MRI Explainer for Pediatric Brain Tumors: Segmentation, Saliency, and Slice-Level Captions

Yinuo Zhang
MSML 640 – Project Report

December 10, 2025

Abstract

Pediatric brain MRIs are difficult for non-experts to interpret: even when a segmentation model performs well, the outputs are often just probability maps and binary masks. This project builds a compact, end-to-end "MRI explainer" pipeline for pediatric brain tumor imaging using the ASNR–MICCAI BraTS 2023 Pediatric (BraTS-PED) dataset. The system converts 3D T2-FLAIR volumes into 2D slices, trains a lightweight 2D U-Net for whole-tumor segmentation, applies Grad-CAM on encoder features to highlight influential regions, and finally generates short, rule-based captions that describe the size and location of the highlighted area on each slice.

On a subject-level validation split of the BraTS-PED training data, the model achieves a mean slice-wise Dice coefficient of approximately 0.80 for whole-tumor segmentation (1377 validation slices) after training for 10 epochs on 5322 training slices using CPU-only hardware. Qualitative examples show that the segmentation and saliency maps are often well-aligned with the ground-truth tumor regions, while failure cases reveal typical under- and over-segmentation patterns. The project demonstrates how a relatively simple architecture and training setup can be extended into an interpretable, slice-level explainer suitable for interactive exploration via a Gradio web interface.

1 Introduction

Brain tumors in pediatric patients are commonly monitored using MRI, but understanding the resulting images is challenging for families and non-specialists. In machine learning research, most work focuses on improving segmentation metrics such as Dice similarity, yet the outputs—logits, probability maps, and masks—are not designed to be communicated to non-experts.

This project explores a more *explainable* view of pediatric brain tumor MRIs by building a pipeline that, for each MRI slice, produces:

1. A tumor segmentation mask.
2. A saliency heatmap indicating where the model is focusing.
3. A short, human-readable caption describing the highlighted area.

The primary technical goals are:

- Implement a 2D U-Net for whole-tumor segmentation on the BraTS-PED dataset.
- Design Grad-CAM style saliency maps on encoder features to visualize model focus.

- Build a rule-based captioning module that translates mask and saliency information into simple verbal descriptions.
- Wrap everything in a small Gradio-based demo that allows interactive slice-by-slice exploration.

Although the resulting system is not intended as a clinical tool, it demonstrates how modern segmentation models can be connected to interpretability and user-facing explanations in a way that is more accessible for communication between technical and medical domains.

2 Dataset and Problem Setup

2.1 BraTS-PED Dataset

The project uses the ASNR–MICCAI BraTS 2023 Pediatric (BraTS-PED) dataset, which consists of multi-modal brain MRI volumes for pediatric patients with brain tumors. Each subject provides several modalities, stored as NIfTI volumes:

- T2-FLAIR or T2-F (e.g., `*-t2f.nii.gz`)
- T2-weighted (`*-t2w.nii.gz`)
- T1 native and T1 contrast (`*-t1n.nii.gz`, `*-t1c.nii.gz`)
- A segmentation file with voxel-wise tumor labels (`*-seg.nii.gz`)

In this project, T2-F (`t2f`) is used as the primary imaging modality because it provides good contrast between tumor and surrounding tissue. The segmentation label encodes tumor subregions; for simplicity, all non-zero labels are collapsed into a single "tumor vs. background" binary mask.

2.2 Train/Validation Split

Only the BraTS-PED training split (which contains segmentation labels) is used for supervised learning. A subject-level split is created:

- A subset of subjects (on the order of a few dozen) is reserved for validation.
- All slices from these subjects form the validation set.
- The remaining subjects form the training set.

After preprocessing and slice filtering (Section 3), this yields the following slice counts:

Split	# Slices
Training	5322
Validation	1377

Table 1: Slice-level dataset sizes after preprocessing and filtering.

The main supervised learning task is then:

$$\text{Input: } x \in \mathbb{R}^{256\times256} \quad \longrightarrow \quad \text{Output: } \hat{y} \in [0,1]^{256\times256},$$

where x is a normalized T2-F slice and $\hat{y}$ is a predicted probability map for tumor vs. background.

