Final Project Report

Sicen (Susan) Liu, Tuxun (Nick) Lu, Wenxuan Lu, Xuerui (Jerry) Qian

The project tackles the problem of brain tumor segmentation and tumor type classification. Brain tumors are a type of neurodegenerative diseases that are represented by abnormal glutamate signaling and release in the brain. Early detection and treatment of brain tumors are important for the well-being of patients. Magnetic Resonance Imaging (MRI) is a non-invasive neuro-imaging technique that can efficiently capture the structural organization of brain soft tissues. To ease the diagnosis with MRI scans, especially to determine if a brain tumor is low-grade tumors LGG (grades I and II; LGG being a Low-Grade Glioma), which are generally benign, or high-grade tumors HGG (grades III and IV), which are highly malignant, we developed a pipeline that could recognize tumor regions and classify tumor types.

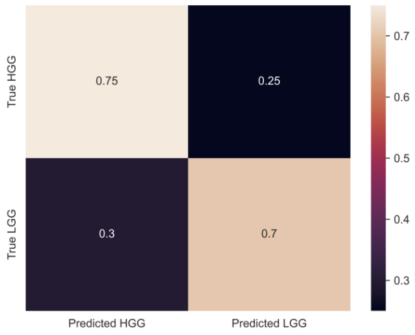
Initial Project Goals & Goals Achieved

The initial goal of the project aims to follow an existing workflow for brain tumor identification and classification using MRI scans [3]. The specific goals are broken down as follows:

- Implement the four key steps: pre-processing, imaging segmentation, feature extraction, and tumor classification.
- Evaluate the result using a variety of evaluation metrics, including accuracy, precision, specificity, sensitivity, F-measure, etc
- Compare the classification part of the algorithm with multiple other dissimilar methods, such as KNN, SVM, and DNN

Due to time and computing resources constraints, we haven't had the opportunities to finish all of them. Specifically, we have finished the following:

- We have implemented the pipeline of the algorithm, including pre-processing, imaging segmentation, feature extraction, and eventually classification
- We evaluated the results of the classification using accuracy, represented by a confusion matrix
- We analyzed our process and found areas for future improvements



Methods

Pre-processing

To obtain a higher accuracy in tumor segmentation and classification, the raw MRI image needs to be denoised. In this project, the non-local means denoising method is applied, and an example of transformation is shown on the right.

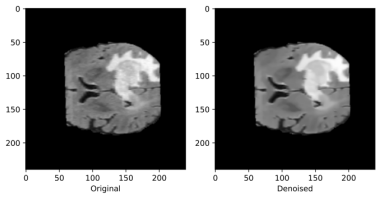
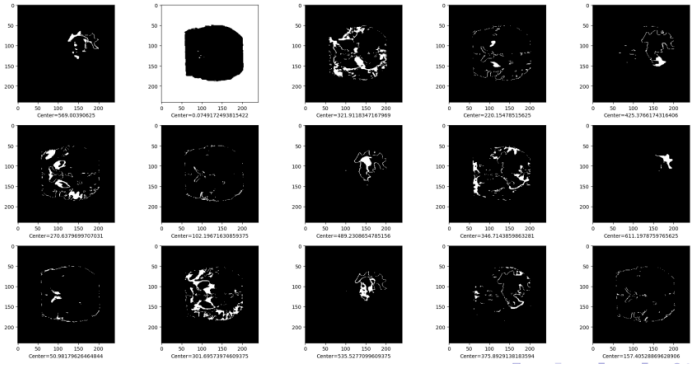


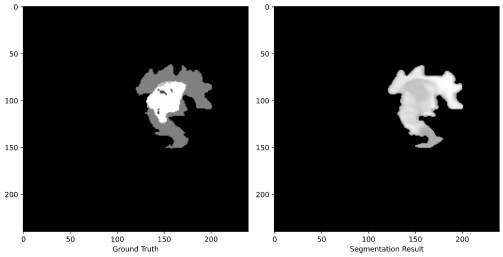
Image Segmentation

The k-means clustering method is used for tumor segmentation, and various post-processing steps are applied to achieve a better performance.

