Tianxiao Zhao

K-means clustering

1. How did you initialize the clustering process and why do you believe this was a good method of doing it?

To initialize the cluster in a good manner in the image orange.jpg, I set two clusters to be [255, 255, 255] and [255, 155, 0] since white and orange are the dominant colors in the image. Other clusters are randomized. If still some clusters are far away from the pixels and no points are assigned to them, they will be replaced by new random centers before computing the means in the iteration.

2. How many iterations L do you typically need to reach convergence, that is the point where no additional iterations will affect the end results?

The number of iterations depends on the value of K, the image and the amount of pre-blurring. Here, I set a threshold (`threshold = 0.01`) to stop the iterations when the maximum value difference of cluster centers on the current and previous time step is below that threshold. The results of convergence iteration for different K and images are listed below (`scale_factor = 1.0, image_sigma = 1.0`).

K	Convergence Iterations			
	`orange.jpg`	`tiger1.jpg`	`tiger2.jpg`	`tiger3.jpg`
2	8	9	9	13
4	16	71	37	35
6	32	37	103	42
8	75	67	68	166
10	89	68	106	58

3. What is the minimum value for K that you can use and still get no superpixel that covers parts from both halves of the orange?

In this case, K = 5 is the minimum value to get no superpixel that covers parts from both halves of the orange. There is a clear boundary between the two halves of the orange when K = 5, while that boundary disappears when K = 4.

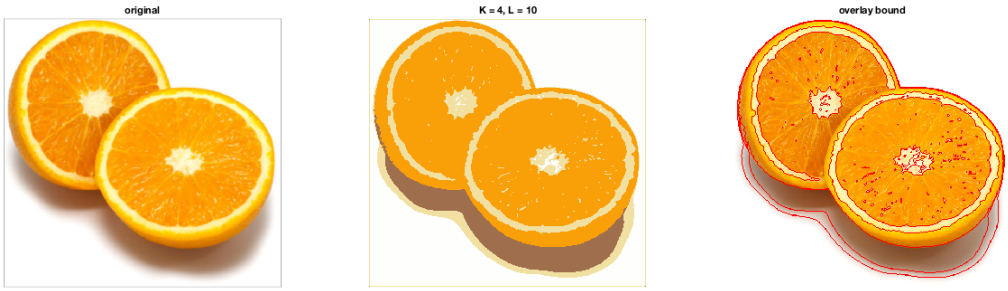

Figure 1: K = 4

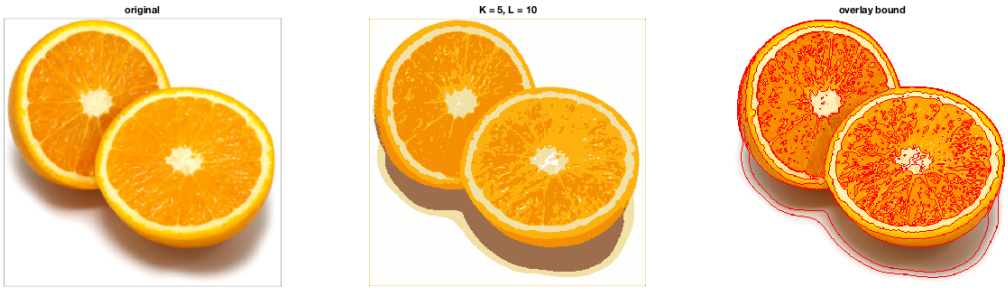

Figure 2: K = 5

4. What needs to be changed in the parameters to get suitable superpixels for the tiger images as well?

Because the tiger images are more diverse in colors than the orange color, we need to increase the cluster number K. As K increases, more iterations will be needed to obtain convergence. So, the number of iterations L should be increased as well.

