CENG 466 Fundamentals of Image Processing Take-Home Exam 4

Mustafa Sezgin
Computer Engineering
e2380863@ceng.metu.edu.tr

I. Object Counting

A. Implementation

The tasks in the first question are implemented using `measure`, and `morphology` packages of Scikit library. The object counting algorithm consists of the following steps:

1) gray scale the image
2) binarize the image
3) apply morphological operations
4) count the connected regions

In this implementation, instead of gray scaling, red channels of the images are used. This approach allows us to work on a gray image with high contrast. See Figures 1-4.

Fig. 1: Gray scale version of the image "A2.png"

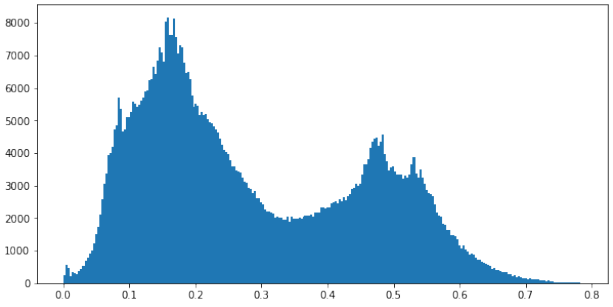


Fig. 2: Histogram of Figure 1

After inspecting the histograms, the images are binarized using a threshold. Figure 5 shows the binarized version of Figure 3

Fig. 3: Red channel of the image "A2.png"

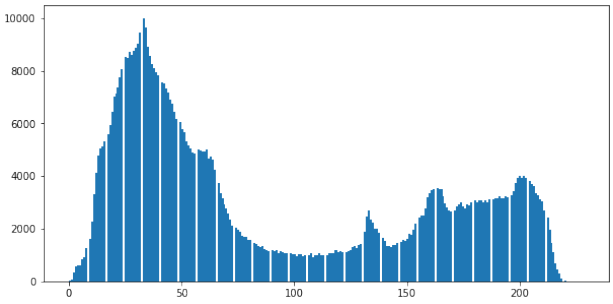


Fig. 4: Histogram of Figure 3

Fig. 5: Binary image

with a threshold value of 144. On the binary image, opening and closing morphological operations are applied. Figure 6 shows the result of morphological orders. The connected

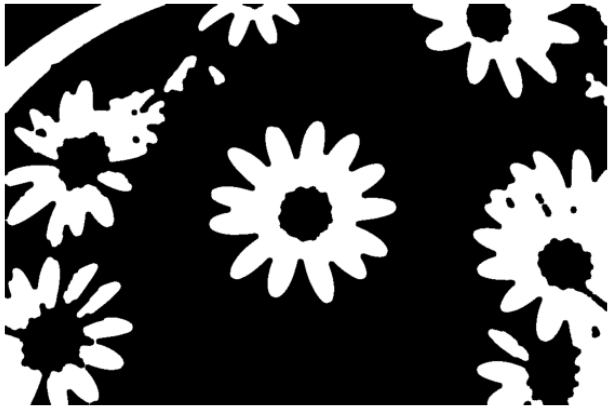


Fig. 6: Result of morphological operations

regions of this image are counted.

B. Drawbacks

The drawbacks of the algorithm used in this question are as follows.

- The morphological operations takes longer when the size of the structuring element get larger.
- The images are not suitable to separate the objects from each other perfectly. Therefore, while trying different morphological operations, some unnecessary regions might not be erased, and some objects might get split into more than one region.

II. Segmentation

A. Implementation

The tasks in the second question are implemented using `color`, `draw`, `future.graph`, `measure`, `segmentation`, and `cluster` packages of Scikit library. The segmentation algorithms consist of the following steps:

1) generate superpixel image using SLIC algorithm to reduce the computational complexity
2) cluster images using mean shift and n-cut segmentation algorithms
3) draw boundary overlay
4) generate tree relationship structure [1]
5) draw region adjacency graph

Mean shift segmentation is one of the various algorithms to group data into clusters. In this case, input images are required to be clustered using mean shift segmentation. To accomplish this, `sklearn.cluster.MeanShift` class is used on the superpixel image. For "B3.jpg" shown in Figure 7, mean shift segmentation maps for bandwidth values 8, 16, and 32 can be seen in Figure 8. As the bandwidth value increases, segments get larger and the image contains less detail.

