# **Results**

| **Model** | **Accuracy** | **F1 Score** | **Notes** |
| --- | --- | --- | --- |
| Multi-Modal (Multi-Layer Perceptron & ALexNet) | 0.8867 | 0.5538 | * i) Trained on metadata using MLP; ii) trained on images using ALexNet * i) Input Metadata features: Age at MRI, Sex, MGMT status, IDH, 1p/19q, MGMT index, OS, with WHO Grade as the target; ii) 2D axial MRI slices as input images * i)Categorical variables were mapped to integers; numeric features standardized using z-score; ii) MRI slices were z-score normalized and resized to 128×128 * i) MLP architecture: 2 hidden layers with ReLU, trained for 50 epochs with Adam optimizer; ii) AlexNet architecture: 5 convolutional layers, 3 fully connected layers with ReLU and dropout, trained for 20 epochs with Adam optimizer |
| AlexNet | 0.89 | 0.7811 | * Trained on 2D axial MRI slices from the T1c modality only * For each patient, selected 2–3 central tumor-containing slices using segmentation masks:   + Identified all axial slices with non-zero tumor   + Chose the middle slice + one before and after (if available) * Slices were z-score normalized and resized to 128×128 * Images loaded as grayscale (in\_channels=1) and passed to a modified AlexNet with a 4-class output (WHO Grade) |
| Multi-Modal MRI Model | / | / | * Add other MRI modalities like FLAIR and T2, assuming patients have them * Stack them as multi-channel input (e.g., [3, 128, 128] for T1c + FLAIR + T2) * Feed into a 2D CNN (e.g., ResNet18) |

# 

# **Overview**

**Dataset**

* UCSF-PDGM dataset: Imported into Google Drive
* Used T1c modality and corresponding tumor segmentation mask for now but can expand
* Used metadata for demographic and diagnostic information

**Patient Validation**

* Wrote a loop to scan all patient folders and defined a “valid patient” using the following criteria:
  + Must have both T1c.nii.gz and tumor\_segementation.nii.gz files
  + Files aren’t empty or corrupted
  + MRI and mask must have matching 3D shapes
    - Want to make sure we’re not measuring tumor voxels in the wrong locations, getting incorrect intensity values, etc
  + Tumor mask must contain non-zero voxels (i.e. there is a tumor)
* Led to 500 valid patients

**Metadata Preprocessing**

**MRI Preprocessing**

* Create a custom BrainTumorDataset class to:
  + Load T1c scans and tumor masks using nibabel
  + Normalize MRI intensity using z-score normalization
  + Convert images and masks into PyTorch tensors

**In case we want to use: Feature Extraction from MRI scans**

| **Feature** | **Description** |
| --- | --- |
| tumor\_volume | Total number of voxels labeled as tumor (volume of the tumor) |
| tumor\_mean\_intensity | Mean intensity in the T1c image restricted to the tumor mask |
| tumor\_std\_intensity | Standard deviation of intensities inside the tumor region |
| bbox\_size\_x/y/z | Width, height, and depth of the tumor's 3D bounding box |
| centroid\_x/y/z | Spatial location of the tumor’s center of mass in 3D space |

## 