Mean-shift segmentation

5. How do the results change depending on the bandwidths? What settings did you prefer for the different images?

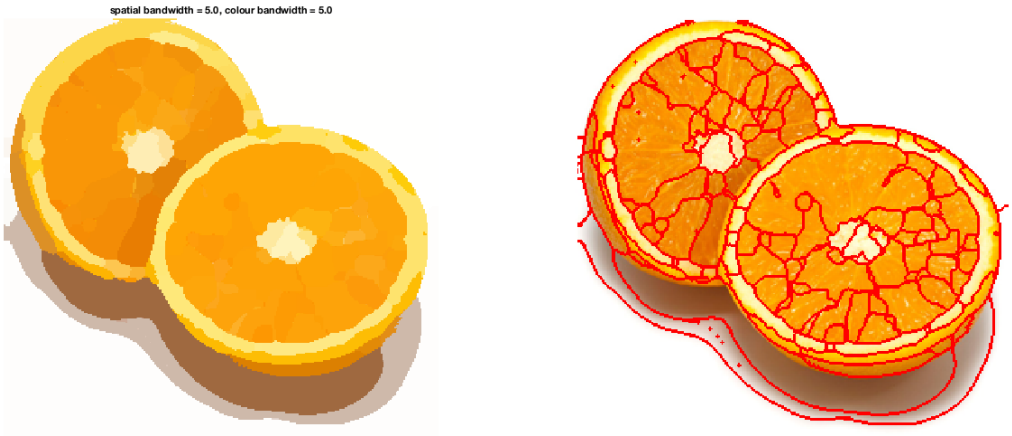

Figure 3: spatial_bandwidth = 5, colour_bandwidth = 5

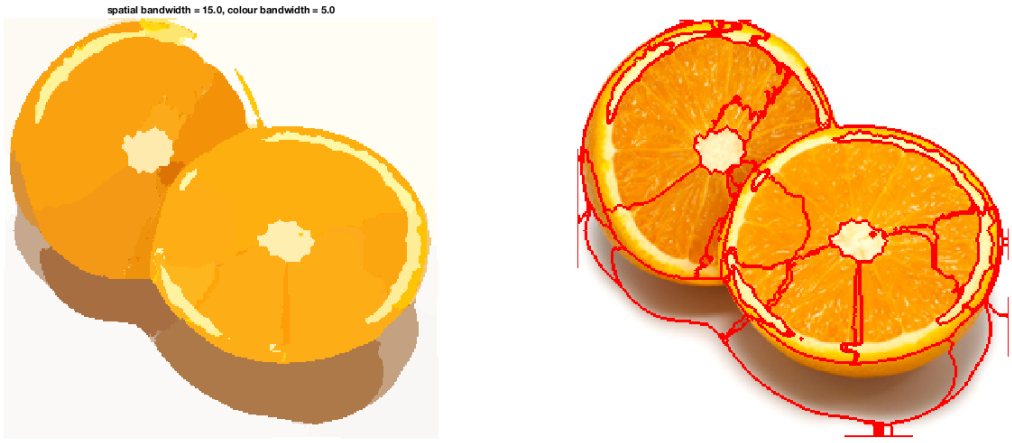

Figure 4: spatial_bandwidth = 15, colour_bandwidth = 5

The value of spatial bandwidth controls the radius of the region of interest. The larger the spatial bandwidth is, the larger the region of interest becomes, which means more pixels are involved in computing the mean and fewer modes will be generated. On the contrary, decreasing the spatial bandwidth will increase the number of modes and segments generated.

