ACML Project Report

Contents

[1. Dataset Source 1](#_Toc199500956)

[2. Dataset Description and Objective 1](#_Toc199500957)

[3. Preprocessing and Data Splitting 1](#_Toc199500958)

[4. Model Architecture and Implementation 2](#_Toc199500959)

[5. Model Training and Hyperparameter Tuning 2](#_Toc199500960)

[6. Training Graphs 3](#_Toc199500961)

[7. Final Evaluation and Results 5](#_Toc199500962)

[8. Conclusion 5](#_Toc199500963)

[9. Code Submission 6](#_Toc199500964)

# 1. Dataset Source

The dataset used for this project was obtained from Kaggle:  
https://www.kaggle.com/datasets/orvile/brain-cancer-mri-dataset  
  
It contains magnetic resonance imaging (MRI) scans of brain tumors classified into three categories:

- Glioma  
- Meningioma  
- Pituitary Tumor  
  
No ethics clearance was required for this dataset as it is publicly available and anonymized.

# 2. Dataset Description and Objective

This dataset comprises MRI brain scans labeled according to the tumor type. The goal is to develop a machine learning model that classifies brain tumors into the three categories with high accuracy.  
  
Medical image classification demands high precision due to its impact on diagnosis and treatment. Thus, our objective was not only high accuracy but also minimal misclassification.

# 3. Preprocessing and Data Splitting

The original images were provided in various sizes and formats. We performed the following preprocessing:

- Resizing all images to 128×128 pixels  
- Normalization with mean = 0.5 and std = 0.5  
- Data augmentation through:  
- Vertical flipping  
- Random rotation by 90°, 180°, or 270°  
  
We combined the original and augmented datasets and split them using 5-fold cross-validation:  
- Training Set: 80%  
- Validation Set: From training set (split into 4:1 folds)  
- Test Set: 20% of the entire dataset

# 4. Model Architecture and Implementation

We built a custom Convolutional Neural Network (CNN) from scratch using PyTorch. The architecture includes:  
  
- Conv2D(3→8) → ReLU → MaxPooling  
- Conv2D(8→16) → ReLU → MaxPooling  
- Flatten → Fully Connected Layer (Linear)  
- Dropout(0.5) → Final Output Layer (3 classes)  
  
This model was chosen to balance complexity and performance, allowing it to learn spatial features critical in medical images.

# 5. Model Training and Hyperparameter Tuning

We initially trained a small model and gradually increased complexity. Training was performed for 10 epochs with early stopping based on validation loss. Key techniques:

- Cross-entropy loss  
- Adam optimizer (better performance vs SGD)  
- 5-fold validation to reduce overfitting  
- Image downscaling improved generalization  
- Training stopped if validation loss increased for 3 consecutive epochs

**Training & Validation Metrics Across Epochs**

|  |  |  |
| --- | --- | --- |
| Epoch | Train Loss | Validation Loss |
| 1 | 0.758 | 0.580 |
| 2 | 0.555 | 0.551 |
| 3 | 0.509 | 0.501 |
| 4 | 0.477 | 0.487 |
| 5 | 0.418 | 0.577 |
| 6 | 0.367 | 0.438 |
| 7 | 0.322 | 0.568 |
| 8 | 0.294 | 0.441 |
| 9 | 0.253 | 0.420 |
| 10 | 0.216 | 0.413 |

# A graph with blue and orange lines AI-generated content may be incorrect.6. Training Graphs

Figure Training and Validation

A graph with a line

AI-generated content may be incorrect.

Figure Accuracy

# 

# 7. Final Evaluation and Results

The model was evaluated on the test set (previously unseen data):  
- Test Accuracy: 83.70%A chart of a number and a number

AI-generated content may be incorrect.

The confusion matrix shows relatively balanced classification across the three tumor types, although some misclassification persists due to the visual similarity in tumor structures

# 8. Conclusion

We successfully implemented a CNN-based classification system for brain tumor MRI scans with:  
- Careful data augmentation  
- Stratified 5-fold validation  
- Effective hyperparameter tuning  
  
The resulting model achieves an accuracy of 83.70% and demonstrates robust performance suitable for further medical research or as a decision support tool.