Answers to questions in Lab 3: Image Segmentation

December 15, 2023

Name: Leonardo De Clara

Instructions: Complete the lab according to the instructions in the notes and respond to the questions stated below. Keep the answers short and focus on what is essential. Illustrate with figures only when explicitly requested.

Good luck!

Question 1.

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

Answer:
The clusters' centers were intialized by randomly picking a pixel from the image and using its RGB value as first center, then by sampling another pixel, different from the previous one and using its RGB value as second center, and so on until K centers were initialized. I believe it's a good method because it avoids sampling duplicate values and it also selects colours that are representative of the image.

Question 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?

Answer:
The number of iterations is strictly related to the image with is being segmented and the number of clusters.

Image	K	Iterations
Orange	2	6
Orange	5	24
Orange	10	53
Orange	15	71
Orange	20	92
Tiger1	2	10
Tiger1	5	54
Tiger1	10	55
Tiger1	15	65
Tiger1	20	46

Table 1: Iterations before convergence

Question 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? Illustrate with a figure.

Answer:
We can observe that with K = 10 clusters we are able to distinguish the two halves most of the times since it doesn't exist any such superpixel that covers both halves.

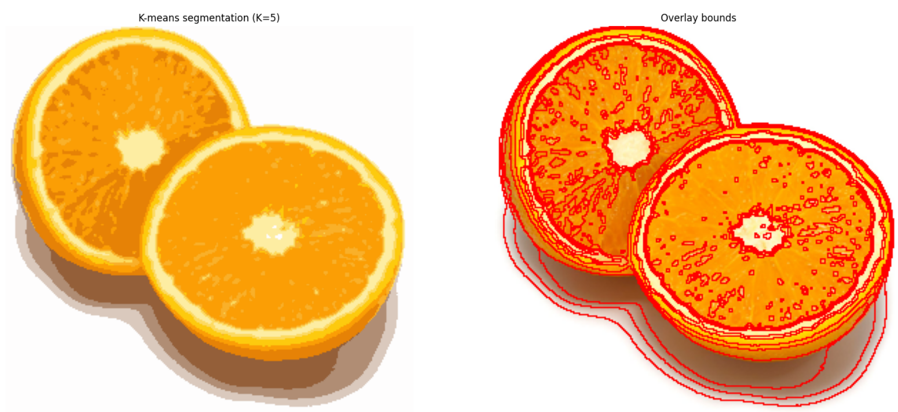


Figure 1: Orange's superpixels with K = 10

Question 4.

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

Answer:
Since the tiger's image is significantly more complex and rich in details compared to the orange one a bigger number of clusters is needed to properly segment it into meaningful regions. Moreover, due to the higher number of centers, a higher number of iterations is needed to reach convergence.

Question 5.

How do the results change depending on the bandwidths? What settings did you prefer for the different images? Illustrate with an example image with the parameter that you think are suitable for that image.

Answer:
A high spatial bandwidth results in large single-colour areas, thus the number of modes will decrease since more pixels will be included in the region of interest. On the other hand, a smaller bandwidth results in more peaks in the density function and a more steep gradient ascent, with more accurate assignment of pixels to modes for what concerns their position. A high colour bandwidth means the image will be smoothened color-wise since it determines the radius for the colour space, thus even significantly different pixels could be considered as a single region as the colour bandwidth increases.

By setting a small spatial bandwidth values such as $\sigma_s^2 = 5$ and a bigger colour bandwidth $\sigma_c^2 = 50$ we are able to segment the tiger's image in many small regions with similar colours.

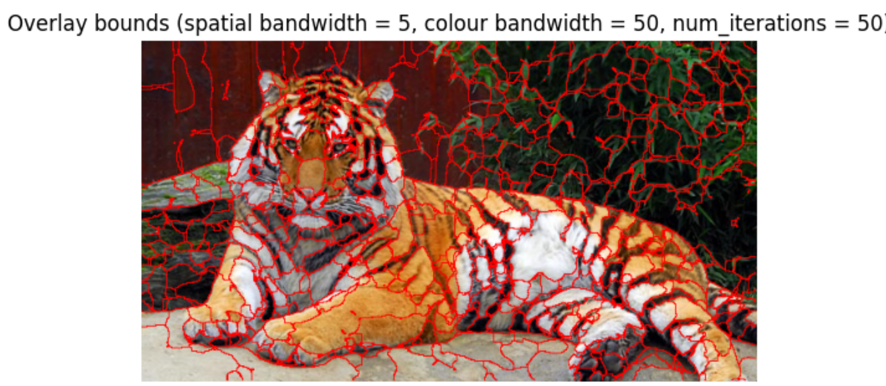


Figure 2: Mean-shift segmentation on tiger image

Question 6.

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

Answer:
The main difference lies on the fact the K-means ignores spatial information and only relies on colour info, while mean-shift take this into account. Due to this feature, K-means is more sensitive to colours outliers. Furthermore, K-means needs a pre-specified number of clusters, whereas mean-shift does not and will find a number of modes unsupervisedly; however Mean-shift needs bandwidth values as input parameters. Once applied to the same image and after some parameter tuning it's possible to show that, despite being very different algorithms, they are able to deliver similar segmentation results, recognizing the main regions in different ways.

Question 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? Illustrate with an example image with the parameters you prefer for that image.

Answer:
The ideal parameter setting depends on the image we are trying to segment because each one of them is characterized by different shapes, contours and colours' shades which need to be taken into consideration while applying the

algorithm. For example, when dealing with images with many complex details the value of `ncuts_thresh` needs to increased, leading to separate parts that look similar but may be not be part of the same structure. Furthermore, more complex images may need more cuts to properly be segmented, therefore `max_depth`, controlling the recursion's depth for the normalized cuts algorithm, needs to be set to a higher value.

