Answers to questions in Lab 3: Image segmentation

Name: Ait lahmouch Nadir **Program:** Machine Learning Master

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?

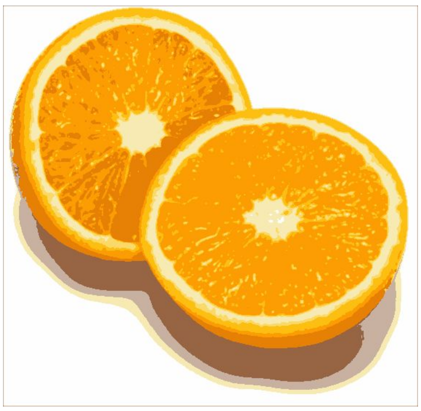
Answers:

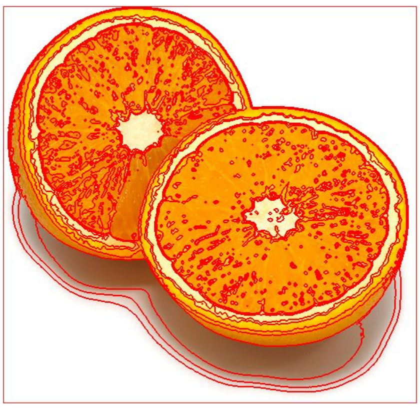
I first tried to initialize the centers by giving random colors (a tuplet of three numbers between 0 and 255) but the problem with this method is that it resulted in empty clusters. For example, for the Orange picture, with L = 10 and K = 8, it always resulted in three empty clusters.
To avoid this problem, instead of taking random colors, we can take random colors of the pictures, and that's what I did. We choose a random color from the picture for each cluster. This way, no cluster will remain empty in the end as it's theoretically impossible.

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?

Answers:

Here's the results for the pictures: orange, tiger1, tiger2 and tiger3.
Let's find the minimum L so we can have convergence in these pictures with a constant K.
Let K = 8 here, scale-factor and sigma equals to 1.










And here's the needed number of iterations for each picture to converge (for K = 8)

Image	Number of iterations L needed
Orange	86
Tiger 1	103
Tiger 2	140
Tiger 3	133

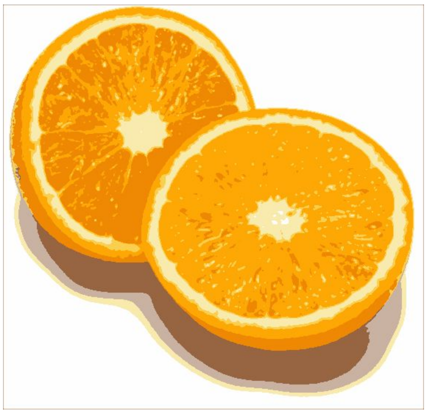
Now for K=16, the results are this :

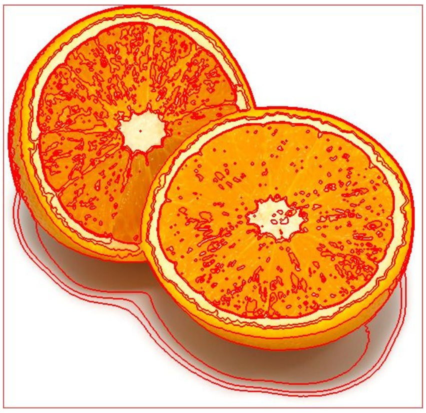
Image	Number of iterations L needed
Orange	283
Tiger 1	312
Tiger 2	329
Tiger 3	355

So when K is higher, the convergence is slower.

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.