For each MRI slice, the intensity values are treated as the input to the k-means clustering algorithm. The choice of the hyper parameter k is crucial to the success of the algorithm because a small k might not be able to differentiate the tumor regions from regions with similar pixel intensities, whereas a large k might break the completeness of a tumor region. Through empirical exploration on our dataset, we choose k = 15, which generates a relatively accurate result on tumor detection. Each of the 15 clusters has a score, in this project, we merge the clusters with top scores to generate the final representation of the segmented tumor. An example of clustering is shown on the right, and the merged result is shown below along with the ground truth.



Due to the heterogeneity of tumor regions and brain structure, k-means clustering may not accurately segment the whole tumor region out. Several problems with the clustering results include: (1) unsmooth or incomplete tumor boundary; (2) unable to segment the whole region within the tumor boundary; (3) due to similarity in pixel intensity, a few distant brain regions are also assigned with the same cluster as the tumor. To alleviate these problems, we apply post-processing steps, including morphology opening and closing, binary fill, and median filter sequentially to improve the segmentation performance.

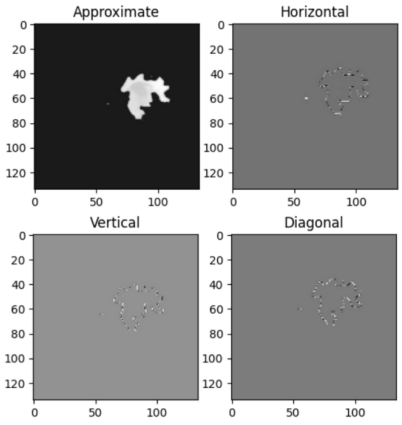


Feature Extraction

Upon finishing the imaging segmentation, we extract three sets of robust features from the tumor regions for each slice using information-theoretic measures, wavelet packet Tsallis entropy (WPTE), and scattering transform (ST).

First, based on the intensity distribution of pixels in the tumor regions, we calculate skewness, kurtosis, entropy, variance, and mean to capture high-level tumor characteristics.

Secondly, we choose the popular Coiflet wavelet and apply the wavelet packet decomposition to tumor regions. One pass through the decomposition algorithm generates four sets of coefficients, representing the low-level signals and the high-level signals consisting of the horizontal, vertical, and diagonal components. An example one-level decomposition is shown on the right. By repeatedly passing the image through high-pass filters and low-pass filters for three times, we obtain 64 sets of wavelet approximation coefficients. The PyWavelets package is used for the wavelet packet decomposition, and we calculate the Tsallis entropy of each set of coefficients to obtain a 64-dim feature vector [1].



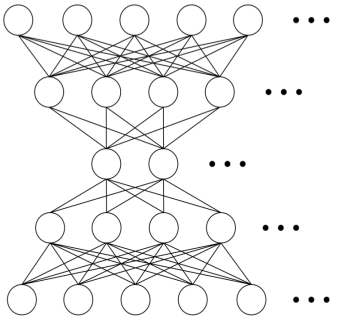
Thirdly, scattering transform is applied to the MRI images to extract representative low-level features. Scattering transform is a complex-valued convolutional neural network with multiple layers of wavelet transforms. The nonlinear signal representation obtained from it is invariant of translation,

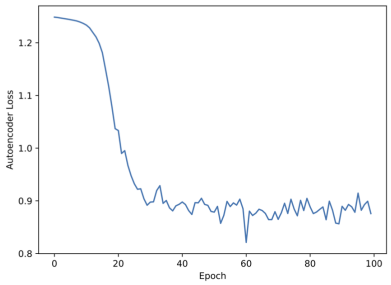
rotation, scaling, and frequency shifting, which is highly suitable for the feature extraction task of brain tumors. The Kymatio package is used to do the scattering transform, which results in a 1401-dim feature vector [2].

Finally, the features extracted from the three parts are combined together to create a design matrix that will be fed into the autoencoder for tumor classification.

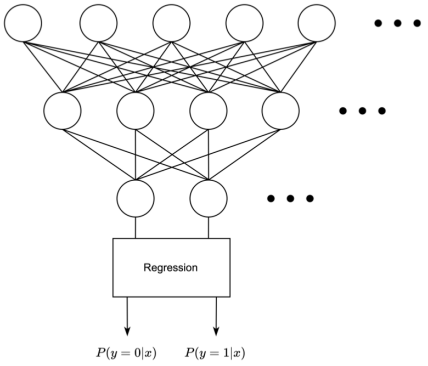
Classification

For the actual classification part, we first train a deep autoencoder (DAE) to learn the compact representations of the features we have extracted. We feed the features into the encoder and then decoder and compare the results with the feature inputs. To measure the difference between feature inputs and reconstructed output, we used MSE loss in training. After several iterations, the loss has converged.