[1]Not implemented

Fig. 7: "B3.jpg"

N-cut segmentation is another one of the various algorithms to group data into clusters. In this case, input images are required to be clustered using n-cut segmentation. To accomplish this, `skimage.future.graph.ncut` function is used on the superpixel image and its region adjacency graph generated by `skimage.future.graph.rag_mean_color`. For "B3.jpg" shown in Figure 7, n-cut segmentation maps for threshold values 0.001, 0.01, and 0.1 can be seen in Figure 9. As the threshold value increases, segments get smaller and the image contains more detail.

Boundaries of regions are drawn on images using `skimage.segmentation.mark_boundaries` function.

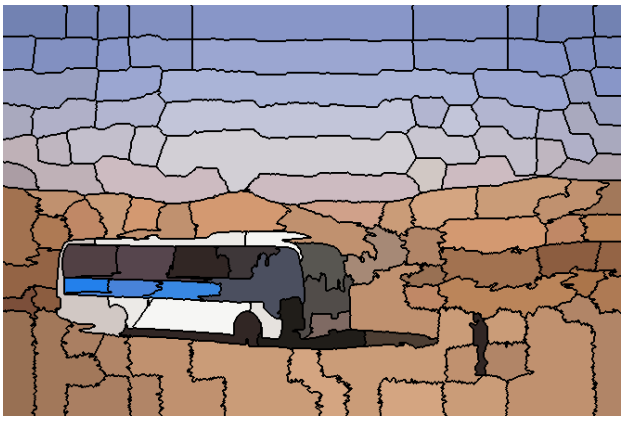
Region adjacency graph is a data structure where nodes represent regions and edges exist between two adjacent regions. The centroids of the regions are calculated using `skimage.measure.regionprops` function. For each region, a small circle is drawn at the centroid. For each pair of adjacent regions, a line is drawn between the two centroids. Circles and lines are drawn using the related functions from `skimage.draw` package. The region adjacency graphs of mean shift and n-cut segmentations of "B3.jpg" can be seen in Figure 10 and Figure 11.

For "B1.jpg", "B2.jpg", and "B4.jpg", mean shift and n-cut segmentations and their region adjacency graphs can be seen in Figures 12-15.

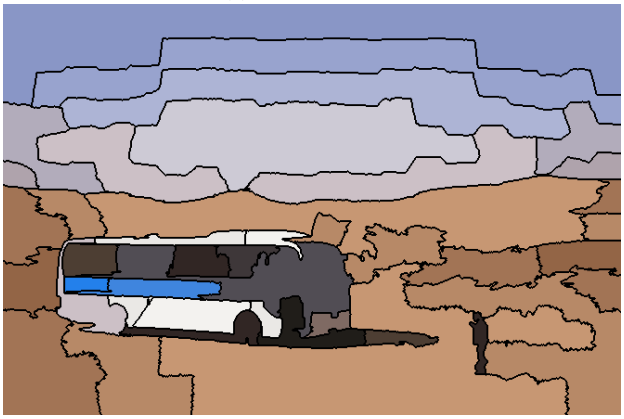
B. Drawbacks

The drawbacks of the algorithms used in this question are as follows.

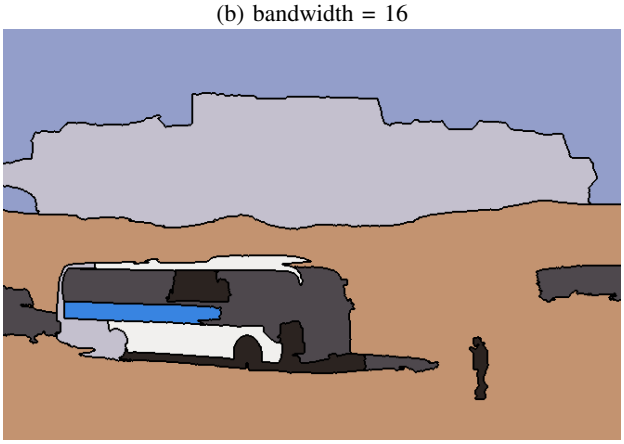
- The algorithms take ages to run even though the superpixel images are used. Question 2 takes nearly 1 hour to produce 24 images in total. Therefore, I have used smaller resolution versions of input images, which takes 5 minutes to run.



