Deep Learning for Brain Tumor Detection Using MRI

Leo Yao yy3959@nyu.edu
Tianyu Zhao tz2263@nyu.edu
Yumo Li yl10192@nyu.edu

Table of CONTENTS

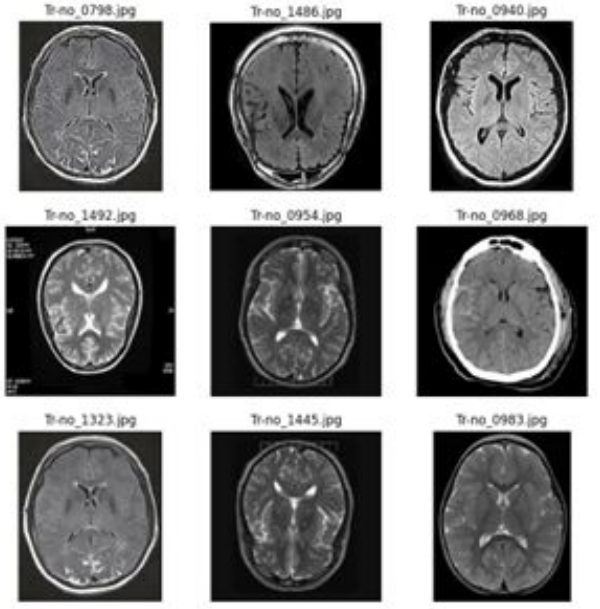
01

Introduction

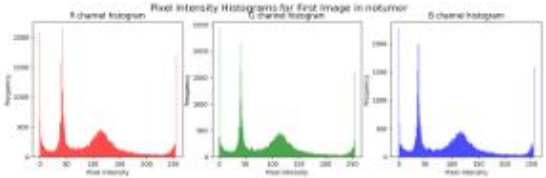
Aim

- High-impact domain
- Motivation
- Clinical focus
- Data availability
- Project goal
- Method stack

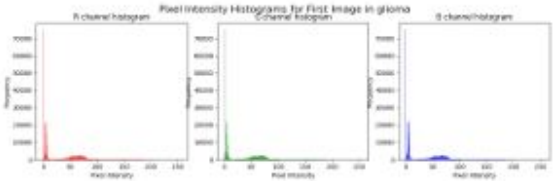
About our data



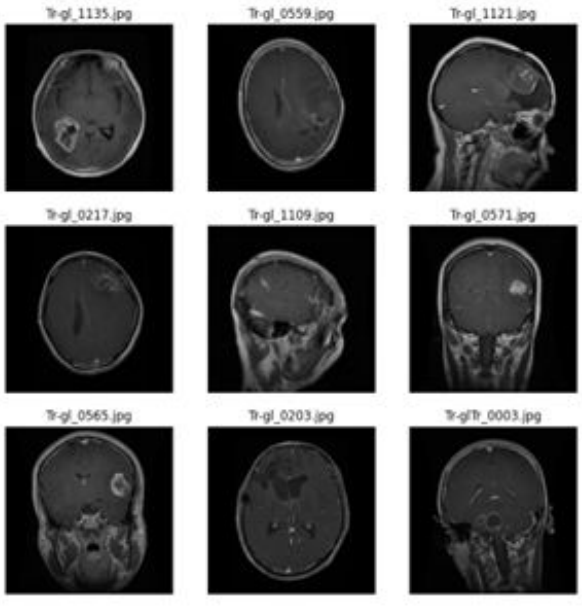
(a) No Tumor Sample Images



(b) No Tumor RGB Pixel Intensity Distribution



(d) Glioma RGB Pixel Intensity Distribution



(c) Glioma Sample Images

Methodology Used

- Data Augmentation and Data Preprocessing
- ResNet
- U-Net and Faster R-CNN

Paper Referenced

- Abdusalomov et al., 2023:
  - *Brain Tumor Detection Based on Deep Learning Approaches and Magnetic Resonance Imaging*
  - The original paper that we referenced

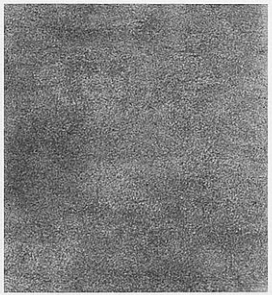
- Ronneberger et al., 2015:
  - U-Net paper

- Shaoqing Ren et al., 2015:
  - Faster R-CNN paper

DATA AUGMENTATION

adding variation,
e.g. noise, blur, motion, etc.

