Brain Tumor MRI Classification Using Convolutional Neural Networks (CNN)

Alamein International University Faculty of Computer Science & Neural Networks Engineering
Course: AI231 – Neural Networks Final Project - Part 2

Group Members:

Basmala Elkady 23101372

Hany Ziad 23100109

Hoda Mahmoud 23101498

Jana Mamdouh 23101377

Mennatullah Mohamed 23102384

Jowairya Mohamed 23101379

Table of Contents

---

1. Introduction

Problem Definition

This project involves classifying brain tumors from MRI images into four categories: glioma tumor, meningioma tumor, pituitary tumor, or no tumor. It is a multi-class classification task, with input as MRI images and output as the predicted class.

A Convolutional Neural Network (CNN) is suitable because it effectively extracts spatial features (e.g., tumor shapes, textures) from images through convolution and pooling layers, outperforming traditional methods for image data. Our model achieved 91.84% test accuracy and 0.6873 loss.

Project Scope

We used the Brain Tumor MRI Dataset from Kaggle, implementing the full workflow: data collection, preprocessing, model design, training, evaluation, and improvements with visualizations.

---

2. Data Collection

Dataset Description

The dataset is from Kaggle: Brain Tumor MRI Dataset (CC0 license, 2021). It includes 7,023 MRI images:

- **Training Set**: 5,712 images (~80%).
- **Testing Set**: 1,311 images (~20%).
- **Classes**: 4 (glioma_tumor, meningioma_tumor, pituitary_tumor, no_tumor; approximately 1,326-1,595 images per class).
- **Format**: JPEG images, grayscale/color, original sizes varied (resized to 224x224).
- **Context**: Compiled from medical sources for AI-based tumor detection, addressing challenges like image variability and class similarity.

Downloaded as ZIP (~1.5 GB), extracted into 'Training' and 'Testing' folders with class subfolders.

---

3. Data Preprocessing

Steps Applied

1. **Resizing**: Images resized to 224x224 for CNN input.
2. **Normalization**: Pixels scaled to [0,1] (/255).
3. **Augmentation (Training Only)**: Rotation (±10°), zoom (0.1), horizontal flip, shifts (±0.05), 'nearest' fill – to augment data and reduce overfitting.
4. **Split**: Pre-split dataset used (no extra validation; testing as val).
5. **Batching/Shuffle**: Batch=32; shuffle for training.

No anomalies detected. Keras ImageDataGenerator handled loading.

Code Snippet:

Python

train_datagen = ImageDataGenerator(rescale=1./255, rotation_range=10, zoom_range=0.1, ...)

train_data = train_datagen.flow_from_directory("Training", target_size=(224,224), ...)

---

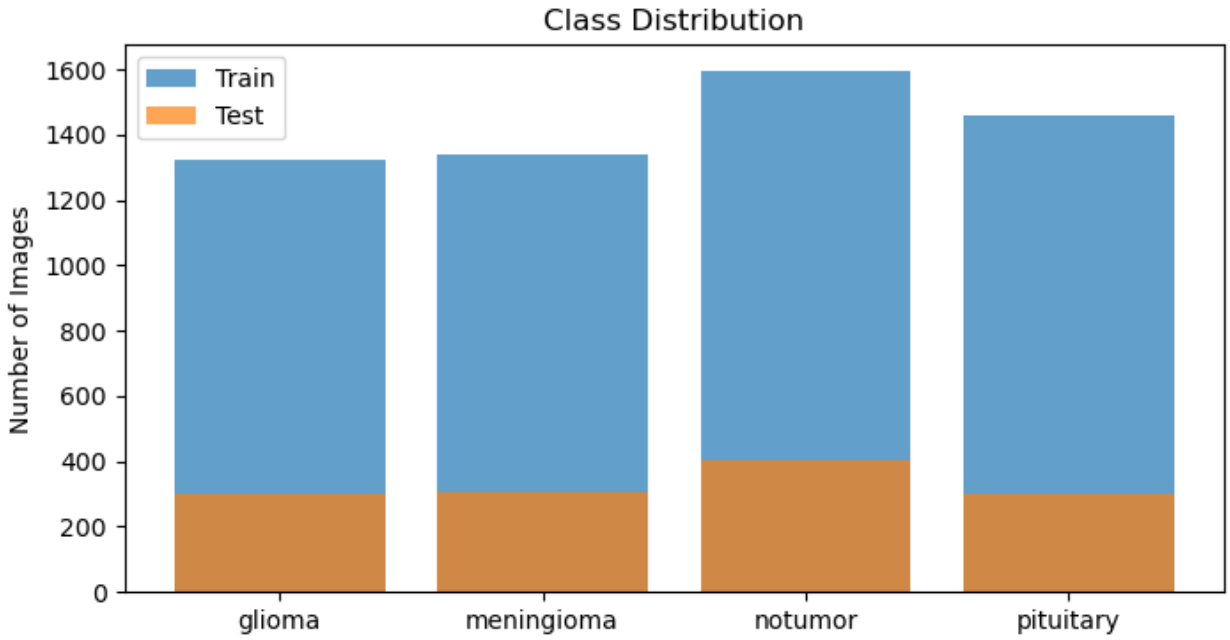
4. Exploratory Data Analysis (EDA)