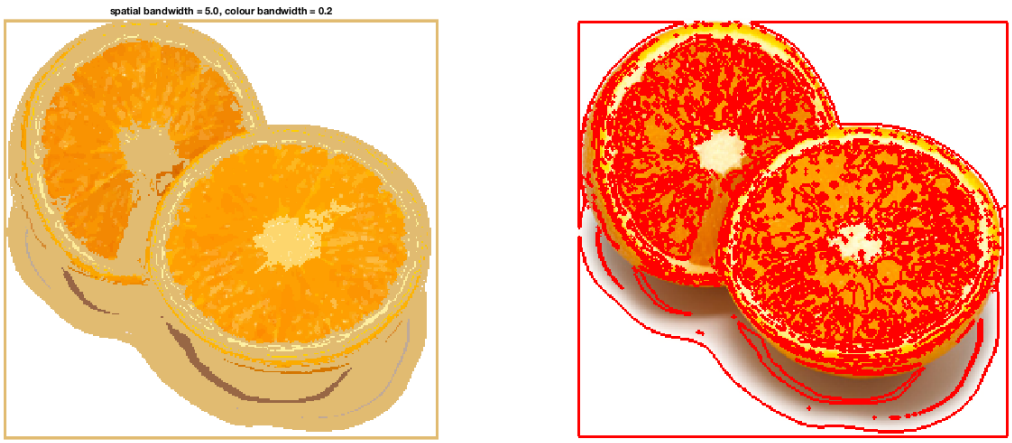

Figure 5: spatial_bandwidth = 5, colour_bandwidth = 0.2

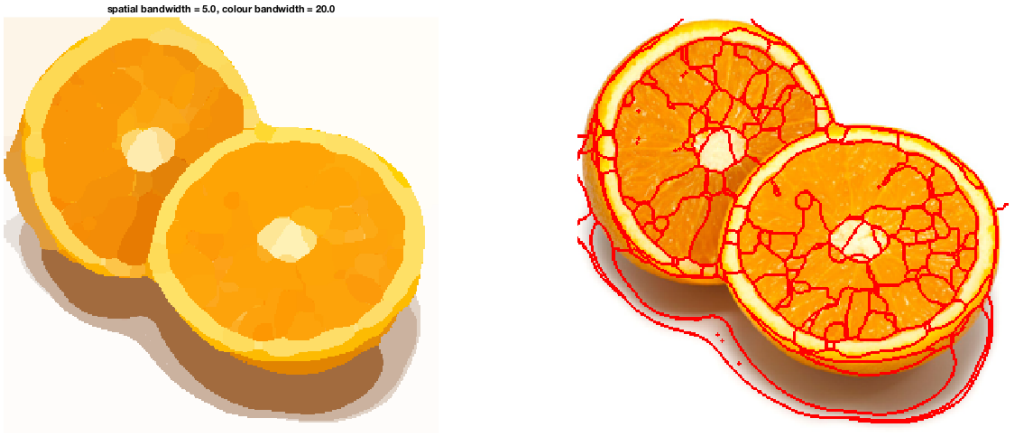

Figure 6: spatial_bandwidth = 5, colour_bandwidth = 20

As the color bandwidth increases, the image gets smoothed more and the color approximation becomes better.

6. What kind of similarities and differences do you see between K-means and mean-shift segmentation?

Similarity: both treat the color and/or position of pixels as samples from a probability distribution and try to determine its clusters or modes;

Difference: mean-shift can't output a predefined number of segments or clusters while K-means could generate K clusters as defined in the beginning; K-means only takes color dimension into consideration while mean-shift also uses the spatial information of the pixels. This means that the segments generated by K-means could span over multiple regions while they spatially concentrate by mean-shift segmentation.

Normalized Cut

7. Does the ideal parameter setting vary depending on the images? If you look at the images, can you see a reason why the ideal settings might differ?

