In the name of Allah

Artificial Intelligence (AI) - Final Project #02

Color Quantization Using K-Means Clustering

Introduction

Color quantization is a process that reduces the number of distinct colors used in an image while maintaining the visual quality as much as possible. This project implements a color quantization technique using the K-means clustering algorithm.

Methodology

Step 1: Image Preprocessing

1. **Load Image**: An image is read using OpenCV library of Python.
2. **Convert Color Space**: The image is converted from BGR (Blue-Green-Red) to RGB (Red-Green-Blue) color space.
3. **Flatten Image**: The 3D image array is reshaped into a 2D array where each pixel is represented by its RGB values.

Step 2: K-Means Clustering

The K-means algorithm partitions the pixel data into k clusters, where each cluster represents a distinct color in the quantized image.

1. **Initialize Centroids**: Randomly select k pixels as initial cluster centers (centroids).
2. **Assign Pixels to Nearest Centroid**: Calculate the Euclidean distance between each pixel and each centroid, and assign each pixel to the nearest centroid.
3. **Update Centroids**: Calculate the mean of the pixels in each cluster and update the centroids to these means.
4. **Check for Convergence**: Repeat steps 2 and 3 until the centroids no longer change or a maximum number of iterations is reached.

Step 3: Image Quantization

1. **Replace Pixels**: Each pixel in the original image is replaced by the nearest centroid color from the K-means result.

2. **Reshape Image**: The 2D pixel array is reshaped back into the original image dimensions.

Step 4: Evaluation

1. **Compression Ratio**: The compression ratio is calculated as the ratio of the original image size to the quantized image size.
2. **Visual Comparison**: Quantized images for different values of k are compared visually.

Results

Comparison with Different K Values

- **Original Image Size**: 215KB (e.g., for MyForza.png)
- **Quantized Image Sizes and Compression Ratios**:

K values	Quantized Image Size	Compression Ratio
2	26KB	8.27
4	40KB	5.38
8	68KB	3.16
User Input (e.g., 16)	112KB	1.92

Visual Comparison

The above figure shows the original image and the quantized images with different values of k. As k increases, the visual quality of the quantized image improves, but the file size also increases.

Trade-offs between Image Quality and File Size

- **Low K Values**: With smaller k values, the file size is significantly reduced, but the image quality degrades due to fewer colors representing the image.
- **High K Values**: Increasing k values improves the image quality, as more colors are available to represent the image. However, this also increases the file size, reducing the compression ratio.

Challenges Faced

1. **Image Loading**: Initially faced issues with loading images, especially handling incorrect paths or unsupported formats.

2. **Convergence Criteria**: Determining the appropriate criteria for K-means convergence was challenging to ensure both accuracy and efficiency.
3. **Performance**: K-means clustering can be computationally intensive, especially for large images and high k values. Optimization techniques or more efficient algorithms like K-means++ could be considered for future improvements.
4. **Color Representation**: Ensuring that the quantized colors represented the original image's colors as closely as possible while balancing file size and quality.

Conclusion

This project successfully implemented color quantization using K-means clustering. By experimenting with different values of k, we observed the trade-offs between image quality and file size. The methodology demonstrated the effectiveness of K-means clustering in reducing the number of colors in an image while maintaining a balance between compression and visual quality.

Good luck! ☺

Seyed Mahdi Mahdavi Mortazavi

Student number: 40030490

Summer 2024