Answers to questions in
Lab 3: Image segmentation

Name:Rithika Harish Kumar   Program: TCSCM

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.

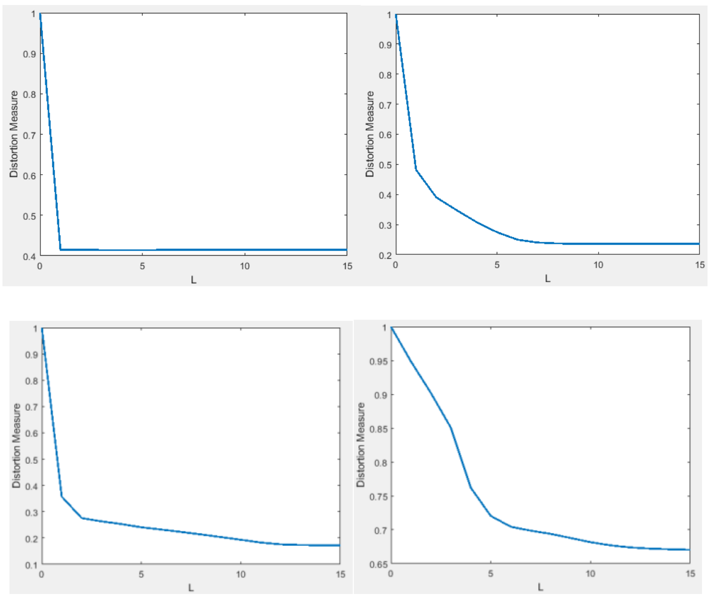
Good luck!

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

Answers:
Kmeans does not converge to global optimum. Based on the initialization of cluster centres it converges to different local optimas. Forgy method was used for cluster initialization. It randomly selects the initial K clusters from the datapoints. The algorithm is run several times and the one with lowest distortion is used.

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:
Kmeans converges in local minimums. The number of iterations depends on the value of K, image and pre-blurring. As K increases, the number of iterations (L) increases. L is inversely proportional to the amount of pre-blurring. Images with K = 3,4,5,6 respectively.



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:
To get superpixels that does not cover both halves, we have to increase the K value. such that extremely small difference between colour result in different superpixel, because of which there are breaks in links. To reduce the line breaks, sigma can be lower than 1. For orange image, K= 6 is the minimum value to get no superpixel that covers parts from both halves of the orange. The boundary disappears when K = 5.

K = 2

K=3

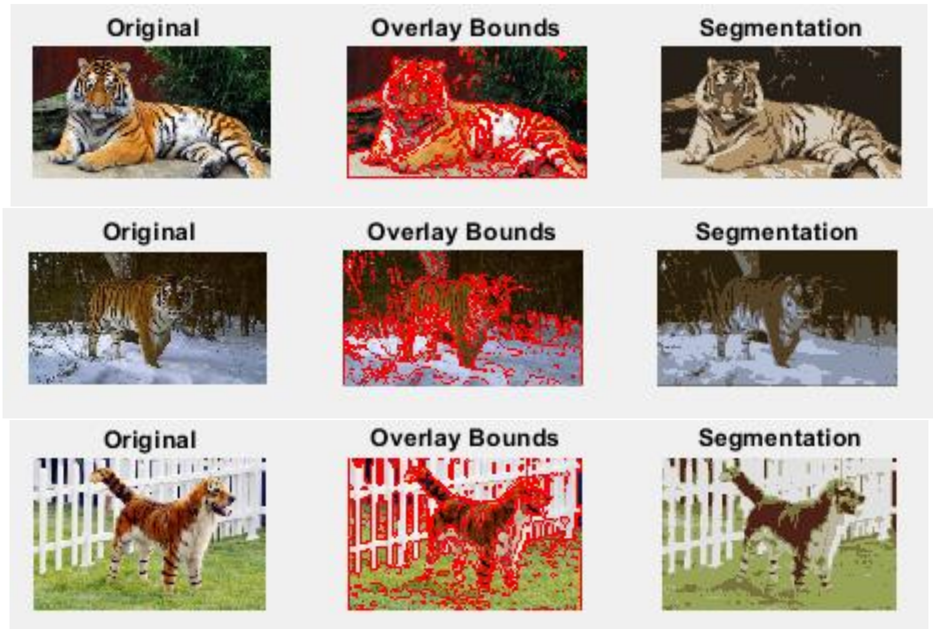
K=4

K=5

K=6

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

Answers:
As the other images have many colours compared to the orange image, increasing K can help. But the fast change in colours will result in superpixels covering small areas. So, bluring the image(increasing sigma) priorly may give better result.
For multicoloured images kmeans is not suitable for segmentation as it does not contain information on pixel position.



