Classification of Types of Brain Tumors in MRIs Through Machine Learning Algorithms and Feature Extraction Methods

Tommy Chung, Yi Jin, Patrick Liu
Project Repo: https://github.com/PatrickLiu022/Brain-Tumor-Analysis

Research Questions

1. What radiomic features are in brain tumors MRIs to train a machine learning model?

   A total of 122 radiomic features were extracted from a pair of brain MRI and mask file with feature extraction method provided by PyRadiomics. We decreased the number of features into 65 to remove the potential of overfitting and to increase the accuracy.

2. Which machine learning algorithm would work well for a multi-class classification of brain tumors in MRIs?

   We used Decision Tree Classifier to train a model to classify three types of tumors with given features. A model with 9 depth, 95 leaf nodes, and 64 features had the highest accuracy of about 0.85 accuracies.

3. What features are significant in the model and why?

   As a result of feature importance analysis, the most significant feature in our model was Gray Level Co-occurrence Matrix Cluster Shade. It scored 0.31 feature importance, which is twice of the second most significant feature.

Motivation and Background

Brain tumors are abnormal cells in your brain that grow out of control. The most common types of brain tumors are meningiomas, gliomas, and pituitary adenomas. To briefly introduce the characteristics of them, Meningioma tumors are mostly located in the outer three layers of tissue between the skull and the brain that cover and protect the brain just under the skull. Gliomas are called

intra-axial brain tumors because they grow within the substance of the brain and often mix with normal brain tissue. A pituitary tumor is an abnormal growth of cells in the pituitary gland, which is located in the center of the brain behind the nose and eyes.

While some brain tumors are benign and harmless, a lot of the brain tumors are very dangerous. For example, gliomas, the most fatal brain tumor, develops very quickly. They can cause headaches, seizures, loss of senses, and even death. Every year, about 10600 people are diagnosed with brain tumors. Brain tumors have become the biggest cancer killer for people under 40. To diagnose brain tumors, your physicians will do imaging studies such as MRI or CT and try to take a small sample of the tumor for examination under a microscope. This process of removing a sample of the tumor can be dangerous to patients and also time and resources-consuming for doctors. If we could identify the types of tumors only with the medical images, we could save a lot of time and resources. The earlier it is discovered and treated, the better prognosis it will have, and therefore more lives will be saved.

The main goal of our study is to build a machine learning model that can detect and differentiate the types of brain tumors with MRIs. This model can be used to build a computer diagnosis system that can diagnose and alert doctors and patients much faster than humans. It will also facilitate matching medical resources to the most needed. With early detection and fast diagnosis, the patients will have a better chance of survival.

Dataset

https://www.kaggle.com/awsaf49/brain-tumor

Our data set contains 3064 T1-weighted contrast-enhanced MRI images from 233 patients with three types of brain tumors and mask images indicating the region of interest of the tumors. Three tumors we will be analyzing are meningioma, glioma, and pituitary tumors. There are 708 slices of meningioma, 1426 slices of glioma, and 930 slices of the pituitary tumor as shown in Figure 1.

The MRI and mask images have a size of 512 * 512 and are stored as .npy files. The MRI images were taken from different axises from a patient. Mask file shows the border and the region of the cancer. In Figure 2, you can see the images and masks for those three kinds of brain tumors.

