Answers to questions in Lab 3: Image segmentation

Name: LASSARAT Côme & CARON Arthur

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!

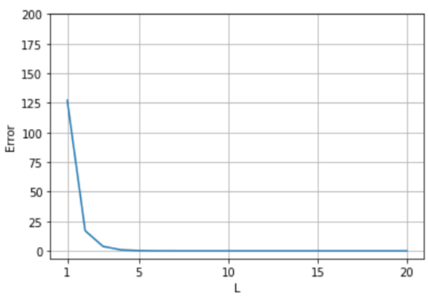
Question 1: How did you initialize the clustering process and why do you believe this was a good method of doing it?

We initialized the clustering process by settling the clusters' centers randomly. In our case, we believed it was a good idea because it is a naive and simple way to do it. However, its downside is that the centers are not spread enough in the entire space. To solve this issue, other methods can be used (K-means ++)

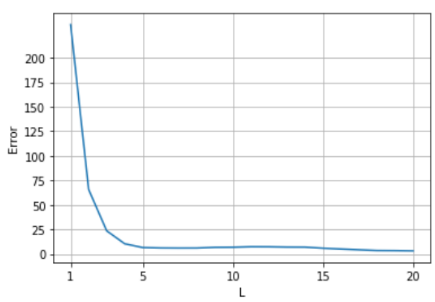
Question 2: How many iterations L do you typically need to reach convergence, that is the point where no additional iterations will affect the end results?

The number of iterations until convergence depends on the **complexity of the image** and on the **number of clusters used.**

orange.jpg ("simple picture")



K = 2

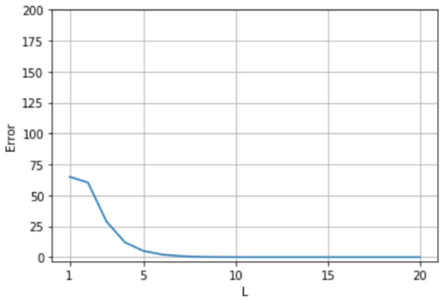


K = 20

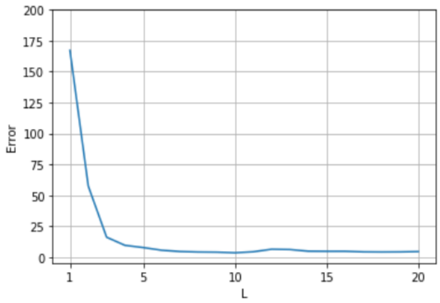
For higher K, we usually see a "bump" after a few iterations: the algorithm didn't converge yet. (We also tried to use a threshold to demonstrate that).

Example: K=10

tiger1.jpg ("complex picture")



K = 2



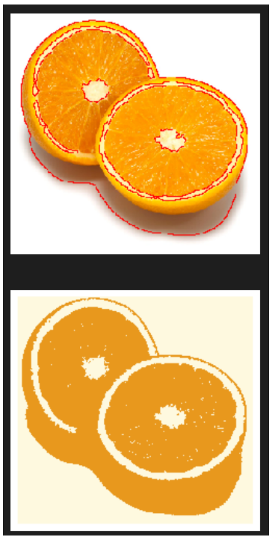
K = 20

The number of iterations until convergence also depends on **the amount of Gaussian blur**: more blur equals less complexity and thus implies less iterations.

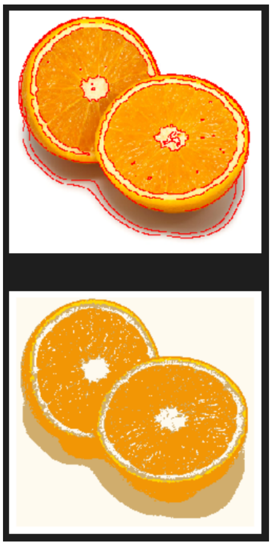
Question 3: What is the minimum value for K that you can use and still get no superpixel that covers parts from both halves of the orange? Illustrate with a figure.

Answers:

It depends a lot on the initialization of the clusters' centers. For the seed chosen below, K = 4 is sufficient.