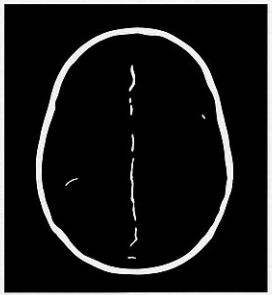
↓



DATA PREPROCESSING

content removal
/ downsizing

↓



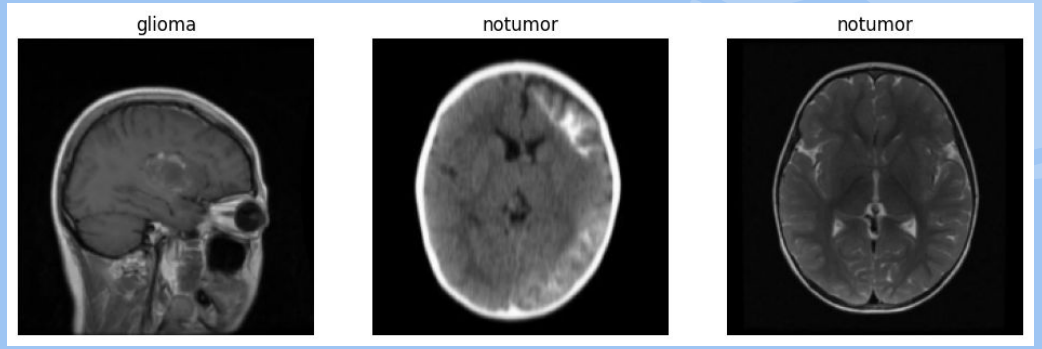
02

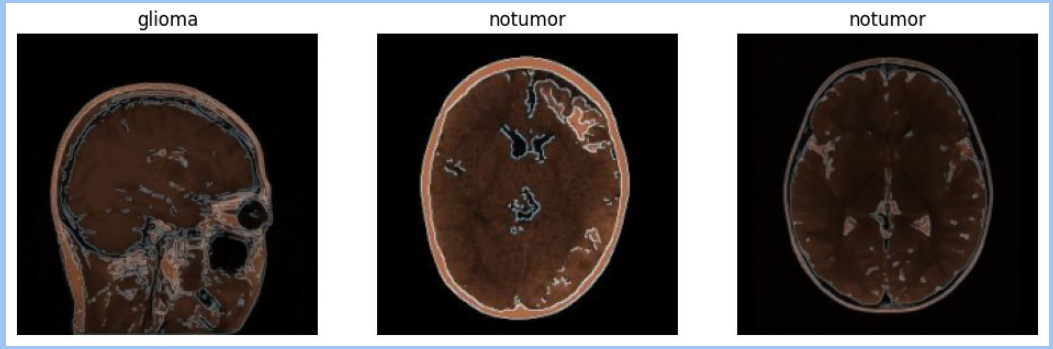
Data Augmentation and Data Preprocessing

Data Augmentation (+)

- TorchIO:
  - Random noise
  - Random blur
  - Random motion
  - Mix of the above

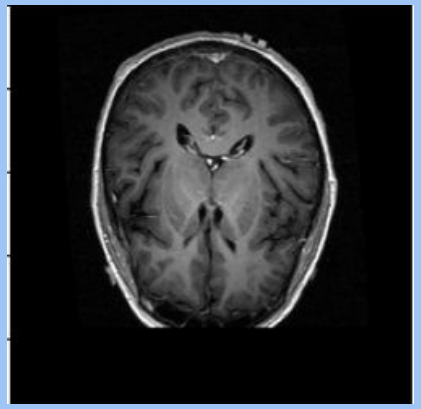
- Canny Edge Detector:
  - Edge augmentation