3 Preprocessing and Slice Extraction

All volume preprocessing and slice extraction is implemented in `src/data/prepare_slices.py`. The pipeline converts 3D NIfTI volumes into 2D slice-level PNGs suitable for training a 2D U-Net.

3.1 Normalization and Slicing

For each subject:

1. Load T2-F volume $V \in \mathbb{R}^{H \times W \times D}$ and the corresponding segmentation volume $S \in \{0,1\}^{H \times W \times D}$.
2. Apply per-volume intensity normalization:

$$V' = \frac{V - \mu_V}{\sigma_V + \varepsilon}, \qquad V'' = \frac{V' - \min(V')}{\max(V') - \min(V') + \varepsilon},$$

   where μ_V and σ_V are the mean and standard deviation, and ε is a small constant to avoid division by zero. V'' is thus approximately in $[0,1]$.
3. Slice along the axial axis (axis 2), yielding (V''_k, S_k) pairs for $k = 0, \ldots, D-1$.
4. For each slice pair (V''_k, S_k):
   - Resize the image to 256×256 pixels using bilinear interpolation.
   - Resize the mask to 256×256 using nearest-neighbor interpolation.

3.2 Slice Filtering

To avoid training on slices where there is almost no tumor, a minimal area filter is applied. Let M_k be the resized binary mask for slice k, and let $A_k = \sum_{i,j} M_k(i,j)$ be the number of tumor pixels in that slice. Slices with $A_k < A_{\min}$ are discarded. In the final configuration, $A_{\min}$ is set to a very small value (effectively keeping all slices with any non-zero tumor), but the option remains available to focus training on slices with more substantial lesions.

3.3 Output Structure

For each dataset split (train/validation), the script writes PNG images to:

```
data/slices_ped_t2f_{split}/images/
data/slices_ped_t2f_{split}/masks/
```

with file names of the form:

```
BraTS-PED-<SUBJECT>-slice_<K>.png
```

This layout is compatible with a simple PyTorch `Dataset` implementation that pairs image and mask PNGs by filename.

4 Model Architecture and Training

4.1 2D U-Net for Segmentation

The segmentation model is a standard 2D U-Net implemented in `src/models/unet.py`. The network takes a single-channel 256×256 input and outputs a single-channel 256×256 logit map.

The encoder consists of repeated `Conv--BatchNorm--ReLU` blocks with max pooling:

- Encoder levels: $32 \to 64 \to 128 \to 256 \to 512$ channels.
- Each level uses two 3×3 convolutions with padding 1.
- ReLU activations are used with `inplace=False` to avoid issues with backward hooks for Grad-CAM.

The decoder mirrors the encoder with transposed convolutions for upsampling and skip connections from the corresponding encoder feature maps. The final layer uses a 1×1 convolution to map back to a single-channel logit output.

4.2 Loss Function

The model is trained using a combination of:

- Binary cross-entropy with logits ($\mathcal{L}_{\text{BCE}}$).
- Soft Dice loss ($\mathcal{L}_{\text{Dice}}$).

For a predicted probability map p and ground-truth binary mask y, the soft Dice coefficient is:

$$\text{Dice}(p, y) = \frac{2 \sum_{i,j} p_{ij} y_{ij} + \varepsilon}{\sum_{i,j} p_{ij} + \sum_{i,j} y_{ij} + \varepsilon},$$

and the Dice loss is defined as $\mathcal{L}_{\text{Dice}} = 1 - \text{Dice}(p, y)$. The total training loss is:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{Dice}}.$$

4.3 Optimization and Hardware

All experiments are run on CPU due to GPU compatibility issues (the available GPU, an NVIDIA GeForce GTX 980M, is not supported by the installed PyTorch CUDA build). The training script (`src/train.py`) is configured as follows:

- Optimizer: Adam.
- Learning rate: 1×10^{-3}.
- Batch size: 4.
- Epochs: 10.
- Device: CPU (via `CUDA_VISIBLE_DEVICES=""`).

Each epoch takes approximately 45–60 minutes on CPU for 5322 training slices and 1377 validation slices.

