

Exercise: K-Means Image Quantization and Color Manipulation

Objective

In this exercise, you will apply K-Means clustering to an image to quantize its colors. You will then perform the following tasks:

1. **Read an image:** Load an image from a file.
2. **Quantize colors:** Reduce the number of colors in the image using K-Means clustering.
3. **Convert to black and white:** Replace the clustered colors with only black and white.
4. **Change primary colors:** Modify the clustered colors to a different set of colors.
5. **Detect contours (if possible):** Highlight edges or contours of objects in the image using simple mathematical techniques.

Instructions

1. **Read an image from a file:**
 - Use a library such as PIL (Pillow) or another alternative to read the image into a numerical format.
2. **Perform color quantization using K-Means:**
 - Convert the image into an array of pixel values.
 - Apply K-Means clustering to group the pixels into k color clusters.
 - Replace each pixel in the image with the centroid of the cluster it belongs to.
3. **Convert the image to black and white:**
 - Define a threshold to map pixels to either black or white based on their brightness.
4. **Detect contours (if possible):**
 - Use simple mathematical techniques like detecting edges by calculating pixel intensity differences.

Deliverables

- A Python script that performs all the above tasks.
- Visual outputs showing:
 - The original image
 - The quantized image
 - The black and white image
 - The contour-detected image (if applicable)

Guidelines

- Use only NumPy and PIL (Pillow). Do not use OpenCV.

- Implement K-Means manually or use an available K-Means function from a library.
- Save and display all output images.