**Convolutional Neural Network (CNN) Binary Brain Tumor Classification**

William W. Winters

Anderson College of Business and Computing, Regis University

MSDS 686 Deep Learning

Dr. Ghulam Majtaba

March 9, 2025

**Introduction**

Intercranial neoplasms (brain tumors) are among the most devastating diseases that result in a significant reduction of quality of life or life expectancy. Often symptoms such as headaches, changes of personality, irritability and other symptoms are not correctly diagnosed as being brain tumors. In addition, some tumors may be difficult to identify on MRI imagery and are overlooked in the diagnostic process. Furthermore, the golden standard of care brain tumors is 100% resection (removal) to prevent reoccurrence. Therefore, being able to accurately interpret and diagnose residual tumors from postoperative MRI brain images is essential to treating brain tumors.

Many Kaggle studies have been conducted to classify brain tumors using convolutional neural networks (CNN)s, but most of these focused on classifying the type of tumor into for groups: glioma, meningioma, pituitary, and no tumor (Roy, et al., 2024). In this study the emphasis will be in developing a CNN that can examine MRI images and determine if a tumor exists or not. The goal is to use this model as a diagnostic tool that can be used in the initial examination or postoperative evaluation to determine if any residual tumor remains.

**Methods**

A review of literature was conducted to determine the current state of using CNNs for image classification. Several peer-reviewed articles were reviews and the methods discussed were included in the model the author produced. The main hurdle for this study was obtaining enough brain MRI images from reliable sources. Many public brain MRI datasets contained mistakes and were of unknown origins. The dataset used in this analysis is a combination of two sources, Viradiya (2021) and Bhuvaji (2020). Most images were taken from Viradiya’s dataset with the *no\_tumor* class copied from Bhuvaji’s dataset. This was performed to bring the *negative* class to around the same number as the *positive* class in the training and testing datasets. After data curation the total number of images are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Training** | **Testing** | **Total** |
| Negative | 2,065 | 522 | 2,587 |
| Positive | 2,011 | 502 | 2,513 |

The datasets can be accessed [here](https://drive.google.com/drive/folders/1o98tM0BlMP5Y4YcHaSkTKVWMAo7NpU0F?usp=drive_link)

The image path and label information were loaded into *Pandas’ DataFrames* to allow for Exploratory Data Analysis (EDA). A validation dataframe was taken from the training one in a standard 80/20 split. The three dataframes were then used to create image pipelines that could be fed into the CNN model.

**Model Architecture**

A functional API approach was utilized for architecture definition. It provides an easy-to-follow and logical flow between the layers and provides more flexibility than other methods. The final model contains four convolutional layers with pooling and dropout. These layers are normalized, flattened and then fed into two dense layers with dropout. The final dense layer or output layer is using *softmax* for its activation instead of *relu* like the previous layers.