Data Set:

Downloaded the data set from Roboflow

@misc{

mri-rskcu_dataset,

title = { MRI Dataset },

type = { Open Source Dataset },

author = { Brain MRI },

howpublished = { \url{ https://universe.roboflow.com/brain-mri/mri-rskcu } },

url = { https://universe.roboflow.com/brain-mri/mri-rskcu },

journal = { Roboflow Universe },

publisher = { Roboflow },

year = { 2023 },

month = { jun },

note = { visited on 2024-07-17 },

}

Classes: Contain 2 classes

- **Eye**
- **Tumor**

YOLO V6:

Clone pretrained repository of Yolov6 from
https://github.com/meituan/YOLOv6

Train on custom data with

- Batch = 32
- Epochs = 100
- Img-size = 416
- Optimizer = SGD

Time Taken 35-40 minutes

Results of training:

```
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.549
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.888
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.656
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.193
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.593
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.506
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.642
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.669
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.509
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.686
Results saved to runs/train/yolov6s_results
Epoch: 99 | mAP@0.5: 0.888061679556391 | mAP@0.50:0.95: 0.5493450608610314
```

Results of Validation:

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.549
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.888
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.656
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.193
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.592
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.506
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.642
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.668
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.509
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.684
```

Conclusion of Yolov6s Base Model:

YoloV6 perform worst on grayscale Dataset. Not even able to predict the grayscale data.

YOLO V7:

Clone pretrained repository of Yolov6 from

https://github.com/WongKinYiu/yolov7

Train on custom data with

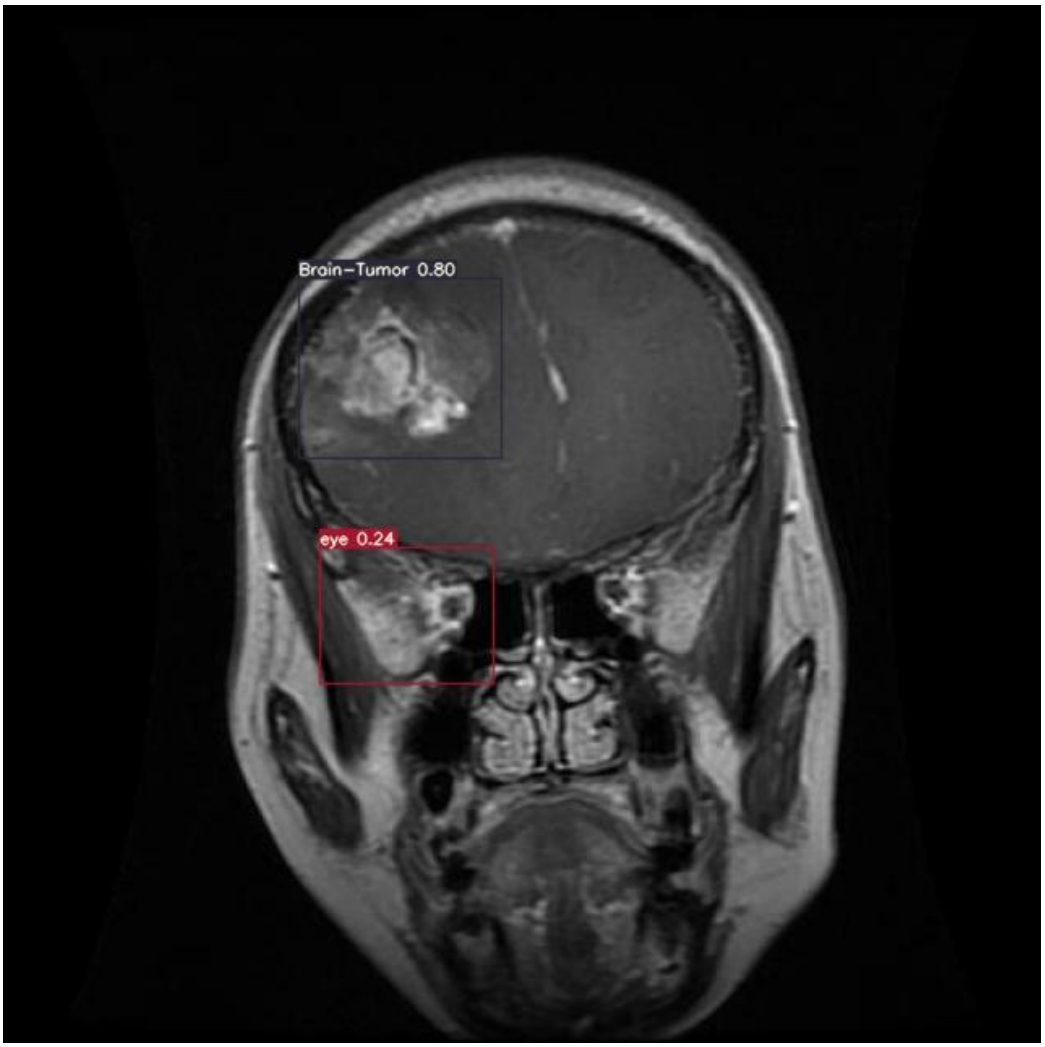
- Batch = 16
- Epochs = 55
- Img-size = 640
- Optimizer =adam

Time Taken 55-60 minutes

Results

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 5/5 [00:04<00:00, 1.01it/s]
all	151	178	0.935	0.823	0.875	0.55
Brain-Tumor	151	155	0.924	0.864	0.928	0.627
eye	151	23	0.946	0.783	0.823	0.473

Detection



Conclusion:

Need high computing power as the number of epochs increases from 55

