Brain Tumor Detection Using CNN

Project Overview

This project focuses on developing a Convolutional Neural Network (CNN) model to classify brain tumor MRI images. The goal is to detect and classify brain tumors into four categories: glioma, meningioma, no tumor, and pituitary. The project involves data preprocessing, model development, training, evaluation, and visualization of results using TensorFlow and Keras.

Dataset

The dataset used in this project consists of MRI images categorized into four types:

- Glioma
- Meningioma
- No Tumor
- Pituitary

The data is organized into training, validation, and test sets, with data augmentation applied to enhance model generalization.

File Structure

```
/kaggle
|-- /input
|   |-- /brain-tumor-mri-dataset
|-- /working
|   |-- /new_dataset
|       |-- /TRAIN
|       |   |-- /GLIOMA
|       |   |-- /MENINGIOMA
|       |   |-- /NOTUMOR
|       |   |-- /PITUITARY
|       |-- /VAL
|       |-- /TEST
|-- final_model.h5
```

Dependencies

Ensure the following Python packages are installed:

- TensorFlow
- Keras
- NumPy
- Matplotlib
- scikit-learn

Installation

Install the necessary packages using pip:

```
pip install tensorflow keras numpy matplotlib scikit-learn
```

Usage

1. **Download and Extract Data:**

   The script automatically downloads and extracts the dataset from the specified URL. Ensure you have access to the dataset.

2. **Data Preparation:**

   The data is split into training, validation, and test sets. Data augmentation is applied to the training data to improve model robustness.

3. **Model Definition:**

   A custom ResNet152V2 model is implemented with convolutional and residual blocks, including dropout and batch normalization layers.

4. **Training:**

   Train the model with the training set and validate using the validation set. Adjust the number of epochs and batch size as needed.

5. **Evaluation:**

   Evaluate the trained model on the test set to measure accuracy and loss.

6. **Results:**

   - The trained model is saved as `brain_tumor_detector_v1.keras`.
   - Training and validation accuracy/loss are plotted for analysis.

Results

- **Model Accuracy:** The accuracy of the model on the test set is printed after evaluation.
- **Plots:** Training and validation accuracy/loss are plotted for each epoch.

Acknowledgments

- The dataset is provided by Kaggle and other sources.
- TensorFlow and Keras for deep learning functionalities.
- Matplotlib for data visualization.
