### Assignment Document: Object Detection Task Using LGG Dataset

#### **Background**

Object detection is a fundamental area in computer vision, where the goal is to locate and identify objects in an image. In this assignment, students will leverage the **LGG Segmentation Dataset**, which contains medical imaging data of low-grade gliomas (LGG). This task will enable students to bridge the gap between segmentation and detection by designing bounding box annotations from segmentation masks and training an object detection model.

#### **Objectives**

Understand the basic principles of object detection, including bounding box generation, feature extraction, and classification.

Learn preprocessing techniques for medical imaging data and adapt segmentation masks for detection purposes.

Gain hands-on experience with popular object detection frameworks, such as YOLO, Faster R-CNN, or SSD.

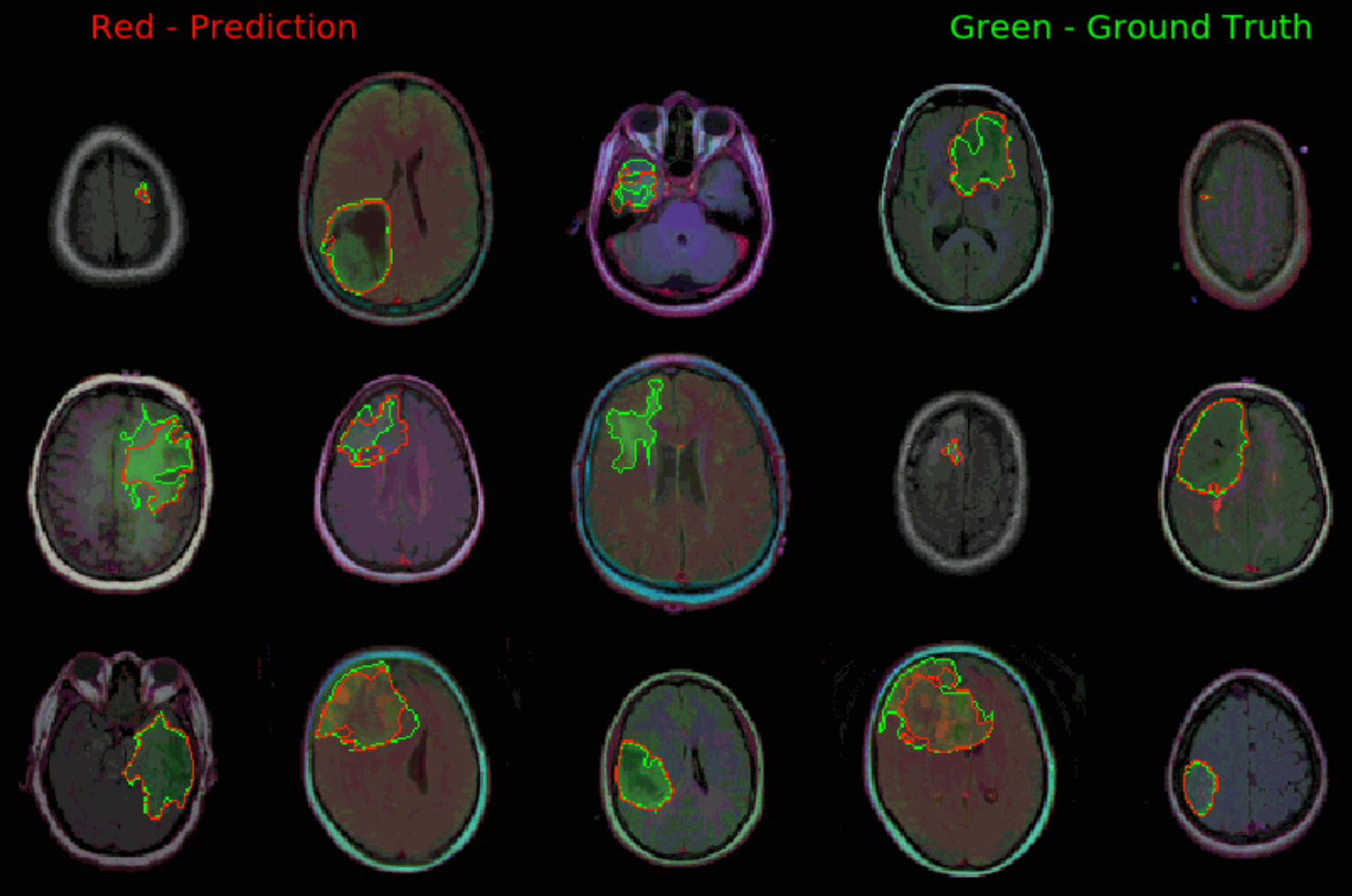
Develop critical thinking skills in handling real-world challenges, such as imbalanced data and domain-specific optimization.

#### **Dataset Overview**

Name: LGG Segmentation Dataset

Content: MRI scans and segmentation masks highlighting LGG regions.

This dataset contains brain MR images together with manual FLAIR abnormality segmentation masks.  
The images were obtained from The Cancer Imaging Archive (TCIA).  
They correspond to 110 patients included in The Cancer Genome Atlas (TCGA) lower-grade glioma collection with at least fluid-attenuated inversion recovery (FLAIR) sequence and genomic cluster data available.  
Tumor genomic clusters and patient data is provided in data.csv file.  
For more information on genomic data, refer to the publication "Comprehensive, Integrative Genomic Analysis of Diffuse Lower-Grade Gliomas" and supplementary material available at [https://www.nejm.org/doi/full/10.1056/NEJMoa1402121](https://www.nejm.org/doi/full/10.1056/NEJMoa1402121" \t "https://www.kaggle.com/datasets/mateuszbuda/_blank)



#### **Assignment Instructions**

**1. Data Preparation**

Slice Extraction:

Extract 2D slices from the 3D MRI scans.

Save each slice as a separate file in an image-friendly format (e.g., PNG).

Bounding Box Generation:

Use segmentation masks to calculate bounding boxes for the regions of interest (ROIs).

Save annotations in formats such as COCO (JSON) or Pascal VOC (XML).

Ensure each annotation contains:

The bounding box coordinates (x\_min, y\_min, x\_max, y\_max)

The class label (e.g., "LGG").

**2. Task Requirements**

**Model Selection**

Choose one object detection framework to complete the task.

Options include:

YOLOv5: Focuses on real-time, high-speed detection.

Faster R-CNN: Known for high accuracy and detailed feature extraction.

SSD (Single Shot MultiBox Detector): Balances speed and precision.

**Model Implementation**

**Training:**

Split the dataset into training (70%), validation (20%), and testing (10%).

Apply data augmentation techniques, such as rotation, flipping, and brightness adjustment, to improve model generalizability.

Train the chosen model using pre-trained weights as initialization (e.g., COCO weights).

**Validation:**

Monitor key metrics such as mean Average Precision (mAP), recall, and precision.

Adjust hyperparameters (e.g., learning rate, batch size) to improve model performance.

**Testing:**

Evaluate the model on the test set and provide both quantitative metrics and qualitative outputs (detected bounding boxes overlaid on test images).

#### **Submission**

* Submit the Python code(.ipynb) with proper comments and structure.
* Submit the trained weights of your model.
* Ensure the code is formatted and structured logically.
* Submit a experiment report for your test results and experiment analysis.