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:
Spatial bandwidth controls the area of the region of interest. The larger the spatial bandwidth, the larger the region of interest becomes, which implies fewer modes(no of segments). Vice versa, decreasing the spatial bandwidth, increases the number of modes and segments generated. As the color bandwidth increases, the image gets smoothed more and the color approximation becomes better.

S=4 C=4

S=5 C=20

S=10 C=5

S=20 C=5

S=10 C=30

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

Answers:

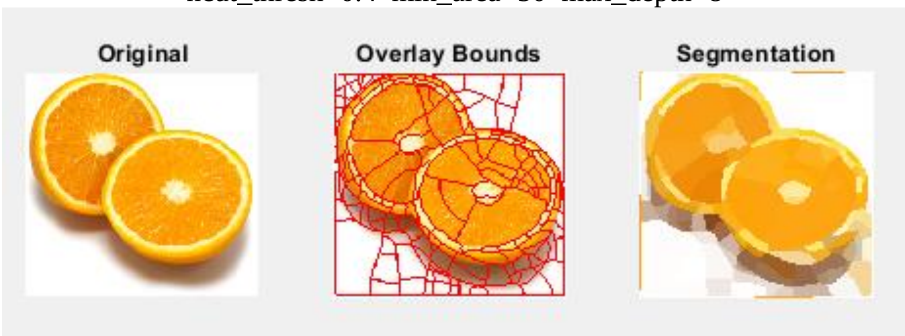
Similarities: Both are unsupervised algorithms that converge in local optimum. Both used for vector quantisation but not very optimal for object segmentation.

Differences:
K-means only takes color dimension into consideration while mean-shift also uses the spatial information of the pixels. This means that the segments generated by K-means could span over multiple regions while they spatially concentrate by mean-shift segmentation.
Kmeans is sensitive to initialization and outliers and generates K clusters as defined.
MeanShift sensitive to predefined area of window(2-colour and spatial) and can control the importance of the neighbouring while clustering but we cannot explicitly control the number of segments.
Kmeans assumes equal covariance which yields spherical clustering of the datapoints.
Mean shift does not have this problem but it does not scale well with the dimensionality of the feature space.
MeanShift is more computationally expensive than Kmeans.

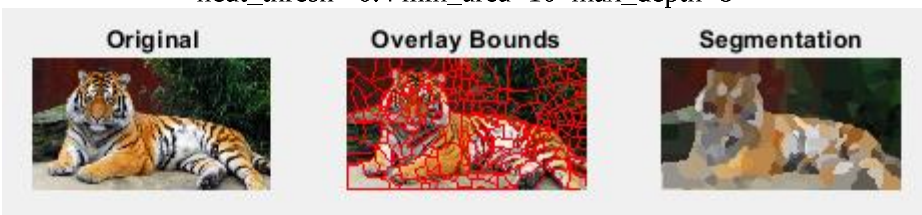
Question 7: Does the ideal parameter setting vary depending on the images? If you look at the images, can you see a reason why the ideal settings might differ? Illustrate with an example image with the parameters you prefer for that image.

Answers:
Yes, the ideal parameter setting vary depending on the images. The reason maybe due to the colour variation in the image, relative contrast between objects, size of object of interest, etc. The min_area depends on the size of the features and the complexity of the feature's structure, ncut_thresh depends on the color variation of the image.

ncut_thresh=0.4 min_area=50 max_depth=8



ncut_thresh= 0.4 min_area=10 max_depth=8



ncut_thresh=0.5 min_area=30 max_depth=8

ncut_thresh=0.4 min_area=20 max_depth=8

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

Answers:
max_depth, ncuts_thresh and min_area are parameters responsible to control the recursive subdivision.
The max depth specifies the depth of recursion, that is the number of subdivision.
The cutting threshold is the upper limit (the largest Ncut(A,B) value) which will restrict the segment to further partitioning. It ensures that it will not partition a segment that is highly likely to be an object in the image.
The minimum area of segment determines the minimum number of points that a segment could have, avoiding small partitions.

Generally, the bigger we set the cutting threshold and the larger we set the minimum area of segment, the bigger segments we will get. So, in any case we could say that if the segments are too small, we would like to increase ncuts_thresh or increase min_area.