K = 2



K = 4

For other seeds, we usually need a dozen clusters.

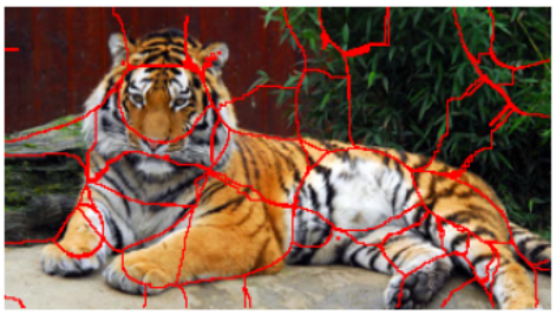
Question 4: What needs to be changed in the parameters to get suitable superpixels for the tiger images as well?

Because the tiger image is more complex (more colors, more designs, …) we need to increase the number of clusters K and thus the number of iterations L in order for the algorithm to converge.

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.

Answers:

The higher the colour bandwidths, the less details we have and the more the image is blurred. Example of an image with a really high colour bandwidths(600) and a normal spatial one(10)



When the spatial bandwidth gets higher, the zones sharing the same colour get bigger,

A good result can be found with a spatial bandwidth of 6 and a colour bandwidth of 80

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

Answers:

Mean shift segmentation uses more information, because it takes into account the position of pixels, whereas mean shift only considers the colours. Thus it is more spatially coherent.

The parameters are not the same for the two algorithms, we don't specify the number of different nodes for the mean shift segmentation.

Both methods use and update clusters of pixels to represent the image.

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.

The ideal parameters vary according to the image.

maxi_depth

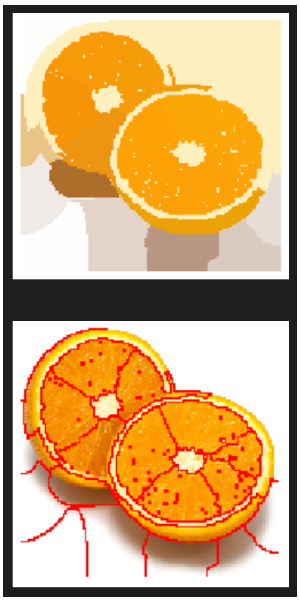
maxi_depth = 12

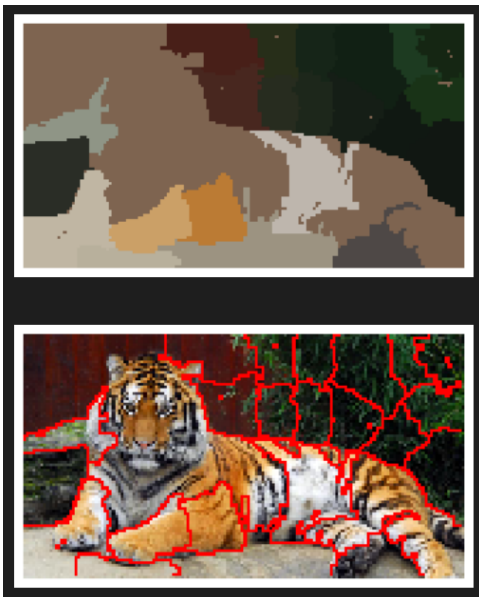
maxi_depth = 5

→ more depth = more segments

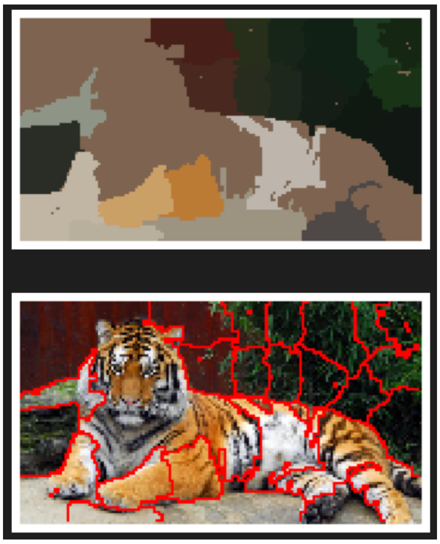
The more the image is complex, the more we need a deeper depth:

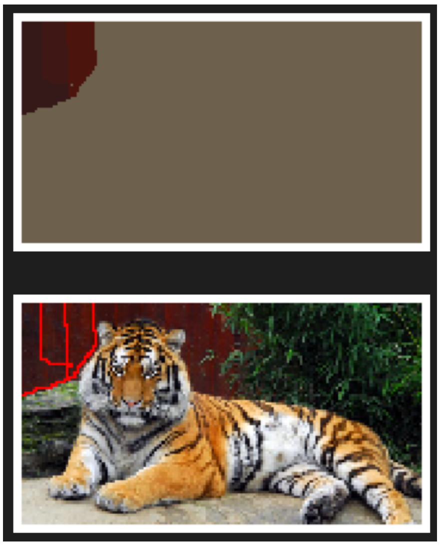
maxi_depth = 5





Color_bandwidth

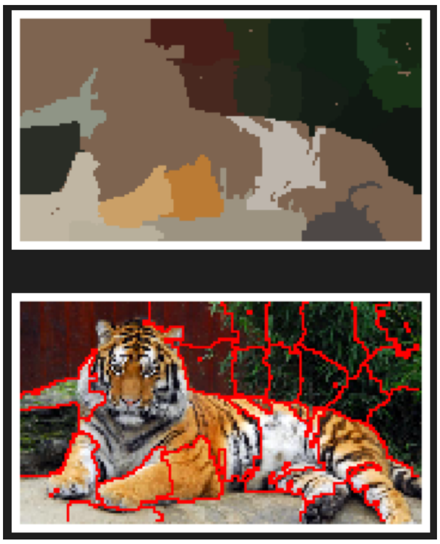


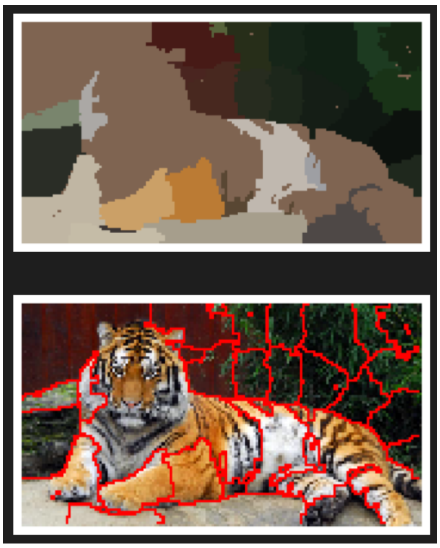


Color_bandwidth = 20

Color_bandwidth = 5

Min_area controls the minimum size of segments

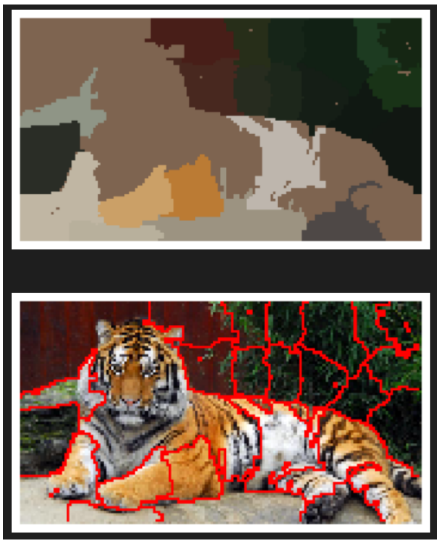


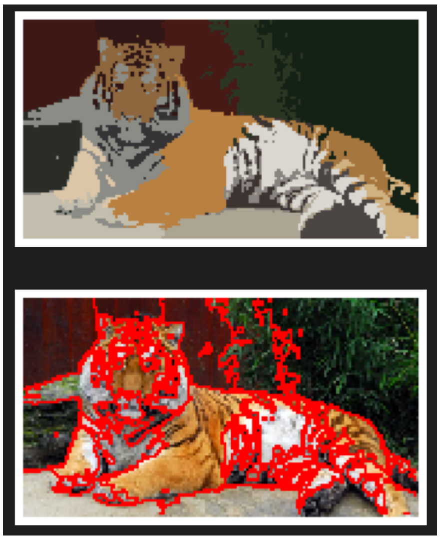


min_area = 200 **min_area = 20**

We observe more segments and thus more details when min_area decreases (but it takes longer to compute)

