Integrated Libraries, Modules, Functional Dependencies

In the implementation of the brain tumor detection project, several Python libraries and modules were integrated to facilitate the development and training of deep learning models. The primary dependencies include:

1. TensorFlow and Keras:

- Purpose: TensorFlow is a powerful open-source machine learning library, and Keras is an API that runs on top of TensorFlow, simplifying the model-building process.

- Usage: Both TensorFlow and Keras were fundamental for implementing and training convolutional neural networks (CNNs) for image classification.

- Implementation: The code extensively utilizes TensorFlow and Keras for defining the architecture of deep learning models, specifying layers, and managing the training process.

2. Scikit-learn:

- Purpose: Scikit-learn provides a set of tools for machine learning, including utilities for data preprocessing, model evaluation, and metrics calculation.

- Usage: Scikit-learn was employed for tasks such as data augmentation, feature scaling, and splitting the dataset into training and testing sets.

- Implementation: The library's functionalities contributed to preparing the dataset and enhancing the robustness of the models through techniques like data augmentation.

3. ONNX (Open Neural Network Exchange):

- Purpose: ONNX is an open format for representing deep learning models, enabling interoperability between different frameworks.

- Usage: The project explored the conversion of TensorFlow models to the ONNX format for improved inferencing speed and cross-platform support.

- Implementation: ONNX was used to convert TensorFlow models, and the ONNX Runtime was employed for model inferencing, with a specific focus on execution speed.

4. Matplotlib and Seaborn:

- Purpose: Matplotlib and Seaborn are plotting libraries for creating visualizations and graphs.

- Usage: These libraries were employed to generate visualizations for model evaluation metrics, such as accuracy and loss, providing insights into the performance of different CNN architectures.

- Implementation: Matplotlib and Seaborn were utilized to plot graphs showcasing the results obtained during model training and evaluation.

5. Pandas:

- Purpose: Pandas is a data manipulation library that provides data structures like dataframes, facilitating data handling and analysis.

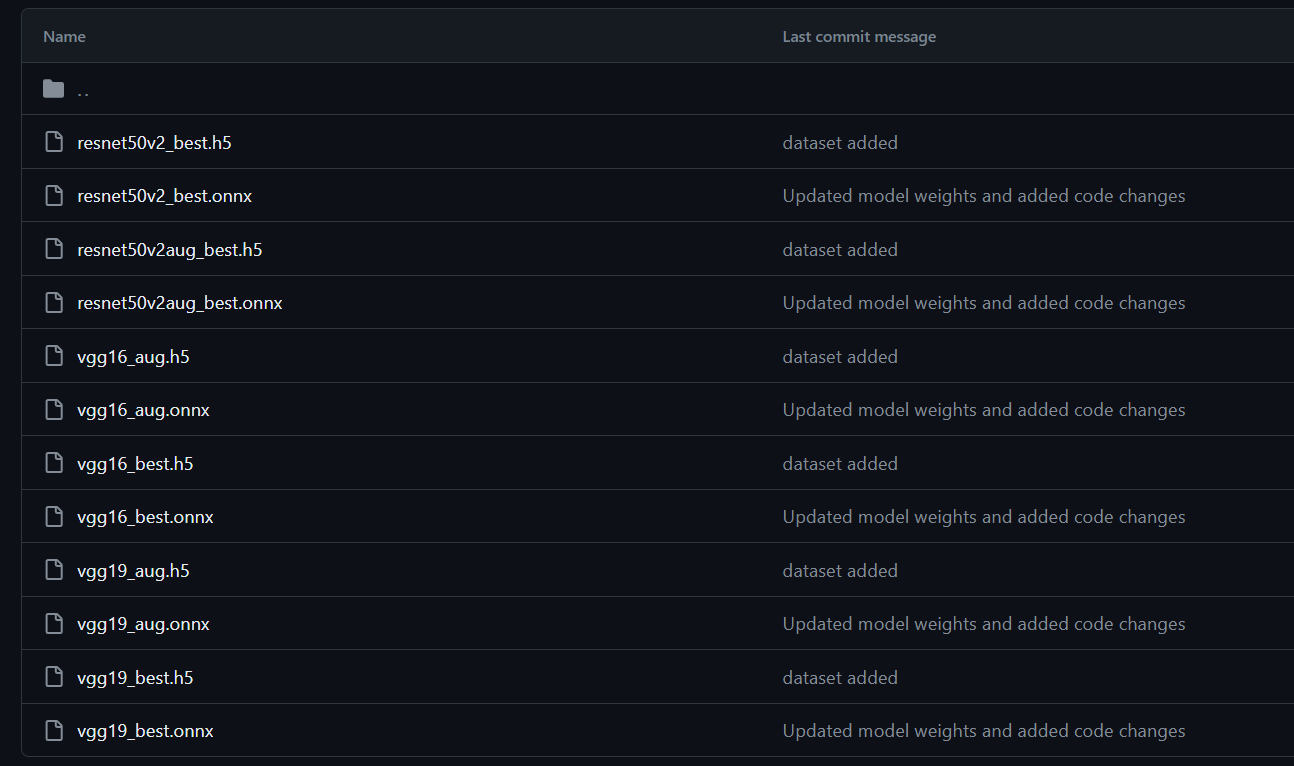
- Usage: Pandas was used for organizing and manipulating tabular data, especially during the analysis of model results and the creation of performance summary tables.

- Implementation: Data collected during model evaluation was structured and analyzed using Pandas, aiding in the generation of comprehensive results tables.

6. Deep Learning Models:

* Models Used: ResNet50V2, VGG19, VGG16.
* Purpose: These pre-trained deep learning models were employed for transfer learning, leveraging their learned features for the task of brain tumor detection.
* Implementation: The selected models were fine-tuned on the brain MRI dataset, and their performance was evaluated based on training and validation accuracy.

Also, model weights need to be uploaded in the google colab file before making predictions:



(available upon request; due to size)