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

Answers:

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

If assoc(V) is the total number of edges in graph, then

$$\mathrm{assoc\ (V)} = \mathrm{assoc\ (A,V)} + \mathrm{assoc\ (B,V)} - \mathrm{cut(A,B)}$$

We get,

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

Except assoc(A,V) all are considered as constants. To minimize Ncut(A,B) we could compute the derivative:

$$\frac{d}{dassoc(A,V)}\mathrm{Ncut\ (A,B)} = \frac{-cut(A,B)}{assoc(A,V)^2} + \frac{cut(A,B)}{(\mathrm{assoc\ (V)} + \mathrm{cut(A,B)} - \mathrm{assoc\ (A,V)})^2} =$$

$$\frac{cut(A,B)(\mathrm{assoc\ (V)} + \mathrm{cut(A,B)})(-2\mathrm{assoc\ (A,V)} + \mathrm{assoc\ (V)} + \mathrm{cut(A,B)}))}{(\mathrm{assoc(A,V)}\ (\mathrm{assoc(A,V)} - \mathrm{assoc\ (V)} - \mathrm{cut(A,B)}))^2}$$

For, $\frac{d}{dassoc(A,V)}\text{Ncut}(\text{A},\text{B}) = 0$

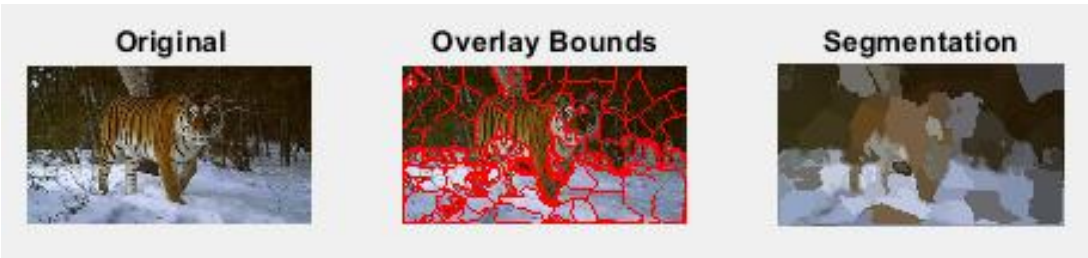
$$assoc(A,V) = \frac{assoc(V) + cut(A,B)}{2}$$

Thus assoc(B,V) = assoc(A,V)

So, the Normalized Cut prefers cuts of approximately equal size theoretically, but this does not always happen in practice because finding the optimal cut is a NP-hard problem.

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

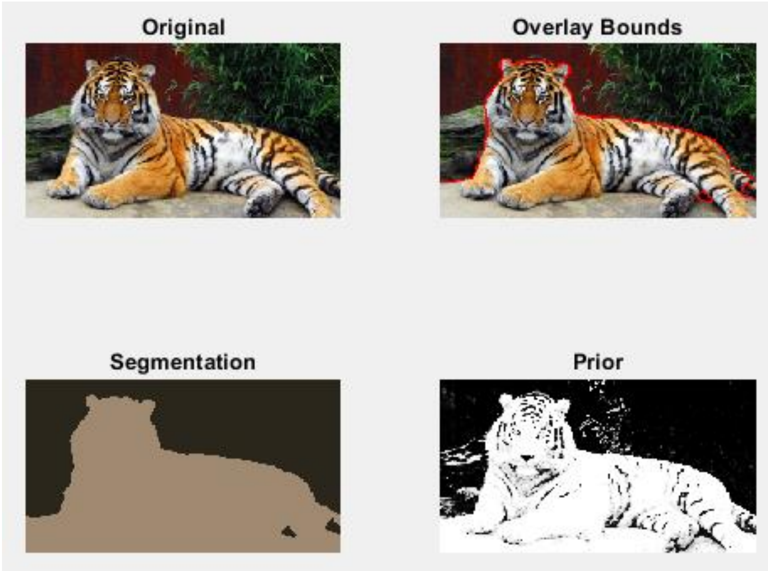
Answers:
The larger the radius the more pixels are considered, the number of segments decreases. For small radius, the weight matrix becomes sparser with small values. Thus shorter computational time.
The radius in the figures below was set to 6, 12 respectively all the other parameters remained constant.