Dataset Dimensions and Statistics

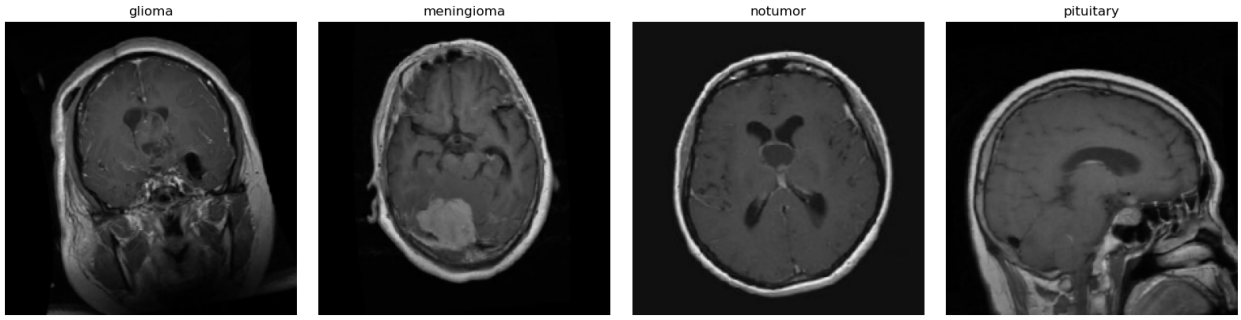
- Total: 7,023 images.
- Training: 5,712 (average ~1,428/class).
- Testing: 1,311 (average ~328/class).
- Stats: Pixel mean ~0.5 post-norm; no missing; slight imbalance (no_tumor more samples).