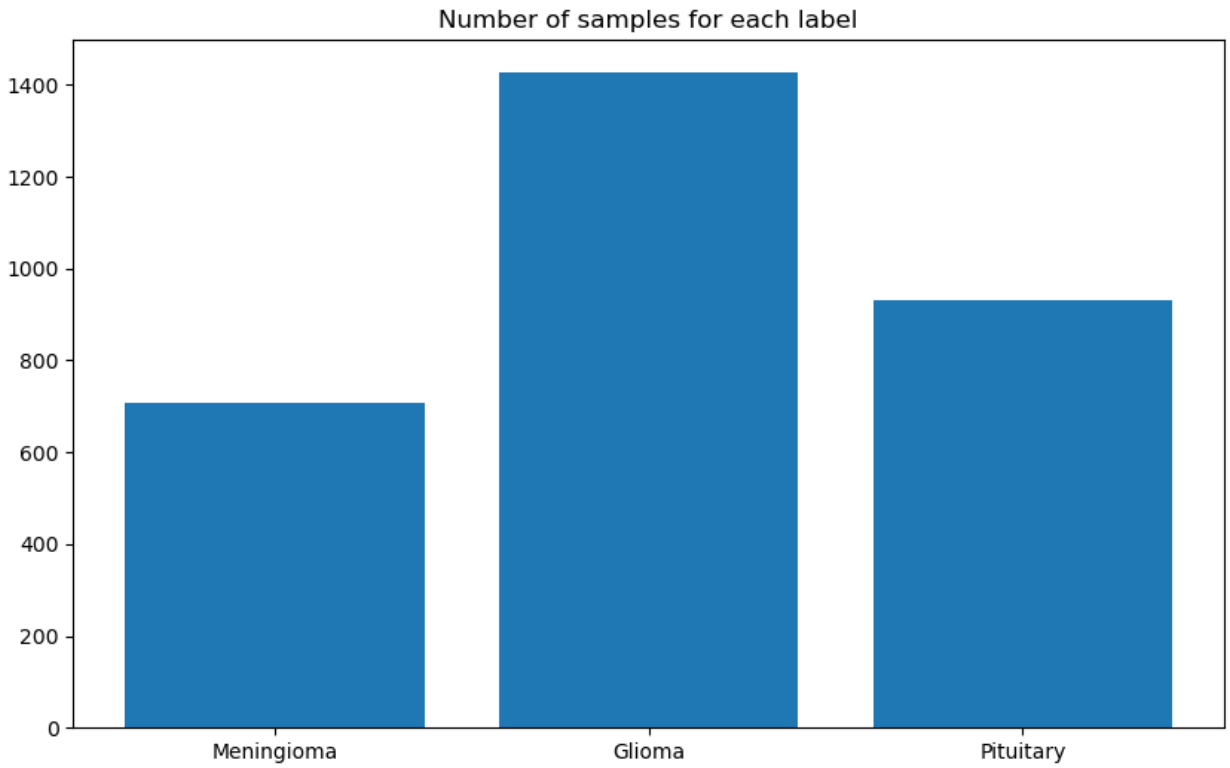

Figure 1: Number of samples for Meningioma, Glioma and Pituitary tumors

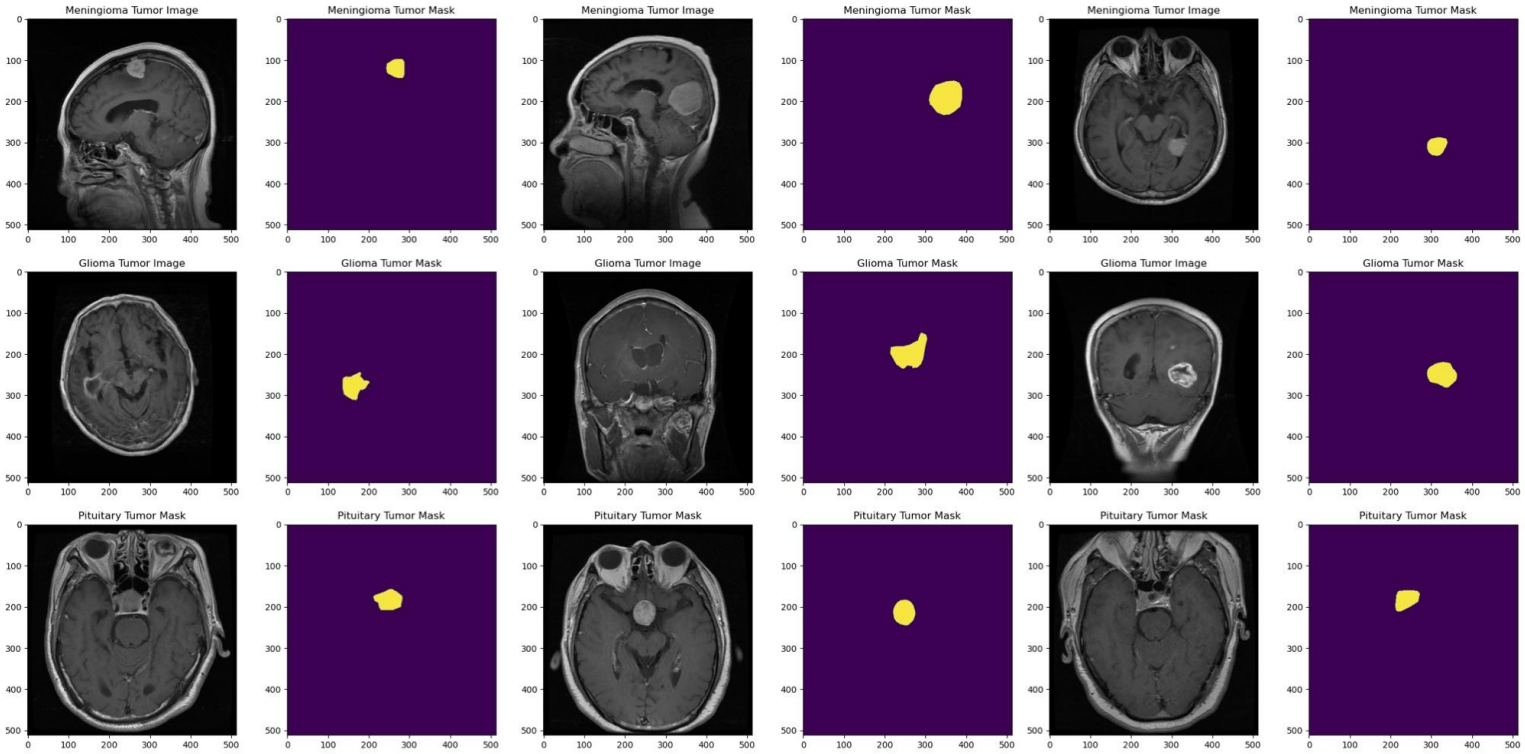

Figure 2: Examples of brain scans for Meningioma, Glioma and Pituitary tumors

Methodology

We used a Python library, PyRadiomics, which is specialized in medical images to extract radiomic features from MRIs and masks of brain tumors. It extracted more than 100 features per image, including First Order Statistics, Gray Level Co-occurrence Matrix (GLCM), Gray Level Run Length Matrix (GLRLM), Gray Level Size Zone Matrix (GLSZM), Neighbouring Gray Tone Difference Matrix (NGTDM), and Gray Level Dependence Matrix (GLDM). These sets of features, excluding first-order, are based on second-order statistics, which describe correlations between pairs of pixels in different aspects such as homogeneity or uniformity.

Since we have more than a hundred features, we should reduce the number of features to avoid overfitting of the trained model as well as to discard useless features. We threw away all columns with non-numeric values. Using the feature selection method, we removed all features whose variance doesn't meet threshold p = 0.8. After that, we used a heat map to see how features are correlated.

$$\mathrm{Var}[X] = p(1-p)$$

We used Decision Tree Classifier to classify the types of brain tumors with the features. We fit and train the model with 70% of the data, leaving the remaining 30% as the test data. Then we considered the hyperparameters to prevent overfitting and to get the highest accuracy of what type of disease should be diagnosed for the patient. We compared the accuracy of the model for the test set with different numbers of max depth, max features, and max leaf nodes to see if the model can predict at a 70% level. We also compared how important each feature is in predicting the types of tumor using the feature importance method.

Afterward, we made another model that identifies the tumor and tumor type only with a given scan. We can do this by highlighting the masks on the image directly. To highlight the region of abnormality, we first transformed the mask data from booleans to the number 1 for regions with no tumor and 2 for regions with the tumor. Then we sum up each row for each image in the image file and divided by the constant 3064 to get numbers closer together.

Having our setup complete, we used a deep learning model to predict which image contains which tumor. To do this, we specified in our hyperparameters to have 4 hidden layers with 50 nodes each and split up images and their true labels to 70% for training and 30% for testing. After this training, we could now have our model predict the label of an image. To help visualize the prediction, we highlight the tumor area and give the plot a title for the true tumor label and the predicted label.

Result

1. What radiomic features are in brain tumors MRIs to train a machine learning model?

As a result of feature extraction, a total of 122 radiomic features were extracted from a pair of brain MRI and mask files. It is highly expected that our model will cause an overfitting issue without feature selection because Pyradiomics provides more than a hundred features. As a part of feature selection, we deleted inappropriate data such as non-integer and redundant values, and 99 features were left. After, we removed features with low variance. As a result, 65 features were left to train a model. We calculated the correlation between the features using a heatmap.

