Computer vision Lab 3

Andrej Wilczek andrejw@kth.se
Ilian Corneliussen ilianc@kth.se

November 2019

1 K-means clustering

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

Answer: We implemented two different approaches for initializing the clusters. First we tried a completely unbiased method was to simply choose a random value for R,G and B for the center of each cluster. The second method was to split the image into segments and calculate the mean value of RGB-values for all pixels in each segment. Centers where then chosen from a Gaussian distribution around each mean. This method intended to add some bias to the initialization in order to avoid getting cluster centers with pixel values that are not present or close to present in the original image. The second method did however not improve results considerably and therefore the first method was used when answering question 2-4.

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: We used two methods to decide the number of iterations, L, that we typically needed to reach convergence. The first method was visual inspection, where one could no further observe any more convergence after around $L = 20$, see Figure 1. The second method was comparing the position of cluster of $L = 1000$ and iterate L_i until the absolute error became lower than 1, which happened around $L = 70$, see Figure 2. Where the absolute error was calculated as shown below,

$$e = \sum_{j=1}^{rgb} \sum_{i=1}^{n} |c - c_i|, \tag{1}$$

where rgb is equal to the number of colors, n is the number of pixels, and c_i is the cluster center for iteration i.

Figure 1: Visual inspection for $L \in (1, 5, .., 35, 40)$.

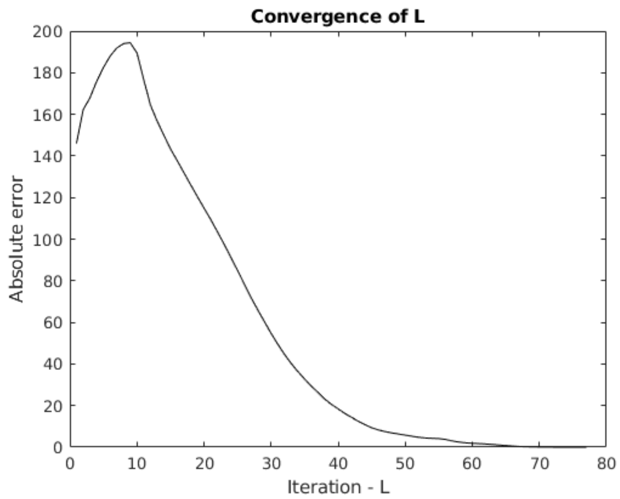

Figure 2: Absolute error function for convergence for iteration L.

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: From Figure 3 there is a clear distinction between using $k = 7$ compared to $k = 8$, where $k = 7$ have issues of having a superpixel which disappears when using a higher number of clusters, i.e. 8 clusters.

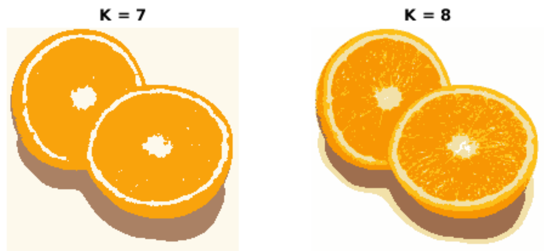

Figure 3: Images illustrating the minimum K-value.

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

Answer: The images of the tiger has a lot more detail and a larger number of different distinct colors then the orange. This requires a larger number of clusters (K) in order to obtain a satisfactory result.

2 Mean-shift segmentation

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: The spatial bandwidth determines the window size around each pixel that we calculate the mean for and thus a large spatial bandwidth will result in fewer and larger segments. The color bandwidth smooths the colors in the image, i.e. a small color bandwidth results in a larger number of distinct colors and a large color bandwidth gives a smaller number of distinct colors. The best results for the tiger image where obtained in Figure 5 with color bandwidth = 1 and spatial bandwidth = 4.5, i.e. quite low values for both bandwidths in order to capture the details in the image.

Figure 4: Image of tiger with different color bandwidth.

Figure 5: Image of tiger with

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

Answer: Both methods captures the essential color scheme of the picture. The biggest difference is that the mean-shift segmentation has a blurring effect of the picture and capture only the most important features of the pictures on local areas of the picture, see Figure 6. K-means does not take into account the spatial distance between segments but only finds the distinct different colors and mean them to their average color, see Figure 6.

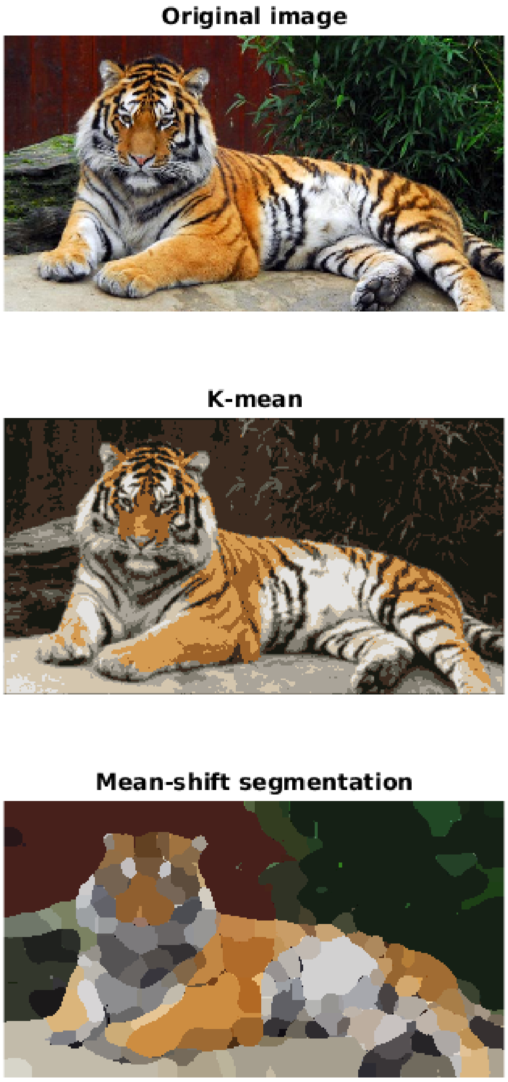

Figure 6: Tiger with K-means and mean-shift.