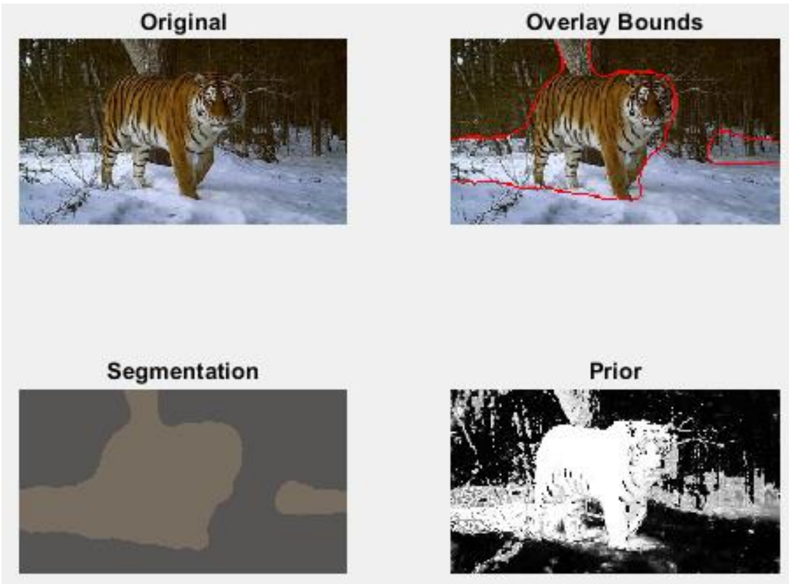
Question 11: Does the ideal choice of *alpha* and *sigma* vary a lot between different images? Illustrate with an example image with the parameters you prefer.

Answers:
The alpha and sigma varies between different images. If we look at the pairwise edge cost equation: we can see that sigma controls the strength of the similarity between the colors of two pixels.
The bigger the value of σ the stronger similarity. This will result in more difficult segmentation between different pixels. So choose a value not too high (to allow further segmentation) and at the same time not too small so as to not enhance the correlation between the colors of neighboring pixels at extrema.
The alpha determines strength of the maximum connection between neighboring pixels. The higher the value of alpha the more difficult the connectivity will break. Too high values will not give fine details the proper attention.

