## Project Progress Report

# Brain Tumor Detection: Machine Learning Classification

**University of Wisconsin - Madison**

Computer Science 539 - Introduction to Neural Networks

Group 6

Nicholas Beninato - Department of Computer Sciences - Undergraduate
Sebastian Murrell - Department of Computer Sciences - Undergraduate
Garrison Waugh - Department of Computer Sciences - Undergraduate

Saturday, November 19th, 2022

# 1 - Abstract

The goal of this project is to create a convolutional neural network that will be trained for multi-class classification on several images of human brain MRIs. This image set will contain MRIs for individuals with three types of brain tumors — glioma, meningioma, and pituitary — in addition to a control sample of individuals with no tumor at all. The ultimate goal will be the classification of glioma, meningioma, pituitary, or no tumor. As of this progress report, we have followed the example model from @jaykumar1607 on Kaggle to implement our own CNN to perform binary classification on an image containing a tumor or no tumor. We will later use what we learn to perform multi-class classification of these tumors.

## 2 - Introduction

Medical imaging analysis, while not a new field, is certainly a growing discipline with enormous potential for saving lives. With that being said, this abundant growth has not come without shadows of doubt around the technology — and for good reason. It is no surprise that patients would want another human to review MRI images, cell biopsies, or other scans to assure their cancer diagnosis, rather than relying on a pre-trained algorithm to confirm their diagnosis. In the United States today, cancer of all forms remains one of the leading causes of death across all cohorts [1]. And, while a doctor is still a necessary middle-man when diagnosing cancer with an algorithm, new research suggests that machine learning algorithms can diagnose diseases consistently with medical professionals [2]. Promising results with other diseases begs the question: why not apply deep learning methods to cancer image classification?

The current workflow when it comes to a brain tumor diagnosis typically begins with a series of neurological tests to assess brain function regarding balance, vision, and reflexes. Then, based on how these tests go, a doctor will order a scan of some sort (typically an MRI) and review the imaging results to determine if there is a tumor present in the brain. Then, finally, the doctor may order a biopsy of the suspected tumor to assess its type and whether or not it is cancerous [3].

It is easy to see where one may apply deep learning principles here. Instead of having doctors review these MRI images, healthcare professionals could instead look to deep learning algorithms to help assess these images. In fact, a recently published study found that deep learning models trained for brain tumor classification performed better in classifying and identifying brain tumors than neuroradiologists. Additionally, neuroradiologists that were provided with this same model outperformed their counterparts without the model in accuracy of diagnosing brain tumors [4].

# 3 - Method

## Data

Original Datasets

This dataset is a compilation of three existing datasets from Kaggle and FigShare. Altogether, it contains 7022 MRI brain scans in a JPEG format. This directory contains 4 sub-directories, each of which corresponds to one of three types of brain cancer, with the fourth and final sub-directory containing images of brain scans with no brain tumor present. This collection of images is a composite of 3 separate data repositories found online, each one providing varying brain scans:

Dataset 1: Brain Tumor Dataset

Jun Cheng, Southern Medical University, Guangzhou, China

Cheng's provides a large portion of the entire sample. The imaging comes from brain scans of 233 different patients with different forms of brain tumors: meningioma (n=708), glioma (n=1426), and pituitary tumor (n=930).

Dataset 2: Brain Tumor Classification

@sartajbhuvaji on Kaggle

This dataset contains several more samples from each of the tumor types, as well a control set. The control samples (n=500) come from Navoneel Chakrabarty's Kaggle dataset (here). The remaining samples (meningioma (n=937), glioma (n=926), pituitary (n=901)) come from a private dataset published by Swati Kanchan, though these images are publicly available in the main dataset above.