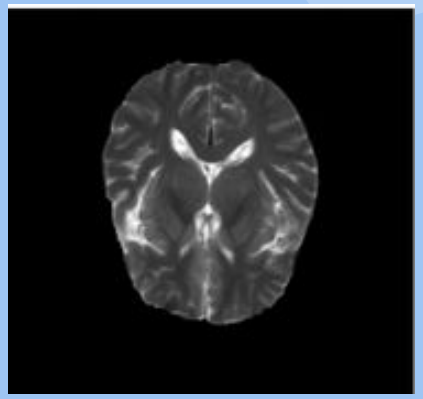




Data Preprocessing (-)

- Skull Stripping



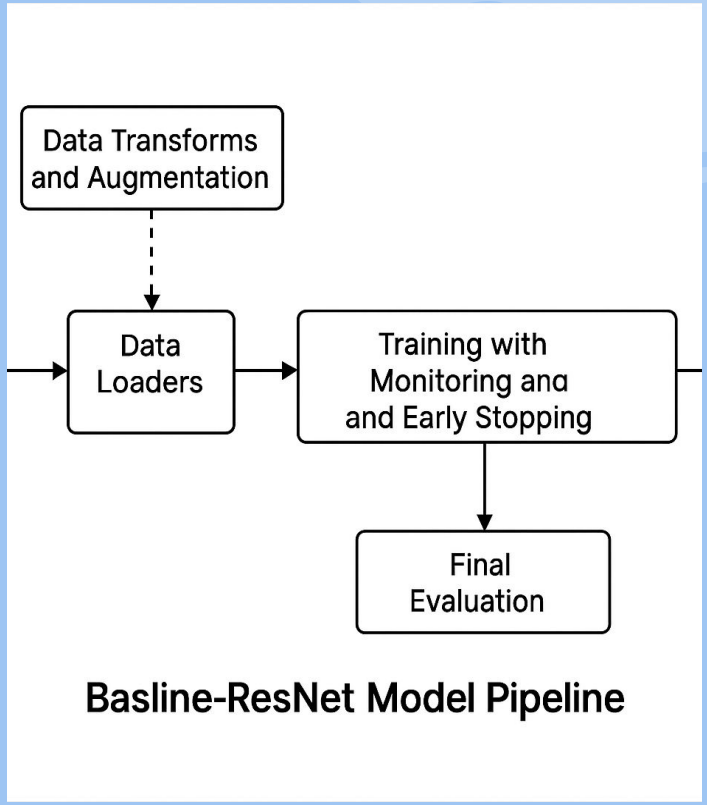


03

Baseline Model

Baseline ResNet Model – Pipeline & Dataset Preparation

- Data Transforms & Augmentation:
  - Training set:
    - Resize, random rotation, affine shift, horizontal flip
    - Normalize to standard mean and std
  - Validation/Test set:
    - Resize and normalize only
- `BrainTumorDataset`:
  - Walks through class folders
  - Loads and labels images (OpenCV + RGB + PIL)
  - Applies specified transforms



Training, Monitoring & Evaluation

Model Architecture:

- Pre-trained **ResNet-50 (ImageNet weights)**
- Replaced final layer with:
    - Dropout + Linear head → **4-class output**

Training Configuration:

- **Loss:** CrossEntropyLoss
- **Optimizer:** Adam
- **LR Scheduler:** ReduceLROnPlateau
- **Monitoring:** Loss, accuracy, confusion matrix (per epoch)

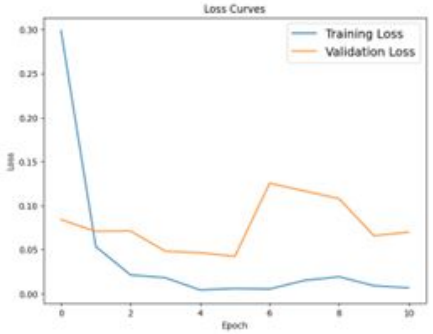
Training Loop:

