K-Means for Image Color Quantization

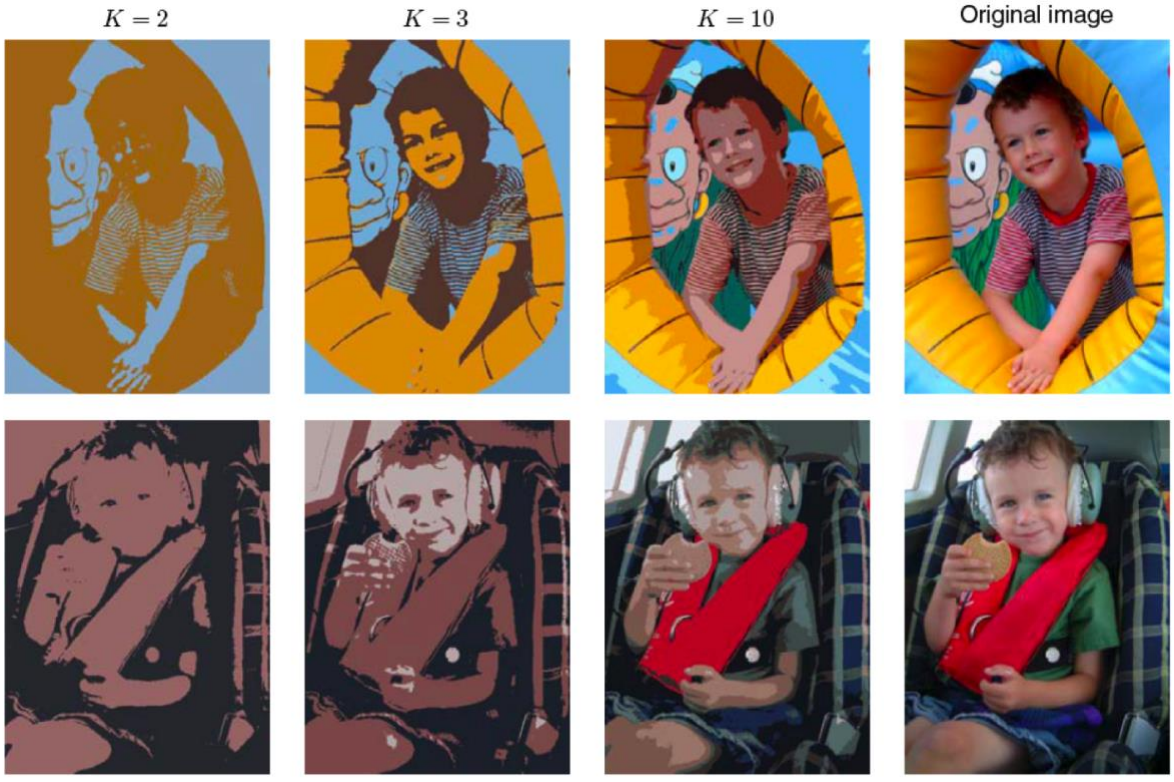
Due Date: 23:59, Oct. 01, 2024
Total Points: 100

Introduction:

In the field of image processing, a common problem is determining how to display a color image on a device that can only display a limited number of colors without sacrificing much image quality. Color images are typically stored as three parallel matrices where each matrix represents the red, green, and blue (i.e., RGB) components of the image. Each component can range from 0 to 255 which means that 256^3 colors can be represented. Since the human eye can't distinguish nearly that many unique colors, it makes sense to choose a limited number of colors to represent a color image.

Using k-means, combinations of colors can be quantized into a certain number of levels. This works well because the human eye can't perceive the full color spectrum. In the context of k-means, these quantized color levels would be the centroids. For each pixel, the closest centroid is determined by treating each pixel as a vector of <r,g,b> and using the distance formula to find the distance between the pixel and each centroid. The algorithm assigns each centroid a color value that represents the average of all the pixels that were closest to that centroid.

The images below (from our lecture) show the result of using k-means to quantize a color image by choosing different values of k (2, 3, 10).



Requirements:

1. In the first step, you need to manually implement the K-Means algorithm (30 points). You also need to implement Image Color Quantization as a function (20 points) using your K-Means algorithm. Note that the empty cluster center should be checked and avoided.
2. Apply your function on the given image (test.jpg) by setting the numbers of clusters to 2, 8, 16, and 32, respectively. Illustrate the original image and images with color quantization. (20 Points)
3. Randomly pick the colors of pixels as centroids (instead the centroids calculated by the K-Means algorithm) and conduct color quantization (10 Points). Show the original image and images with color quantization with K=2, 8, 16, and 32, respectively (10 Points). Compare the image generated using K-Means algorithm and randomly picked centroids (5 Points).
4. Pick an image you like and conduct the Image Color Quantization using K-Means. (Pick any values of K you like.) (5 Points)

Project Information:

You should manually implement the K-Means algorithm. You can only use packages: numpy (for calculation and implementation of K-Means algorithm), matplotlib (for image load and show), sys, and os. Citation is a must if you find any resources helpful. Write everything in a single Jupyter notebook and add comments for your code if needed.

Submission Instructions:

1. Click the Save button at the top of the Jupyter Notebook.
2. Please make sure to have entered your name and ID in the file.
3. Select Cell -> All Output -> Clear. This will clear all the outputs from all cells (but will keep the content of all cells).
4. Select Cell -> Run All. This will run all the cells in order and will take several minutes.
5. Make sure all your solutions are there and displayed correctly.
6. Submit your notebook and the image you picked on Canvas as a *.zip file.
7. Make sure your Canvas submission contains the correct files by downloading it after posting it on Canvas.