facilitate complex neurosurgical decision-making, the platform allows for precise visualization of deep brain structures, aiding in the determination of optimal surgical corridors and risk minimization.

6. Interactive Training & Communication: The system adapts its high-resolution rendering capabilities for medical education, providing an intuitive interface for teaching complex neuroanatomy to trainees and explaining surgical procedures to patients.

5.3 Expected Impact

The successful implementation of this system is expected to have significant impacts across multiple domains:

Medical Education: Medical students and residents will gain access to realistic, interactive 3D tumor cases that can be explored from any angle, providing superior spatial understanding compared to traditional textbook diagrams or 2D images.

Patient Communication: Patients and their families will benefit from intuitive visualizations that clearly demonstrate tumor location, size, and relationship to critical brain structures.

Clinical Workflow: Integrated into the neurosurgical planning pipeline, this system delivers rapid visualization tools that serve as a robust augmentation to traditional imaging modalities.

References

1. Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., & Maier-Hein, K. H. (2021). nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature Methods, 18(2), 203-211.

2. Menze, B. H., Jakab, A., Bauer, S., et al. (2015). The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE Transactions on Medical Imaging, 34(10), 1993-2024.

3. Bakas, S., Akbari, H., Sotiras, A., et al. (2017). Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features. Scientific Data, 4, 170117.

4. Unity Technologies. (2023). Unity Real-Time Development Platform. Retrieved from https://unity.com

5. Meta Platforms, Inc. (2024). Meta Quest 3 Technical Specifications. Retrieved from https://www.meta.com/quest/quest-3/

6. Anwar, M., et al. (2023). Brain Tumor Segmentation in Multimodal MRI Using U-Net Layered Structure. Computers, Materials & Continua, 74(3), 5267–5281.

7. Wang, Z., Zhang, Z., Liu, J., & Yi, X. (2023). Research on Segmentation Method of Brain Tumor Image Based on Deep Learning. 2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS), 118–122.

8. Yan, B.B. et al. (2022). MRI Brain Tumor Segmentation Using Deep Encoder-Decoder Convolutional Neural Networks. In: Crimi, A., Bakas, S. (eds) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2021. Lecture Notes in Computer Science, vol 12963. Springer, Cham.

9. Zhang, F., Sun, Z., Wang, T. (2022). Brain Modeling for Surgical Training on the Basis of Unity 3D. In: Yang, S., Lu, H. (eds) Artificial Intelligence and Robotics. ISAIR 2022. Communications in Computer and Information Science, vol 1701. Springer, Singapore.

10. H. Gore, U. Chaskar, B. Borotikar and K. Bhole, "Medical Image Visualization using Augmented Reality," 2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI), Wardha, India, 2024, pp. 1-6

11. Abidin, A. Z., Naqvi, R. A., Haider, A., Kim, D. Y., Jeong, J. W., & Lee, S. W. (2024). Recent deep learning-based brain tumor segmentation models using multi-modality magnetic resonance imaging: a prospective survey. Frontiers in Bioengineering and Biotechnology, 12, 1392807.

12. Queisner, M., & Eisenträger, K. (2024). Surgical planning in virtual reality: a systematic review. Journal of Medical Imaging, 11(6), 062603.

13. Ahsan, R., Shahzadi, I., Najeeb, F., & Omer, H. (2024). Brain tumor detection and segmentation using deep learning. Magnetic Resonance Materials in Physics, Biology and Medicine, 38(1), 13-22.

14. Hoebel, K. V., Patel, J. B., Beers, A. L., et al. (2024). A review of deep learning for brain tumor analysis in MRI. npj Precision Oncology, 8, 5.

15. Peng, M. J., Chen, H. Y., Leung, F., et al. (2024). Virtual reality-based surgical planning simulator for tumorous resection in FreeForm Modeling: an illustrative case of clinical teaching. Quantitative Imaging in Medicine and Surgery, 14(2), 2060-2068.

16. Ujiie, H., Kato, T., Hu, H. P., et al. (2024). Developing a Virtual Reality Simulation System for Preoperative Planning of Robotic-Assisted Thoracic Surgery. Applied Sciences, 14(3), 973.

17. Bakhuis, W., Sadeghi, A. H., Moes, I., et al. (2023). Preparing for the future of cardiothoracic surgery with virtual reality simulation and surgical planning: a narrative review. Shanghai Chest, 7, 27.

18. Kersten-Oertel, M., Chen, S. J., Drouin, S., Sinclair, D. S., & Collins, D. L. (2019). Virtual interaction and visualisation of 3D medical imaging data with VTK and Unity. Healthcare Technology Letters, 6(1), 1-6.

19. Zhang, Y., Tang, W., Chen, S., et al. (2023). Online view enhancement for exploration inside medical volumetric data using virtual reality. Computers in Biology and Medicine, 163, 107193.

20. Khan, M. F., Iftikhar, A., Anwar, H., & Ramay, S. A. (2024). Brain Tumor Segmentation and Classification using Optimized Deep Learning. Journal of Computing & Biomedical Informatics, 7(01), 632-640.