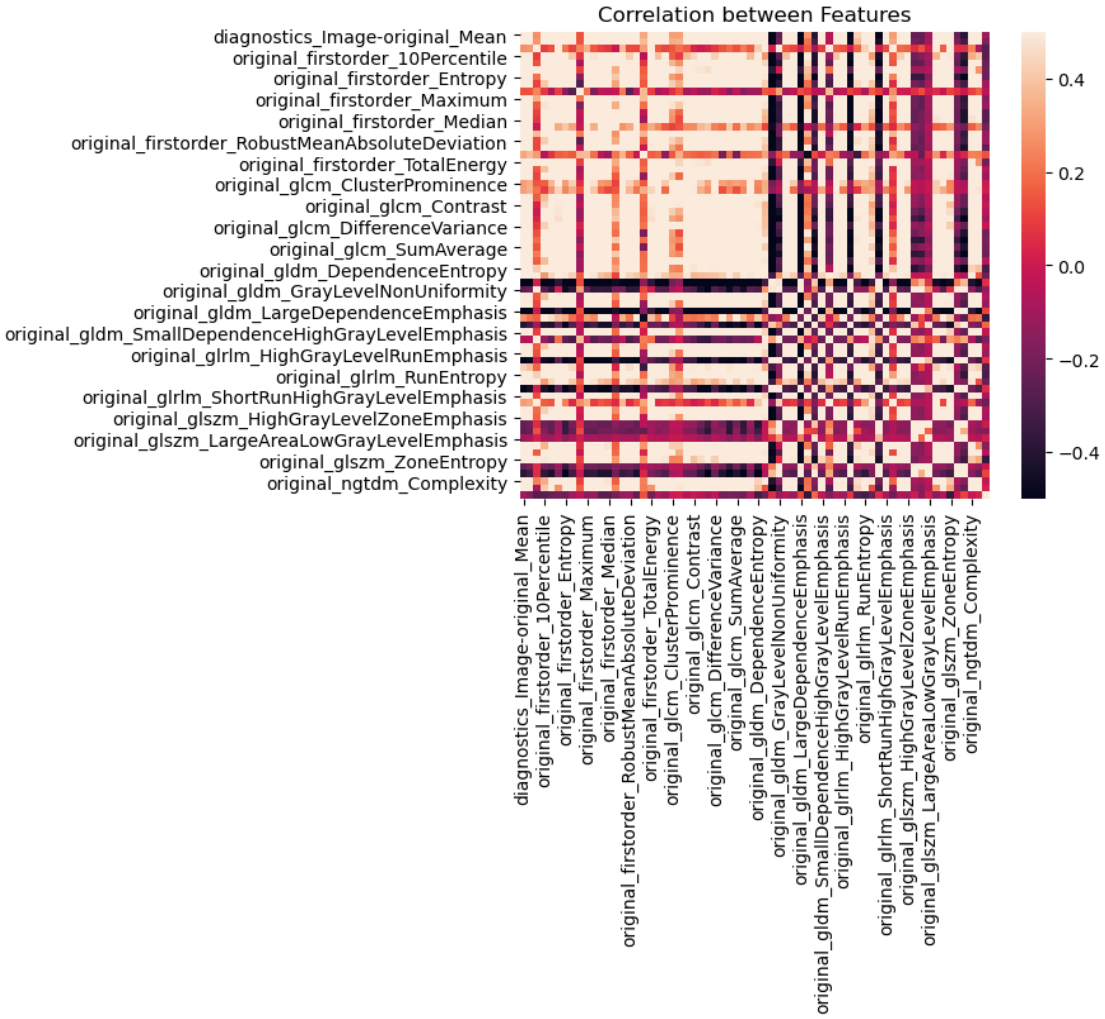

Figure 3: Correction Map between Features

We can see both strong positive and strong negative correlation between features on the heatmap. PyRadiomics does convolution with several kernels to the given images and generates features based on

the relationship between pixels. To introduce seven kernels that are used for our features, first-order features are in concern of single-pixel rather than any other relationship between pixels of an image. A GLCM represents the number of times the combination of specific two numbers occurring in two neighboring pixels in the image. For a GLSZM, it provides a size of consequence and collinear pixels that share the same gray level intensity, where pixels are considered connected if the distance is 1. Alike to GLSZM, a GLRLM quantifies pixels' length that has the same gray level values, but the pixels only have to be consecutive to be counted. NGTDM compares a gray value of a pixel and an average value of neighboring pixels and stores it as a form of a matrix. For the last one, GLDM measures the number of connected pixels, which are dependent on the center pixel.

2. **Which machine learning algorithm would work well for a multi-class classification of brain tumors in MRIs?**

We used Decision Tree Classifier to train our model to classify the types of tumors for given features. We compared the accuracy of the model as the max depth, max features, and max leaf nodes changes. As a result, we concluded that our model has the highest accuracy of 85% at max depth = 9, max feature = 64, and max leaf nodes = 95, which is shown at figure 4. Although this value can be differentiated due to the randomness of the splitting data set and training model, it successfully exceeds our expected accuracy of 70%. Our ways to extract, reduce features, and train the model worked well in the classification of types of brain tumors in MRIs.

