Image Processing Brain Tumors

Alex Lai

- **What is the problem you want to solve?**
  - To make an image-processing brain tumor predictive model to automate on scale.
  - The accuracy has to be above 95% to be considered successful. This will be completed within the next 3 months.
  - Stakeholders would be hospitals and doctors.
  - The solution space will most likely be creating the model using Python, Pandas, and CNN.
  - Constraints would be the limit amount of image data to train and the quality of the image data.
- **Who is your client and why do they care about this problem? In other words, what will your client do or decide based on your analysis?**
  - My client would be hospitals wanting to
    - scan through their existing history of MRI brain scans, but the treatment
    - Or quickly scan through a new patient's brain MRI scan.
  - Yes a human can do it but if you have 10,000 images, then you would want to automate this tedious process and save manpower, time, and money.
- **What data are you using? How will you acquire the data?**
  - I will be using this Dataset from Kaggle.
    - https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset
  - which was provided by the UCI Machine Learning Repository.
    - https://ieee-dataport.org/documents/brain-tumor-mri-dataset
  - This data under the license CC BY 4.0 ATTRIBUTION 4.0 INTERNATIONAL Deed. I would need to give the appropriate credit. The data is free and ethical to use.
    - https://creativecommons.org/licenses/by/4.0/

- **Briefly outline how you'll solve this problem. Your approach may change later, but this is a good first step to get you thinking about a method and solution.**
    - **Data Wrangling**:
        - Preprocess images (resize, normalize, augment) and split into training, validation, and testing sets.
    - **EDA:**
        - Count the number of images, classes, and check image dimensions, file sizes, class imbalance.
    - **Feature Engineering:**
        - Then standardize the data to make it compatible to be inputted into the predictive model.
        - Will split the data set into a training dataset (80% of the total) and a testing dataset (20% of the total). I will design and train the model til I get the desired accuracy.
    - **Modeing**:
        - Use CNNs for classification (e.g., ResNet), U-Net for segmentation, or YOLO/Faster R-CNN for detection.
        - Train with appropriate loss functions, optimizers, and learning rate schedules; monitor metrics to avoid overfitting.
        - Assess performance using metrics like accuracy, F1-score, Dice coefficient, IoU, or mAP based on the task.
        - Fine-tune hyperparameters, use transfer learning, and apply regularization techniques for improved results.
    - **Documentation**:
        - Present results with confusion matrices, ROC curves, segmented overlays, or detection bounding boxes.
        - Summarize methodology, results, and insights with suggestions for future improvements.
- **What are your deliverables?**
    - I will deliver a GitHub link to my code, slide deck, and project report.