Radius: maximum neighboring distance

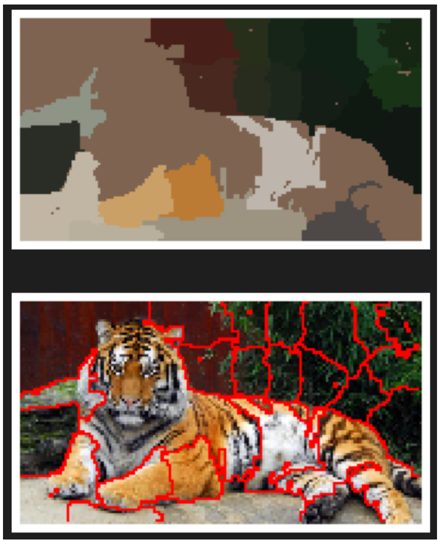


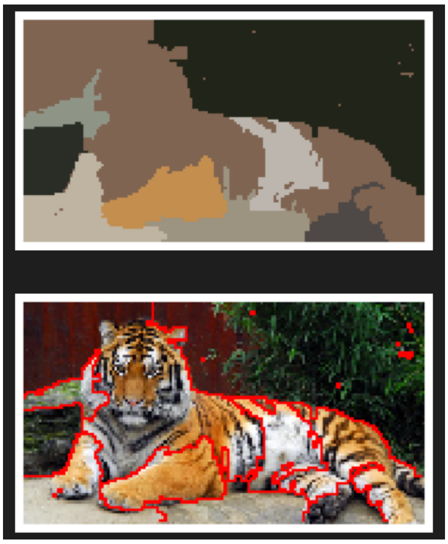


radius = 1 **radius = 10**

When the radius increases, more far away neighbors are taken into account.

ncuts_thres: controls the maximum allowed value for Ncut(A,B) for a cut to take place.





ncuts_thres = 0.1 **ncuts_thres = 0.001**

When ncuts_thres increases, more similar areas are likely to be cut

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

- ncuts_thres
- max_depth
- min_area?

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

Answers:

If we take the expression of Ncut(A,B), we can derive it by assoc(A,V) and solve the equation $\frac{\delta Ncut(A,B)}{\delta assoc(A,B)} = 0$ (we want to minimize Ncut (A,B)), its solution is assoc(A,V) = assoc(B,V): cuts have approximately an equal size.

Question 10: Did you manage to increase *radius* and how did it affect the results?

When the radius is increased, computation time increases and there is more segments.

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.

Answers:

alpha is the maximum cost of an edge.
sigma determines how much the edge cost decays for decreasing similarity between neighboring pixels σ.

When alpha increases, the maximum cost of edges increases. That means it gets very complicated to cut edges because their cost is high, particularly within homogenous areas. Thus when alpha decreases, more areas are splitted.

For a given similarity in color of two pixels, if sigma increases, the cost increases so it gets complicated to cut edges.

Efficient values of alpha and sigma seem not to vary much if the image is different. What is needed is to redefine the rectangular area

---

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

Answers:
The results get considerably worse for K lower than 3.

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!

Answers:

Advantage: easy to apply when we know we want to extract an object in a foreground of a picture

Downside: Sometimes, when the algorithm is applied to a picture, there is nothing particular to extract in the foreground.

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!!

Answers:

Similarities:

- The different methods use clusters of pixels based on similarities of neighbour pixels
- Normalized Cut and energy-based segmentation uses graphes (edges and vertices)

Differences:

- Segmentation using graph-cuts is a probability-based algorithm
- Image positions are taken into account is Mean-shift segmentation
- In graph-cut, we need to indicate the area when we want the segmentation to be done
- We can control the number of segments in the normalized cut algorithm, compared to the others

---