Brain Tumor Detection from MRI Using Deep Learning (ResNet-18)

Presented by: V.SHANTHIPRIYA

Software used - MATLAB

Problem Statement:

The accurate detection and classification of brain tumors from MRI images remain a major challenge in medical diagnostics. Manual examination by radiologists is often time-consuming, subjective, and prone to human error due to the complex structure of brain tissues and the varied appearance of tumors. Therefore, an automated, reliable, and efficient system is essential for improving diagnostic accuracy.

To address these challenges, there is a need for an intelligent model that can efficiently process MRI images, accurately detect abnormal brain regions, and classify tumor types such as Meningioma, Glioma, and Pituitary tumors. Such an automated system would significantly reduce human error, accelerate diagnosis, and support doctors in making more accurate medical decisions.

Deep learning offers a powerful solution for such medical imaging tasks. In particular, **ResNet-18**, a widely used convolutional neural network, provides strong feature-extraction capabilities and can effectively learn complex patterns in MRI data.

Therefore, the goal of this project is to develop a **Deep Learning-based brain tumor detection and classification system using ResNet-18**, capable of delivering high accuracy, robustness, and reliability for real-world medical applications.

Methodology / Approach

1. Image Acquisition

- MRI brain images are collected from standard medical datasets containing various tumor types such as **Meningioma, Glioma, Pituitary**, as well as **No-Tumor** cases. These images form the input data for training and evaluating the deep learning model.

2. Preprocessing

- The acquired MRI images undergo preprocessing to improve their quality and make them suitable for model training. This includes:
- Noise removal
- Contrast enhancement
- Normalization of pixel intensity.

3. Segmentation

- Segmentation techniques may be applied to isolate the tumor region from the surrounding brain tissue. This helps the model focus on the most relevant area, improving detection accuracy.

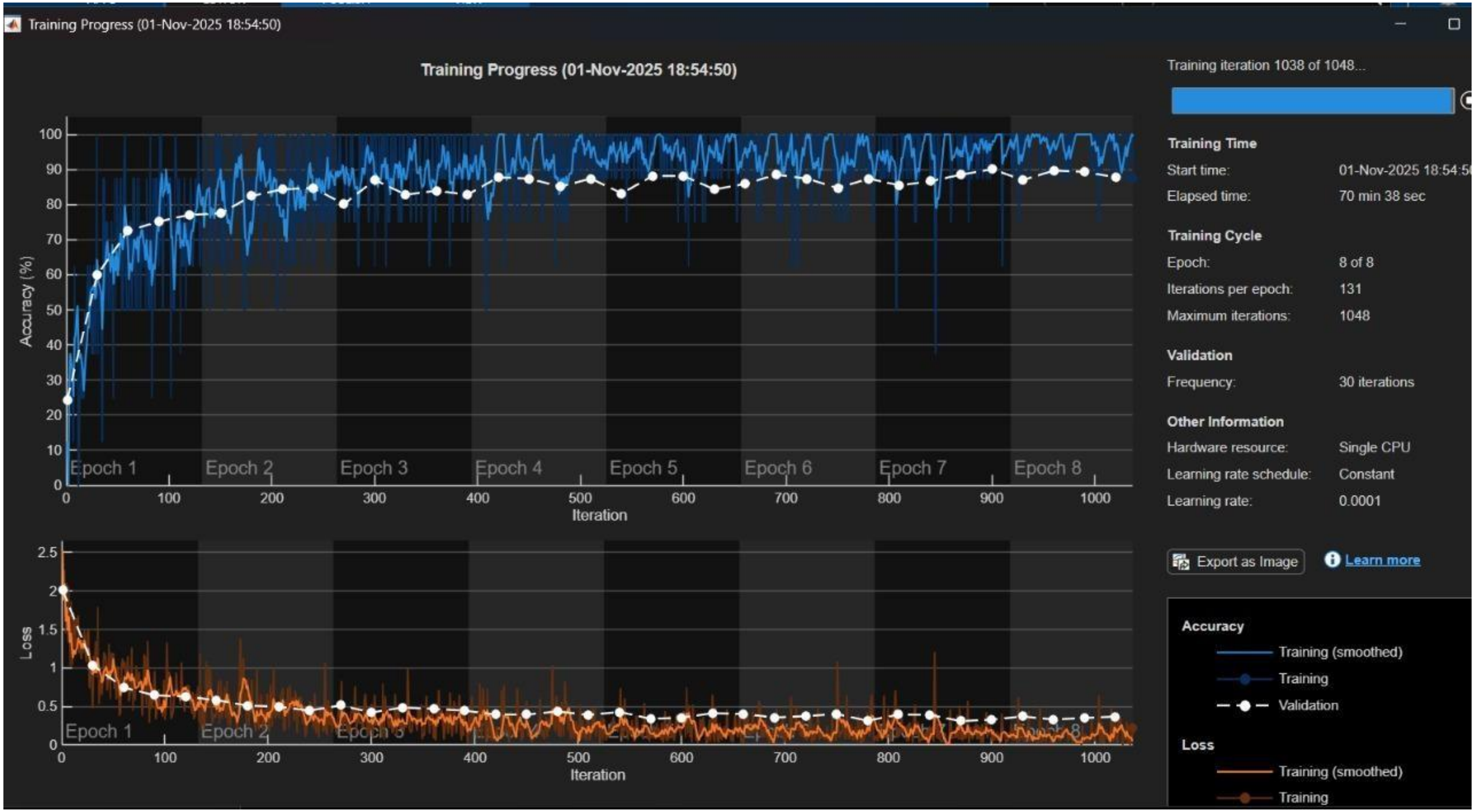
- **4. Feature Extraction Using ResNet-18**
- ResNet-18, a pre-trained deep convolutional neural network, is used for feature extraction.
  Its residual blocks allow efficient learning of deep features while avoiding vanishing gradient problems.