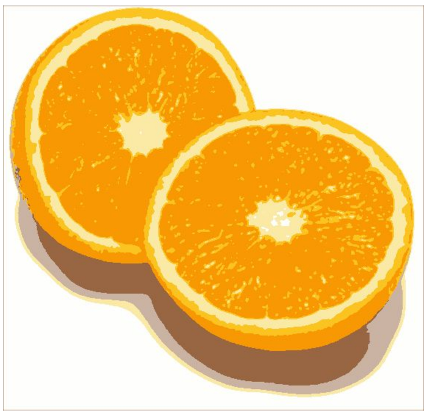
Answers:
When having K=7, there's no such superpixel that covers parts from both halves of the orange. Look at the image below :

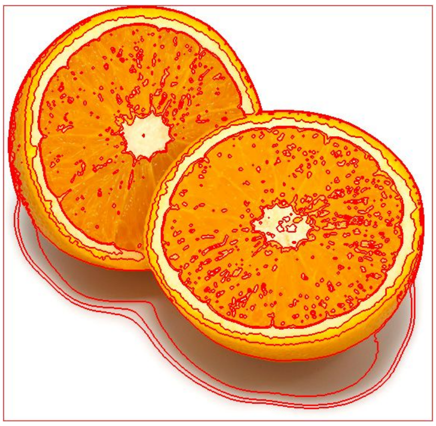




K-means with K=7

But when we change K to K = 6, we can see the formation of a superpixel regrouping the inside and the peel from respectively the left and right halves of the orange, as it is showed in the image below :





K-means with K=6

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

Answers:
Compared to the orange pic, the tiger images has many components, many colors and much more noise to reduce. And that's probably why the orange pic needs less iterations to converge. Let's take for example the image 'tiger3' and let's what parameters would be best for it.
One first intuition is to increase the blurring scale to reduce noise and enlarge the clusters. We will let scale-factor equal to 1 as its only benefit is increasing the clustering speed.

Scale = 1 :

Scale = 2 :

Scale = 4 :


We can see that as we increase the scale, we enlarge the neighbouring clusters and reduce the noise, it gives us a clearer and smoother clustering.
But we shouldn't increase a lot so we won't lose some important details in the image.
We can say from what we saw above that taking **scale = 4** is a good value for the scale parameter for tiger3.

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:

Let's first see what happens when we have a constant colour bandwidth and vary the spatial one.

–

Clusters, σ_s=5

Clusters overlaying the picture, σ_s=5

Clusters, σ_s=10

Clusters overlaying the picture, σ_s=10

We can see that the larger the spatial bandwidth is, the less are the number of modes (and therefore the larger the segments are). We can say that this bandwidth is responsible for the number of modes or how large the regions are.

Let's now test how the colour bandwidth is changing the clustering.

We notice that the color bandwidth somehow controls the smoothness of the segmented regions. When we increase it, these regions get smoother.

The couple of parameters that i think is suitable for the image is having
the color bandwidth = 5 and the spatial_bandwith = 10. It gives a good clustering when looking at the clusters alone.

Clusters, σ_b=5, σ_s=10

Clusters overlaying the picture, σ_b=5, σ_s=10

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

Answers:

Similarities:

- They both determine superpixels based on parameters of the picture (colors for k means here, and colors and coordinates for the mean-shifted).

Differences:

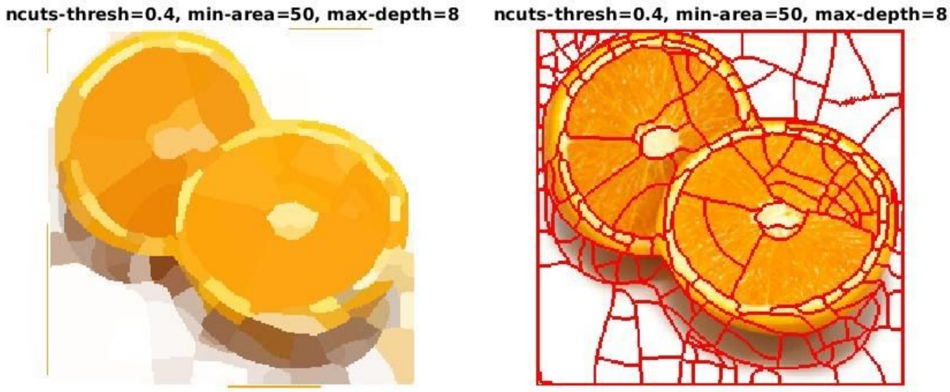
- The mean-shift method also takes the position of the pixels into account to determine the superpixels while K-means just focuses on the color.
- We can predefine the number of clusters k we want with Kmeans while we can't control that in the mean-shift method.
- The clusters in Kmeans can be discontinuous over the image while with the mean shifted method, the segments are concentrated in particular regions.