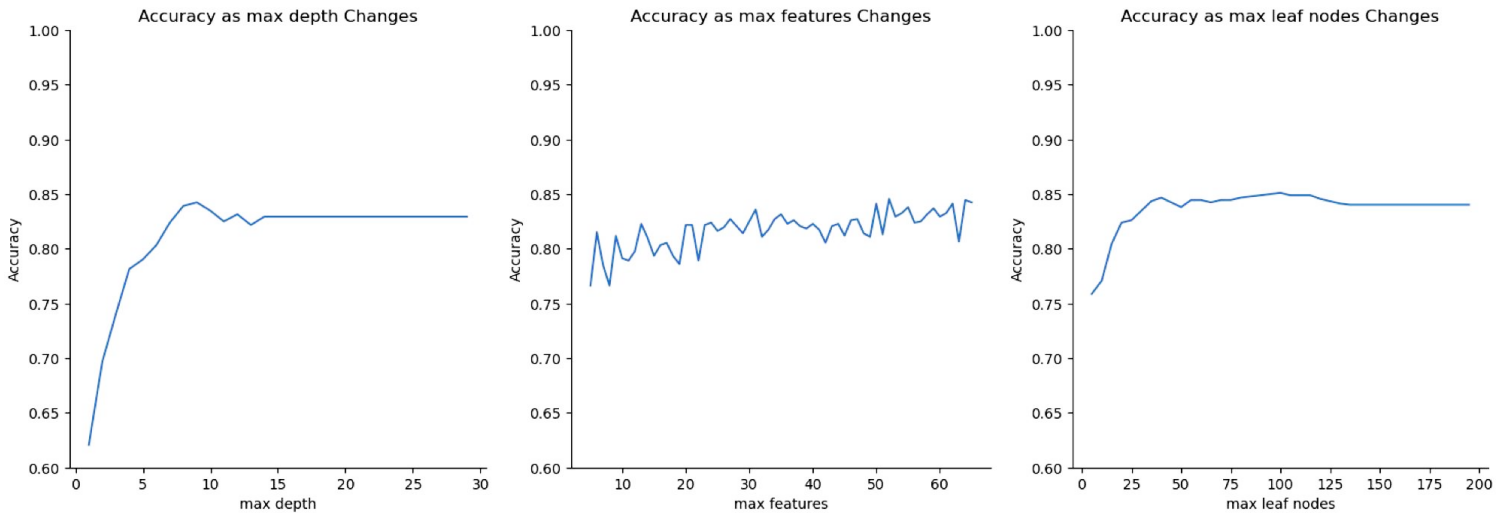

Figure 4: Accuracy as Hyperparameters Change

We analyzed how successfully the model predicted each category of brain tumor with the normalized confusion matrix of the predicted label and test label. As shown in figure 5, our model had the highest accuracy at predicting the glioma tumor and the lowest accuracy at predicting the meningioma tumor. It is not surprising to have this result because we had 1426 images of glioma but only 708 images for meningioma. A larger number of samples of certain labels can cause biased results in the prediction. Nonetheless, 78% accuracy of correct prediction for meningioma cancer is still greater than what we expected, so our model is not too biased to reject our result.

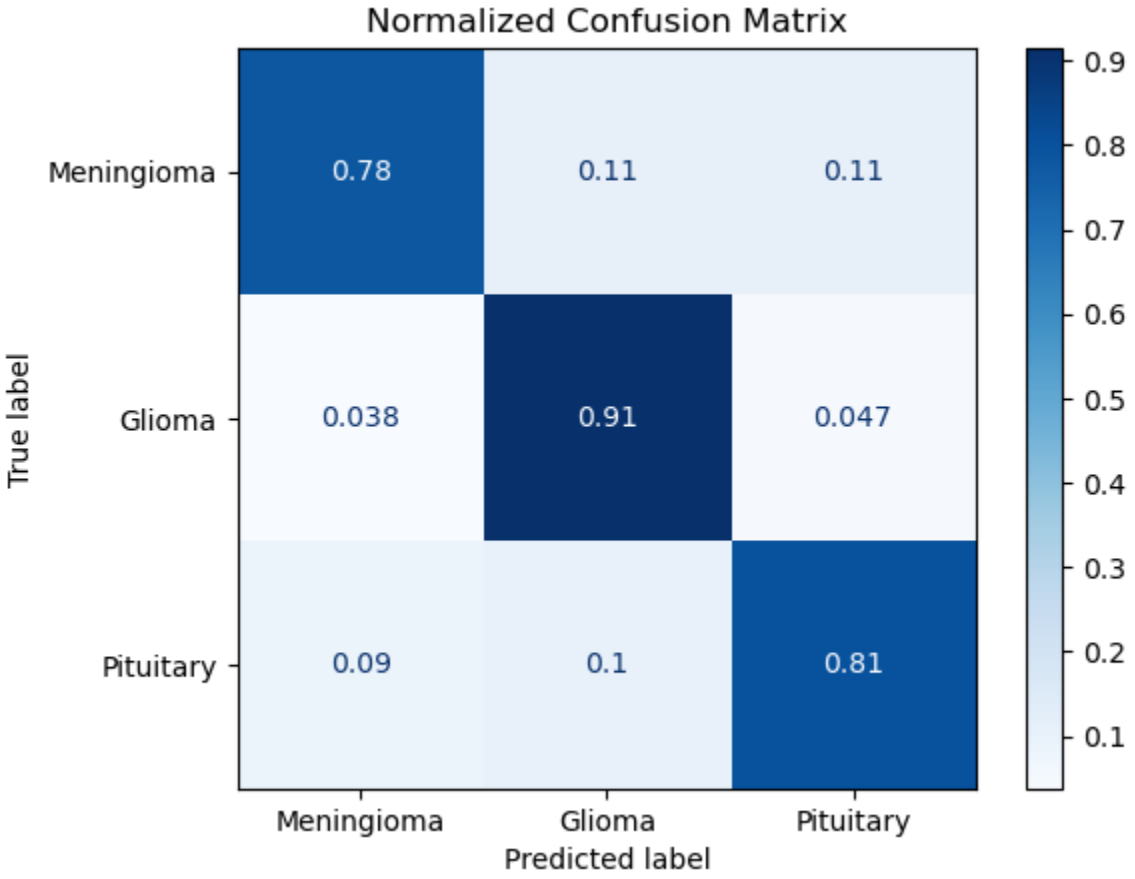

Figure 5: Normalized Confusion Matrix of True label and Predicted label

Our model predicted pituitary better than meningioma cancer. The pituitary can be much easier to predict because it is located only in the pituitary gland, which is in the center of the brain behind the nose and eyes. Due to the small size of the pituitary gland, it can be hard to capture cancer itself. However, since we are trying to know the type of tumors, it is much easier because all pituitary tumors have the exact location. The other two tumors grow in various locations compared to the pituitary, so

this might be a reason why predicting pituitary tumors had higher accuracy than predicting meningioma tumors.