- Epoch-wise:
    - Train → Evaluate → Log metrics → Adjust LR
    - Save model if val loss improves > 0.001
    - Early stopping after `patience` epochs of no improvement

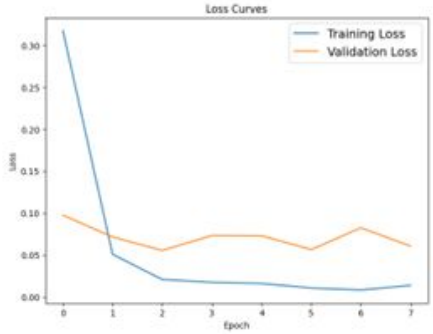
Final Evaluation:

- Load best model → Run on test set
- Report
    - Loss, accuracy
    - **Confusion matrix**
    - **Precision, recall, F1-score** per class

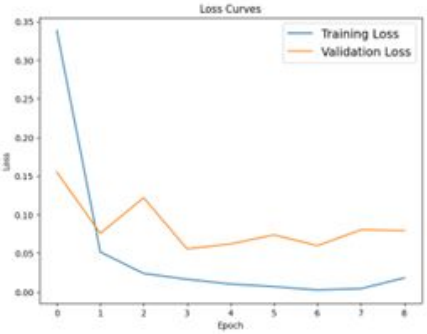
Comparison of Data Augmentation Techniques