Dataset 3: Brain Tumor Detection 2020

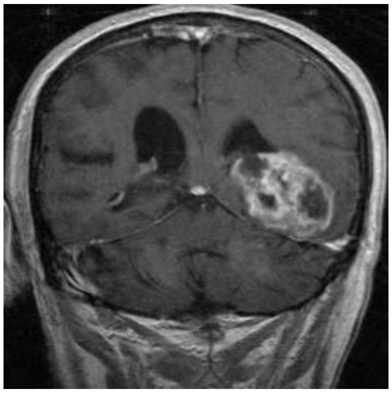
@ahmedhamada0 on Kaggle

This dataset contains data for binary classification of brain tumors. As such, we are only going to be using the "no tumor" subset (n=1500) in order to expand the size of our control group. We are opting not to use the images labeled as having a brain tumor as it would require a professional to manually label 1500 more images of brain tumors.

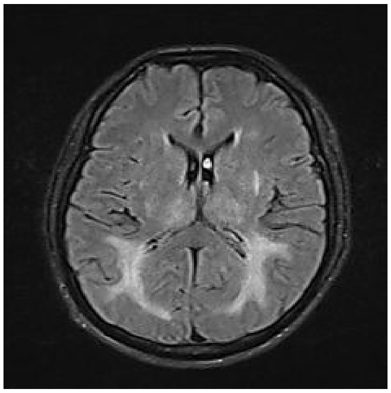
Main Dataset

To generate our data, we combined these 3 datasets. We went through each, which were composed of various formats, such as jpeg, matlab, and png, with many different shapes and channel numbers. As such, our first step was to make our data consistent in format and shape. In terms of format, we needed to first convert our .mat files (from the figshare dataset) into JPEG images. Following this, our newly converted JPEG images, along with several images from the other datasets, contained only one channel as they were mapped to be grayscale. To combat this, we stacked these grayscale images, adding two channels so that all images contain three channels. Our final challenge was regarding the shape of the images. Images originally ranged in size from around 100 x 100 to over 1300 x 1300. Additionally, not all of the images were square in nature. We used OpenCV to resize all images to 150 x 150 x 3. Once we had all images in a consistent format, we flattened them and saved them as a csv with the label as the first column and each pixel in their own column.

Data Example



**Meningioma Positive MRI**



**Healthy Brain MRI**

Tumors appear as large white masses superimposed on the surfaces of the brains. Due to this phenomena, our model is able to achieve relatively high accuracy through the use of average pooling to detect significant changes in grayscale coloring for each convolution.

## Algorithm/Program

### Model

For our model, we plan on constructing and training a convolutional neural network (CNN) that is able to successfully classify the existence of tumors as well as their type from an MRI image. Currently, there is a large amount of work available online in the field of tumor classification, most of which takes a similar approach to that of our group. As mentioned in the introduction, this is a widely used method in the medical field due to its accuracy and cost-effectiveness. For example, the National Institute of Health released a study outlining the use of CNN's in tumor identification, specifically highlighting the contrasts between discriminate image analysis such as K-Nearest Neighbors (kNN), Artificial Neural Networks (ANN) and CNN's [5].

Our current iteration of this model uses binary classification to determine presence of tumors in our MRI scans. We have taken inspiration from the works of Jay Kumar and implemented transfer learning in the development of this binary classifier in the form of EfficientNetB0. We chose this model as a base for our classifier as it has already been trained on ImageNet and therefore has weights that will be useful in image classification. By fine tuning to the specifics of brain scans using this pretrained model, we hope to experience much more efficient training times than if we were to start from scratch.

| | |
|---|---|
| Total Params: | 4,050,852 |
| Trainable Params: | 4,008,829 |
| Non-trainable Params: | 42,023 |