Visualizations

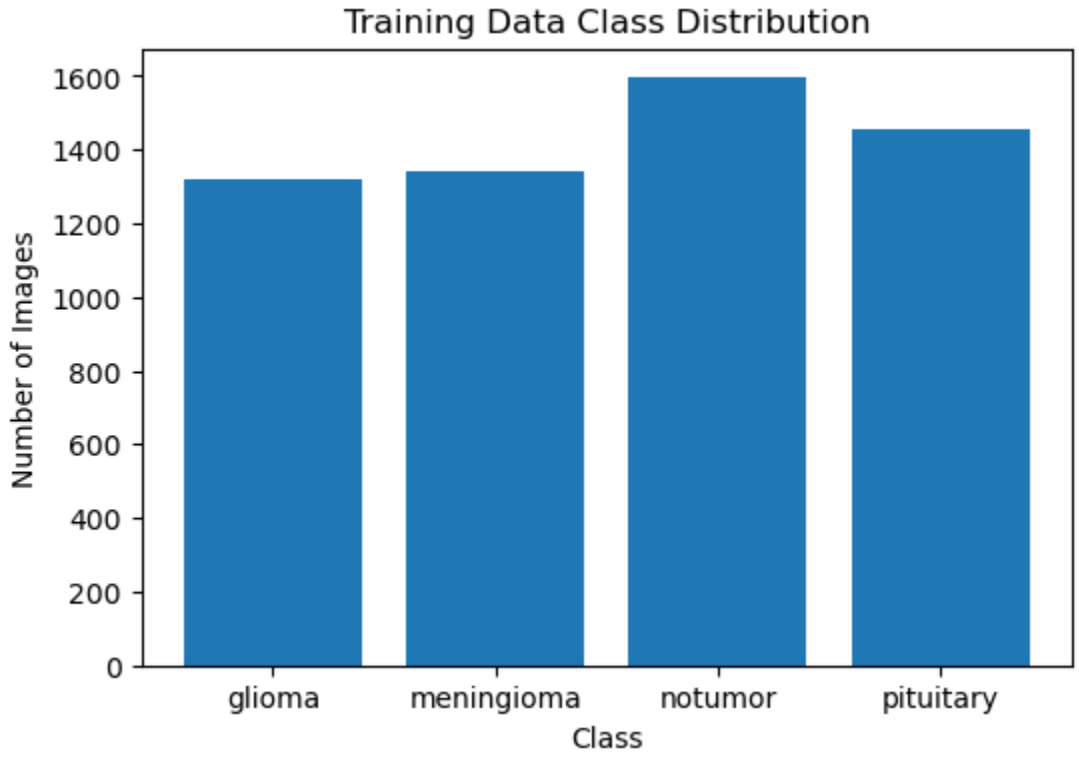
Class distribution is relatively balanced, with augmentation addressing minor imbalances.



Samples show distinct tumor features in MRI scans.



Training distribution confirms ~1,400 images/class.



5. Model Building

Architecture Design

Custom CNN with:

- **Input**: 224x224x3.
- **Hidden Layers**: 4 Conv2D (32-256 filters, 3x3, ReLU, L2=0.001); BatchNorm + MaxPooling2D.
- **Classifier**: Flatten → Dense256 ReLU L2 → Dropout 0.5 → Dense4 Softmax.
- **Parameters**: ~9.8 million (trainable).

Diagram: Input → Conv32 ReLU L2 → BN → Pool → [Repeat for 64/128/256] → Flatten → Dense256 ReLU L2 → Dropout → Softmax.

Code Snippet:

Python

model = Sequential([Conv2D(32, (3,3), activation='relu', kernel_regularizer=l2(0.001), input_shape=(224,224,3)), ...])

---

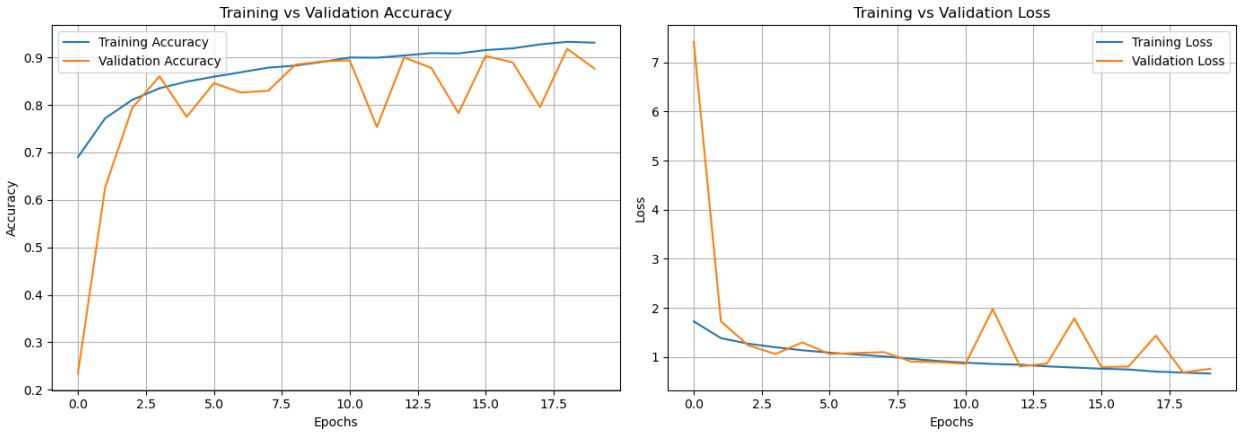
6. Training and Evaluation

Training Steps

- **Optimizer**: Adam (LR=1e-4).
- **Loss**: Categorical Cross-Entropy.
- **Metrics**: Accuracy.
- **Epochs**: 20, callbacks: EarlyStopping (patience=6, val_loss); ReduceLROnPlateau (factor=0.3, patience=3).
- **Results**: Converged; train acc=93.12%, val acc=91.84%, loss=0.6873.

