Ghulam Ishaq Khan Institute of Engineering Sciences and Technology.

Name: Ahmad Azzam

Reg No.:2023064

Subject: DSA

Faculty: FES

Tumor Detection in Brain MRI Images Using Image Segmentation and Flood Fill Algorithm (DFS)

1. Algorithm Integration

For this project, the primary algorithm implemented is **Flood Fill using Depth-First Search (DFS)**, designed to detect and isolate tumor regions in grayscale MRI brain images. The steps include:

- **Preprocessing**: Images are first converted to grayscale and thresholded to enhance contrast between normal and abnormal tissue.
- **Flood Fill with DFS**: The algorithm begins at a seed pixel, checking connected pixels (typically 4- or 8-directionally) for similarity in intensity values. DFS is applied using an explicit stack data structure (`std::stack`) rather than recursion, to give better control and prevent stack overflow on large regions.
- **Region Marking**: Tumor-like areas are colored or highlighted for visualization.

This approach allows precise traversal and segmentation of irregular regions, which is common in tumor boundaries.

2. Optimization & Efficiency Analysis

To analyze performance and optimize, the following comparisons and observations were made:

- **DFS vs BFS**: DFS was chosen for its lower memory footprint in deep but narrow regions, which is common in organic, tumor-like patterns. BFS, while effective, can become memory-intensive as it stores a wider frontier at each level.
- **Stack Implementation**: Using a manual stack avoids system recursion limits, improving robustness and runtime performance in large images.

- **Threshold Tuning**: Adjustable intensity threshold values were tested to balance under-segmentation (missing parts of the tumor) and over-segmentation (including non-tumor tissue).
- **Time Complexity**: Approx. O(n), where n is the number of pixels in the segmented region.
- **Space Complexity**: O(n) for visited pixels and the explicit stack.

Further experiments will include performance profiling on MRI images of different sizes and resolutions.

3. Edge Case Handling

Multiple edge cases have been tested or planned for testing:

- **Multiple Tumors**: The algorithm can detect multiple disconnected regions by applying DFS from multiple seed points.
- **Tumor at the Image Border**: Ensures the algorithm correctly detects regions even if they extend to image edges.
- **No Tumor Present**: Handled by setting a minimum region size; false positives are discarded.
- **Highly Noisy Images**: Noise filtering (e.g., Gaussian blur) is optionally applied before segmentation to improve accuracy.

Further enhancements are planned to automate seed point selection using histogram analysis or contours.

4. Implementation of Course Concepts

This project is strongly rooted in the **Data Structures and Algorithms (DSA)** concepts discussed in class:

- **Stacks**: Manual implementation of DFS using the stack data structure for iterative traversal.
- **Vectors**: Use of `std::vector` for dynamic storage of connected components and pixel positions.

- **Graph Traversal (DFS)**: The MRI image is treated like a 2D grid graph where each pixel is a node and adjacency is defined by pixel connectivity.
- **Algorithm Design**: Custom thresholding and control flow to adapt DFS to a flood fill scenario.

The project directly applies classroom knowledge of graph traversal, memory management, and complexity analysis in a real-world application.

Current Status and Next Steps

- **Completed**: Basic flood fill implementation using DFS, test cases for various input images, manual seed point detection.
- **In Progress**: Automatic seed selection, refining thresholding, benchmarking with other segmentation techniques (e.g., region growing, contour-based).
- **Next**: Implement GUI-based interaction using OpenCV, finalize performance comparison charts, write final report and documentation.