AI TechStack 2025
Week 3 Task

August 17, 2025

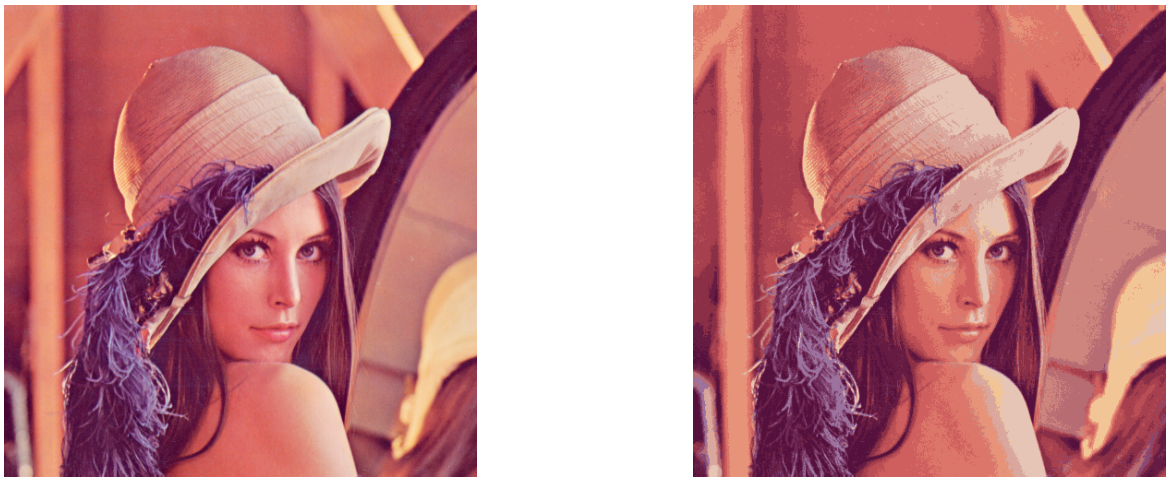
1 Color Image Quantization

Objective

Reduce the number of colors in two test images to 16 distinct colors using the K-means clustering algorithm, implemented from scratch, to explore clustering in image processing for efficient compression with minimal visual quality loss.

Task Description

You are provided with `lena.png` and `peppers.tif` in the `images` directory. Reduce each image's colors to 16 using K-means, implemented without machine learning libraries like `scikit-learn`. Use `numpy`, `pandas`, `matplotlib`, or `opencv` for array handling and visualization. Ensure code allows easy modification of parameters like K and input image. Log the L2 norm per iteration in `L2_norm_log.txt` and optionally visualize quantized images, controlled by a flag.



(a) Original Image (`lena.png`) (b) Quantized Image (`lena_quantized.png`)

Figure 1: Example of original and quantized images for the Color Image Quantization task.

Steps to Follow

1. **Data Preparation:** Load images and extract RGB pixel values as 3D data points.
2. **K-means Clustering:**
   - Randomly initialize 16 centroids.
   - Assign pixels to nearest centroids using L2 norm: $\sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}$.
   - Update centroids as the mean of assigned pixels.

- Repeat until convergence or a set iteration limit.
- Log L2 norm per iteration.

3. **Image Reconstruction:** Replace pixel colors with their cluster centroid colors. Optionally visualize results if flag enabled.
4. **Evaluation:** Compute total L2 norm between original and quantized images.

Improvements

Consider enhancements like better centroid initialization or alternative clustering methods to address K-means limitations (e.g., sensitivity to initial centroids).

Evaluation and Submission

Submissions are ranked by total L2 norm and execution time on an additional test image, with bonus points for innovative improvements. Submit:

- K-means code (implemented from scratch).
- Quantized images (`lena_quantized.png`, `peppers_quantized.png`).
- `L2_norm_log.txt` with per-iteration L2 norms.
- Brief report on approach, improvements, L2 norms, and execution time.

Ensure code is documented and allows easy parameter changes.

2 Content-Aware Image Retargeting

Objective

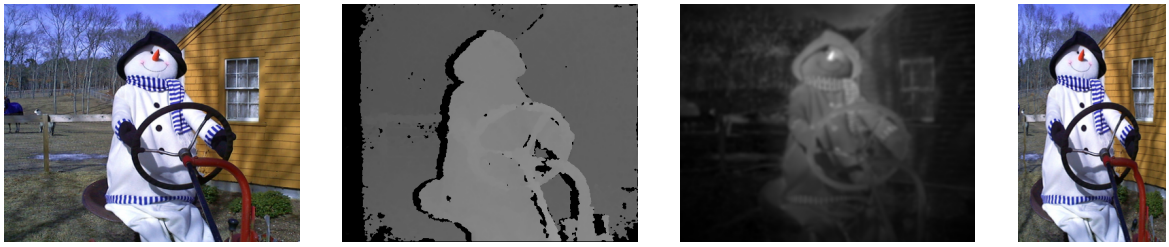
Reduce the width of three test images by a specified number of columns using the Seam Carving algorithm, preserving key content with minimal distortion, to learn dynamic programming and energy-based resizing techniques.

Task Description

You are provided with three datasets (`Baby`, `Diana`, `Snowman`) in the `images` directory, each containing:

- `Baby.png`, `Diana.png`, or `Snowman.png`: Original image.
- `*_DMap.png`: Optional depth map.
- `*_SMap.png`: Optional saliency map.

Reduce each image's width by `seams_number` columns (e.g. 200) using Seam Carving, implemented from scratch without libraries like `opencv`'s seam-carving functions. Use `numpy`, `matplotlib`, or `opencv` for basic processing and visualization. Optionally use depth or saliency maps or alternative energy functions. Ensure code supports adjustable `seams_number` and energy function. Visualize seams if a flag is enabled.



(a) Original (b) Depth Map (c) Saliency Map (d) Retargeted

Figure 2: Example images from the `Snowman` dataset and the retargeted result for the Seam Carving task.

Steps to Follow

1. **Data Preparation:** Load the original image and optionally depth/saliency maps.
2. **Energy Map:** Compute using gradient-based methods (e.g., Sobel filters), depth/saliency maps, or other approaches.
3. **Seam Carving:**
   - Find the lowest-energy vertical seam using dynamic programming.
   - Remove the seam, updating the image.
   - Repeat for `seams_number` iterations.
   - Visualize seams if flag enabled.
4. **Image Reconstruction:** Output the resized image with reduced width.

Improvements

Explore enhancements like advanced energy functions (e.g., saliency-based), forward energy criteria, or simultaneous width/height resizing.

Submission Guidelines

Submit:

- Seam Carving code (implemented from scratch).
- Resized images (`Baby_resized.png`, `Diana_resized.png`, `Snowman_resized.png`).
- Brief report on approach, energy function used, and visual quality observations.

Ensure code is documented and supports adjustable parameters.