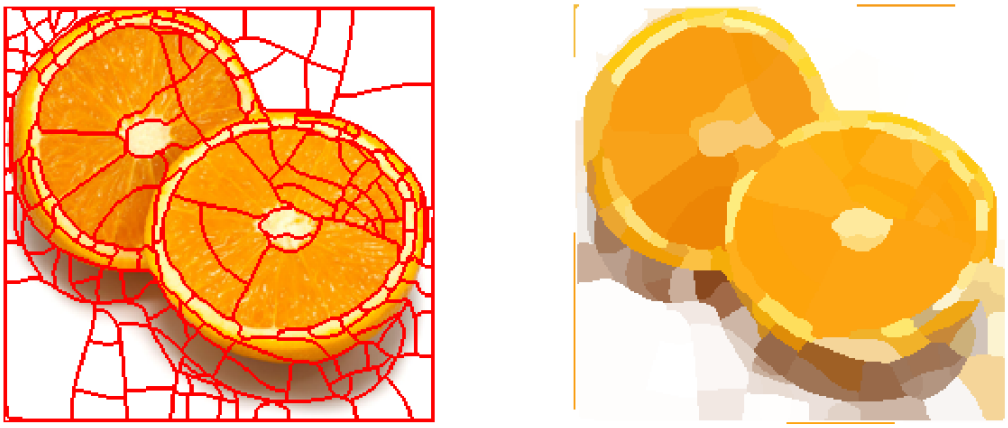

Figure 7: `ncuts_thresh = 0.4, min_area = 50, max_depth = 8`

Figure 8: `ncuts_thresh = 0.4, min_area = 20, max_depth = 8`

Figure 9: `ncuts_thresh = 0.5, min_area = 30, max_depth = 8`

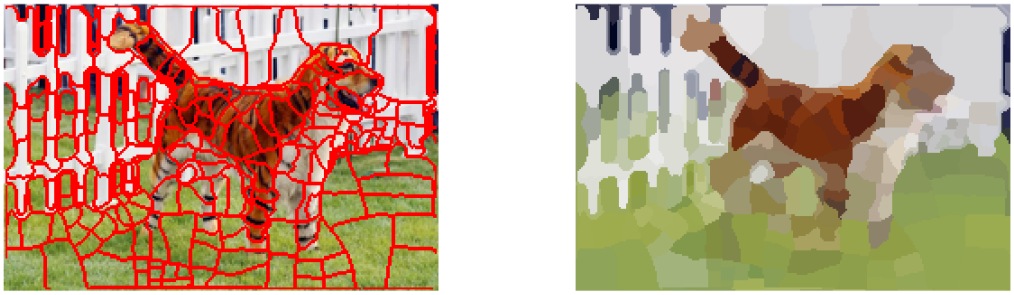

Figure 10: `ncuts_thresh = 0.4, min_area = 20, max_depth = 8`

The ideal parameter setting vary depending on the images. Parameter `min_area` depends on the size of the features and the complexity of the feature's structure; parameter `ncut_thresh` depends on the diversity in color of the image. Since the images of tigers are more diverse in color and rich in details, more accurate cuts will be required to get suitable segments. So, we need to decrease the `min_area` and increase the `ncut_thresh` to preserve the characteristics of the shape and color.

8. Which parameter(s) was most effective for reducing the subdivision and still result in a satisfactory segmentation?

Parameter `min_area`, `ncut_thresh` and `max_depth` are candidates for reducing the subdivision and resulting a satisfactory segmentation.

9. Why does Normalized Cut prefer cuts of approximately equal size? Does this happen in practice?

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

Since both $assoc(A,V)$ and $assoc(B,V)$ include those edges whose one vertex is from the opposite subset of vertices, $assoc(V) = assoc(A,V) + assoc(B,V) - cut(A,B)$ should hold. Then we could get

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(V) - assoc(A,V) + cut(A,B)}$$

To minimize $Ncut(A,B)$, we could compute the derivative:

$$\frac{dNcut(A,B)}{dassoc(A,V)} = \frac{cut(A,B)(assoc(V)+cut(A,B))(-2assoc(A,V)+assoc(V)+cut(A,B))}{assoc^2(A,V)(assoc(V)-assoc(A,V)+cut(A,B))^2} = 0$$

Therefore, $assoc(A,V) = \frac{1}{2}\big(assoc(V)+cut(A,B)\big) = \frac{1}{2}\big(assoc(A,V)+assoc(B,V)\big)$, which means $assoc(A,V) = assoc(B,V)$. So, the Normalized Cut prefers cuts of approximately equal size theoretically, but this does not always happen in practice because of the NP-hard problem.

10. Did you manage to increase radius and how did it affect the results?

Increasing `radius` will include more neighborhood pixels into computation, and will result in a decrease on the number of segments, but the color approximation will sort of get ruined.