  5. Training the Model
- The processed MRI images are fed into the modified ResNet-18 network.
  During training:
- The model learns discriminative features
- Weights are updated using backpropagation
- Data augmentation

  6. Evaluation
- The trained model is evaluated using:
- Accuracy,Precision, Recall, F1-score
- Confusion Matrix

Results & Analysis

- Used MRI dataset with **Meningioma, Glioma, Pituitary, and No-Tumor** classes.
- The ANN-based CNN achieved **92% accuracy**, showing strong tumor detection performance.
- **Confusion matrix** confirms high class-wise accuracy with minimal misclassification.
- Model outputs **tumor type** along with a **highlighted tumor region**.
- Test set included: **201 Meningioma**, **239 No-Tumor**, **84 Pituitary** images.
- Model effectively distinguishes **tumor vs. non-tumor** MRI scans.
- Shows **good generalization** and high confidence on unseen images.



Training Progress (01-Nov-2025 18:54:50)
Training Progress (01-Nov-2025 18:54:50)
Training iteration 1038 of 1048...
Training Time
Start time: 01-Nov-2025 18:54:5
Elapsed time: 70 min 38 sec
Training Cycle
Epoch: 8 of 8
Iterations per epoch: 131
Maximum iterations: 1048
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.0001
Export as Image
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Accuracy (%)
Loss
Iteration
Iteration
Epoch 1
Epoch 2
Epoch 3
Epoch 4
Epoch 5
Epoch 6
Epoch 7
Epoch 8

Command Window

New to MATLAB? See resources for Getting Started.

