**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

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

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. The golden-standard of care brain tumors is 100% resection (removal) to prevent reoccurrence, and being able to accurately interpret and diagnose residual tumor from postoperative MRI brain images is essential to treating them.

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 diagnostic tool that can be used in the initial examination or postoperative evaluation to determine if any residual tumor remains.

**Data**

The dataset used in this effort is a combination of two data repositories available on Kaggle.com. Two Kaggle sources were combined to form the final dataset used in this project and it contains a total of 5,100 images. The original data sources can be found in the links below.

[Brain Tumor Classification (MRI) (Bhuvaji, 2020)](https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri)

[Brain Tumor Dataset (Viradiya, 2021)](https://www.kaggle.com/datasets/preetviradiya/brian-tumor-dataset)

A majority of the data was taken from Viradiya’s dataset and a few images were taken from Bhuvaji’s no\_tumor class. The following is a breakdown of how the images were used

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Training** | **Testing** | **Total** |
| Negative | 2,065 | 522 | 2,587 |
| Positive | 2,011 | 402 | 2,513 |
| **Total** | 4,076 | 1,024 | 5,100 |

Table 1 - Image Counts

The images from Bhuvaji’s dataset were randomly selected and added to this project’s dataset to help balance out the classes. A copy of the dataset used in this project can be found on the author’s [Google Drive](https://drive.google.com/drive/folders/1o98tM0BlMP5Y4YcHaSkTKVWMAo7NpU0F?usp=drive_link).

**Methods**

A review of current peer-review literature was reviewed to determine current trends in MRI brain tumor classification using convolutional neural networks and sources of brain tumor MRI images. From this research it was determined that many of the public image datasets are available on [kaggle.com](https://www.kaggle.com/search?q=brain+tumor+mri+image+datasets+in%3Adatasets). After the datasets where procured, each one was evaluated for suitability to this study. Items considered included how well the datasets were documented, labeled and traceable. Sets that were not up to research standards were discarded and the final dataset was constructed.

Since images are involved and this is a classification problem, a convolutional neural network (CNN) is the best choice for this type of deep learning problem (Chollet, 2021). After dataset curation and model architecture selection, a Jupyter-lab notebook was used to develop the model’s architecture, clean and process data, perform EDA, train the model and evaluate its performance.

**Model Architecture**

Final CNN architecture of the model used in this project consists of the following key components:

1. **Convolutional Layers**: This model uses four convolutional layers with a combination of max pooling.
2. **Activation Functions**: The convolutional layers are using the ReLU (Rectified Linear Unit) activation function, which is well suited to this type of binary classification problem
3. **Pooling Layers**: Max pooling layers are used to reduce the spatial dimensions of the feature maps while retaining the most important features
4. **Flattening Layer**: Is a crucial component in a CNN and its primary purpose is to transform the output of the convolutional and pooling layers into a one-dimension array that can be fed into a fully connected layer
5. **Dropout**: Is a technique used in CNNs to prevent overfitting and improve generalization
6. **Fully Connected Layers**: Two fully connected layers are being used to perform final classification and produce the results.

**Model Training**

To help prevent overfitting two callback functions were developed. The first one is designed to stop the training process when validation loss stops decreasing and the second will reduce the learning rate until validation loss plateaus. The author adjusted several arguments for both functions until satisfactory results with training were obtained. In addition, to the callbacks, the Adam optimizer was utilized for model training and the default learning rate of 0.001 was utilized. The beta\_1 and beta\_2 arguments were adjusted in an attempt to improve performance. Furthermore, the model uses the categorical cross entropy loss function with metrics of accuracy, precision, recall, and AUC.

Model training with the Adam optimizer and the two callback functions averaged between 35 to 45 epochs per training session. Plots of training/validation loss, accuracy, precision, recall and AUC all indicate the model is well fitted with the data being used.

**Results**

Standard evaluation methods were used to determine the accuracy of the model. The following results were observed:

* **Accuracy**: The model consistently scored ***above 0.90 (90%)*** on the test data. This shows the model is generalizing the test data well and is accurately predicting classes
* **AUC**: Area Under Curve is a representation of the classifier’s performance and indicates how well classification is taking place. The model used in this project consistently scored ***above 0.90 (90%)*** in this category.
* **F1 Score**: Is a measure of the model’s accuracy on a classification problem. This project’s model consistently scored ***above 0.90 (90%)*** in this area.
* **TPR**: True Positive Rate is the measure of how well the model predicts true positives. Here again the model consistently performed ***above 0.90 (90%)***.

Analysis

Based on the training plots and the results of evaluating the model against the test dataset, the model is consistently making accurate predictions over 90% of the time.

* **TNR**: True Negative Rate determines how well a model predicts negative classes. Like the TRP reading, this model consistently scored ***above 0.90 (90%)*** in this area.