(a) bandwidth = 8

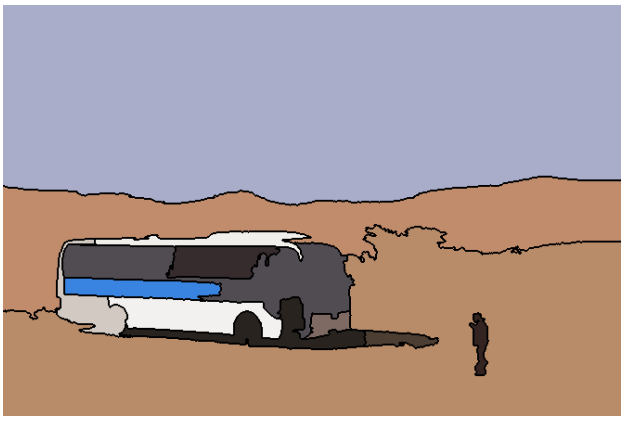


(b) bandwidth = 16

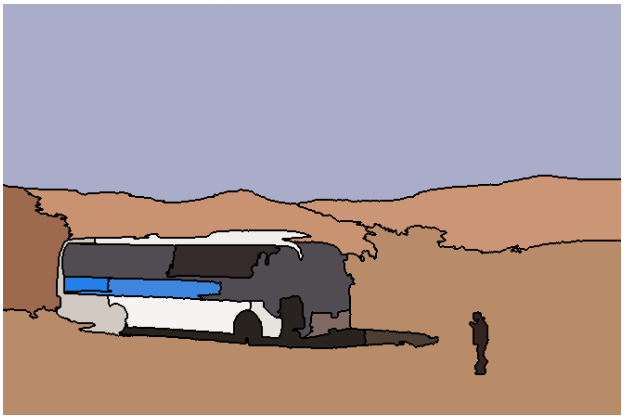


(c) bandwidth = 32

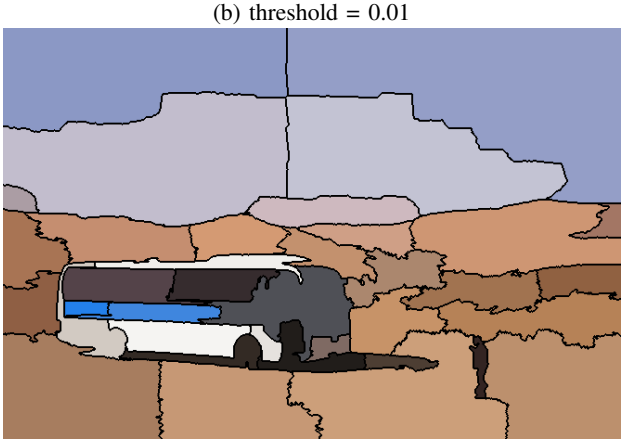
Fig. 8: Mean shift segmentation of “B3.jpg”



