**Documentation**

**Brain Tumor Classification**

**using**

**Convolutional eXtreme Gradient Boosting Architecture**

**(https://www.github.com/ManojGowda27/Brain-Tumor-Classification)**

**Project Description**

* A Brain tumor is considered as one of the aggressive diseases, among children and adults. Brain tumors account for 85 to 90 percent of all primary Central Nervous System (CNS) tumors. Every year, around 11,700 people are diagnosed with a brain tumor.
* And hence, it becomes an important area of research in the field of medical imaging for early diagnosis and categorization of brain tumors, which in turn aids in choosing the most practical course of therapy to protect patients' lives.
* Our team has come up with three different models to compare amongst ourselves in identifying which model performs best in classification.

**Dataset**

The dataset consists of two folders: Training and Testing which contains the different Brain Magnetic Resonance Images (MRI).

The Training folder contains four subfolders: glioma, meningioma, no tumor and pituitary.

The Training folder consists at total of 5712 images and the Testing folder consists of 1310 images.

**Source**: <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>

**Data Preprocessing**

For every image, the following preprocessing steps were applied:

* Crop the part of the image that contains only the brain (which is the most important part of the image).
* Resize the image to have a shape of (224, 224, 3) = (image\_width, image\_height, number of channels): because images in the dataset come in different sizes. So, all images should have the same shape to feed it as as input to the neural network.
* Apply Normalization: to scale pixel values to the range 0 - 1.

**Deep Learning Architecture**

**Convolutional eXtreme Gradient Boosting:**

* Convolutional Neural Networks, or CNNs, have grown in popularity in recent years for image classification and object identification applications, as many of you may already be aware. CNNs are made to recognize and extract characteristics from pictures automatically, but they can overfit and need a lot of computing power to train.
* XGBoost can help in this situation. A potent machine learning method called XGBoost makes predictions by using decision trees. It has been utilized in a number of applications, including fraud detection and customer churn prediction, and is renowned for its speed and accuracy.

**Advantages**

* ConvXGB combines the benefits of XGBoost with CNNs. ConvXGB's CNN component extracts features from pictures or other forms of input, and the XGBoost component uses these characteristics to produce predictions. ConvXGB, which combines these two techniques, provides a number of advantages over conventional CNN or XGBoost models, including increased accuracy, decreased overfitting, and quicker training periods.
* Preprocessing the data for training is one computational technique that is relevant. For picture data, this can include scaling the photos to a standard size or using techniques for data augmentation to expand the training set. Another approach involves modifying a pre-trained CNN using transfer learning on a fresh dataset, which can cut training time and boost accuracy.

**Model Implementation**

Implementing ConvXGB includes several steps.

1. Preprocessing: Preprocessing is the process of getting data ready for training. For picture data, this can entail scaling the photos to a standard size or using techniques for data augmentation to expand the training set.
2. Training the CNN: The CNN part of the model must first be trained before moving on to ConvXGB. In order to do this, the training data must be fed into the CNN, and the weights must be updated depending on the discrepancy between the expected and real labels.
3. Extracting Features: The next stage is to utilize the CNN to extract features from the training data after it has been trained. This entails running the CNN with the training data and capturing each layer's output.
4. Training the XGBoost model: The XGBoost part of the model must be trained as the last stage in the ConvXGB training process. This entails feeding the XGBoost algorithm with the characteristics that were retrieved from the CNN.
5. Evaluation: It's critical to assess the model's effectiveness on a validation set once it has been trained. Metrics like accuracy, precision, recall, and F1 score can be used to achieve this.

**Accuracy and Loss**

Chart, line chart

Description automatically generated

Chart

Description automatically generated

**Confusion Matrix**

Chart, treemap chart

Description automatically generatedChart, treemap chart

Description automatically generated**CNN** **XGBoost**

**Results**

Based on the information provided, it appears that the ConvXGB model developed has achieved good accuracy rates.

* Training accuracy rate of 99.30 %.
* Testing accuracy rate of 97.17 %
* The validation accuracy rate of 98.58 %.