In the following example, since the image is quite simple and not rich in details, we can afford to set low `ncuts_thresh` and `max_depth` values (0.1 and 10).

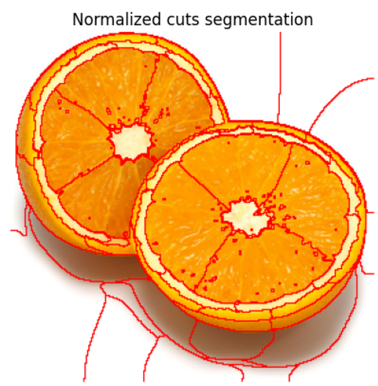


Figure 3: Orange, `ncuts_thresh` = 0.1, `max_depth` = 10, `radius` = 10

Question 8.

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

Answer:

The most effective parameters were the `min_area` value controlling the minimum size of a segment and the `max _depth` value controlling the number of recursive calls of the algorithm. As the first one increases the number of macro regions decreases while being consistent with the image's features, while a decrease in the second one results in fewer segments as fewer cuts will be performed.

Question 9.

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

Answer:
We can define the associativity of a set of vertices V as

$$assoc(V) = assoc(A, V) + assoc(B, V) - cut(A, B)$$

where A, B are two subsets of V. We are trying to minimize $Ncut$, defined as

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

By substituing $assoc(B, V)$, deriving $Ncut(A, B)$ with respect to $assoc(A, V)$ and setting the result equal to 0 we obtain that $assoc(A, V) = assoc(B, V)$. However, this result is feasible only when the `max_depth` value is set to its maximum value, which is not the case for our tests.

Question 10.

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

Answer:
By increasing the radius the algorithm needs to consider larger neighbourhoods, leading to increase in computational complexity. The results show bigger and more defined image regions as the radius increases.

Question 11.

Does the ideal choice of alpha and sigma vary a lot between different images? Illustrate with an example image with the parameters you prefer.

Answer:
The choice of α and σ deeply affect the result: since the first controls the maximum cost of an edge, by increasing it cutting across similar pixels surfaces becomes more difficult. σ, on the other hand, controls how much the cost decays for decreasing similarity between neighbouring pixels: by decreasing it separating pixels will be cheaper. These parameters need to be properly tuned for each specific image, otherwise the segmentation will not be as accurate.

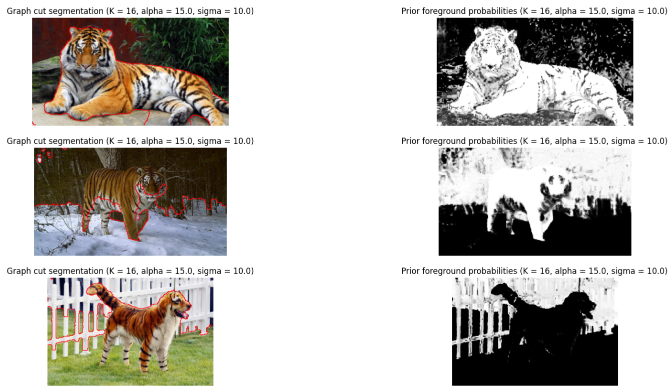
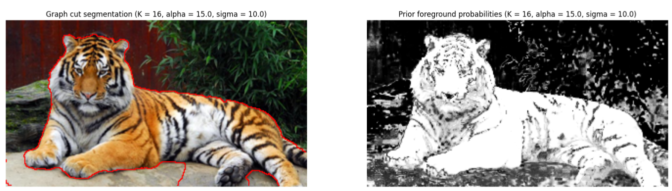
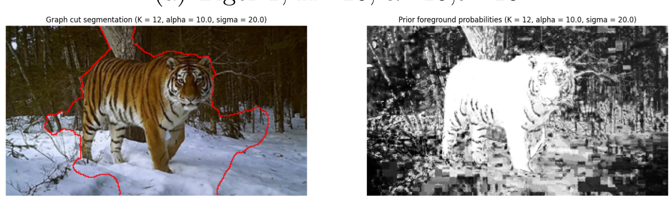


Figure 4: Same values of α and σ for different images



(a) Tiger 1, K=16, α=15,σ=10



(b) Tiger 2, K=12, α=10,σ=20

(c) Tiger 3, K=5, α=12,σ=15

Question 12.

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

Answer:
For values of K greater or equal to 3 we obtain good results with not many significant differences. $K = 2$ is not able to provide a good representation of any image.

Question 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!

Answer:
Whether the benefit is worth the effort really depends on the input image: if there is some clear object that can be boxed inside the bounding rectangle, then the accuracy of the segmentation will be deeply improved by a more specifically defined input rectangle. Images where foreground and background are not as clearly separated will be harder to segment regardless of the input, therefore making it not worth the effort.

Question 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!!

Answer:
The main differences lie on the fact that K-means and Mean-shift are clustering methods, while Normalized Cut and Graph Cuts are graph-based methods which use a graph to represent the image, while others do not. Both Mean-shift and Graph Cuts use the Gaussian distribution to model data. Furthermore, K-Means is the only method among those tested in this lab that does not take into account the spatial information about pixels. Concerning prior information, Graph Cut is the only algorithm that needs prior data about foreground and background pixel distribution. All algorithms aim at minimizing/maximising some function: K-Means tries to minimize the distance between pixel values and centers, Mean-shift tries to maximise the density function through gradient ascent and the graphs methods minimize a function related to edge costs.