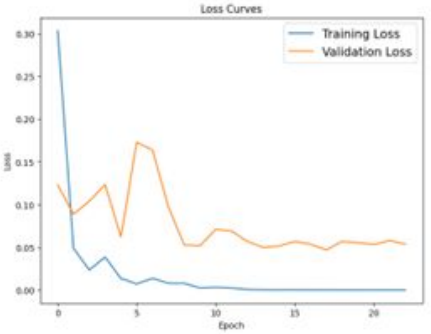
(a) Baseline - Loss



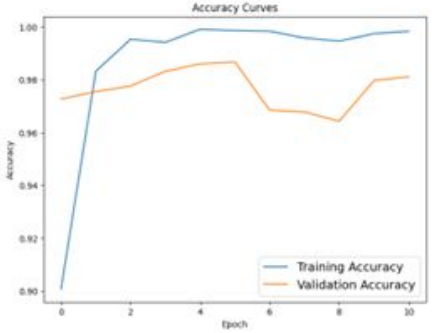
(b) Blurred - Loss



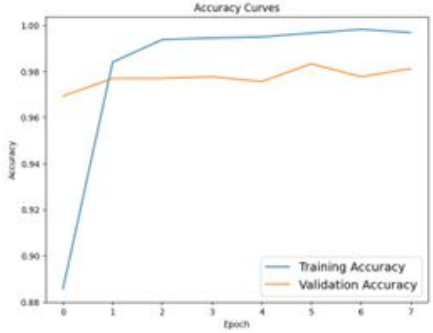
(c) Noised - Loss



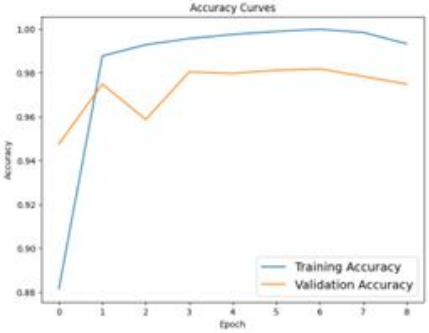
(d) Edged - Loss



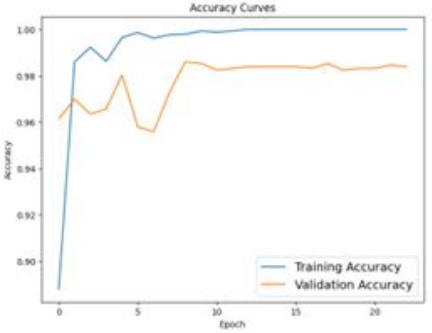
(e) Baseline - Accuracy



(f) Blurred - Accuracy



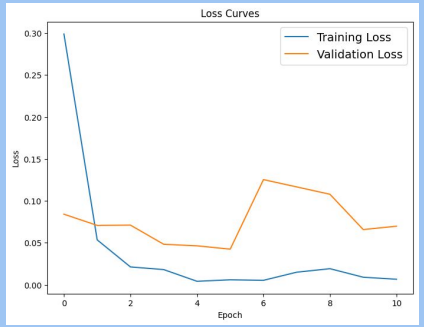
(g) Noised - Accuracy

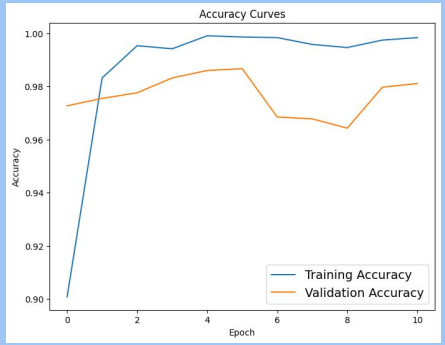


(h) Edged - Accuracy

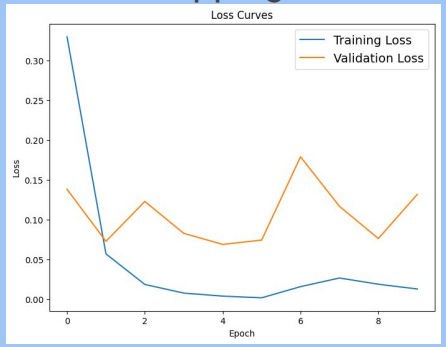
Comparison of Data Preprocessing Techniques