Or batch size increase from 16

YOLO V8:

Clone pre trained repository of Yolov8 from https://github.com/ultralytics /ultralytics

And installed the yolo v8 from this official repository

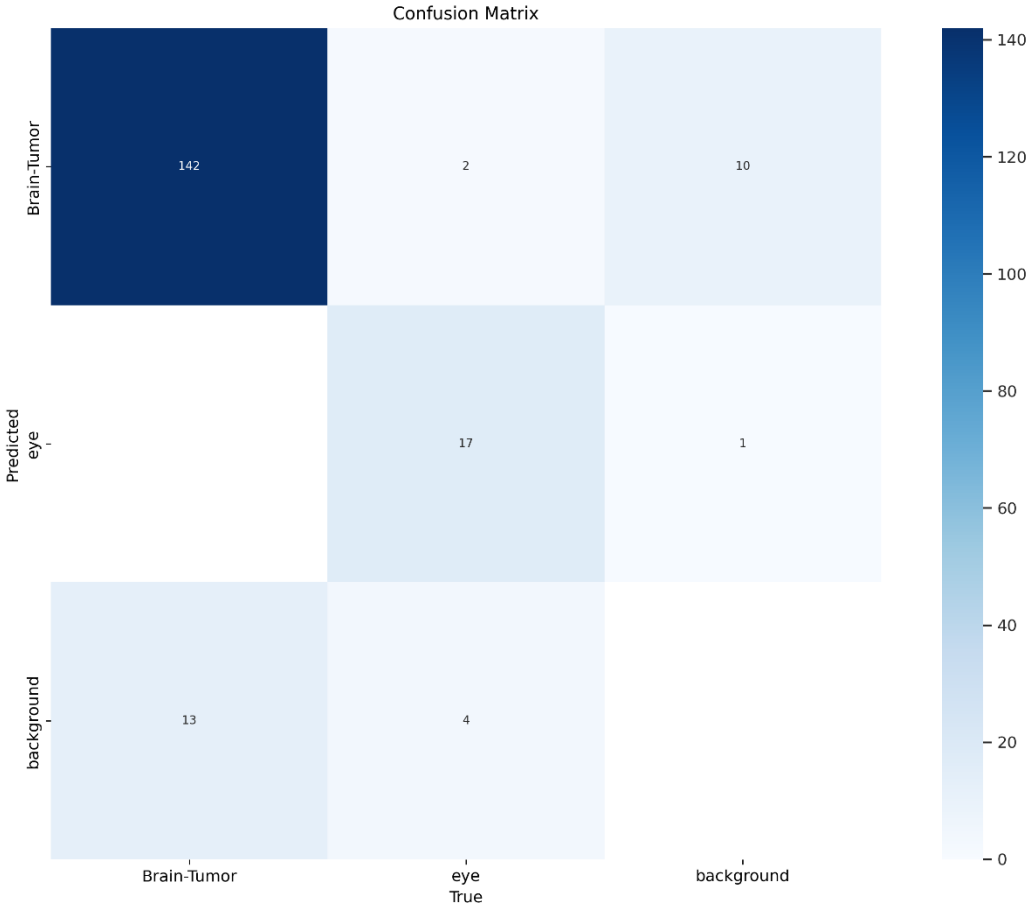
Train on custom data with

- Batch = 16
- Epochs = 25
- Img-size = 800
- Optimizer = ADAM-W

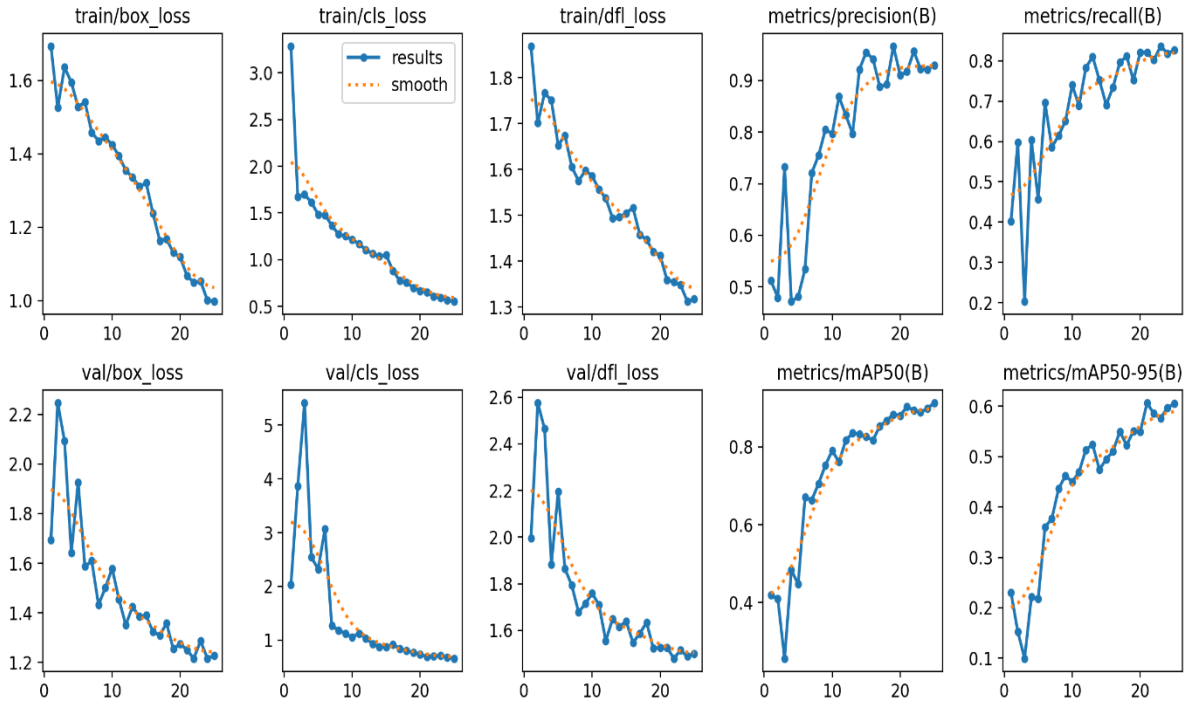
Time Taken 10-15 minutes

Results

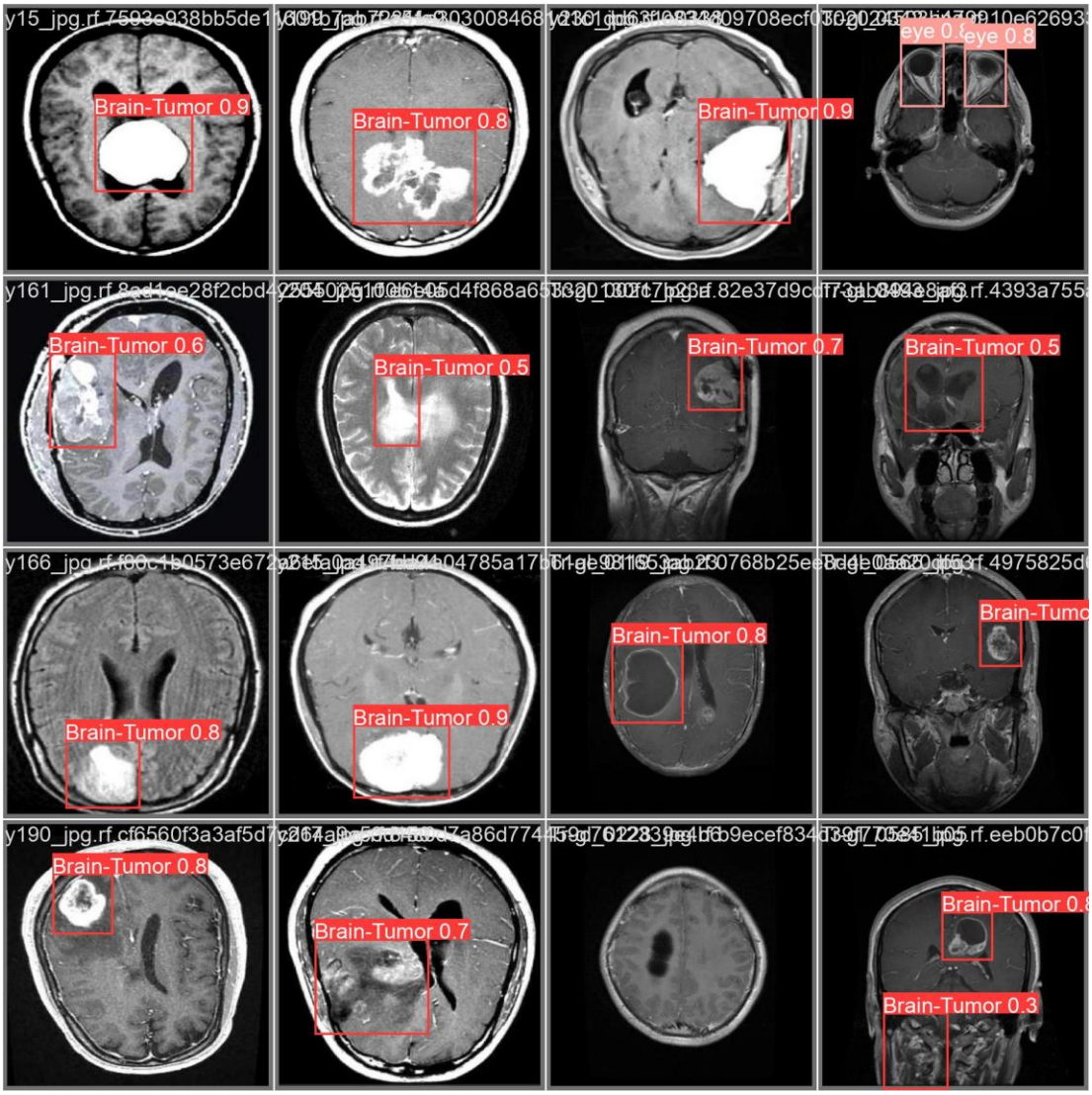
Confusion matrix



Graphs


train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)

Detection



Conclusion

Till now v8 performed best on this dataset with overall mAP of 0.904

YOLO V10:

Clone pretrained repository of Yolov10 from

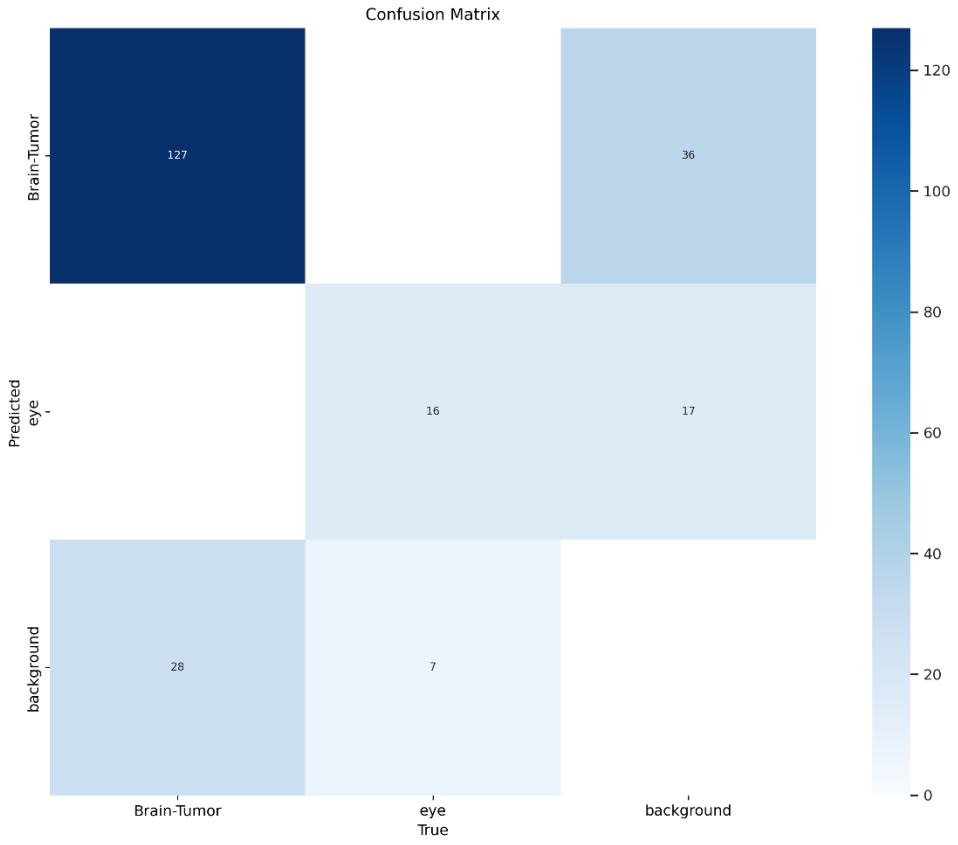
https://github.com/tHU-MIG/yolov10.git

Train on custom data with

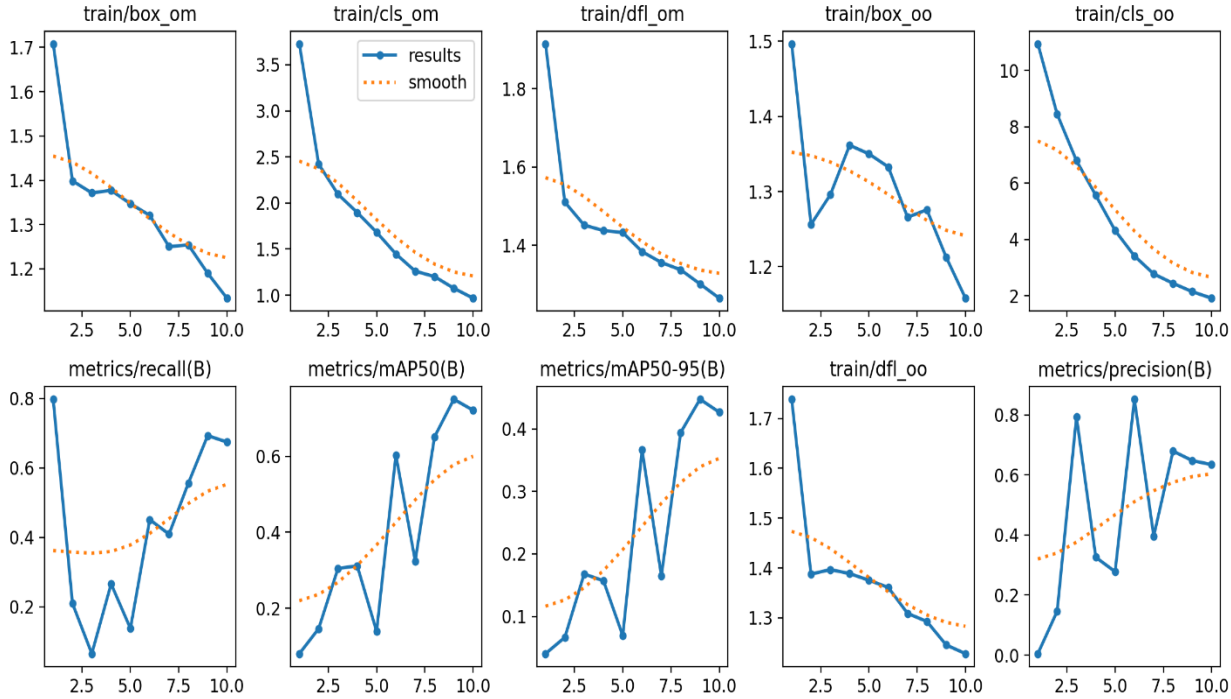
- Batch = 32
- Epochs = 10
- Img-size = 640
- Optimizer = adam-w
- Time Taken 10-15 minutes

Results

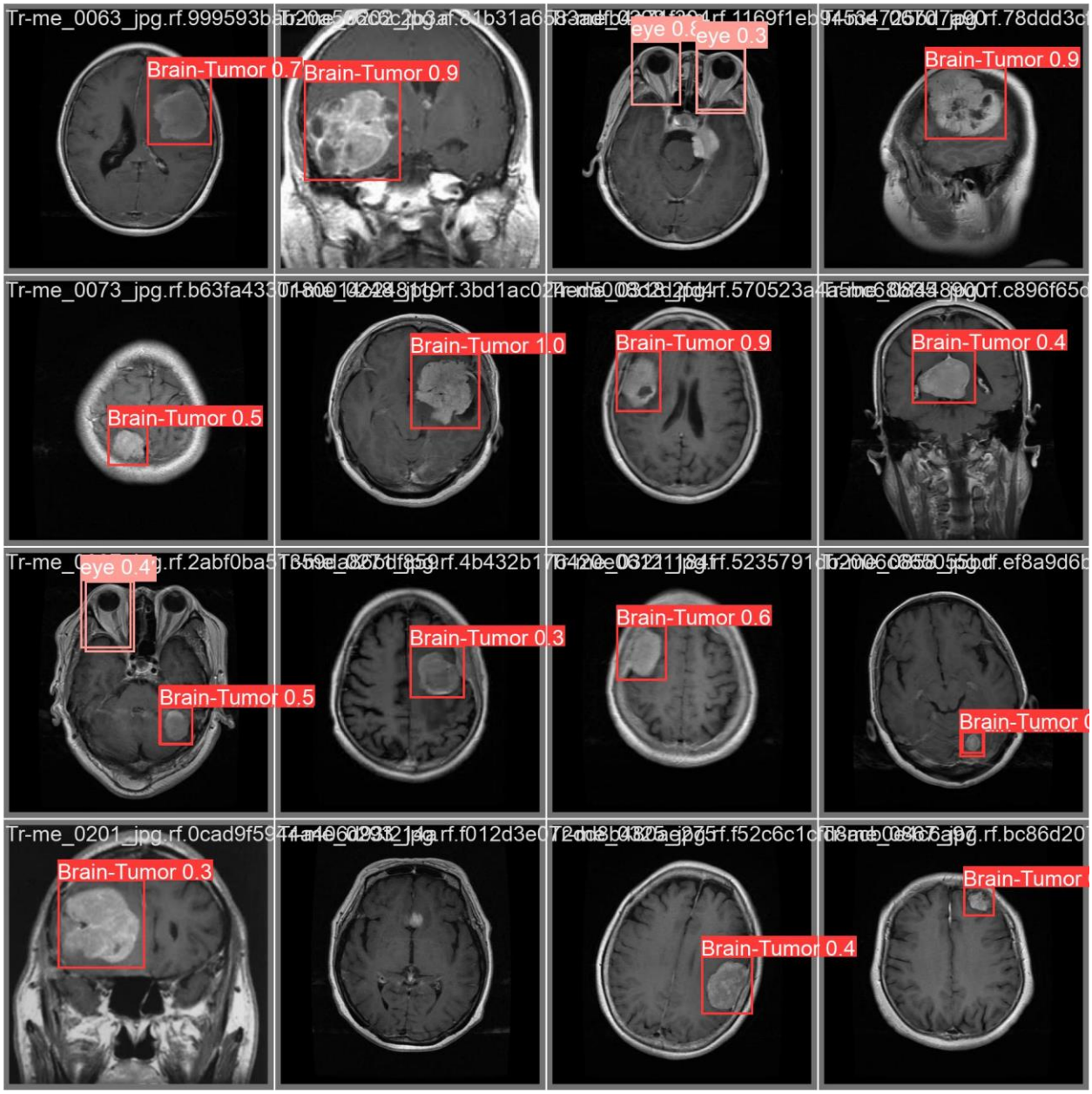
Confusion matrix



Graphs


train/box_om
train/cls_om
results
smooth
train/dfl_om
train/box_oo
train/cls_oo
metrics/recall(B)
metrics/mAP50(B)
metrics/mAP50-95(B)
train/dfl_oo
metrics/precision(B)

Detection



Conclusion

The base model with 10 epochs did not perform well than v8 and v9 To increase its performance epochs should be increased