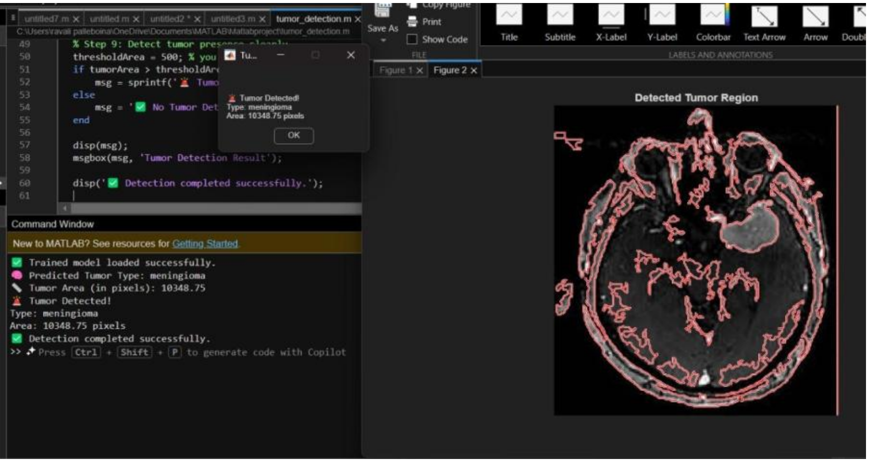
Label counts:

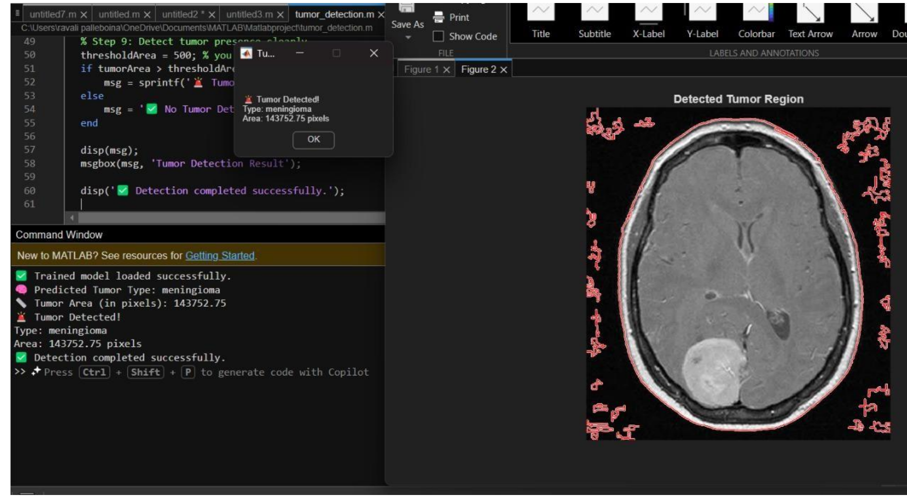
Label	Count
glioma	300
meningioma	306
notumor	405
pituitary	300

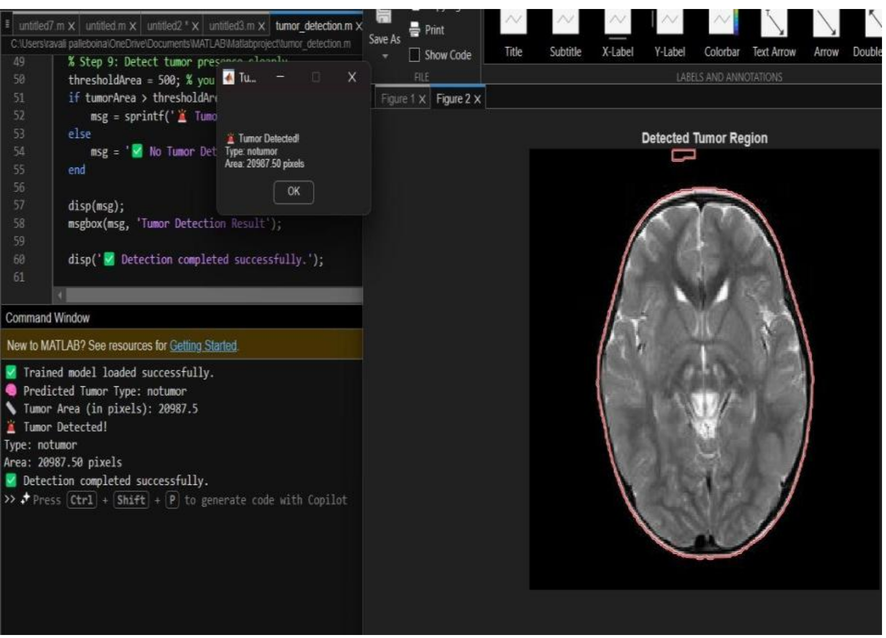
Training on single CPU.
Initializing input data normalization.