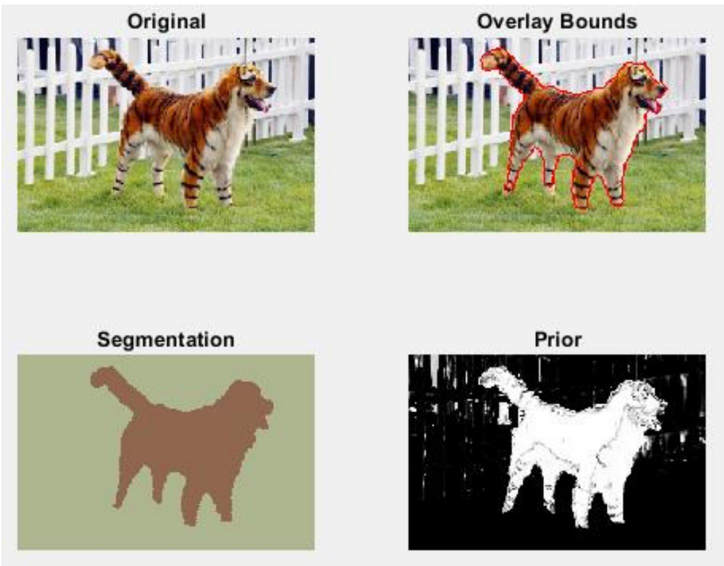
a=40 s=5



a=30 s=25



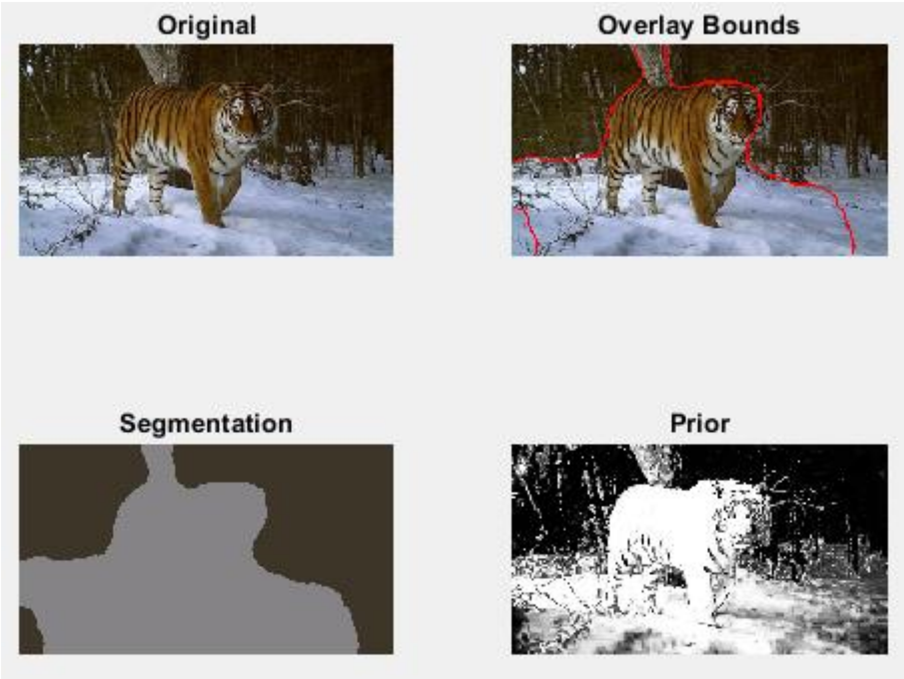
a=12 s=14



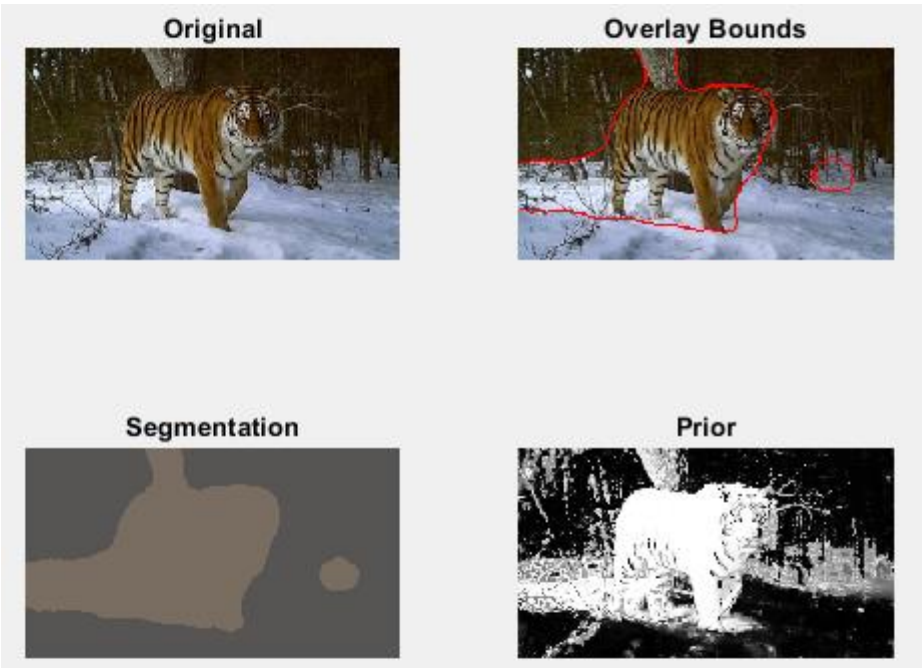
Question 12: How much can you lower K until the results get considerably worse?

Answers:
The lack of color variation in orange image gives good results even for K=2.But for the tiger's images which have more variation in color in both the foreground and the background need higher K values.

K=2



K=4



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:
Segmentation is supervised as we define the rectangle and give prior information to the system. The results we get with this method are better than the previous methods.
If most parts of the foreground objects are located within the rectangle, then it is helpful to use the information to obtain an accurate training set, and provide classification probability for each pixel in the image. Otherwise, it won't be a good idea to use this defining rectangle since the result from the training data will not build a reliable inference to the whole image.

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:
K-means: Does not take into account the spatial information. Computationally simple. Requires user to set the number of clusters and is sensitive to initial cluster centres. Tries to minimize distances.

Meanshift: takes spatial information into account. It uses gradient ascent similar to a minimization problem. Hyper parameter optimization is harder.

Normalized cut: Tries to minimize the cut based on affinity and color differences of neighbourhood. Hyper parameter optimization is harder.

Energy based segmentation with graph cut: Similar to normalized cut only in the way that they use the linking between the datapoints as a measure of similarity/difference. Minimizes the the energy which is represented by the similarities between the neighbourhoods and the priors. Supervised bayesian probabilistic approach which introduces prior to the system. This method outperforms the above segmentation techniques.

All do not guarantee convergence to global optimum.