3 Normalized Cut

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 using the parameters you prefer for that image.*

Answer: Yes, the ideal parameter settings does depend on the choice of image. Depending on the size of the features of interest the min_area parameter should be tuned accordingly. Further, the color_bandwidth should be tuned to match the color scheme depth of the picture, i.e. if the colors of the pictures are close to each other the color variance should be smaller etc. This can be observed by inspecting Figure 7 compared to Figure 9. In Figure 7 the color scheme is more diverse and therefore the color_bandwidth is smaller.

Figure 7: color_bandwidth = 14.0, radius = 8, ncuts_thresh = 0.2*0.15, min_area = 200*0.2, max_depth = 10, scale_factor = 0.4, and image_sigma = 2.

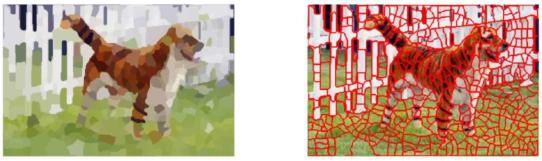

Figure 8: color_bandwidth = 20.0, radius = 4, ncuts_thresh = 0.4, min_area = 100, max_depth = 15 scale_factor = 0.6, and image_sigma = 1.5.

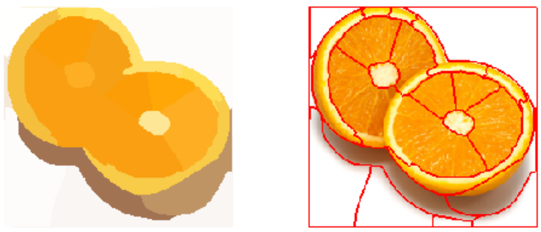

Figure 9: color_bandwidth = 24.0, radius = 6, ncuts_thresh = 0.2*0.3, min_area = 200*0.3, max_depth = 10, scale_factor = 0.4, and image_sigma = 2*1.

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

Answer: As stated in the instructions, the parameters ncuts_thresh, min_area and max_depth influence the subdivision. This is reasonable since ncuts_thresh defines the threshold for when we make a cut, min_area defines the minimum area of segments and max_depth the number of recursive splitting. Combining appropriate values of these parameters can result in less subdivision with satisfactory results. Furthermore we found that increasing the radius also can reduce the subdivision without affecting the results in a negative way.

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

Answer: This is because when minimizing $Ncut(A, B)$ through setting the derivative $\frac{dNcut(A,B)}{dassoc(A,\mathcal{V})}$ equal to zero we end up with the expression $assoc(A, \mathcal{V}) = assoc(B, \mathcal{V})$. In practice this does not always happens since it's NP-hard which is why we approximate using eigencomputations.

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

Answer: Increasing the radius will, a stated above, reduce the subdivision. This is because the radius defines how large the area of neighboring pixels are and a larger radius will include a larger area of pixels. This results in fewer and larger segments. However increasing the radius decreases computational speed.

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 appropriate choice of alpha and sigma does depend on the image. Alpha controls the maximum edge cost. A large value for alpha means that the cost for separation is higher and therefore results in a smoothed blob-like foreground segment and respectively a small alpha results in a lot of separation with lots of tiny background segments inside the foreground. Sigma controls the speed of the decay and a low value of sigma makes the cost more sensitive to an increased difference between pixels. Depending on how big difference there is between the desired foreground and background segment in the image alpha and sigma should be tuned accordingly.

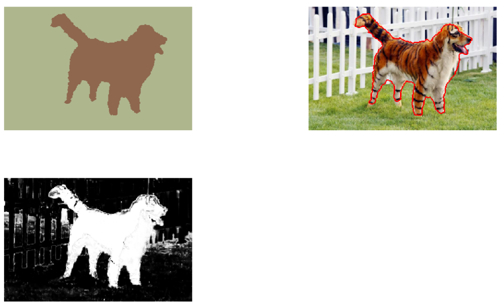

Figure 10: alpha = 20, sigma = 8.

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

Answer: We could lower K to 3 before getting bad results. When K was 2 we got just two colors, one for the background and one for the object.

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: This depends on what kind of application this method should be implemented on if you want an unsupervised algorithm one should preferably use the earlier mentioned method. But, if the application allows for user input the accuracy can be improved significantly by using Graph Cut segmentation, for example, Photoshop is such an application where it could be beneficial to use Graph Cut segmentation.

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: All four methods divide the image into segments/clusters mainly based on similarities/differences in pixel values. Mean-shift, normalized cut and graph cut also take the spacial dimension into account. Both normalized cut and graph cut are graph based methods and views the image as a network of vertices with edges connecting them. Graph cut is supervised and requires an input from the user whereas all other methods are unsupervised.