

Report: Brain Tumor Detection Using YOLOv8

1. Preprocessing Steps

Dataset Description: The dataset is structured into two main folders:

- **train:** Contains MRI images for training, along with their respective labels.
- **valid:** Contains MRI images for validation, along with their respective labels.

The images represent two classes:

- **positive:** MRI images indicating the presence of a tumor.
- **negative:** MRI images without a tumor.

Each folder contains two subfolders: **images** (containing the MRI scans) and **labels** (containing corresponding annotation files in YOLO format).

Preprocessing Steps:

1. Image Resizing:

- All images are resized to a common dimension of 640x640 pixels for input into the YOLOv8 model.

2. Normalization:

- Pixel values were normalized to range between [0, 1] by dividing the pixel intensity by 255.

3. Augmentation (optional):

- Techniques such as flipping, scaling, and rotating were applied to increase the dataset's diversity and help the model generalize better.

4. Annotation:

- Each image has a corresponding **.txt** label file in YOLO format. Each line in the label file corresponds to an object in the image, and contains:
 - The class label (0 for "no tumor" and 1 for "tumor").
 - Bounding box coordinates normalized to the image size.

2. Model Architecture

Pre-trained Model Used: YOLOv8 (You Only Look Once, version 8)

- We used the **YOLOv8s** pre-trained model, which is lightweight and efficient for detecting objects in real-time.
- **Modification:**
 - The model's output layer was adjusted to fit the binary classification task (tumor vs no-tumor). The original YOLOv8 model was pre-trained on the COCO dataset, but we fine-tuned it to detect brain tumors from MRI images.

Why YOLOv8:

- **Speed:** YOLOv8 is known for its speed, which makes it suitable for real-time object detection.
- **Accuracy:** It balances accuracy and computational efficiency, especially when fine-tuned for specific tasks like medical image detection.

3. Training Process

Hyperparameters:

- **Batch size:** 16
- **Image size:** 640x640 pixels
- **Epochs:** 30
- **Learning rate:** 0.01
- **Optimizer:** SGD with momentum (0.937)
- **Loss functions:** Standard YOLO loss functions for object detection (bounding box regression, classification, objectness).

Data Augmentation:

- Random image augmentations, including flips and rotations, were applied during training to improve the model's robustness.

4. Evaluation Metrics

After training, the model was evaluated using various metrics to assess its performance:

1. Accuracy:

- Measures the proportion of correctly classified images.

2. Precision:

- $\text{Precision} = \text{True Positives} / (\text{True Positives} + \text{False Positives})$
- Indicates the model's ability to correctly identify tumors without including too many false positives.

3. Recall:

- $\text{Recall} = \text{True Positives} / (\text{True Positives} + \text{False Negatives})$
- Measures the model's ability to detect tumors (sensitivity).

4. F1-Score:

- $\text{F1-Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$
- The harmonic mean of precision and recall, providing a balance between the two.

5. Observations:

- The model showed high recall for tumor detection, but slightly lower precision, indicating it caught most tumors but included some false positives.

5. Instructions for Running the Code

1. Clone the Repository:

Clone the YOLOv8 repository to set up the environment for training:

2. Prepare the Dataset:

- Place your `train` and `valid` folders in the required format with subfolders for `images` and `labels` inside each.
- Ensure the `data.yaml` file is correctly configured with paths to these folders.

3. Train the Model:

Run the training command in your terminal:

4. Evaluate the Model:

After training, evaluate the model on the validation set:

5. Test the Model:

To test the model on a new set of images:

6. Conclusion

The YOLOv8 model was successfully fine-tuned to detect brain tumors from MRI images. The model showed promising results, with high recall, making it useful for medical applications where sensitivity is crucial. However, further tuning is required to improve precision and reduce false positives.