

# Advanced Image Processing Assignment #4

**Shikhar Vashishth**

M.Tech CSA - 13374

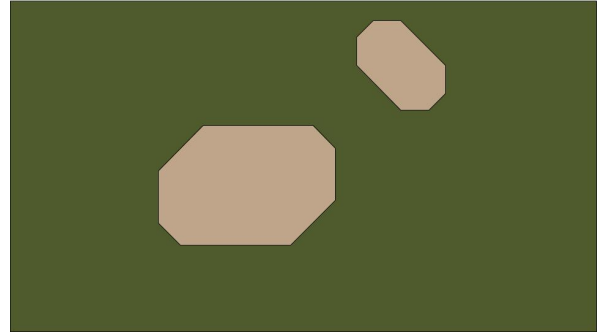
## Problem 1

### Part (a): Segmentation with a simple image

I chose the following simple image and its synthetic version for clustering.



(a) Simple Image



(b) Synthetic Image

Figure 1: Images for clustering

$$w_{ij} = e^{\frac{-\|\mathbf{F}(i) - \mathbf{F}(j)\|_2^2}{\sigma_I}} * \begin{cases} e^{\frac{-\|\mathbf{X}(i) - \mathbf{X}(j)\|_2^2}{\sigma_X}} & \text{if } \|\mathbf{X}(i) - \mathbf{X}(j)\|_2 < r \\ 0 & \text{otherwise,} \end{cases}$$

Feature Similarity term
Spatial Proximity term

**Parameters:**  $\sigma_I$ ,  $\sigma_X$ ,  $r$ , (*thresh*) threshold for the 2nd eigen vector.

### Results

Before applying N-cut algorithm, I first down-sampled the original image for reducing the computation. I have tried segmenting image with and without using spatial information. From the results below, we can see that although there is not much difference in the case of synthetic image but with the original image the segmentation results are better if we use the spatial information. Instead of directly using RGB values, I have used HSV values and have defined feature vector as:

$$F(i) = [v, v.s. \sin h, v.s. \cos h]$$

**Parameters used:**  $\sigma_I = 5, \sigma_X = 5, r = 0.5, thresh = 0$

In the figure below, the first column shows the down-sampled version of the original image used for segmentation. The other two columns show the two segments obtained from the algorithm. Overall, for both the images the results are quite satisfactory.

Without using spatial information

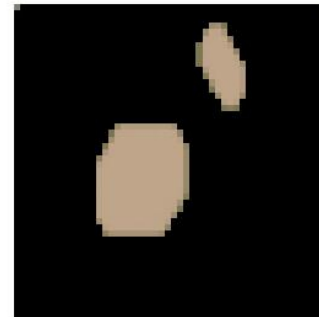
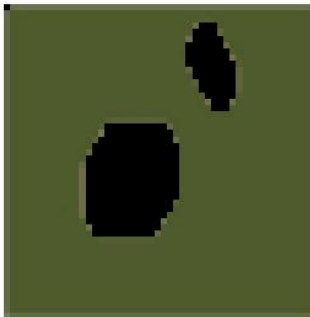
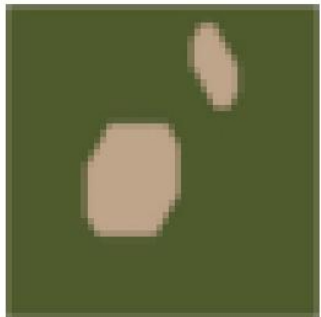
**Original Image**



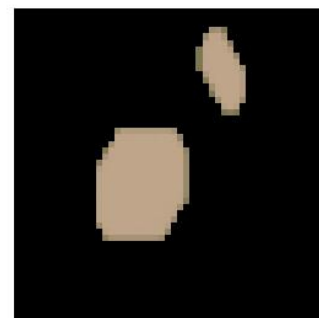
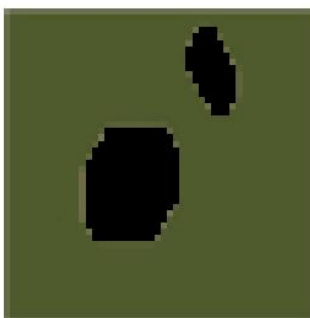
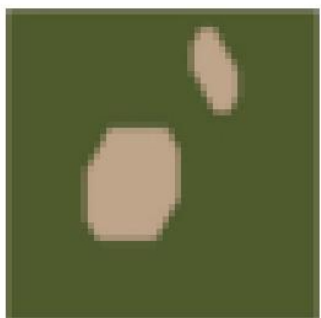
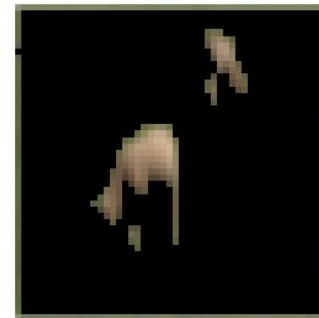
**Segment 1**



**Segment 2**



Using spatial information



**Observations** In the above cases, I have used the same value of  $\sigma$  for both feature and spatial similarity, but instead if we use a smaller value of  $\sigma_I$  that would imply giving more emphasis to feature similarity compared to spatial similarity. The following results shows that how that would affect our segmentation results. We can see that some part of animal got segmented in the wrong segment.

