Shiraz University

Artificial Intelligence - Spring 1403
Machine Learning Project

Deadline: 26th Tir

Overview

Color quantization is the process of reducing the number of distinct colors used in an image while preserving its visual appearance as much as possible. This technique is widely used in image compression, reducing the size of image files, and improving the efficiency of image processing algorithms.
In this project, you will implement a color quantization algorithm using K-Means clustering. **K-Means clustering** is an unsupervised machine learning algorithm that partitions data into a predetermined number of clusters. By applying this algorithm to the colors in an image, you will be able to reduce the image to a limited palette of colors.

Objectives

1. **Understand the K-Means Clustering Algorithm**: Learn how the K-Means algorithm works and how it can be applied to color data.

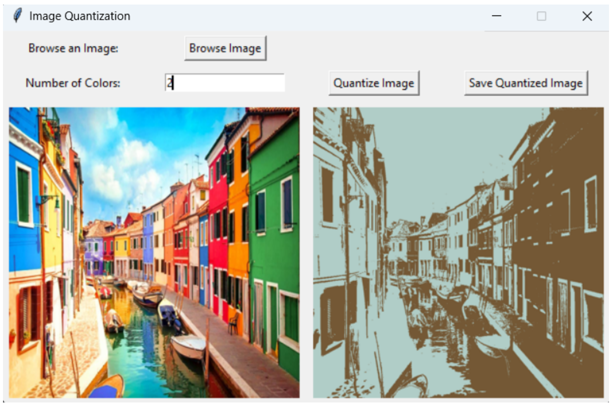
2. **Implement the K-Means Algorithm from Scratch**: Write code to perform K-Means clustering without using any built-in library functions.

3. **Perform Color Quantization**: Apply your K-Means implementation to an image to reduce its color palette.

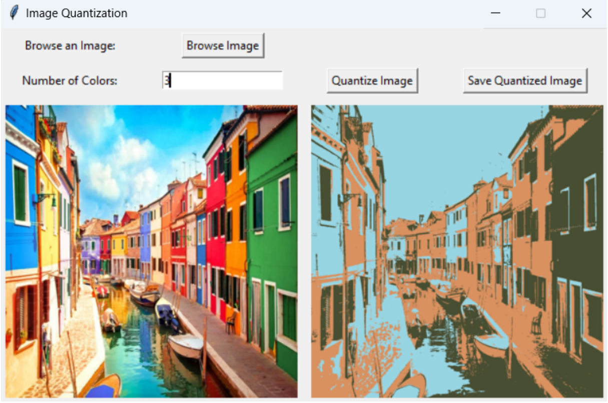
4. **Evaluate and Visualize Results**: Assess the quality of the quantized images and visualize the results.

5. **Improve Image Compression**: Understand how color quantization can reduce the size of image files.

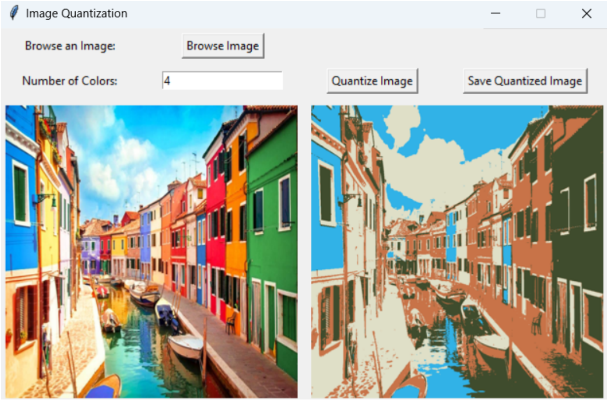
Test Cases



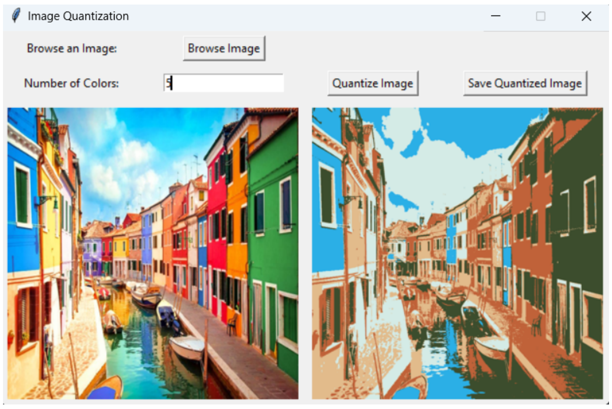
Number of Colors: 2



Number of Colors: 3



Number of Colors: 4



Number of Colors: 5

Requirements

1. **Programming Language**: Python
2. **Libraries**:
   - (a) NumPy for numerical computations
   - (b) OpenCV or PIL for image processing
   - (c) Matplotlib for plotting and visualization

Steps

1. **Setup the Environment**
   - (a) Install the necessary libraries using pip.
2. **Load and Preprocess the Image**
   - (a) Read an image using OpenCV or PIL.
   - (b) Convert the image from RGB (or BGR) to a 2D array of pixels (each pixel being a 3D vector of R, G, B values).
   - (c) Also retrieves the dimensions of the image.
3. **Implement K-Means Clustering**
   - (a) **Initialization**: Randomly initialize the centroids (color clusters).
   - (b) **Assignment**: Assign each pixel to the nearest centroid.
   - (c) **Update**: Update the centroids based on the mean of the assigned pixels.
   - (d) **Iteration**: Repeat the assignment and update steps until convergence or for a fixed number of iterations.
4. **Create the Quantized Image**
   - (a) Replace each pixel in the original image with the nearest centroid color from the K-Means result.
   - (b) Reshape the 2D pixel array back into the original image shape.
5. **Evaluate the Results**
   - (a) Display the original and quantized images side by side for visual comparison.
   - (b) Calculate the compression ratio by comparing the file sizes before and after quantization.
6. **Experiment with Different K Values**
   - (a) Repeat the process with different values of K (e.g., 2, 4, 8, 16, 32) and observe how the image quality and file size change.

Deliverables

1. **Code Implementation**: A Python script or Jupyter notebook with the complete implementation of the K-Means clustering and color quantization process.
2. **Report**: A short report (2-3 pages) documenting:
   (a) The methodology used.
   (b) The results obtained with different K values.
   (c) The trade-offs between image quality and file size.
   (d) Any challenges faced during the implementation.
3. **Visualization**: Visual comparisons of the original and quantized images for different K values. Include plots showing the clustering of colors.

Conclusion

This project will help you understand the practical applications of K-Means clustering in image processing and compression. By implementing the algorithm from scratch, you will gain a deeper understanding of how K-Means works and the challenges involved in clustering data. Experimenting with different values of K will also provide insights into the balance between image quality and compression efficiency. Good luck, and have fun with your implementation!

Good Luck,
AmirHossein Roodaki