Computer Assignment 5

Due Date:
03/23/2018

1 Image color quantization using K-means algorithm

First read the following page to understand how to use `Kmeans()` program in `sklearn` and understand all options.
sklearn.cluster.KMeans
Then read the the example given below, which quantizes all colors in an image into K colors.
Color-quantization Example

Note that the example first use `Kmeans.fit()` to generate the codebook (i.e. all quantized colors or cluster centroids) from colors of a subset of pixels in the image. It then applies this codebook to the entire image to find the cluster label for every pixel using `Kmeans.predict()`. Finally, it reconstructs the color of each pixel from its label based on the codebook (in the function `recreate_image()`). The program also implements an alternative benchmark color quantization approach, which takes the colors of every K pixels as the codebook. It compares the reconstructed image using the codebook designed by the K-means algorithm with this benchmark algorithm.

- Run this program for a selected color image and for two different K values, e.g. K=64 and K=256. Compare the resulting quantized images by the K-means and the benchmark algorithm at the same K, and using the same algorithm at different K.
- The program uses the default method for selecting the initial centroids, which is K-means++, and 10 trials of initializations. Modify it so that the program uses random selection and with 1 trial only. Compare the resulting quantized image with what you obtained in previous part. Also, compare the error reduction curve (error vs. iteration time) with these two different initialization method, when both use only one trial. The error is the quantization mean square error between the original feature vectors and the cluster centroid. In the kmeans function, this is the output denoted as `inertia_`. You can do this for K=64 only.

To modify initialization method, you could set `init ='random'` or `init = 'k- means++'` and set `n_init=1`, in kmeans function. To record the error reduction vs. iteration time, set `verbose = 1` to display the error on the screen. Try running the algorithm a couple of times using different random initializations by changing `random_ state` each time you run your program. Discuss your findings based on both the error curves and the quantized images.

2 Image segmentation considering both color homogeneity and region connectivity

The example program in Prob. 1 quantizes a pixel based on its color only. It is likely to result in image segments that are noisy. One way to improve the segmentation result is by considering not only the color but also the position of a pixel. Modify the previous program so that you include not only the color but also the image coordinate at a pixel in the feature vector of the pixel, when you use the `Kmeans.fit()` to design the codebook, and when you use the `Kmeans.predict()` to find the label of each pixel. However, after you obtain the cluster label of each pixel, you only need to use the color portion of the codeword to reconstruct the quantized color of each pixel.

To control the relative contribution of the color and the image coordinate, you should normalize the image coordinate value so that it has a range that is a scaled version of the color range. For example, each color can have a range of 0 to 255. Assume your image horizontal coordinate goes from 1 to width, and vertical coordinate goes from 1 to height. You want to scale the coordinate by x'=(x/width)*255*w, y'=(y/height)*255*w, where "w" is a user-selected parameter that controls the relative contribution of the color vs. coordinate in the segmentation result. Use the K-means++ option for initialization and set K=64. Apply your program on the same image as in Prob. 1, compare the results obtained with different w, e.g. w=0.25, 0.5, 1.0, with the result obtained in Prob. 1. You only need to do this for K=64 and you should use the default initialization method.

Please submit your report and the source code. Your report should show the original image and segmented image (and for Prob. 2, the error reduction curves) for different parts. Comment on the effect of different parameters.