Tumor Detection in Medical Images Using YOLOv8 Object Detection

BOBORICA MIHAI-ALEXANDRU
Faculty of Mathematics and Computer Science
University of Bucharest
mihai-alexandru.boborica@s.unibuc.ro

***Abstract*—This project presents a deep learning approach for detecting tumors in medical imaging using the YOLOv8 object detection model. The model was trained and finetuned on a custom-labeled dataset following a strict train/validation/test split. Performance metrics and model behavior were evaluated, and an interface was developed using Streamlit to demonstrate live inference on uploaded images.**

I. Introduction

Early and accurate detection of tumors in medical images is a critical task in medical diagnostics. We leverage the power of YOLOv8—a real-time object detection model—for this purpose. The model was trained using labeled bounding box annotations on radiographic or pathology image datasets. The datasets were sourced from multiple domains including brain tumors and stroke imaging.

II. Dataset and Preparation

The dataset used in this project was sourced from:
https://www.med.upenn.edu/cbica/brats2020/data.html
https://www.nature.com/articles/s41597-022-01875-5
https://aimi.stanford.edu/datasets/brainmetshare
These include high-resolution images with tumor annotations. Images were preprocessed and split into three sets:

- **Training Set**: 70%
- **Validation Set**: 20%
- **Test Set**: 10%

Images were resized to 640×640 pixels, and labels were converted into YOLO-compatible text files with class ID and bounding box coordinates (center x, center y, width, height — all normalized).

III. Methodology

A. Model Architecture

We selected the YOLOv8m model from Ultralytics due to its balance between speed and accuracy. The model was initialized with pretrained weights and finetuned on our custom dataset.

B. Training Setup

Training was performed using the following parameters:

- **Model**: YOLOv8m
- **Epochs**: 100
- **Batch Size**: 4
- **Image Size**: 640
- **Optimizer**: SGD
- **Learning Rate**: 0.001
- **Device**: NVIDIA GTX 1650

Early stopping and data augmentation (flip, mosaic, scale) were applied.

The training script was implemented using the Ultralytics API:

```
from ultralytics import YOLO

def train_bmshare():
    model = YOLO('yolov8m.pt')

    model.train(
        data='brats_data.yaml',
        epochs=100,
        imgsz=640,
        batch=4,
        lr0=0.001,
        device=0,
        workers=4,
        project='runs/train',
        name='brats_yolov8m_advanced_on_brats',
        exist_ok=True,
        augment=True,
        patience=15,
        optimizer='SGD',
        save=True,
        verbose=True,
    )

if __name__ == '__main__':
    train_bmshare()
```

During training, I experimented with different parameters such as batch size and image resolution to investigate their effects on the model's performance and convergence speed.

IV. Experimental Results

TABLE I: Model Performance on Validation Set

Metric	Value
Precision	0.644
Recall	0.446
mAP@0.5	0.471
mAP@0.5:0.95	0.229

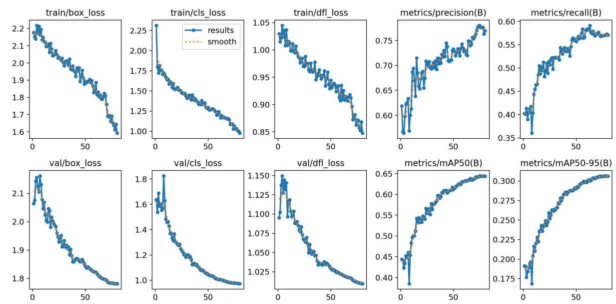

Fig. 1: Training and validation loss over 100 epochs

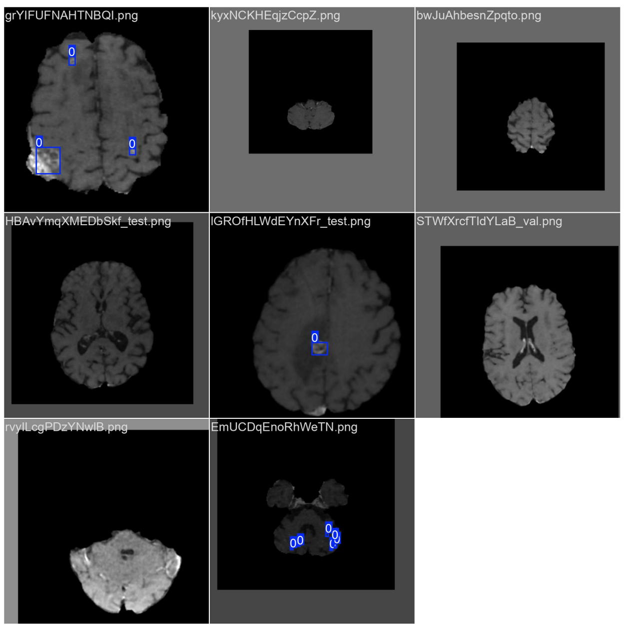

Fig. 2: Example of a training batch showing input MRI images along with bounding box annotations.

V. Interface

A lightweight web interface was built using Streamlit. Users can upload an image, and the trained YOLOv8 model will return predicted bounding boxes with class labels and confidence scores.

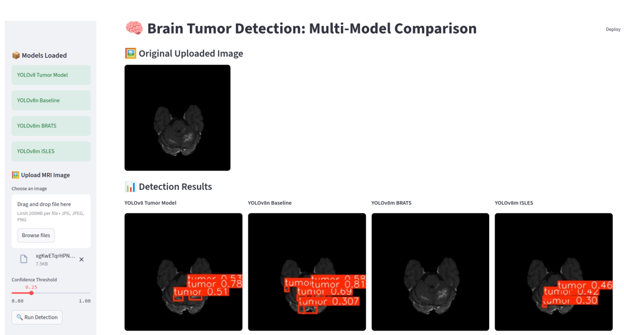

Fig. 3: Streamlit web interface for tumor detection

VI. Conclusion

This project demonstrates the effectiveness of YOLOv8 in detecting tumors in medical images. While the model achieved moderate precision, the recall and mAP scores suggest potential for improvement through further training or model scaling. Future work will focus on refining augmentations, hyperparameter tuning, and exploring segmentation models for more granular localization.

References

- G. Jocher et al., "Ultralytics YOLOv8," GitHub, 2023. https://github.com/ultralytics/ultralytics
- BMShare Dataset: https://aimi.stanford.edu/datasets/brainmetshare
- BraTS Dataset: https://www.med.upenn.edu/cbica/brats2020/data.html
- ISLES Dataset: https://www.nature.com/articles/s41597-022-01875-5
- Streamlit documentation: https://streamlit.io