4.4 Validation Metric: Dice Coefficient

For evaluation on the validation set, the predicted logits are passed through a sigmoid function and thresholded at 0.5 to obtain binary masks. For each slice, the Dice coefficient between predicted and ground-truth masks is computed, and the reported val_dice for an epoch is the mean Dice over all validation slices.

5 Interpretability via Grad-CAM

5.1 Grad-CAM on Encoder Features

To visualize which regions of a slice are most influential for the model's prediction, Grad-CAM is applied to a late encoder block. The implementation resides in `src/interpret/gradcam.py`. The main idea is:

1. Choose a target encoder layer (the final convolution in the last encoder block).
2. For an input slice x, compute feature maps $A \in \mathbb{R}^{C\times H'\times W'}$ at this layer and the final logits z.
3. Define a scalar target t as the mean predicted tumor probability over the output map.
4. Backpropagate $\frac{\partial t}{\partial A}$ to obtain gradients $G \in \mathbb{R}^{C\times H'\times W'}$.
5. Compute channel-wise weights by spatial averaging: $w_c = \frac{1}{H'W'}\sum_{i,j} G_{cij}$.
6. Form the class activation map:

$$\mathrm{CAM}(i,j) = \mathrm{ReLU}\left(\sum_c w_c A_{cij}\right).$$

7. Normalize CAM to $[0,1]$ and upsample to the pixel space of the input slice.

This CAM is used in two ways:

- As a saliency heatmap overlay (primarily in internal analysis, not the final overlay shown in the demo).
- As an input signal to the captioning module to decide whether the model's focus is concentrated or diffuse.

6 Rule-Based Caption Generation

6.1 Features for Captioning

The captioning module is implemented in `src/caption/templates.py`. For each slice, it receives:

- The normalized input image x.
- The predicted probability map p (or binary mask $\hat{y}$).
- Optionally, the Grad-CAM saliency map CAM.

From these inputs, it computes:

- The relative area of the highlighted region: $\text{area} = \frac{1}{HW}\sum_{i,j} \mathbf{1}(\hat{y}_{ij} = 1)$.
- The centroid of the predicted region (c_x, c_y) when the mask is non-empty.

- A coarse left/right/center classification based on c_x relative to the image width.
- Whether the saliency map is concentrated (e.g., proportion of pixels with CAM > 0.7).

6.2 Caption Template

Using these features, the module chooses phrases for:

- Size: "small highlighted area," "moderate highlighted area," or "larger highlighted area."
- Location: "on the left side," "on the right side," or "near the center."
- Saliency: an optional clause describing that the heatmap shows where the model is most focused.

A typical caption produced by this template is:

> "This slice shows a moderate highlighted area on the right side. The overlay marks tissue the model considers unusual. The heatmap shows regions the model found most influential."

While this is a simple rule-based system, it serves to illustrate how segmentation and saliency information can be translated into natural language.

7 Experiments and Results

7.1 Training Dynamics

Table 2 summarizes the training loss and validation Dice over 10 epochs for the final configuration trained on all available training slices.

Epoch	Train Loss	Val Dice
1	0.5519	0.5173
2	0.3830	0.6396
3	0.3453	0.6781
4	0.3158	0.7152
5	0.2913	0.7371
6	0.2733	0.7403
7	0.2620	0.7704
8	0.2426	0.7767
9	0.2427	0.7938
10	0.2293	0.8002

Table 2: Training loss and validation Dice over 10 epochs on the full slice training set (5322 train slices, 1377 validation slices).

The training loss decreases steadily across epochs, while the validation Dice improves from approximately 0.52 in the first epoch to 0.80 by epoch 10 (Figure 1). The improvement begins to plateau around epochs 8–10, suggesting that the model is approaching its best performance under this configuration.

