Brain Tumor Detection Rubric

DS 4002 – Fall 2024 - Instructor: Ashley Whitehouse
Due: TBD
Submission format: Upload a link to your GitHub repository on Canvas

Individual Assignment

General Description: Submit a link to your GitHub repository containing your completed case study, which demonstrates your ability to use convolutional neural networks to classify MRI brain scan images.

Preparatory Assignments – Everything you have learned thus far as a 2nd year data science student.

Why am I doing this? This is your opportunity to apply what you've learned to a real-world scenario that you may encounter in future classes or in your career. This will also be a great learning opportunity for you to develop your analytical and technical skills and to gain experience working with convolutional neural networks (CNNs).

What am I going to do? You will obtain the data for this project from the GitHub repository, which can be found here: https://github.com/awhitehouse1/DS4002-CS3. You will then build a CNN of your choosing using the provided dataset to classify MRI brain scans as "tumor" or "no tumor". You will also document your methodology and results in a PDF. Finally, you will create a README containing the GitHub file structure, instructions for running your code, and your references.

Tips for success:

- Experiment with hyperparameters, such as the number of epochs, batch size, number of filters, etc. to increase the accuracy of your model.
- If your accuracy is low even after trying to optimize your model, don't worry! You can reflect on reasons the accuracy of your model might be low, such as overfitting/underfitting or model architecture, in your PDF report.

How will I know I have Succeeded? You will meet expectations on this brain tumor detection case study when you follow the criteria in the rubric below.

Spec Category	Spec Details
Formatting	● GitHub Repository: Make sure it includes the following ○ The MRI images dataset provided to you in the GitHub link ○ Your code source file ○ PDF report with your methodology and key findings ○ A README.md with project structure, instructions for replication, and references
Source Code File	● Goal: Write well commented code that classifies MRI brain scan images as tumor or no tumor

	● Write your code in Python using Google Colab or a Jupyter Notebook. ● Turn in as a Python script or notebook (.py or. ipynb file) ● Include preprocessing, such as data augmentation and feature scaling ● Build a CNN using TensorFlow/Keras, using an architecture of your choice. You will explain the modeling choices you make in the PDF report. ○ Some CNN architectures you may consider are ResNet50, VGG19, MobileNet, etc. ● Include comments that explain how your code works. Make sure to describe any preprocessing steps and hyperparameters. ● Create and include visualizations showing the accuracy and loss of your training and validation sets over multiple epochs. You will interpret these graphs in your PDF report.

PDF Report	● Goal: Explain your methodology and present the results (PDF format)
	● Methodology section:
	  ○ Explain the following:
	  ▪ Preprocessing steps (data augmentation, feature scaling, etc.)
	  ▪ Train-test split
	  ▪ Why you chose your CNN model
	  ▪ Any model optimizations you applied
	  ▪ Any callbacks or checkpoints you used
	● Results section:
	  ○ Explain the key findings
	  ○ Interpret your model's performance using accuracy and loss. If your model's performance is low (below 60%), reflect on possible causes for this and how it could be improved in the future.
	  ○ Include and discuss visualizations from your Python source file

README.md	● Goal: Orient people to your repository and give credit to your references ● Use markdown headers to divide content ● Section 1: Project Structure ○ Provide an outline of the files included in your repository. You should briefly explain the contents of each file.

	● Section 2: Instructions for replicating your results ○ Give explicit step-by-step instructions for reproducing the results of the case study. Explain where to find the data, which code should be run, and how to run it. ● Section 3: References ○ All references should be listed ○ Use IEEE Documentation style

Acknowledgements: Special thanks to Jess Taggart from UVA CTE for coaching on making this rubric. This structure is pulled from Streifer & Palmer (2020).