**Model Summary**

### Platform

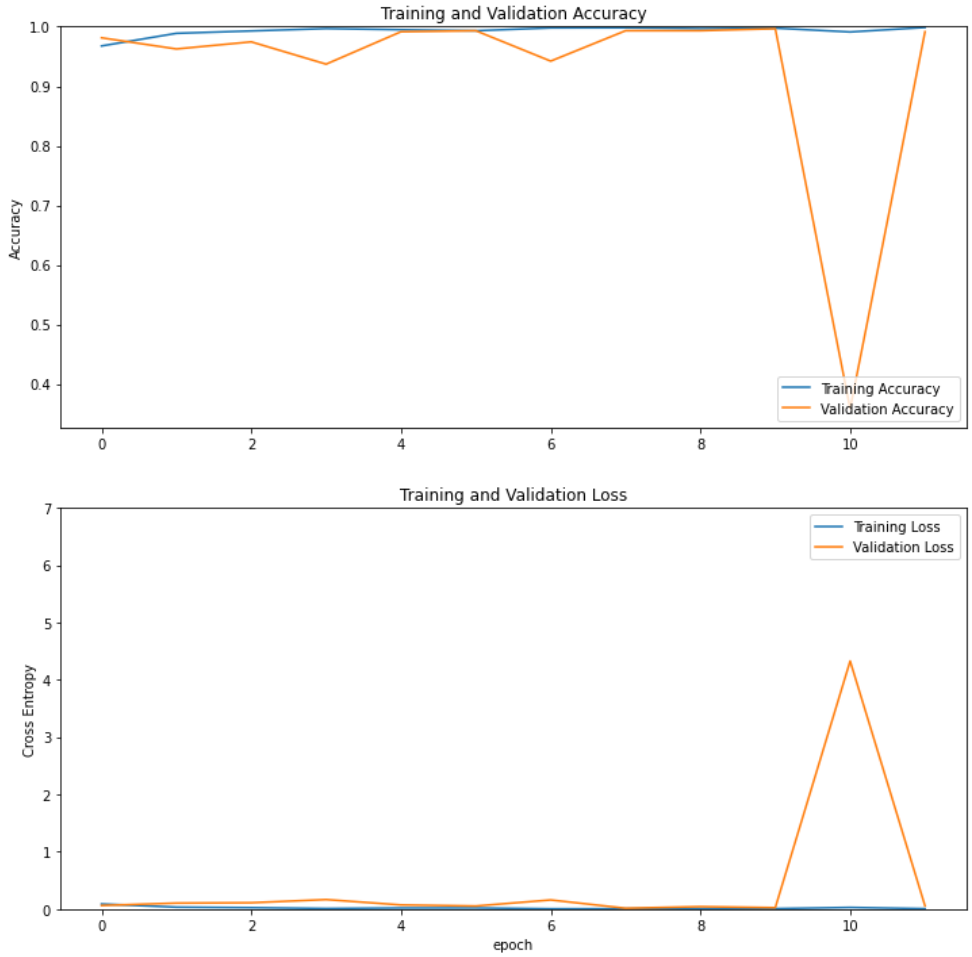
For this project, we have written the majority of our code in IPython-Notebooks (ipynb) hosted in Google Colab. Hosting our ipynb in Colab allows them to be collaborative, whereas an ipynb hosted on a virtual machine or on a local machine will not allow for collaborative development. Additionally, the use of notebooks as opposed to modules allows us to make small changes to our model efficiently. This code has also been uploaded to our GitHub repository (here). In addition to the code for our model, the code used for transforming our original files into the consistent images used for training, evaluation, and testing has also been uploaded to the GitHub repository.