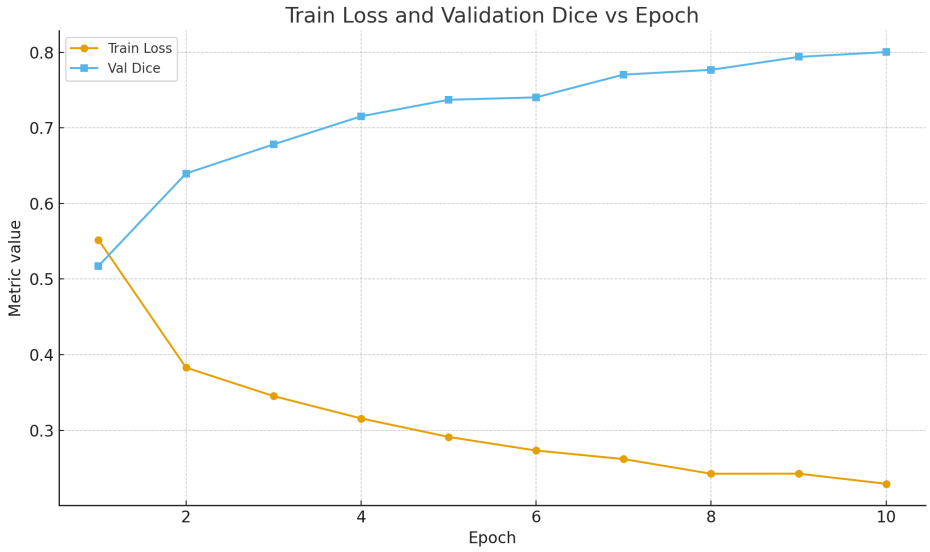

Figure 1: Training loss and validation Dice vs. epoch for the final 2D U-Net configuration.

7.2 Quantitative Performance

The final model achieves a mean slice-wise validation Dice of 0.8002 on the held-out validation set. For a lightweight 2D U-Net operating on single-modality T2-F slices, this is a strong baseline for whole-tumor segmentation in the pediatric setting.

No additional baselines were implemented in this project, as the main focus is the end-to-end explainer pipeline. However, the reported Dice provides a reasonable reference point for the quality of the segmentation outputs that underlie the saliency maps and captions.

8 Qualitative Analysis

8.1 Successful Example

One representative success case comes from subject `BraTS-PED-00016-000`, slice 29. In this slice, the model's predicted mask aligns closely with the ground-truth tumor region. When visualized as a grayscale brain slice with red prediction and green ground truth, the overlap region appears predominantly yellow, indicating strong agreement between prediction and ground truth.

The corresponding caption correctly identifies the highlighted region as a moderate area located near the center-right of the brain. This combination of accurate mask, coherent saliency map, and sensible caption demonstrates that the pipeline can provide an interpretable and internally consistent explanation for many slices.

8.2 High-Dice Example

Another slice, from subject `BraTS-PED-00079-000`, slice 29, also shows high Dice performance. The tumor is relatively compact and well-contrasted. The model's mask covers the core of the tumor with minimal false positives, and the Grad-CAM heatmap is concentrated over the same region.

This yields a near-ideal visual explanation: the red prediction, green ground truth, and yellow overlap essentially coincide, and the caption correctly describes the lesion as a localized highlighted area on one side of the brain.

8.3 Failure Example

A more challenging case is observed for subject `BraTS-PED-00108-000`, slice 42. In this slice, there is a clear mismatch between the predicted mask and ground truth:

- The model over-segments parts of the lesion, resulting in disjoint red-only and green-only regions.
- The Grad-CAM heatmap partially focuses on non-tumor structures.

These failure examples are informative: they illustrate that even when the overall validation Dice is high, individual slices can contain meaningful errors that would be important in a real-world setting.