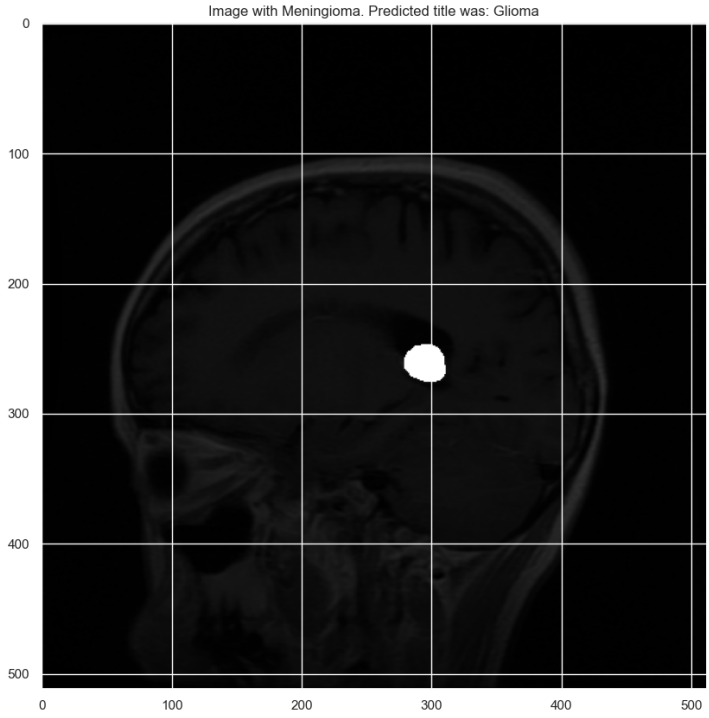

Figure 6: MRI of Meningioma as which MLP model predicted Glioma

In addition to our machine learning model, we wanted to try different methods other than the feature extraction method in differentiating brain tumors, so we made a neural network model that can work only with MRI images. We used MLP Classifier to train a model, which at first showed only 30% of accuracy. However, after trying different hyperparameters, we found that 4 hidden layers with 50 nodes would score the highest accuracy rate which is around 58% ~ 60%. Our machine learning model which used features from MRIs and masks did better in identifying the types of brain tumors than the neural network model. With these two models, we are confident that our study contributes to the development of a computer-aided diagnosis system that can help those in the medical fields in the future.