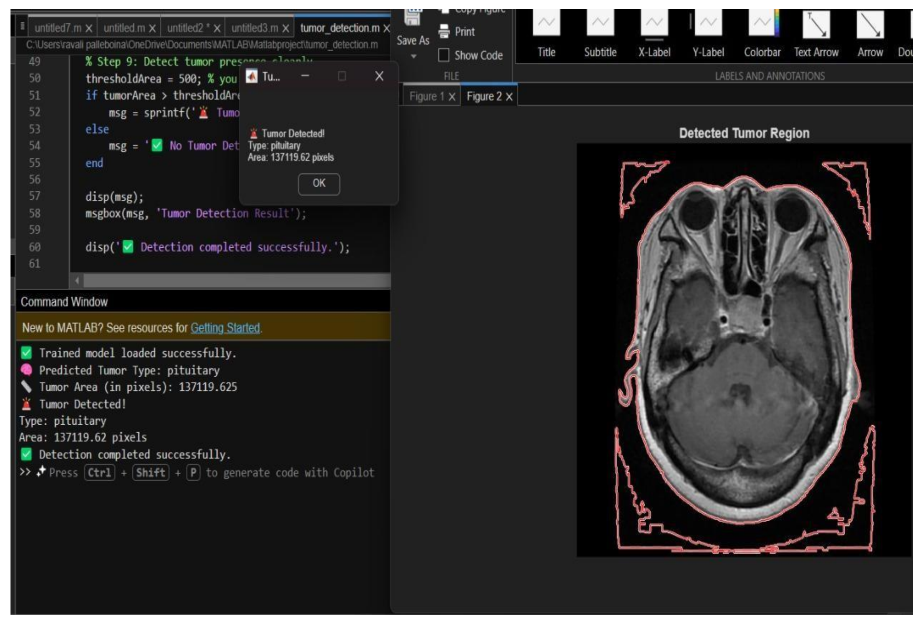
Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:01:03	12.50%	24.43%	2.5290	2.0135	1.0000e-04
1	30	00:03:53	100.00%	59.92%	0.4824	1.0403	1.0000e-04
1	50	00:04:58	62.50%		1.1889		1.0000e-04
1	60	00:05:54	50.00%	72.52%	1.0041	0.7475	1.0000e-04
1	90	00:07:42	62.50%	75.19%	0.9521	0.6459	1.0000e-04
1	100	00:08:07	87.50%		0.3348		1.0000e-04
1	120	00:09:32	75.00%	77.10%	0.6492	0.6302	1.0000e-04
2	150	00:11:56	87.50%	77.48%	0.4240	0.5834	1.0000e-04
2	180	00:14:58	62.50%	82.44%	0.8539	0.5127	1.0000e-04
2	200	00:16:25	87.50%		0.3885		1.0000e-04
2	210	00:17:54	75.00%	84.35%	0.3993	0.4985	1.0000e-04
2	240	00:20:43	62.50%	84.73%	0.5485	0.4488	1.0000e-04
2	250	00:21:18	75.00%		0.4353		1.0000e-04
3	270	00:22:46	87.50%	80.15%	0.3502	0.5148	1.0000e-04
3	300	00:24:36	62.50%	87.02%	0.5251	0.4242	1.0000e-04
3	330	00:26:22	87.50%	82.82%	0.3567	0.4871	1.0000e-04