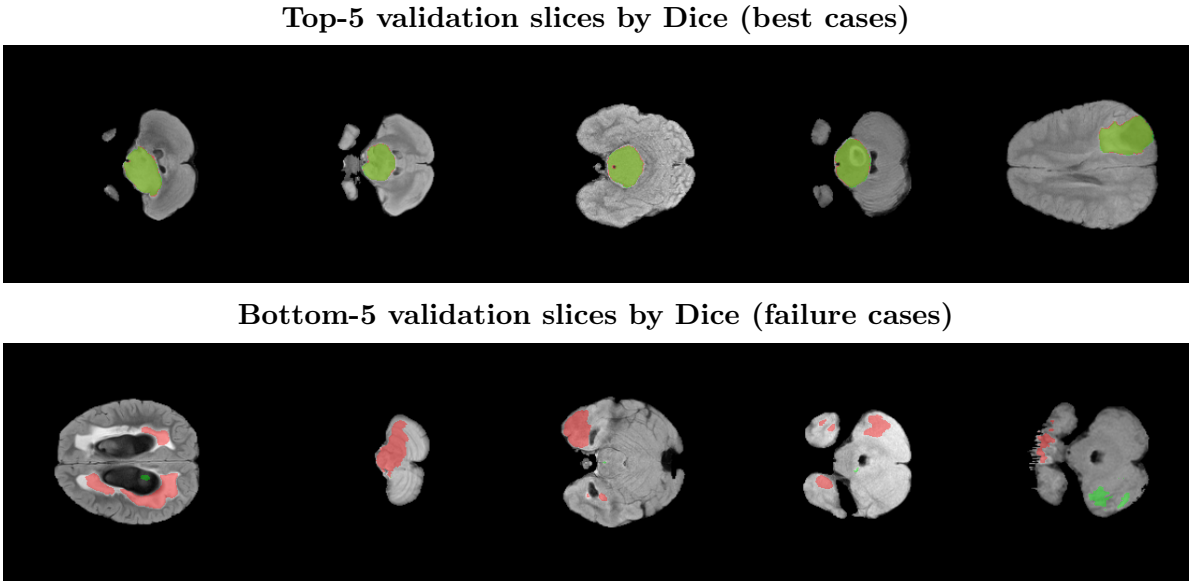

Figure 2: Top-5 and bottom-5 validation slices by Dice score. Each panel shows the T2-F slice with prediction in red, and ground truth in green. Best cases show large overlapped regions with minimal disagreement, while worst cases reveal under-segmentation, over-segmentation, and shape errors.

9 Interactive Demo

9.1 Gradio Application

To make the explainer pipeline easy to use, an interactive web demo is implemented with Gradio in `demo/app.py`. The interface supports:

- Uploading a single-modality MRI volume (e.g., a T2-F NIfTI file).
- Optionally uploading a trained checkpoint (`.pt`) file.

- Browsing slices along the axial axis using a slider.
- Viewing a gray-scale slice with a red segmentation overlay, and a green overlay for ground-truth if a BraTS-style `*-seg.nii.gz` file is present.
- Reading the generated caption for the currently selected slice.

Internally, the app:

1. Loads and normalizes the NIfTI volume.
2. Looks for a segmentation volume with a matching subject prefix.
3. Runs the 2D U-Net and Grad-CAM for each slice.
4. Stores all slices, masks, saliency maps, and ground-truth masks in a shared state to enable efficient slider updates.

9.2 Use Cases

The demo supports several use cases:

- Rapid visual inspection of how the model behaves across all slices of a volume.
- Qualitative debugging of segmentation and saliency alignment.
- Showing example slices to non-experts with accompanying captions.