3. **What features are significant in the model and why?**

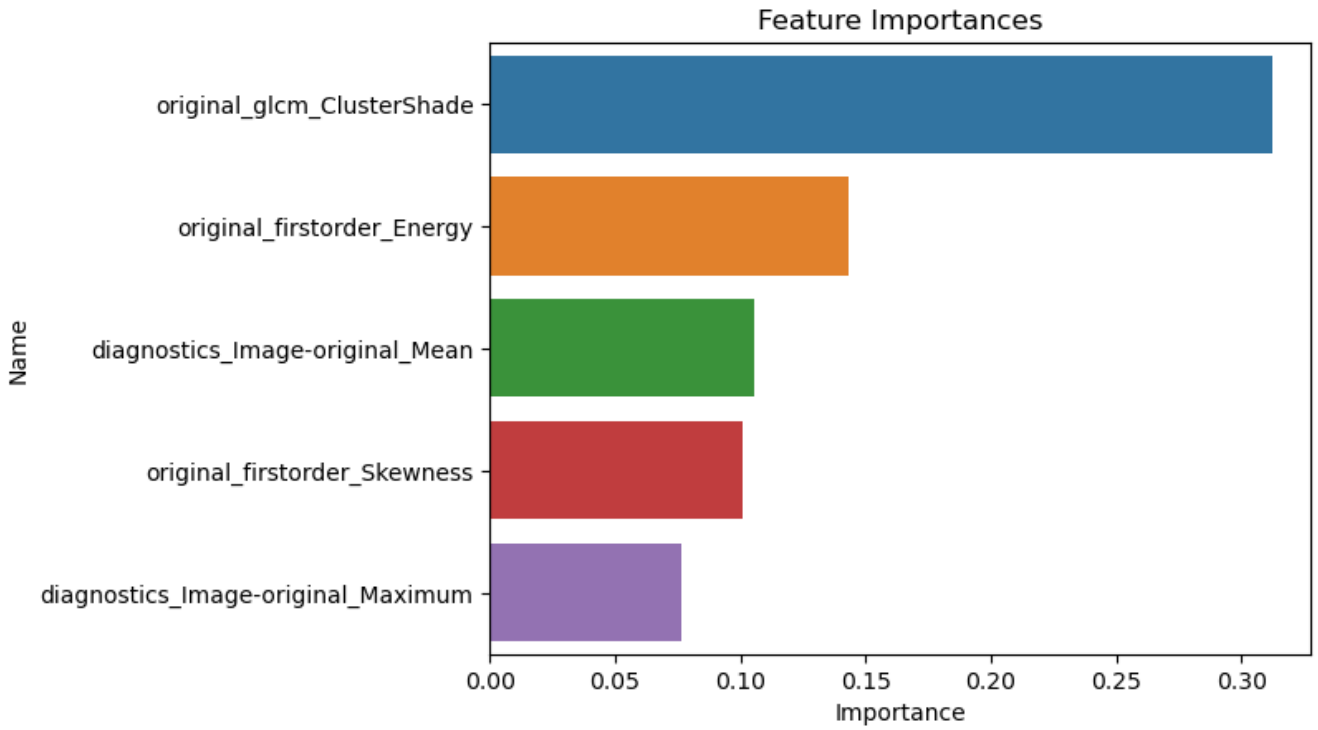

Figure 7: Features with Importance greater than 0.05 in our model

We measured the importance of each feature from zero to one, where zero indicates that the feature is doing no work in the classification and one means the single feature can decide the correct answer. Among 65 features, there were only five features having feature importance higher than 0.05. The most significant feature in our model was Gray Level Co-occurrence Matrix ClusterShade (0.31). It scored more than twice of First Order Energy, the second most significant feature (0.11). Based on the result, there is a significant distinction between MRIs of different brain cancers, and GLCM ClusterShade was the best feature that can recognize the differences.

