Assignment 3

Sanqiang Zhao ([saz31@pitt.edu)](mailto:saz31@pitt.edu))

1.

related code: Color\_quantization\_with\_K-means.py

I calculate the error and generate correponding image regarding to k equals from 1 to 14.

The image is named as fish1.jpg, fish2.jpg, fish3.jpg .... fish15.jpg in the part 1 folder.

And following lists error for different k.

41426.796688277216

31918.9587801534

26856.30462323593

23743.954460556015

21500.43239527721

19629.375928780642

18605.40184219362

17617.94044079938

16842.357425469843

16199.20233043216

15591.17234251903

15077.634472789385

14648.895946748044

14216.55522041165

Based on the result, it is obvious that the error decrease and output image turns to clear as increasing k.

The reason is as k increase, the output image turns to more close to original image due to more variance of the output image (think of k = 1, it is nothing but mean of all pixels and when k = 2, it is mean of two most variant pixels and so on).

2.

related code: Edge\_detection\_and\_circle\_detection.py

Mostly, what I do is use detectCircles to find threshold with clear edge since we only interested in circle (e.g. in Jupiter.jpg, we doesn`t interested in the edge in the right-bottom planet but more interested in the planet contour but some threshold will ignore the right big planet). And then I use detectCircles with suitable radius for finding the circle.

Note that the threshold and radius need to try multiple times, and image in part 2 fold is my result. egg\_r\_3\_5.jpg means plot radius from 3 to 5.

Actually the result is not perfect and I think the reason could be from edge detection especially for egg.jpg. Because there are many circles in the image so that space between circles could be detected as circle by algorithm. Also edge of contour and edge of circle cannot be differentiated.

3.

related code: Spatial\_Pyramid\_Match.py

Based on my experiment, Kmeans of k = 100 definitely generate more variant features than that of k = 3, and also obtain better performance.

For k = 3, the accuracy is 11% and k = 100 the accuracy is 20% when splitting the data into training / test by 50% / 50%.

However, I think size of training data is still play an important role here. Since when I split the data into 80% / 20% for training / test, I can obtain better performance, which accuracy is 26%.