DD2423 Lab 3

Giovanni Castellano

January 2023

K-means clustering

Question 1

I initialized the centers of the clusters randomly but close to the data points. In particular, for each of the three dimensions of the space where the data lives, I computed the interval of the space where the samples were distributed and then I drew the position of the center from a uniform distribution within that interval. In this way, the centers are not too far from the data.

Question 2

The number of iterations needed for convergence depends a lot on the number of clusters and on the image. In particular, if the number of clusters is more than needed it will take more time to converge. I observed that for the orange image with $K = 2$ the number of iterations is 9. However, this number can change slightly depending on how we check for convergence.

Question 3

The minimum number of K is 19. In figure 1, We can observe the result that we get if we set K equal to 18.

Question 4

Decreasing the number of clusters to 3 is enough to get good results. (figure 2)

Mean-shift segmentation

Question 5

We can observe in figure 3 and figure 4 that increasing the bandwidth leads to fewer segments. This happens because the function to optimize becomes smoother. If we increase the spatial bandwidth pixels of the same color but far away will be segmented together, this happens because the function becomes smoother on the spatial axis.

The best results I found for the different images are the following (figure 5):

tiger1: space bandwidth=12, color bandwidth=51
tiger2: space bandwidth=24, color bandwidth=12
tiger3: space bandwidth=49, color bandwidth=4

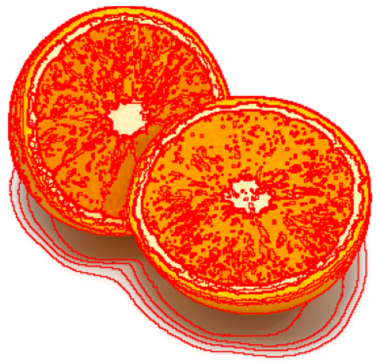

Figure 1: If we set K=18 we get superpixels that covers parts from both halves of the orange. (L = 50 seed = 60, scale=0.5, sigma= 0.4)

Figure 2: K-means results on tiger image with K=3

Figure 3: how the segmentation changes as we increase the color bandwidth (from left to right)

Figure 4: how the segmentation changes as we increase the spatial bandwidth (from left to right)

Figure 5: best mean-shift results

Question 6

The main differences are that using mean-shift we cannot control the number of clusters. Moreover, mean-shift is much more computationally expensive. Another important difference in our implementation is that we are not taking the spatial information into consideration in the k-means clustering, however, we could also decide to do it.

Normalized Cut

Question 7

Yes, the ideal parameters depend on the image. For instance, in the tiger image, we have a lot of colors and details therefore we will want to set a lower min_area parameter to allow for smaller segments. Also, depending on how many details we want to identify we can change the similarity threshold on the cut. (figure 6)

Question 8

The max_depth parameter is the most important to reduce the subdivision. But also the other parameters need to be adapted to the image to get good results.

Question 9

Let's suppose that all the weights on the edges are equal to one. Moreover, let's say that $cut(A, B) = a$ where a represents the number of edges. We can write:

$$Ncut(A, B) = \frac{a}{a + x} + \frac{a}{a + y} \tag{1}$$

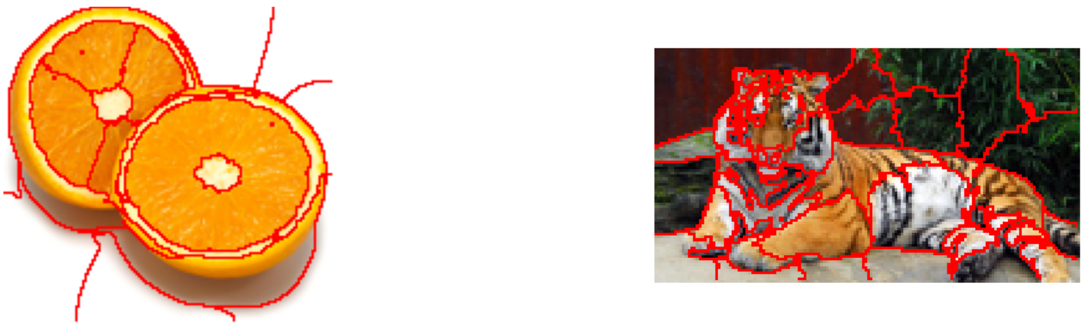

Figure 6: Normalized cut results. Orange (ncuts_thresh=0.1, min_area=150.) Tiger (ncuts_thresh=0.2, min_area=100.)

where x is the number of edges in A and y is the number of edges inside B. We want to prove that $Ncut(A, B)$ is minimized when A and B contain the same number of edges (that is, when $x = y$). If we call k the total number of edges in the graph, we can write:

$$Ncut(A, B) = \frac{a}{a + x} + \frac{a}{k - x} \tag{2}$$

computing the derivative with respect to x and setting it equal to zero we have:

$$-\frac{a}{(a + x)^2} + \frac{a}{(k - x)^2} = 0 \tag{3}$$

$$(a + x)^2 = (k - x)^2 \tag{4}$$

$$(a + x) = (k - x) \tag{5}$$

$$k = 2x + a \tag{6}$$

Moreover, we also know that:

$$a + y = k - x \tag{7}$$

joining equations 6 and 7 we get:

$$a + y = 2x + a - x \tag{8}$$

$$y = x \tag{9}$$

This proves that the two clusters contain the same number of edges. However, in real applications, we cannot assume that the similarity between two pixels is always equal to one therefore the segments will not have equal size.