# 4 - Results

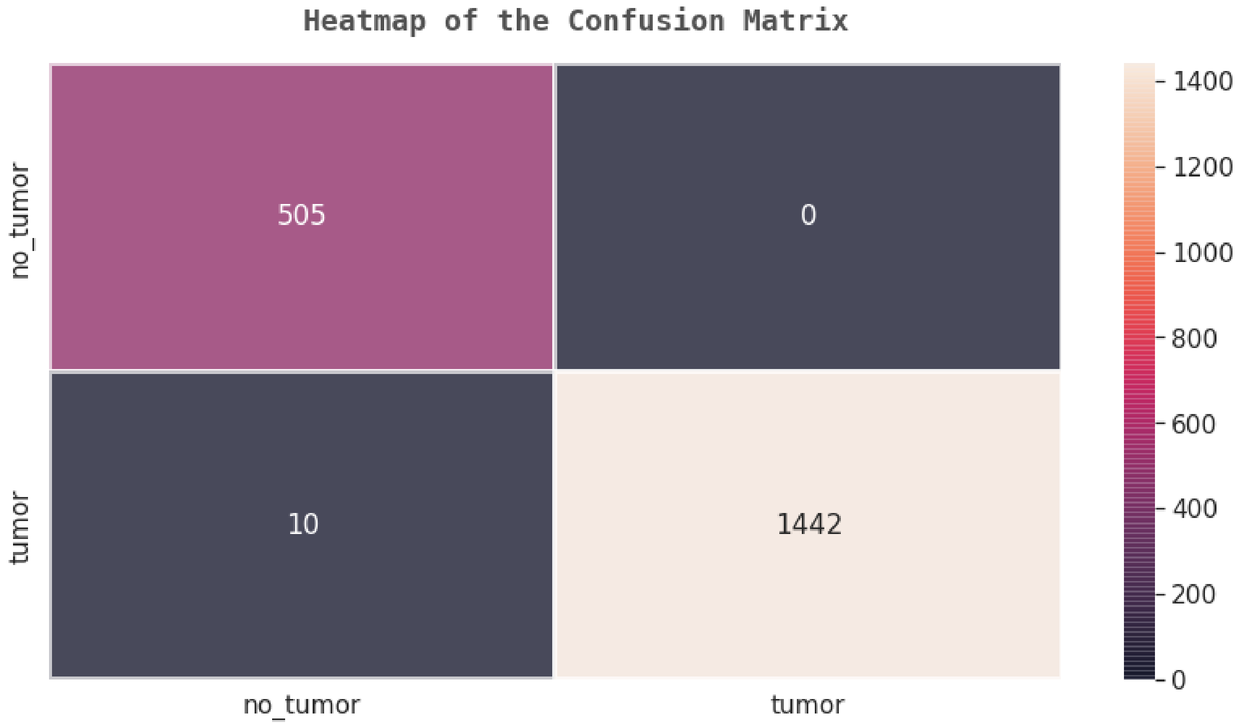
Our baseline was the Kaggle Project here, which achieved accuracy of 98%. Using their code, we were able to replicate the same results on our own dataset. The baseline uses the EfficientNetB0 model with weights from ImageNet.

Our binary classification model was able to achieve an accuracy of  >99%. The confusion matrix and graph below shows the current predictive ability of this model with an output shape of (,2) and the training accuracy per epoch.

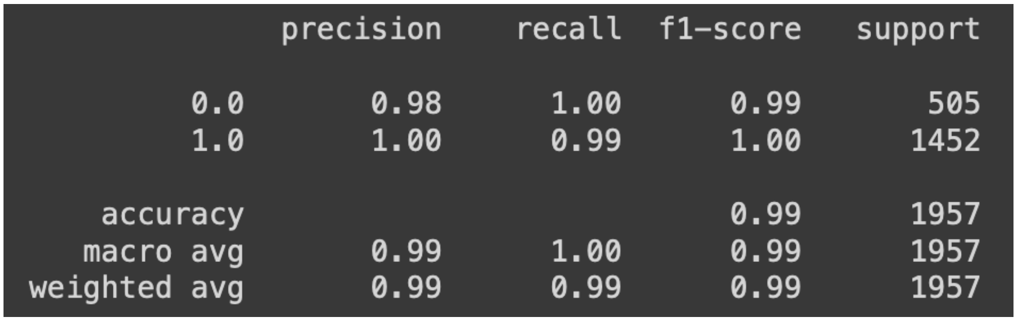
Our model is showing extremely high accuracy for the binary classification of tumors. We believe that this reason for such high accuracy comes from the transfer learning of EfficientNetB0. Since the model comes pre-trained on ImageNet, the accuracy after the first epoch was 97%. With such a high starting classification rate, the model was still able to make gains from its 12 epochs of training, ending at a misclassification rate of less than 0.01. We do not expect such high accuracy from a multi-class classifier. This is because the class variance will be lower as we will be testing for 3 different types of tumors rather than their general presence.



