AI 2024
Week 3 Task

Color Image Quantization

0.1 Objective

The goal of this task is to perform color image quantization using the K-means clustering algorithm. Specifically, you will reduce the number of colors in two given test images to only 16 distinct colors. This task will help you understand how clustering algorithms can be implemented from scratch and applied in image processing for efficient compression while maintaining visual quality.

0.2 Task Description

You are provided with two color images, `lena.png` and `peppers.tif`, located in the `images` directory. Your task is to reduce the number of colors in each image to exactly 16 distinct colors using the K-means clustering algorithm. The key requirements for this task are:

- **Implementation:** You must implement the entire K-means algorithm from scratch. Use of machine learning libraries such as `scikit-learn` is not allowed. However, you may use other libraries like `numpy`, `pandas`, `matplotlib`, and `opencv` for handling arrays, data manipulation, and visualization.
- **Dynamic Code Design:** The code should be designed to allow easy modification of key parameters such as the number of clusters K and the input image. These parameters should be easily adjustable without requiring significant changes to the code structure.
- **Logging and Visualization:** During the execution of the K-means algorithm, you must log the L2 norm at each iteration in a file (e.g., `L2_norm_log.txt`). Additionally, you are encouraged to visualize the quantized image after each iteration. This visualization should be controlled by a flag in your code, allowing it to be enabled or disabled as needed.

0.3 Steps to Follow

1. **Data Preparation:**
   - Load each image from the `images` directory.
   - Extract the RGB color values of all pixels in the image. Each pixel should be treated as a data point in a 3-dimensional space (with dimensions corresponding to Red, Green, and Blue components).
2. **K-means Clustering:**
   - Implement the K-means algorithm from scratch to group the pixels into 16 clusters. The algorithm involves the following steps:
     (a) **Initialization:** Randomly select 16 initial centroids from the pixel data.
     (b) **Assignment Step:** Assign each pixel to the nearest centroid based on the Euclidean distance (L2 norm).
     (c) **Update Step:** Recompute the centroids by taking the mean of all pixels assigned to each cluster.
     (d) Repeat until a termination condition is satisfied, such as the absence of further improvement or the reaching of a specified iteration limit.
     (e) Log the L2 norm after each iteration in a file (e.g., `L2_norm_log.txt`). The L2 norm (Euclidean distance) between two pixels with RGB values (R_1, G_1, B_1) and (R_2, G_2, B_2) is defined as:

$$\text{L2\_norm} = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \tag{1}$$

3. **Image Reconstruction:**
    - Replace the color of each pixel in the original image with the color of the centroid of the cluster to which it has been assigned.
    - The resulting image will have only 16 unique colors.
    - If the visualization flag is enabled, display the quantized image after each iteration.
4. **Evaluation:** Calculate the total distance between the original image and the quantized image using the L2 norm. The total L2 distance for the entire image is the sum of the L2 norms for all corresponding pixels in the original and quantized images.

0.4 Improvements and Extensions

While the basic K-means algorithm is effective, it has certain limitations, such as sensitivity to initial centroids and the assumption of spherical clusters. You are encouraged to explore and implement any improvements to the standard K-means algorithm.

0.5 Ranking and Evaluation Criteria

Your submission will be evaluated based on the following criteria:

- **Total L2 Norm:** The total L2 norm between the original and quantized images for both `lena.png` and `peppers.tif`, as well as an additional test image.
- **Execution Time:** The time taken to run the algorithm on the additional test image.

Submissions will be ranked based on a combination of these metrics, with additional credit given for innovative improvements to the basic K-means algorithm.

0.6 Submission Guidelines

Please submit the following:

- The code used to perform the color quantization. Remember that the entire K-means algorithm must be implemented by you from scratch, without using libraries like `scikit-learn`.
- The quantized images (`lena_quantized.png` and `peppers_quantized.png`).
- A log file (e.g., `L2_norm_log.txt`) containing the L2 norm value after each iteration.
- A brief report explaining your approach, any improvements made to the standard K-means algorithm, and the final distance values for both images, as well as the execution time for the additional test image.

Ensure that your code is well-documented, easy to execute, and allows for easy modification of key parameters like K, input image, and the visualization flag.