(a) threshold = 0.001

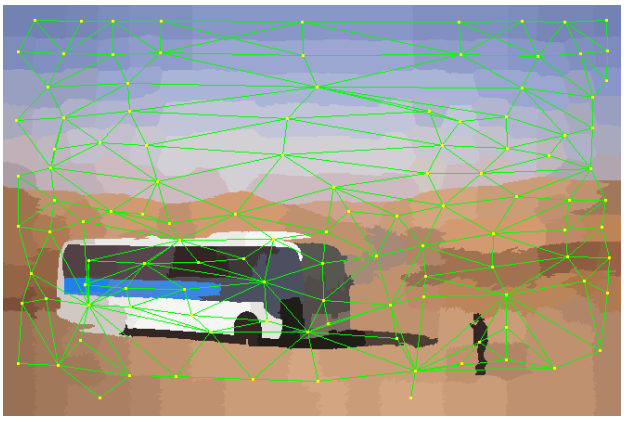


(b) threshold = 0.01

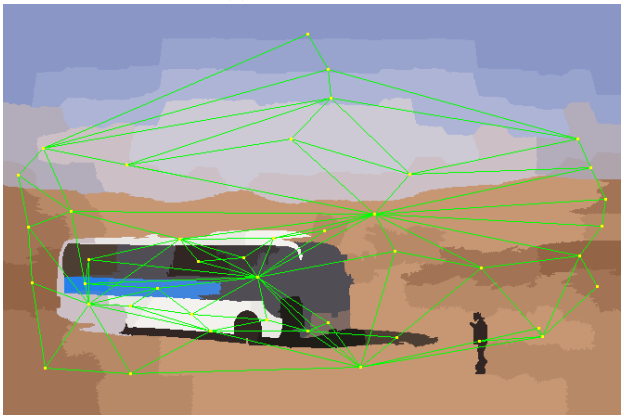


(c) threshold = 0.1

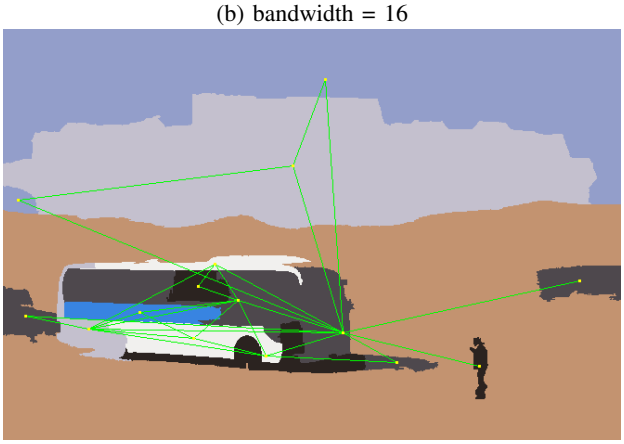
Fig. 9: N-cut segmentation of “B3.jpg”



(a) bandwidth = 8



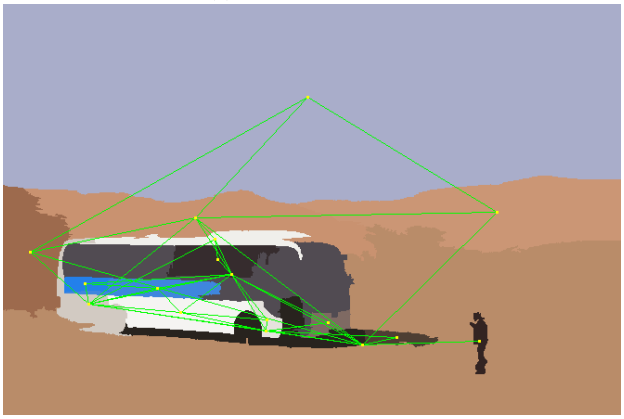
(b) bandwidth = 16



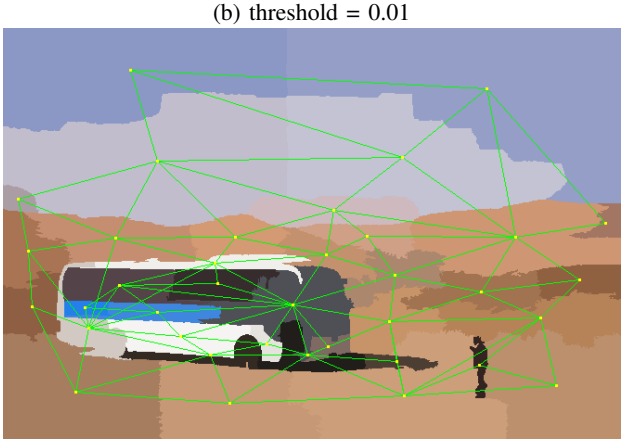
(c) bandwidth = 32

Fig. 10: RAG of mean shift segmentation of “B3.jpg”