**Training/Loss per Epoch**



**Confusion Matrix {no_tumor, tumor}**



```
              precision    recall  f1-score   support

         0.0       0.98      1.00      0.99       505
         1.0       1.00      0.99      1.00      1452

    accuracy                           0.99      1957
   macro avg       0.99      1.00      0.99      1957
weighted avg       0.99      0.99      0.99      1957
```

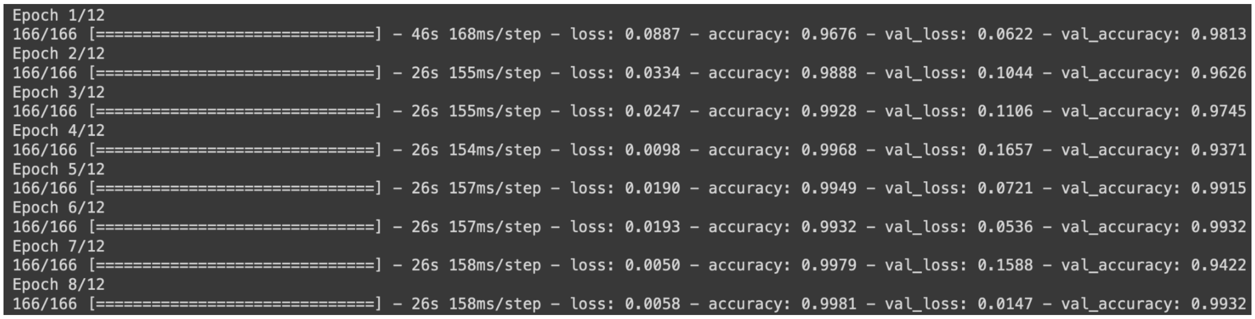
**Model Classification Report**

# 5 - Discussion

For the final iteration of our model, we want to evolve from a binary classifier to multi-class. This means that we will have to retrain our model on 4 classes in the additional layers we will be adding. Since the model is currently at a place where it is able to determine the presence of tumors, its ability to distinguish their type involves the addition of layers while maintaining the current weights for binary classification. This way, we will be able to implement early stopping if it is determined no tumors are present. Another goal for our next model is to add padding to our early convolutions. As shown in our Data Example earlier, the images are not all scaled equally, so zero-padding will be necessary to remove the darker borders which do not contain relevant information for the model.

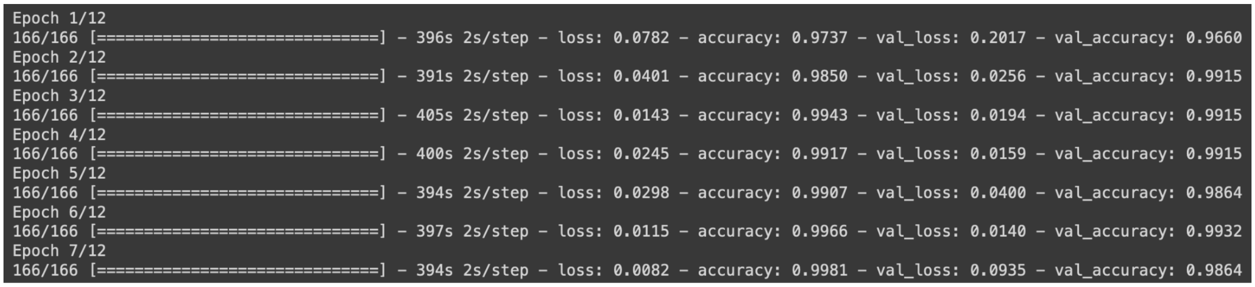
We did encounter some difficulties training our model. The free version of Google Colab did not have enough ram to support learning on our combined csv of all three datasets. We suspect this is because there were over 7000 total images. We overcame this issue by recruiting a GCP virtual machine with a 13GB graphics processing unit. This took our training time per epoch down from 7 minutes to 20 seconds and enabled us to make improvements to our code at a much faster rate.