We then feed the features into the encoder and classify the output with a Logistic Regression algorithm to determine if the tumor is LGG or HGG.



Meanwhile, we use Jaya optimization to optimize this regression. Using Jaya optimization rather than the standard Adam optimizer offers a multitude of benefits, the most obvious being speed and computational efficiency. Since Jaya is not a gradient descent algorithm, there is no need to calculate the partial derivative of the loss function, eliminating the vast majority of the computational time that Adam (or other popular optimizers) utilize. Another advantage is that Jaya optimization does not require the use of hyperparameters. This eliminates the need for hyperparameter experimentation and also reduces overall memory used in the entire network; if we had more time, we could've taken advantage of this and used various threading and/or parallel processing methodologies to reduce computation time by large factors.

An under the radar benefit of Jaya optimization is that (if an optimal solution exists) Jaya optimization can find the optimal solution, it being the global solution, regardless of initial approximation. This is due to the fact that the lack of partial derivatives means that Jaya uses random (but direction-oriented and controlled) movements to find the global optimal solution by recording the best and worst solutions to shift in the appropriate direction. This means that there is no "wrongful" solution that Jaya can provide; a horrible initial approximation coupled with an un-ideal step size will just take longer to complete optimization.

Known Limitations and Future Directions

This project could definitely benefit from more improvements. A list of possible future directions that we have realized are as follows:

- In the current project, we only ran the algorithm on one slice per subject. If more time/computing resources were available, we could run the algorithm on all 260 slices per subject to improve classification performance and gain a more subjective view about strengths and weaknesses of the model.
    - Optimize the efficiency of our model using techniques including (but not limited to): threading, parallel processing, mathematical optimization, c-ython, and better use of resource and computation allocation
- Experiment with other auto-encoder structures
- Run PCA analysis on the features we got from feature extractions to determine dependencies
- Constructing a 3D model of the brain from the 260 slices allowing for (a potential) ICA analysis
- Try other classification algorithms other than logistic regression, such as reinforcement learning
- Identify different regions in a tumor
- Create a novel methodology (besides bias weighting) to solve dataset imbalances
- Construct a predictive model that can not only identify, but also predict from the early states (i.e. before the tumors is so large as it requires a surgical procedure for removal) whether LGG/HGG will develop

Key Takeaways and Advice for Future Students/Instructors

Some key takeaways we have learned in this project are:

- Brain tumor detection and segmentation is a complex problem that requires the usage of multiple methods to achieve a reasonable good performance
- Sometimes over complicated methods result in worse outcomes
- During pre/post-processing it is possible to over-process the image, resulting in loss of valuable information
- Low accuracy rates should not be a cause to lose motivation, as sometimes just tinkering with hyperparameters can lead to a huge jump in accuracy

For students who would be interested in taking this class or future instructors:

- Definitely start early or rent out a GPU as training without a GPU could take virtually forever
- Don’t be restrained by a state-of-art implementation, always feel free to experiment with any idea that you have
- Understand the mechanism behind popular packages (especially in image classification) as then one can tailor by customizing (custom writing) adjustments to a specific dataset and methodology
- Don’t be overwhelmed by the plentitude of possible ways to process image, as each has their own advantages and disadvantages; choose the one that is most suitable, which isn’t always the most popular method

References

[1] Lee, G., Gommers, R., Waselewski, F., Wohlfahrt, K., & O'Leary, A. (2019). PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software, 4*(36), 1237.

[2] Andreux, M., Angles, T., Exarchakis, G., Leonarduzzi, R., Rochette, G., Thiry, L., ... & Eickenberg, M. (2020). Kymatio: Scattering Transforms in Python. *J. Mach. Learn. Res., 21*(60), 1-6.

[3] Raja, P. S. (2020). Brain tumor classification using a hybrid deep autoencoder with Bayesian fuzzy clustering-based segmentation approach. *Biocybernetics and Biomedical Engineering*, *40*(1), 440-453.