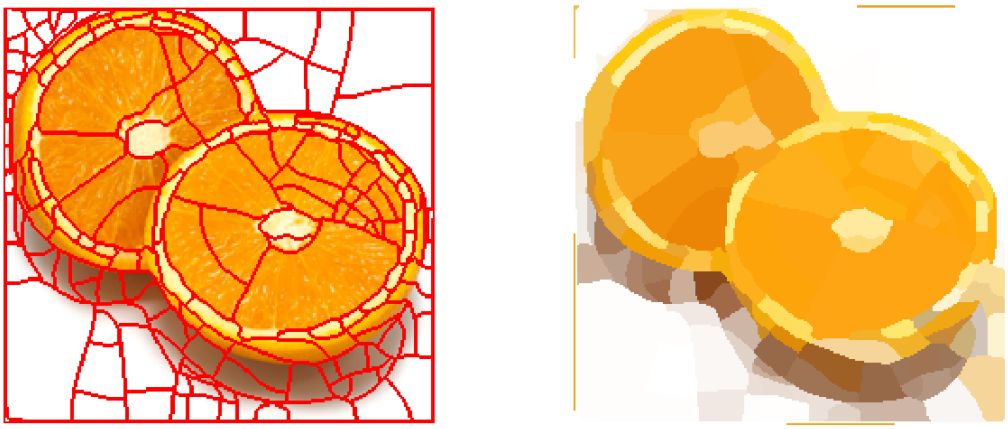

Figure 11: `radius = 3`

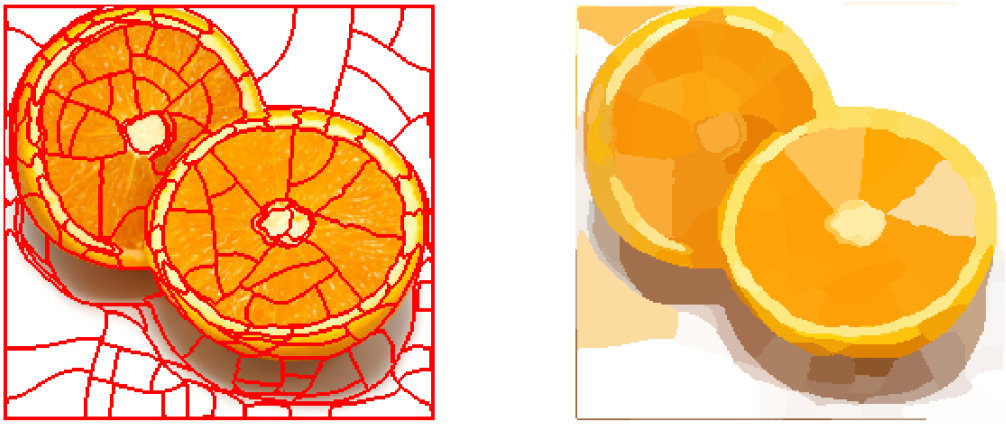

Figure 12: `radius = 6`

Segmentation using graph cuts

11. Does the ideal choice of alpha and sigma vary a lot between different images?

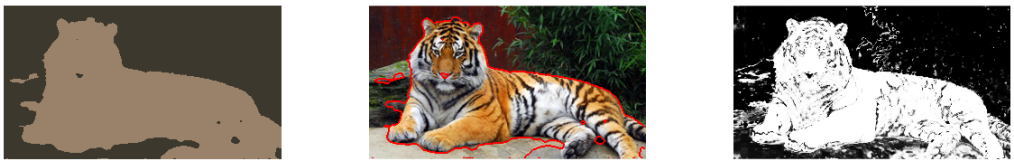

Figure 13: tiger1, alpha = 8, sigma = 10

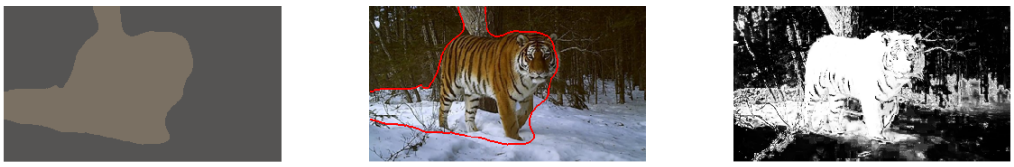

Figure 14: tiger2, alpha = 30, sigma = 25

Figure 15: tiger3, alpha = 12, sigma = 14

For images tiger1.jpg, the ideal value of alpha and sigma should all be around alpha = 8 and sigma = 10; For tiger2.jpg, alpha = 30 and sigma = 25 will be the best choice and alpha = 12 and sigma = 14 for tiger3.jpg. If we decrease the two parameters, more inaccurate subdivision will be presented; if we increase the parameters, the result looks similar to the optimal one but seems a little bit coarse.