At its current state, we believe that we can achieve even better results when predicting for multiple classes as opposed to binary classification. The model already has near perfect accuracy, so any increase in predictive power may appear marginal, however it will be significantly more challenging for our network to predict 4 classes as opposed to 2.



```
Epoch 1/12
166/166 [==============================] - 46s 168ms/step - loss: 0.0887 - accuracy: 0.9676 - val_loss: 0.0622 - val_accuracy: 0.9813
Epoch 2/12
166/166 [==============================] - 26s 155ms/step - loss: 0.0334 - accuracy: 0.9888 - val_loss: 0.1044 - val_accuracy: 0.9626
Epoch 3/12
166/166 [==============================] - 26s 155ms/step - loss: 0.0247 - accuracy: 0.9928 - val_loss: 0.1106 - val_accuracy: 0.9745
Epoch 4/12
166/166 [==============================] - 26s 154ms/step - loss: 0.0098 - accuracy: 0.9968 - val_loss: 0.1657 - val_accuracy: 0.9371
Epoch 5/12
166/166 [==============================] - 26s 157ms/step - loss: 0.0190 - accuracy: 0.9949 - val_loss: 0.0721 - val_accuracy: 0.9915
Epoch 6/12
166/166 [==============================] - 26s 157ms/step - loss: 0.0193 - accuracy: 0.9932 - val_loss: 0.0536 - val_accuracy: 0.9932
Epoch 7/12
166/166 [==============================] - 26s 158ms/step - loss: 0.0050 - accuracy: 0.9979 - val_loss: 0.1588 - val_accuracy: 0.9422
Epoch 8/12
166/166 [==============================] - 26s 158ms/step - loss: 0.0058 - accuracy: 0.9981 - val_loss: 0.0147 - val_accuracy: 0.9932
```

**Training with GCP virtual machine (~26 s / epoch)**



```
Epoch 1/12
166/166 [==============================] - 396s 2s/step - loss: 0.0782 - accuracy: 0.9737 - val_loss: 0.2017 - val_accuracy: 0.9660
Epoch 2/12
166/166 [==============================] - 391s 2s/step - loss: 0.0401 - accuracy: 0.9850 - val_loss: 0.0256 - val_accuracy: 0.9915
Epoch 3/12
166/166 [==============================] - 405s 2s/step - loss: 0.0143 - accuracy: 0.9943 - val_loss: 0.0194 - val_accuracy: 0.9915
Epoch 4/12
166/166 [==============================] - 400s 2s/step - loss: 0.0245 - accuracy: 0.9917 - val_loss: 0.0159 - val_accuracy: 0.9915
Epoch 5/12
166/166 [==============================] - 394s 2s/step - loss: 0.0298 - accuracy: 0.9907 - val_loss: 0.0400 - val_accuracy: 0.9864
Epoch 6/12
166/166 [==============================] - 397s 2s/step - loss: 0.0115 - accuracy: 0.9966 - val_loss: 0.0140 - val_accuracy: 0.9932
Epoch 7/12
166/166 [==============================] - 394s 2s/step - loss: 0.0082 - accuracy: 0.9981 - val_loss: 0.0935 - val_accuracy: 0.9864
```

**Train with Local Machine (~400 s / epoch)**

# 6 - References

[1] Centers for Disease Control and Prevention. (2022, September 6). *FASTSTATS - leading causes of death*. Centers for Disease Control and Prevention. Retrieved October 1, 2022, from https://www.cdc.gov/nchs/fastats/leading-causes-of-death.htm

[2] Gaël Varoquaux, Veronika Cheplygina. (2022, April 12). *Machine learning for medical imaging: methodological failures and recommendations for the future.* Springer Nature Limited. Retrieved October 1, 2022, from https://www.nature.com/articles/s41746-022-00592-y

[3] Johns Hopkins Medicine. *Brain Tumors and Brain Cancer*. The Johns Hopkins University. Retrieved October 1, 2022 from https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor#:~:text=Diagnosing%20a%20brain%20tumor%20usually,%2C%20hearing%2C%20vision%20and%20reflexes.

[4] Gao P, Shan W, Guo Y, et al. *Development and Validation of a Deep Learning Model for Brain Tumor Diagnosis and Classification Using Magnetic Resonance Imaging*. JAMA Netw Open. 2022;5(8):e2225608. doi:10.1001/jamanetworkopen.2022.25608

[5] Díaz-Pernas, Francisco Javier et al. “A Deep Learning Approach for Brain Tumor Classification and Segmentation Using a Multiscale Convolutional Neural Network.” Healthcare (Basel, Switzerland) vol. 9,2 153. 2 Feb. 2021, doi:10.3390/healthcare9020153