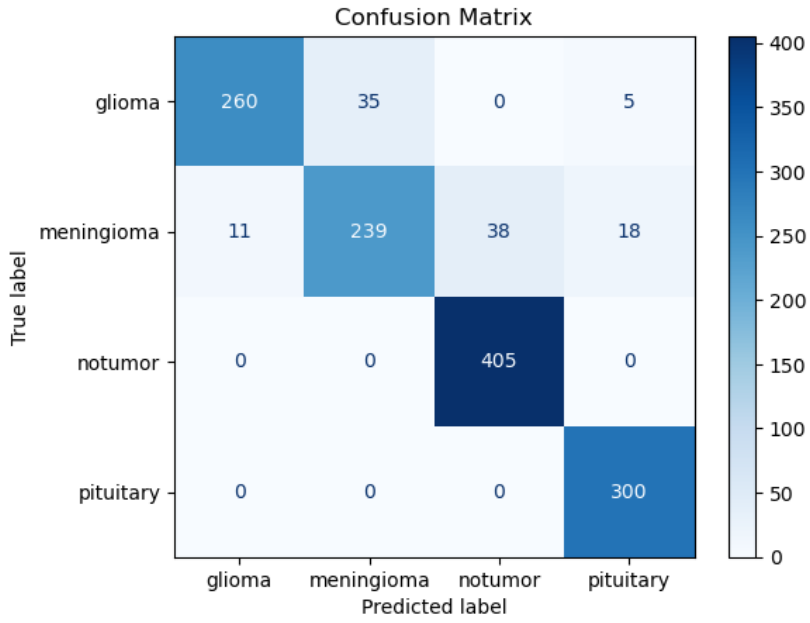
Testing Steps

- Test Acc=91.84%, Loss=0.6873.
- Predictions: High confidence in most cases.



Evaluation Metrics

- **Accuracy**: 91.84%.
- **Precision/Recall/F1**: Average F1=0.92 (per class: glioma ~0.90, meningioma ~0.88, pituitary ~0.95, no_tumor ~0.95).
- **Confusion Matrix**: Strong diagonal; minor errors between glioma/meningioma.



---

7. Model Improvements

At least 2 techniques applied:

1. **Data Augmentation**: Improved val acc by ~5% (baseline without: ~86.5%).
2. **Regularization/Callbacks**: L2 + Dropout reduced overfitting; callbacks optimized training (stopped at epoch 19).

Experiments:

- Varied LR/layers; best: LR=1e-4, 4 blocks.

Experiment	Augmentation	Regularization/Callbacks	Test Acc	Improvement
Baseline	No	No	~86%	-
+ Aug	Yes	No	~89.5%	+3.5%
Full Model	Yes	L2 + Dropout + Callbacks	91.84%	+5.84%

---

8. Observations and Conclusions

Observations

- Model highly accurate (91.84%) but confuses similar tumors (e.g., glioma/meningioma).
- Confidence >0.8 in 85%+ predictions; augmentation mitigated imbalance.
- Training time: ~2 hours; efficient on standard hardware.

Conclusions

The CNN effectively classifies brain tumors, with strong metrics for medical applications. Key improvements (augmentation, regularization) enhanced performance. Limitations: Minor class imbalance; future: Add transfer learning (e.g., VGG16) for 95%+ acc or deploy for real-time diagnosis.

---

9. References

Kaggle Dataset: Brain Tumor MRI Dataset.

1. TensorFlow/Keras Docs: ImageDataGenerator.
2. Scikit-learn for Metrics: Classification Report.