I400/B457 Intro to Computer Vision - Programming assignment #2

This assignment relies on lecture5_1_segment.pptx

Problem 1: K-means clustering [30%]

0. Import SciPy clustering package for the k-means clustering. Check python textbook: page 162 (pdf) or page 128 (paper book).

1. First, generate a set of 100 random points in 2-D space (class 1), following a normal distribution (i.e., Gaussian distribution). Multiply their 2-D coordinates by 2. The following code will do it for you: `class1 = 2.0 * randn(100,2)` . Generate another set of 100 random points (class 2), following a standard Gaussian distribution whose mean is at [5,5]: `class2 = randn(100,2) + array([5,5])` . Generate the third set of 50 random points (class 3), following a standard Gaussian distribution with mean [5,0]: `class2 = randn(100,2) + array([5,0])` .

2. Do k-means (k=3), and visualize the results similar to page 162 (pdf) or page 128 (paper book). Use the functions `kmeans` and `vq`. Save the figure as “2d_clustering.jpg”.

Problem 2: Color-based image segmentation using K-means clustering [40%]

0. Download “zebra.jpg” and “fish.jpg” from Canvas.

1. Treat each pixel in the test image as a 3-D point based on its RGB values.

2. Cluster them using k-means (try k=4,8,12, ...). Compute cluster centers: for example, `centroids,variance = kmeans(features,4)` , and assign each pixel to one of the clusters using `vq`.

3. Visualize the clustering results: prepare a new image. Randomly assign a RGB value per cluster. Make the pixel value of the new image to be the RGB value of the corresponding cluster. Save the best result image as “fish_color_clustering.jpg” and “zebra_color_clustering.jpg”.

The following code may help. Feel free to use it or not.

```
# coloring the pixels
import matplotlib.colors
import random
...
im2 = array(Image.open(imageName).convert('RGB'))
im2 = im2.tolist()
colors =
map(matplotlib.colors.hex2color,random.sample(list(matplotlib.colors.
cnames.values()),k))
for row in range (0,im_height):
        for col in range (0,im_width):
            im2[row][col] = [x * 255 for x in
        colors[clust_id[row,col]]]
im = array(im2).astype(np.uint8)
im = Image.fromarray(im,'RGB')
```

Problem 3: Texture-based image segmentation using K-means clustering [30%]

1. Get your results from the step 2 of Programming Assignment #2. As a result of this step, you obtain a total of 8 different 2-D arrays computed based on 8 filters (i.e., the filter bank).

2. Treat each pixel of the test images as a 8-D data using the result above. Cluster them using k-means as you did in Problem 2. Try multiple k (e.g., 5, 10, ...). Save the best result image as “fish_texture_clustering.jpg” and “zebra_texture_clustering.jpg”.

3. Try doing this with your own images!