(a) threshold = 0.001

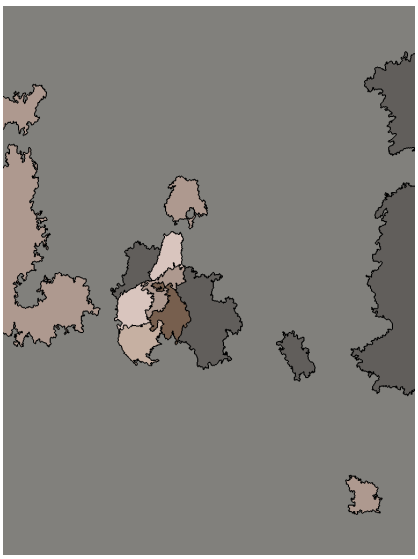


(b) threshold = 0.01



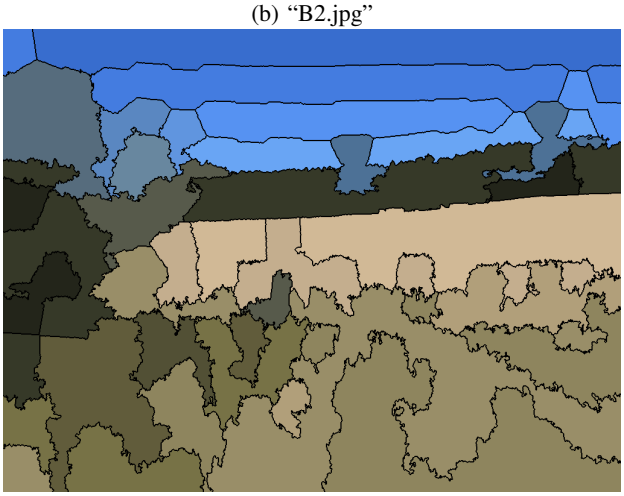
(c) threshold = 0.1

Fig. 11: RAG of n-cut segmentation of “B3.jpg”



(a) “B1.jpg”

(b) “B2.jpg”

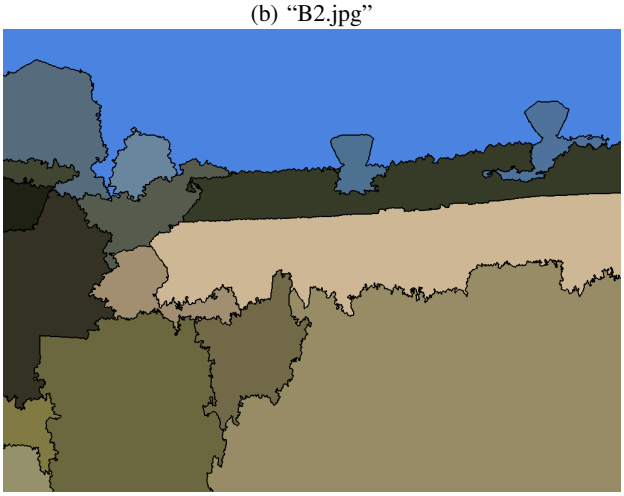


(c) “B4.jpg”

Fig. 12: Mean shift segmentation for bandwidth = 16

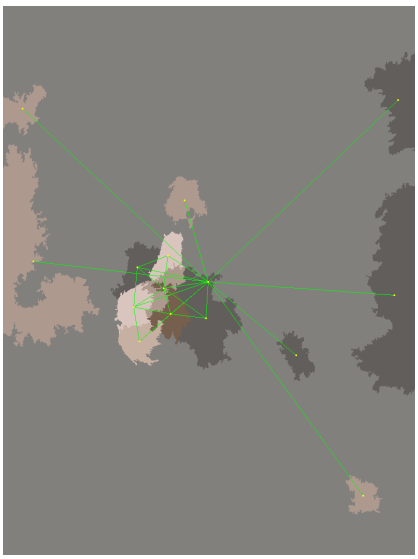
(a) “B1.jpg”

(b) “B2.jpg”