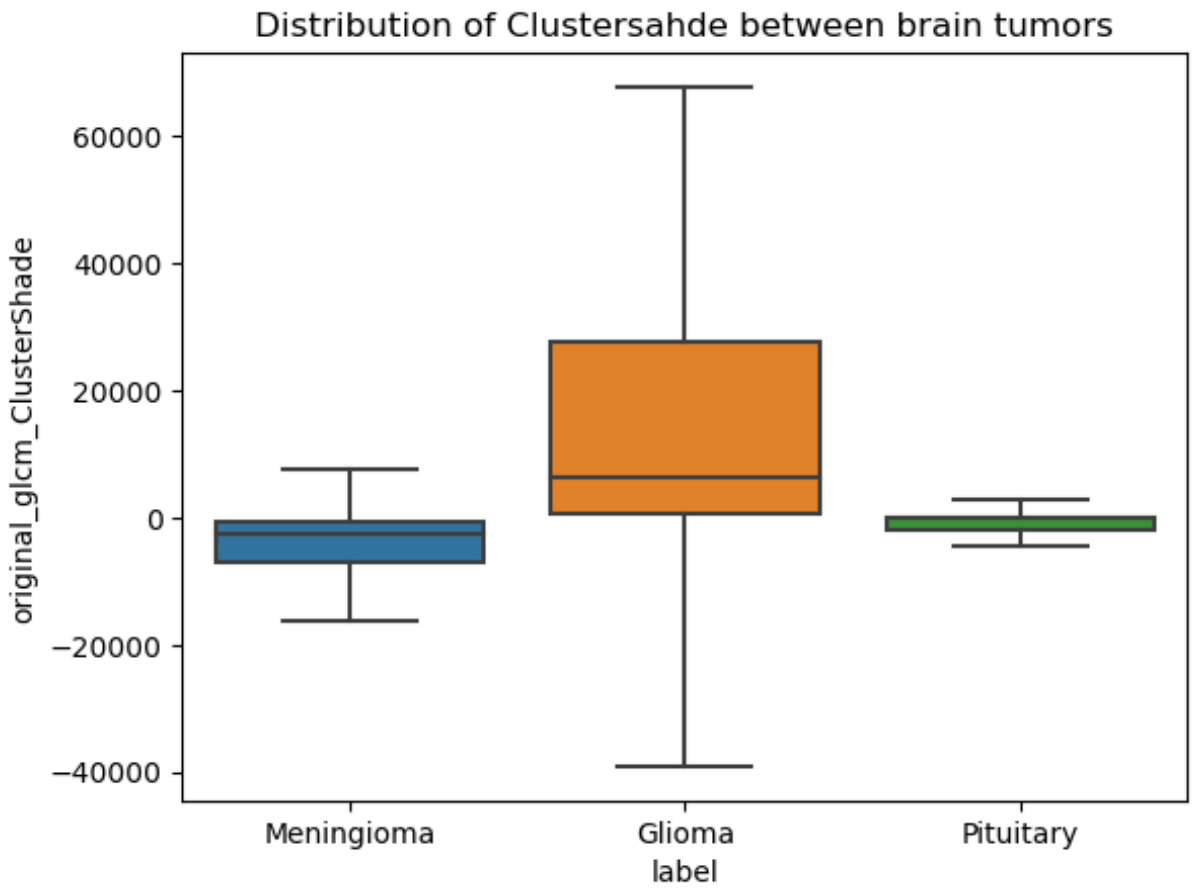

Figure 8: Box plot of Cluster Shade between Types of Brain Tumors

Cluster shade of the image means the asymmetry of the distribution of pixel value of GLCM about the mean. It is a measure of homogeneity and uniformity of the matrix. The image tends to be asymmetric and skewed when the cluster shade is higher. When the MRI image has a high variance rather than the consistency and smoothness of pixels, it would appear the rapid changes of darker and lighter pixels in the region of interest. In figure 7, we can see that Glioma cancer has a relatively higher rate of cluster shade compared to the other two cancers. While most meningioma and pituitary tumors have a negative value of cluster shade, glioma cancers have a significantly greater value of cluster shade, which means glioma cancers are captured in MRIs with higher contrast of pixels. In MRI imaging, the intensity of the object chooses the pixel color of the results. Meningioma and pituitary tumors locate the outer part of the brain, whereas glioma grows within the substance of the brain and normal brain tissue. It makes a difference of glioma showing more contrast enhancement (white on the outside) and necrosis in the middle (looks black on the MRI), as shown in figure 2.

Challenge Goals:

- **New Library**
    - The reason our research questions will meet this challenge is that we are trying to learn to use a new library in python: Pyadiomics. This library allows us to extract radiomic features from our brain tumor scan database and helps us build a machine learning model on identifying different kinds of brain tumors.
    - In our code, we extracted more than 100 features using the Pyradiomics library per image. These sets of features are mostly based on second-order statistics, which describe correlations between pairs of pixels in different aspects such as homogeneity or uniformity. With exploration and experimentation, this library has become a powerful and easy-to-use toolset for us. We are confident in using this library against other image-based datasets. So, it is safe to say that we achieved this goal.
- **Machine learning: Accuracy**
    - This is a challenging goal because we are using machine learning to gain information about our dataset. We will build a model to recognize which type of brain tumor it is given an MRI scan. We will extract the most relevant features using Pyradiomics and build the most accurate model. We aim to improve the accuracy of our model to 70%. We will separate our data into a training set and a testing set, and our goal is that the accuracy of the testing set will be at least 70%.
    - We tried to find the most adequate model for our problem, and as a result, used the Decision Tree Classifier to build our machine learning model. We analyzed it with different settings of its hyperparameters to find the best accuracy: max depth, max leaf nodes, and max features. In general, our accuracy is about 85%. This is well above our goal, which is 70%. We also analyzed which feature was the most important in the model and interpreted why it happened. This is the evidence that we have achieved our goal.

Work Plan

Part 1

1. Feature extraction: 3 hours
    - Using Pyradiomics to extract radiomic features from the images and masks
2. Data Summary and Selection: 3 hours
    - When we have all features extracted, we will casually look over the data to know the correlations and drop insignificant features
3. Machine Learning and Visualize results: 10 hours
    - We'll use a number of different algorithms to train a model and compare a result with ROC curves to know the best model. Afterwards, we'll use the feature importance method to know the most significant feature in the model.

4. Analyze results in words: 10 hours
    - Analyze results with visualizations. Discuss why we got such a result, what can be done to improve and future development.

Part 2

The partitioning of the time on each part was mainly accurate based on our actual experience. We spent most of our time on machine learning as well as analysis. However, we didn't anticipate that multiclass classification was our biggest obstacle. We ended up using way more time than we expected to solve this problem: we used different algorithms and then compared them, but only one worked. Instead, we compared the accuracy of the model depending on different parameters. Also, we used a neural network to train a model that can classify types of brain tumors without the mask files, which we didn't plan in the previous assignments.

We created a common workspace on google colab for our initial jupyter notebook codes, then we converted them into .py files and uploaded to a shared repository on github which we used for our final submission. During the project, we each performed a part for this project. Tommy wrote the machine learning part of the codes. He extracted and selected features, trained and evaluated the ML model. He also wrote the test functions for the ML models and contributed greatly to result 1 and 3 in this report. Patrick wrote the deep learning part of the codes. He trained and evaluated neural network models, organized all of the codes together and uploaded them to github. He also wrote and edited classes and methods comments, result 2 of this report, and instructions on how to run the project in README.md. Yi wrote and edited all of the other parts in the report and documentations of the codes. She also put together the presentation and the slides.

Testing:

We visualized our data set to know if there is an unexpected difference in our data set. Figures 1 and 2 were the results of it, and we checked there are three types of tumor and images match to the corresponding masks.

For our codes, it was hard to write a test function because we couldn't predict what features would be extracted from original data and remained after feature reduction. We decided to concentrate on preventing the loss of data in the process. We made a smaller data set of 30 samples and compared the number of rows after extracting the feature as well as reducing the feature. We also looked over the csv file we created to know if there is a loss of data while saving and reading the file.

We can assure that each image and mask produced a single row of features, and there was no loss of rows while reducing the number of columns. As a result, two models we created were well trained with the correct data.

Collaboration:

We used the codes from lesson slides about how to compare and graph the accuracy of the model depending on different hyperparameters. We also used cse163_utils.py to test our codes.