MAIS 202 Deliverable 2

Eya Ibrahim

1. Problem Statement

The current process of diagnosing brain tumors using MRI scans is often time-consuming and costly. Medical professionals must manually analyze numerous images, which can lead to delays in diagnosis and treatment, as well as increased healthcare expenses. This project aims to develop a machine learning model that efficiently classifies MRI scans, reducing the time required for diagnosis while minimizing costs associated with manual evaluations.

2. Data Preprocessing

I am working with the Brain Tumor MRI Scan Dataset, which contains a total of 5,266 images, categorized into two classes: positive (which means a brain tumor present) and negative (no brain tumor). The initial dataset required some preprocessing to enhance its usability. I removed 50 corrupted images that could not be opened and tried to resize images to the same dimensions for consistency (128x128).

3. Machine Learning Model:

The chosen model for this project is a Convolutional Neural Network (CNN) due to its proven effectiveness in image classification tasks. The CNN architecture consists of the following layers:

Input Layer: 128x128x3 (for RGB images)

Convolutional Layers: TBD…

Activation Function: ReLU (Rectified Linear Unit)

Pooling Layer: Max pooling after each convolutional layer

Flattening Layer: To convert the 2D matrices to a 1D vector

Output Layer: Softmax activation for classification (2 classes)

Im going to implement the model using Keras with a TensorFlow backend. The training/validation/test split is set to 70/15/15 to make the sure the model is tested on unseen data at some point. I will probably have to do some sort of thing to prevent overfitting since a large percentage of my data is used for training.

4. Preliminary Results:

For evaluation, I used accuracy and the confusion matrix as metrics. The preliminary results show an accuracy of **85%** on the validation set, with a precision of **80%** and recall of **78%** for the positive class. The confusion matrix indicates some misclassification between the two classes, particularly with false negatives. For the final presentation, I'll also provide visual representations of the confusion matrix and accuracy graphs to illustrate these findings.

5. Next Steps:

Moving forward, I plan to implement data augmentation techniques (e.g., rotation, flipping) to increase the diversity of the training dataset and potentially improve model accuracy. Additionally, I will explore fine-tuning the model by adjusting hyperparameters and possibly incorporating more advanced architectures, such as transfer learning with pre-trained models. I also need to add some things to the code and continue with some minor tweaks, and monitor its performance.