12. How much can you lower K until the results get considerably worse?

For image tiger1.jpg, tiger2.jpg and tiger3.jpg, K should be 3.

Figure 16: K = 2

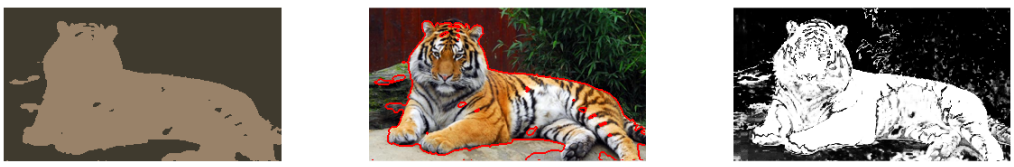

Figure 17: K = 3

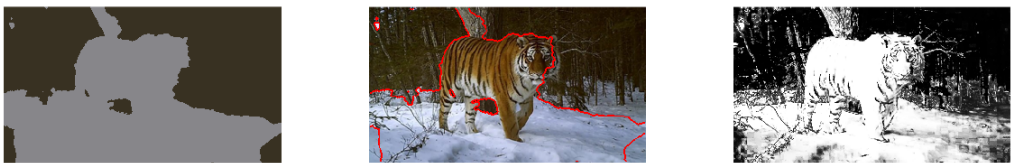

Figure 18: K = 2

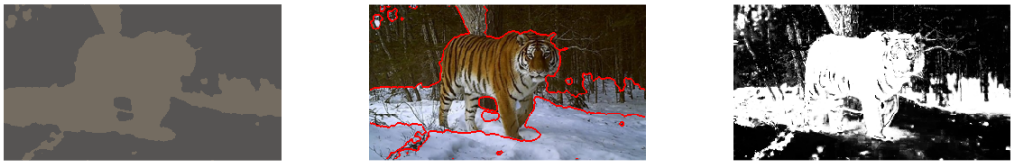

Figure 19: K = 3

Figure 20: K = 2

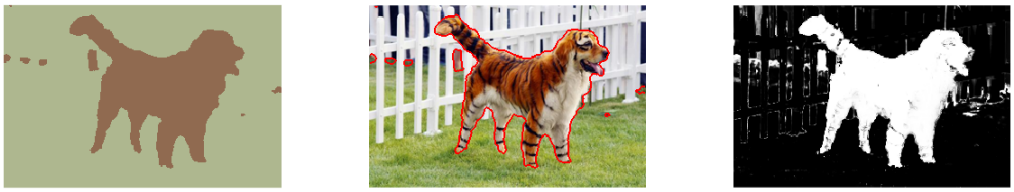

Figure 21: K = 3

13. Unlike the earlier method Graph Cut segmentation relies on some input from a user for defining a rectangle. Is the benefit you get of this worth the effort? Motivate!

If most parts of the foreground objects are located within the rectangle, then it will be much helpful to use the information inside to obtain an accurate training set, and provide classification probability for each pixel in the image. Otherwise, it won't be a good idea to use this bounding box since the result from the training data will not build a reliable inference to the whole image.

14. What are the key differences and similarities between the segmentation methods (K-means, mean-shift, Normalized Cut and energy-based segmentation with Graph Cuts) in this lab? Think carefully!!

Similarity: all the segmentation methods try to label those similar points and group them to form different clusters. In this way, the image will be segmented into various parts.

Difference: K-means doesn't take spatial information into consideration while mean-shift does. Normalized Cut and energy-based segmentation with Graph Cuts treat each image as a graph and construct the weights (edges) based on the similarity between pixels. The difference between this two method is that the Graph Cuts also need prior information of the pixels (probability to be foreground or background element) to obtain better segmentation.