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.

Answers:

ncuts-thresh=0.4, min-area=10, max-depth=8

ncuts-thresh=0.4, min-area=10, max-depth=8



Yes, the ideal parameters vary depending on the picture and its features.

ncuts_threshold: this parameter depends on how much a color varies in a picture. So for pictures with a high variation of colors, we need better cuts and therefore increase this parameter (for example the tiger images).

min_area: this parameter controls the size of our clusters. We need our clusters to catch our image features, so the smaller the features are the smaller we should set this parameter to. For example, for the tiger pictures, we have a lot of small features, therefore we have to decrease min_area to have a better clustering. But for the orange pic, we don't have a lot of variation of features, a higher min_area would be fine here.

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

Answers:

The 3 parameters ncut_threshold , min_area and max_depth are the most effective for reducing the subdivision and still result in a satisfactory segmentation.

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

Answers:
We have :

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

Let |V| be the total sum of all the edges in the graph.
Therefore, we can say that :

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

Let's substitute this result in the first equation :

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

Our goal is to minimize this variable, let's see when that happens. For that, let's derive the equation:

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

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

$$\Rightarrow cut(A,B)(|V| + cut(A,B))(-2assoc(A,V) + |V| + cut(A,B)) = 0$$

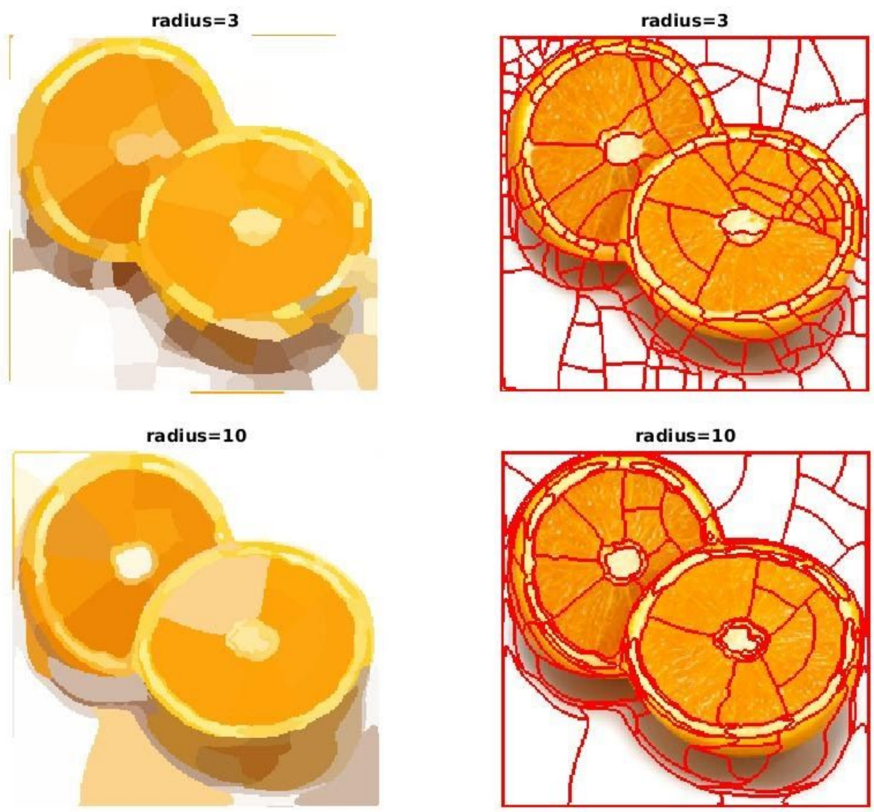
$$\Rightarrow -assoc(A,V) + |V| + cut(A,B) = 0$$

