Image Edge Detection and Orientation Assignment

MM:100 points

Objective:

The objective of this assignment is to challenge your skills in image processing and edge detection by implementing Various algorithms and conducting in-depth analysis. You will work on loading and processing natural images, applying edge detection filters with various orientations, implementing Winner-Takes-All (WTA) and normalization operations, and conducting a comprehensive comparative analysis.

Note: Please refer to the appendix of this assignment to get an idea about which techniques to use. You are free to use other advanced techniques.

Part 1: Image Preprocessing (10 points)

- ➔ Create a Google Drive folder to upload and upload diverse natural images (minimum 10). Ensure the images encompass a wide range of complexities, perspectives, and lighting conditions. For reference you may have a look at these images. The image must be RGB having a minimum resolution of 400x400.
- ➔ Load the images using appropriate libraries and display them with a brief description of their content.
- ➔ Convert each image to grayscale using techniques that ensure optimal contrast enhancement and dynamic range preservation.

Part 2: Complex Orientation Filters (15 points)

- ➔ Select one high-resolution grayscale image from Part 1 for further processing.
- ➔ Implement complex orientation filters based on Gabor wavelets. Create multiple filters (at least 4, at 0, 45, 90 and 135 degrees respectively) with varying orientations and frequencies.
- ➔ Apply each filter to the selected grayscale image using convolution operations.
- ➔ Visualize the filtered images for each orientation, emphasizing the extracted features.

Part 3: Winner-Takes-All and Normalization (20 points)

- ➔ Implement a Winner-Takes-All (WTA) algorithm that considers both magnitude and orientation of the complex filtered images.
- ➔ Normalize the WTA output images to enhance feature visibility while preserving contextual relationships.

- ➔ Visualize the WTA and normalized images, capturing intricate features and texture patterns.

Part 4: Comparative Analysis (15 points)

- ➔ Apply the entire pipeline (image preprocessing, complex orientation filtering, WTA, and normalization) to all loaded images.
- ➔ Compare and analyze the results across different images, focusing on the impact of complexity, texture, and lighting on edge detection.
- ➔ Apply different kinds of noise eg gaussian noise, motion blur etc on the image and then perform edge detection.
- ➔ Do a comparison between the original edge detection and noise influenced edge detection using metrics such as structural similarity index (SSIM) and edge F1-score.
- ➔ Discuss the pros and cons of Gabor-based edge detection in challenging scenarios.

Part 5: Visualization (15 points)

- ➔ Implement visualization techniques to highlight detected edges and features on the original images.
- ➔ Generate gradient magnitude and orientation maps for visualizing the underlying features.
- ➔ Compare and discuss the effectiveness of these visualization methods.

Part 7: Report and Documentation (25 points)

- ➔ Write a comprehensive report (PDF) containing:
- ➔ Introduction detailing the assignment objectives and challenges.
- ➔ Detailed explanations of implemented techniques in each part.
- ➔ Visualization of intermediate and final results for multiple images.
- ➔ In-depth analysis of comparative results.
- ➔ Discussion on the significance and limitations of the Gabor-based edge detection approach.
- ➔ Conclusion summarizing your findings and lessons learned.
- ➔ You are free to use colab notebook for writing and submitting your code. However, this colab notebook will not be accepted as a report.
- ➔ Submission Guidelines:
    - ◆ Submit your complete assignment (code, images, report) by **20th September 2023 EOD**.
    - ◆ The deadline is strict and will not be extended in any case.
    - ◆ Late penalty: 5% per day. After 3 days, the submission will not be accepted.
    - ◆ If code is done on colab, make sure to share the colab file (link) with me before the deadline. Make sure to provide access rights.
    - ◆ In case your file is corrupted or inaccessible, your assignment will not be evaluated.

- All python and notebook files inside the zipped folder must have your roll number as a prefix. Eg: There is a file ans.py, it should be renamed to <your_roll_number>_ans.py
- Supported language: Python (if jupyter notebook, it must be on colab)

Appendix:

Part 1: Image Preprocessing
Techniques for grayscale conversion might involve:

Adaptive Histogram Equalization (AHE) or Contrast Limited Adaptive Histogram Equalization (CLAHE) for dynamic range enhancement while preserving local details.
Applying gamma correction to adjust pixel intensities and improve contrast.

Part 2: Orientation Filters (Gabor-Based Filters)

Apply gaussian smoothing.
Gabor filters are advanced orientation filters that capture both frequency and orientation characteristics. These filters are created using Gabor wavelets, which are complex sinusoidal functions modulated by Gaussian distributions. By varying the parameters of the wavelets, you can extract features at different orientations and frequencies.

Part 3: Winner-Takes-All and Normalization

Techniques for WTA and normalization might include:

Considering both magnitude and orientation information for WTA, allowing for more robust edge extraction.
Using advanced normalization methods like Adaptive Contrast Normalization (ACN) or Perceptual Contrast Enhancement (PCE) to enhance edges while suppressing noise.

Part 4: Comparative Analysis

Metrics for comparative analysis could include:

Structural Similarity Index (SSIM): A metric that quantifies the structural similarity between two images, considering luminance, contrast, and structure.
Edge F1-score: A metric that evaluates the precision and recall of edge detection results, considering true positive, false positive, and false negative edges.

Visualization methods could include:

Overlaying edge-detected results on original images with varying transparency levels to highlight detected features.

Generating gradient magnitude and orientation maps to visualize the strength and direction of edges.