Brain Tumor Segmentation Using YOLOv11 and SAM2

# Overview

This project applies two advanced computer vision techniques—YOLOv11 (You Only Look Once version 11) and SAM2 (Segment Anything Model v2)—to detect and segment brain tumors from MRI images.

## Objectives

- Utilize YOLOv11 for object detection of brain tumors.

- Employ SAM2 for fine-grained segmentation.

- Train YOLOv11 on a custom brain tumor dataset.

- Evaluate segmentation performance visually and statistically.

- Present results through plots, visualizations, and code.

# Environment Setup

Platform: Google Colab

Language: Python

Libraries used: ultralytics, opencv-python, matplotlib, segment-anything, numpy, torch

## Setup Commands

!pip install ultralytics opencv-python matplotlib

!pip install git+https://github.com/facebookresearch/segment-anything.git

# Dataset Structure

/tumor detection/dataset/  
├── train/  
│ ├── images/  
│ └── labels/  
├── test/  
│ ├── images/  
│ └── labels/  
├── valid/  
│ ├── images/  
│ └── labels/

Images are in .jpg format. Labels are in YOLO format (class x\_center y\_center width height).

Classes:  
- 0: Glioma  
- 1: Meningioma  
- 2: Pituitary

# YOLOv11 Training

Model loading and training:

from ultralytics import YOLO  
  
model = YOLO("yolo11n.pt")  
  
results = model.train(  
 data="brain-tumor.yaml",  
 epochs=50,  
 imgsz=640,  
 batch=8,  
 name="yolov11\_brain\_tumor",  
 verbose=True  
)

Training Summary:  
- 50 epochs  
- Batch size: 8  
- Image size: 640  
- Model: YOLOv11n (nano)

# Inference with YOLOv11

results = model.predict(source="/path/to/test/images", save=True)

Result images are saved in runs/segment/predict/.

# Data Distribution Analysis

Class Distribution (Test Set) visualized using Bar and Pie charts.

# Bounding Box Visualization (YOLO Labels)

Bounding boxes drawn using cv2.rectangle and cv2.putText for validation.

# Segment Anything Model v2 (SAM2)

Model setup and usage:

from segment\_anything import sam\_model\_registry, SamPredictor  
  
sam = sam\_model\_registry["vit\_h"](checkpoint="sam\_checkpoints/sam\_vit\_h.pth")  
predictor = SamPredictor(sam)

Segmentation and prompt setup:

predictor.set\_image(image)  
masks, scores, \_ = predictor.predict(point\_coords=input\_point, ...)

# Visual Segmentation Results

Red contour overlays show tumor masks predicted by SAM2 using matplotlib.

# Final Observations

- YOLOv11 effectively localizes tumors.

- SAM2 precisely segments the tumor regions.

- Models are complementary: YOLO for detection, SAM for segmentation.

- Visualizations and metrics validate results.

# Folder Structure for GitHub

Brain-Tumor-Segmentation/  
├── yolov11\_training.ipynb  
├── sam2\_segmentation.ipynb  
├── brain-tumor.yaml  
├── runs/  
├── sam\_checkpoints/  
├── dataset/  
├── README.md

# References

- Ultralytics YOLOv11: https://github.com/ultralytics/ultralytics

- Segment Anything: https://github.com/facebookresearch/segment-anything

- Google Colab: https://colab.research.google.com/

- <https://youtu.be/pFIwBmlm2O4?si=4ZbaA0KfPf9OD3m6>

<https://youtu.be/rPOYIUiij90?si=PObaV0HFtb1Tutfb>

<https://youtu.be/UOoSw9VfdS4?si=2wQHgXssBJ7dMr0V>

<https://youtu.be/5er9ozQdjyk?si=NPFoUcxrzNmT3Y4Y>

<https://www.kaggle.com/code/aryashah2k/tutorial-yolo-v11-sam2-tumor-detection/notebook>