$$\Rightarrow assoc(A,V) = assoc(B,V)$$

Therefore, the Normalized Cut prefers cuts of approximately equal size theoretically.
But in reality, this does not always happen in practice due to the complexity of the problem.

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

Answers:



Increasing the radius results in more pixels into the computation and less segmented areas.
As a result, we get a bad color clustering too.

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:
Here's the best parameters for the tiger images.

Yes it does vary a lot between images (look at the images above).
We know that alpha controls the maximum edge cost and that sigma controls the edge cost decay. Therefore, a too small value for either one of these parameters results in unwanted small segments. In the opposite, a too high value results in less segments but that will miss important details in the image each time we're increasing them.

One other important parameter to yield better results is the control of the area. Indeed, as the foreground changes from pic to another, we should change the area too so we can catch most of the foreground features each time, and that's what I did for the images above.

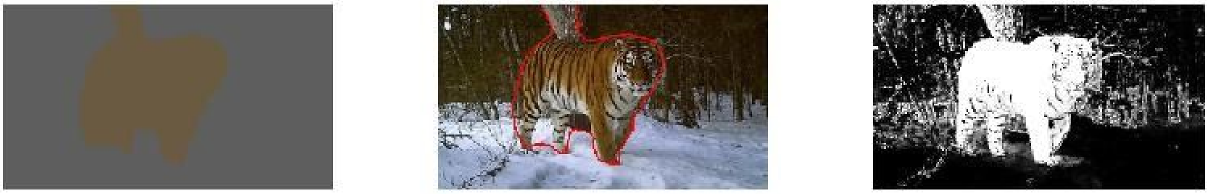
Question 12: How much can you lower K until the results get considerably worse?

Answers:

K=4

K=2

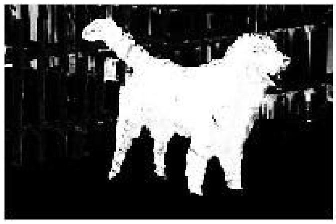
K=16



K=15

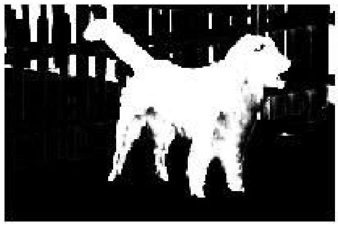
K=2





K=1





From the images above, we can see that our lowest possible values for K for which the resulting segmentation remains descent are:

4 for tiger1

16 for tiger2

2 for tiger3

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:

Yes it's totally worth the effort. As we saw in question 11, manually changing this area can yield a significantly better/worse segmentation. Indeed, if the foreground is not completely covered by the input area the segmentation will be bad. Now otherwise, having most features of the foreground in that area will be so helpful as it will give good information that will yield an accurate training.

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:

Similarity: all methods listed above label similar points and group them to form different clusters. As a result, the image is segmented to different areas.
Also, one similarity between the Normalized Cut and energy-based segmentation with Graph Cuts is that it treats each image as a graph and constructs the weights based on the similarity between pixels.

Differences:
The most important difference resides in **the informations** used by the methods to compute the segmentation. For example, for the K Means method it only uses the color while the mean shift method uses the positional arguments as well. Also, while the graph cuts and Ncut method operate in the same way, the difference is that the Graph Cuts also need prior information of the pixels which is the probability to be foreground or background element to obtain better segmentation.