Question 10

we can observe (figure 7) that by increasing the radius to 5 the results improve slightly however the algorithm needs much more time to run.

Segmentation using graph cuts

Question 11

Yes, we should set these two parameters carefully depending on the image (figure 8). changing the alpha and sigma parameters we can change the weights of the edges connecting nodes representing pixels of the image. Instead, the weights of the edges connecting the pixels with the drain or the source will not change. Therefore, We can exploit these parameters to push the algorithm towards a different cut.

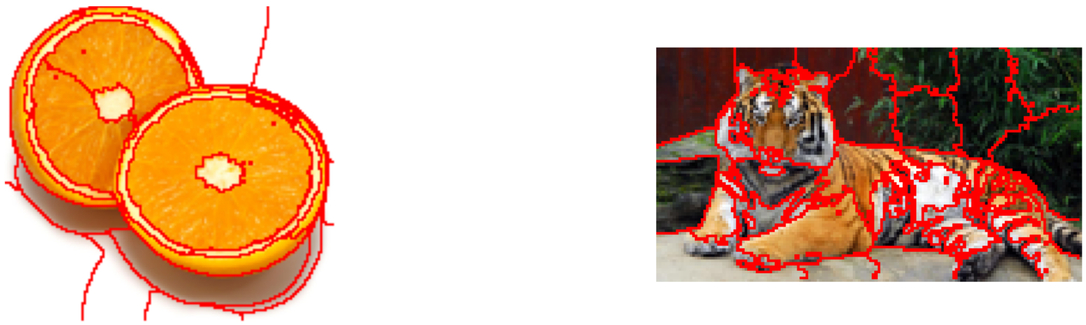

Figure 7: Normalized cut results increasing the radius to 5

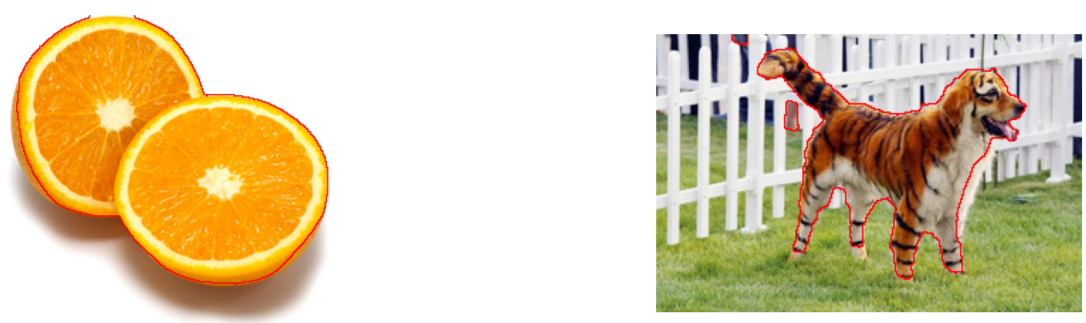

Figure 8: graph cut results. Orange parameters: K=2, alpha=50, sigma=10, area = [150, 110, 500, 350]. Tiger parameters: K=4, alpha=16, sigma=20, area = [100, 110, 570, 290].

Question 12

Ideally, the value of k should represent the number of colors in the background and the number of colors in the foreground. The choice of these parameters depends on the image. In the case of the orange image, we get good results also with k=2. The tiger image instead is more complicated, and we need at least a value of K=4

Question 13

Yes, it is very useful that we can suggest to the algorithm in which part of the image the object is located. However, this can also be seen as a limitation since the algorithm will not work properly if we do not choose a good window.

Question 14

The main difference between the K-mean and mean shift is that in the mean-shift algorithm, we do not choose the number of clusters, moreover, in the mean-shift algorithm, we also take into consideration the position of the pixels. Another important property of mean-shift is that we can adjust the variance parameters. However, mean-shift is computationally expensive since we have to loop over the pixels.
Normalized cut works in a completely different way, but the important property is that we can control the number of segments and the size. Moreover, all the pixels in a segment are close to each other (there are no discontinuities in the segments).
Graph-cut, like Normalized-cuts, works on graphs. The main difference is that we can divide the image into just two segments (foreground and background). Furthermore, we need to define a window to suggest to the algorithm where the object is located. Another important aspect is that Graph-cut outputs a probabilistic model which can be useful to study the image.