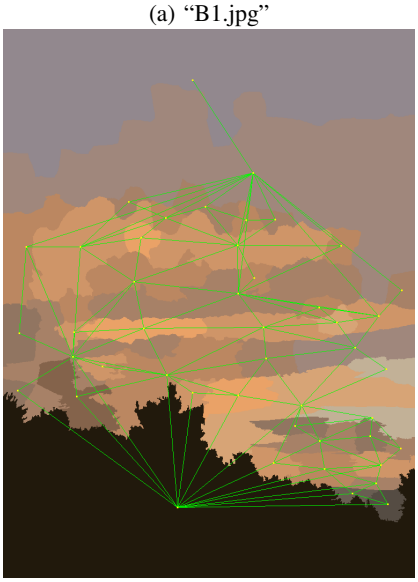


(c) “B4.jpg”

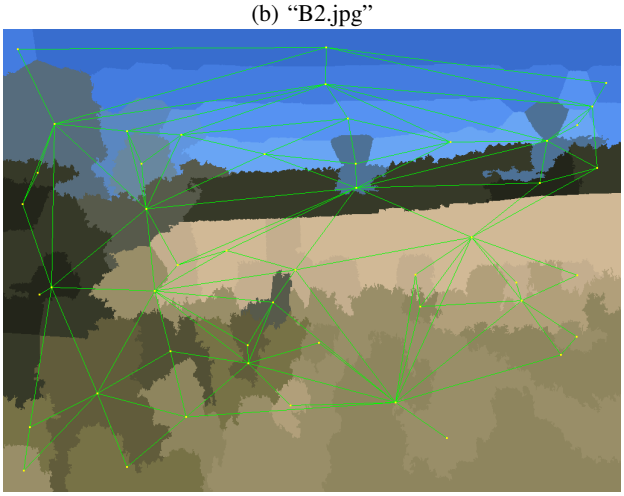
Fig. 13: N-cut segmentation for threshold = 0.01



(a) “B1.jpg”

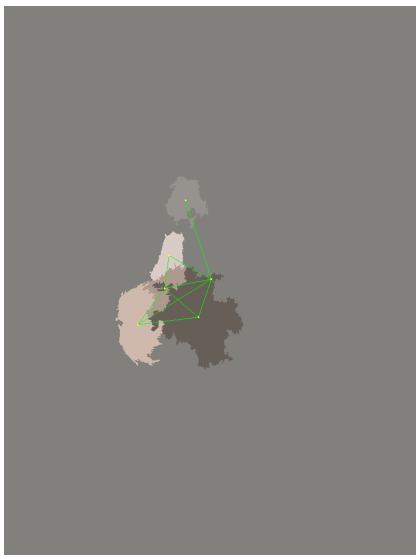


(b) “B2.jpg”



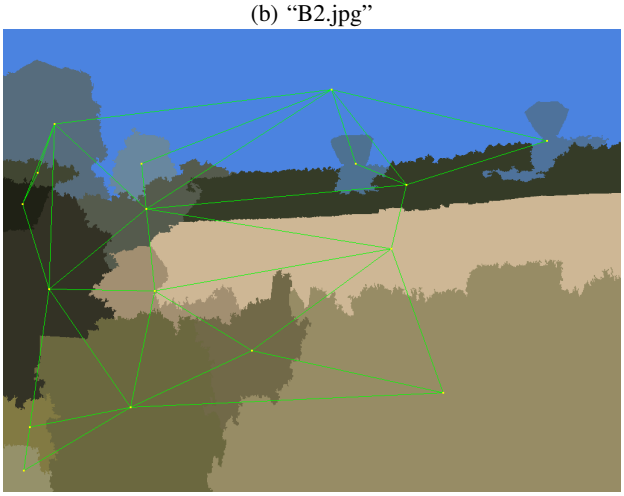
(c) “B4.jpg”

Fig. 14: RAG of mean shift segmentation for bandwidth = 16



(a) “B1.jpg”

(b) “B2.jpg”



(c) “B4.jpg”

Fig. 15: RAG of n-cut segmentation for threshold = 0.01