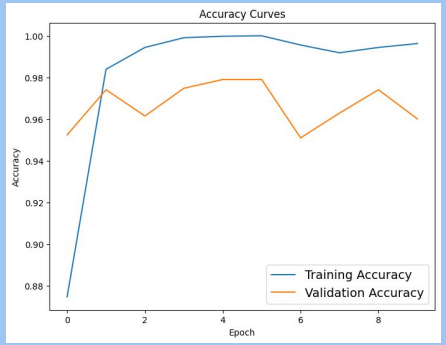
Baseline





Canny Edge Detector + Skull Stripping

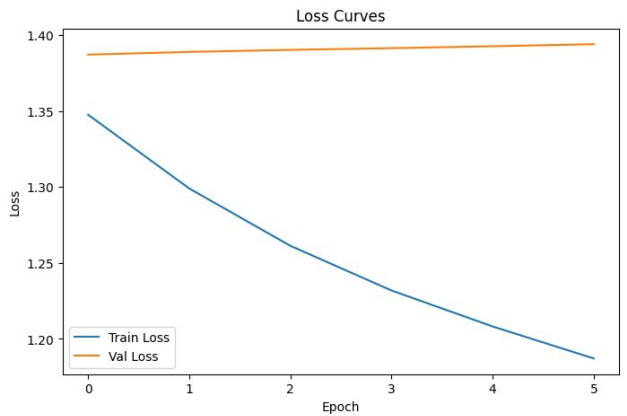


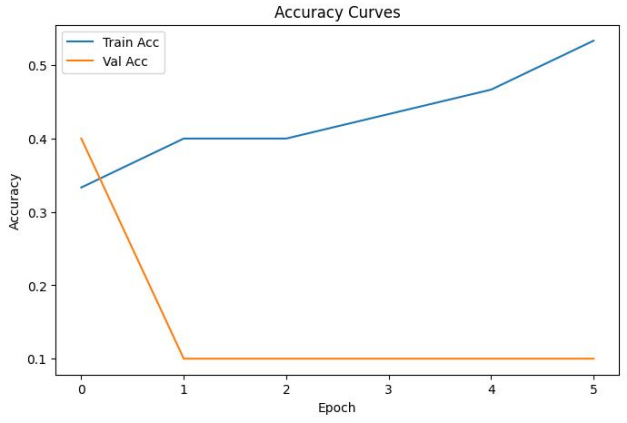


04
OTHER MODELS

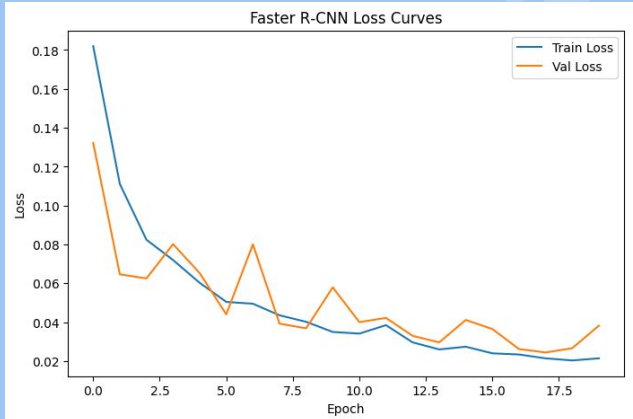
UNET and Faster R-CNN

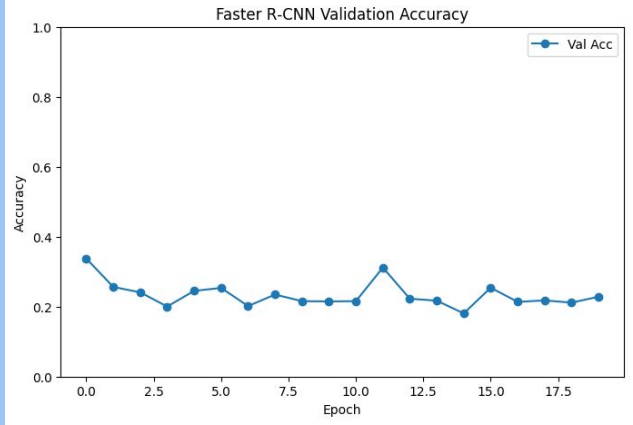
Test accuracy: 0.3750

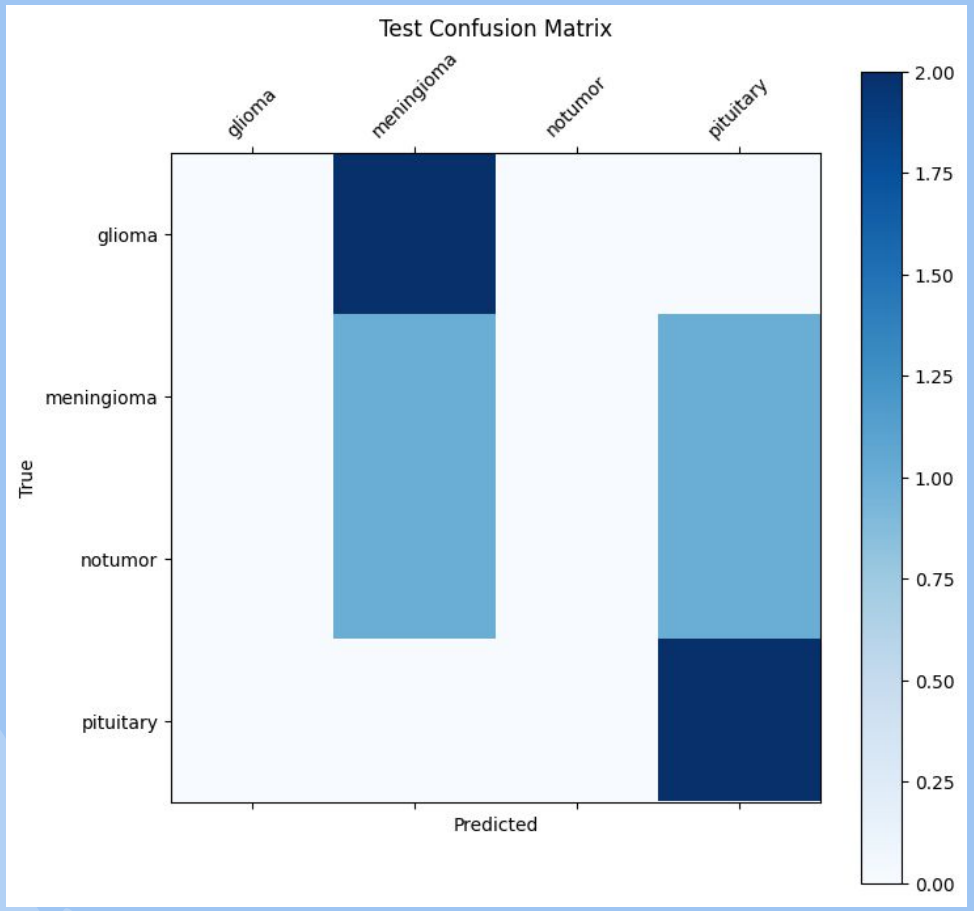


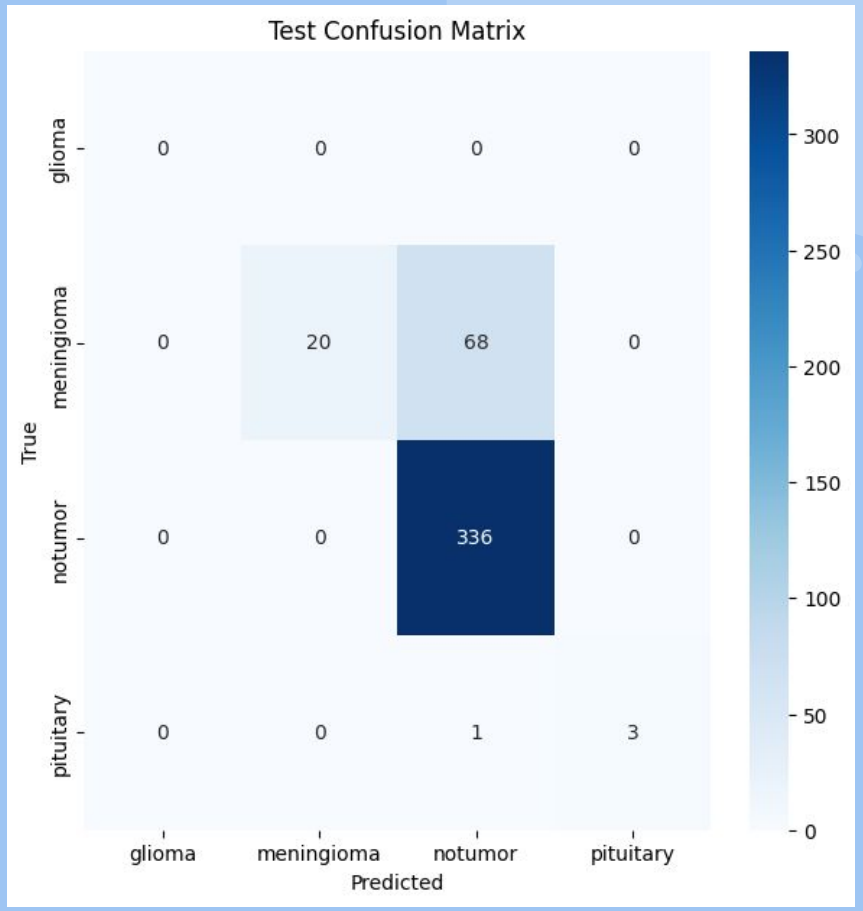


Test accuracy: 0.2283






Test Confusion Matrix
glioma
meningioma
notumor
pituitary
True
Predicted
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00


Test Confusion Matrix
glioma
meningioma
notumor
pituitary
True
Predicted
0 0 0 0
0 20 68 0
0 0 336 0
0 0 1 3
300
250
200
150
100
50
0

THANKS!

GitHub Link:

https://github.com/nyonyoko/Deep_Learning_for_Brain_Tumor_Detection/settings