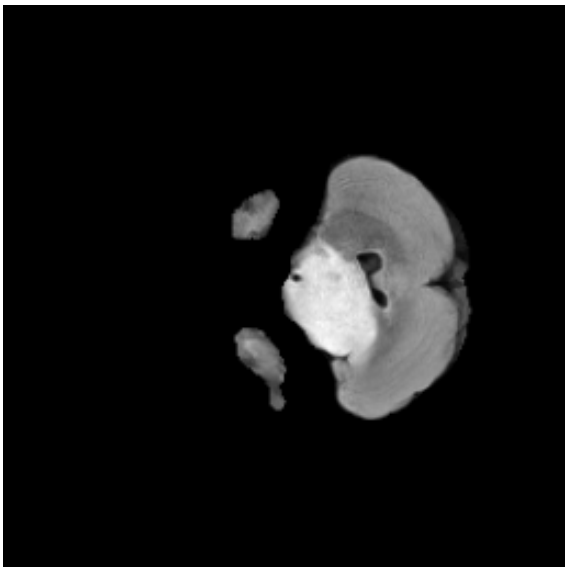

Figure 3: (a) Original T2-F slice
BraTS-PED-00060-000, slice 29

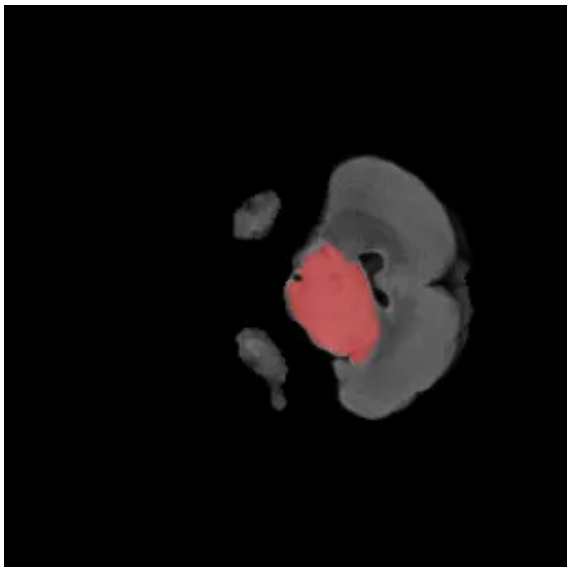

Figure 4: (b) Demo prediction overlay
red = model prediction

Figure 5: Example of the Gradio demo output for BraTS-PED-00060-000, slice 29. The user can browse slices, inspect the segmentation overlay, and read the generated caption for each slice.

10 Discussion and Limitations

10.1 Modeling Limitations

The current system has several modeling limitations:

- **2D-only context:** The model processes each slice independently and does not exploit 3D context across slices. Tumor structures that span multiple slices could be better modeled with 3D U-Nets or 2.5D approaches.
- **Single modality:** Only T2-F is used as input. Incorporating multiple modalities (e.g., T2-F, T1c) could improve segmentation performance and robustness.
- **Simple preprocessing:** The pipeline uses basic intensity normalization and resizing. More sophisticated preprocessing (e.g., skull stripping, bias-field correction, cross-site harmonization) may further improve performance.

10.2 Interpretability Limitations

The interpretability components also have limitations:

- Grad-CAM is a coarse, feature-based method that highlights regions correlated with the model's output, not necessarily the true causal features.
- Saliency maps can be noisy or shift with small changes in the model or input, which can reduce trust in individual explanations.
- The rule-based captioning system is limited to describing size and location; it does not capture richer information such as shape, texture, or clinical context.

11 Future Work

Several directions for future improvement are clear:

- **3D modeling:** Replace the 2D U-Net with a 3D U-Net or 2.5D approach that considers multiple adjacent slices as input, potentially improving segmentation at tumor boundaries.
- **Multimodal input:** Incorporate multiple MRI sequences (e.g., T2-F, T1c) to capture complementary information.
- **Alternative interpretability methods:** Evaluate methods such as integrated gradients, occlusion sensitivity, or attention-based approaches and compare them against Grad-CAM.
- **Learned captioning:** Use a small vision-language model or a lightweight transformer to generate captions conditioned on the image and masks, potentially allowing more flexible, patient-tailored explanations.
- **User studies:** Gather structured feedback from clinicians about which visualizations and captions are most useful, and iteratively refine the explainer design based on that feedback.

12 Conclusion

This project presents an end-to-end MRI explainer for pediatric brain tumors built on top of the BraTS-PED dataset. Starting from 3D T2-F volumes, the system produces 2D slice-level segmentations, Grad-CAM saliency maps, and simple captions that describe the highlighted regions. A lightweight 2D U-Net trained on 5322 slices and validated on 1377 slices achieves a mean validation Dice of approximately 0.80, demonstrating solid segmentation quality for a single-modality, 2D baseline.

Beyond the numeric performance, the project shows how segmentation, interpretability, and natural language can be combined into a compact, interactive tool. While many limitations remain, espe-

cially in modeling and captioning, this pipeline offers a practical foundation for more interpretable MRI-based systems in pediatric neuro-oncology research.