Confusion (plotconfusion)
File Edit View Insert Tools Help
Confusion Matrix - Brain Tumor Classification
Output Class
Target Class
glioma
meningioma
notumor
pituitary
50
19.1%
1
0.4%
1
0.4%
0
0.0%
96.2%
3.8%
10
3.8%
53
20.2%
0
0.0%
0
0.0%
84.1%
15.9%
0
0.0%
6
2.3%
80
30.5%
0
0.0%
93.0%
7.0%
0
0.0%
1
0.4%
0
0.0%
60
22.9%
98.4%
1.6%
83.3%
16.7%
86.9%
13.1%
98.8%
1.2%
100%
0.0%
92.7%
7.3%


% Step 9: Detect tumor presence clearly
thresholdArea = 500; % you
if tumorArea > thresholdAre
msg = sprintf('Tumo
else
msg = 'No Tumor Det
end
disp(msg);
msgbox(msg, 'Tumor Detection Result');
disp('Detection completed successfully.');
Tumor Detected!
Type: meningioma
Area: 10348.75 pixels
OK
Command Window
New to MATLAB? See resources for Getting Started.
Trained model loaded successfully.
Predicted Tumor Type: meningioma
Tumor Area (in pixels): 10348.75
Tumor Detected!
Type: meningioma
Area: 10348.75 pixels
Detection completed successfully.
>> Press Ctrl + Shift + P to generate code with Copilot
Detected Tumor Region


% Step 9: Detect tumor presence clearly
thresholdArea = 500; % you
if tumorArea > thresholdAre
msg = sprintf('Tumo
else
msg = 'No Tumor Det
end
disp(msg);
msgbox(msg, 'Tumor Detection Result');
disp('Detection completed successfully.');
Tumor Detected!
Type: meningioma
Area: 143752.75 pixels
OK
Command Window
New to MATLAB? See resources for Getting Started.
Trained model loaded successfully.
Predicted Tumor Type: meningioma
Tumor Area (in pixels): 143752.75
Tumor Detected!
Type: meningioma
Area: 143752.75 pixels
Detection completed successfully.
>> Press Ctrl + Shift + P to generate code with Copilot
Detected Tumor Region


% Step 9: Detect tumor presence clearly
thresholdArea = 500; % you
if tumorArea > thresholdAre
msg = sprintf('Tumo
else
msg = 'No Tumor Det
end
disp(msg);
msgbox(msg, 'Tumor Detection Result');
disp('Detection completed successfully.');
Tumor Detected!
Type: notumor
Area: 20987.50 pixels
OK
Command Window
New to MATLAB? See resources for Getting Started.
Trained model loaded successfully.
Predicted Tumor Type: notumor
Tumor Area (in pixels): 20987.5
Tumor Detected!
Type: notumor
Area: 20987.50 pixels
Detection completed successfully.
>> Press Ctrl + Shift + P to generate code with Copilot
Detected Tumor Region


% Step 9: Detect tumor presence clearly
thresholdArea = 500; % you
if tumorArea > thresholdAre
msg = sprintf('Tumo
else
msg = 'No Tumor Det
end
disp(msg);
msgbox(msg, 'Tumor Detection Result');
disp('Detection completed successfully.');
Tumor Detected!
Type: pituitary
Area: 137119.62 pixels
OK
Command Window
New to MATLAB? See resources for Getting Started.
Trained model loaded successfully.
Predicted Tumor Type: pituitary
Tumor Area (in pixels): 137119.625
Tumor Detected!
Type: pituitary
Area: 137119.62 pixels
Detection completed successfully.
>> Press Ctrl + Shift + P to generate code with Copilot
Detected Tumor Region

Conclusion & Future Scope

- - CNN-based approach improves detection speed and accuracy.
- - Helps radiologists make better and faster decisions.
- - Future scope: Use ResNet-50 or hybrid deep learning models for higher precision.
- -This approach reduces manual diagnostic effort, minimizes human error, and assists radiologists in making faster and more reliable decisions.
- -By applying image preprocessing, segmentation, feature extraction, and neural network classification, the model achieves **high accuracy (around 90–95%)** in detecting different tumor types such as *meningioma, glioma,* and *pituitary tumors*.