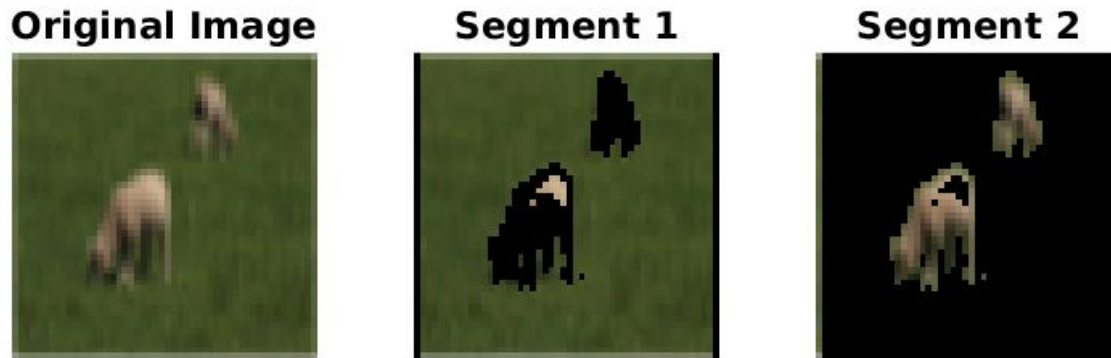


Figure 2: Segmentation with emphasis on feature similarity

## Part (b): Segmentation with a difficult image

I have used the following image for evaluating N-cut algorithm with different parameters:



Figure 3: Complex image for segmentation

### Observations

1. The spatial information is a must for complex images. In the previous part, we got decent results even without using spatial information of the pixels, but in the complex image ignoring spatial information gives unacceptable results.

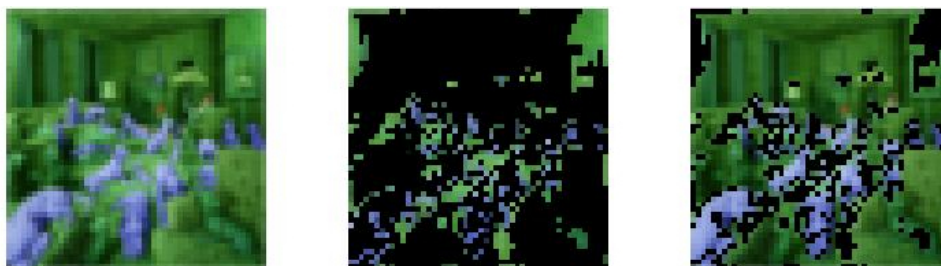
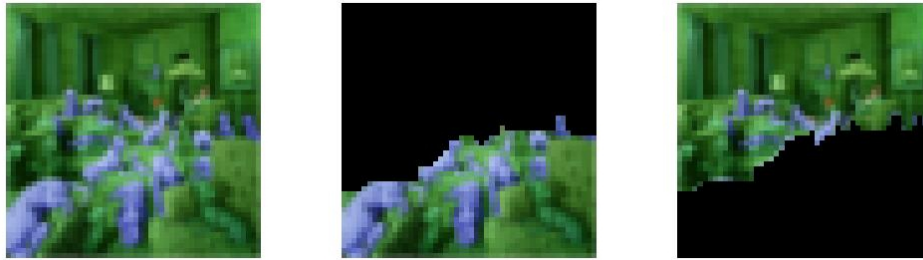


Figure 4: Results with  $\sigma_I = 0.5$  and  $\sigma_X = \infty$

We can see from the above results that because of ignoring spatial information we got discontinuity in the computed segments.

2. **Keeping  $\sigma_I$  very large and trying different values of  $\sigma_X$ .**

Small value of  $\sigma_X$  implies that we are giving more weightage to the spatial similarity between pixels than feature similarity between them.

Figure 5: Results with  $\sigma_I = 15$  and  $\sigma_X = 0.1$ Figure 6: Results with  $\sigma_I = 15$  and  $\sigma_X = 1$ Figure 7: Results with  $\sigma_I = 15$  and  $\sigma_X = 2$ Figure 8: Results with  $\sigma_I = 15$  and  $\sigma_X = 10$ 

The above results shows us that as we increase the value of  $\sigma_X$  the clusters obtained will be better, or in other words, lesser importance to spatial similarity than feature similarity improves the quality of results.





Figure 9: Results with  $\sigma_I = 15$  and  $\sigma_X = 15$

3. In the above experimentation, I also found out that while segmenting the image based on the second eigen vector, it is better to use the mean of the values of eigen vector rather than using 0 always. The following plot fortifies this observation. We can clearly see that the values of eigen vector are not symmetrically distributed around 0.

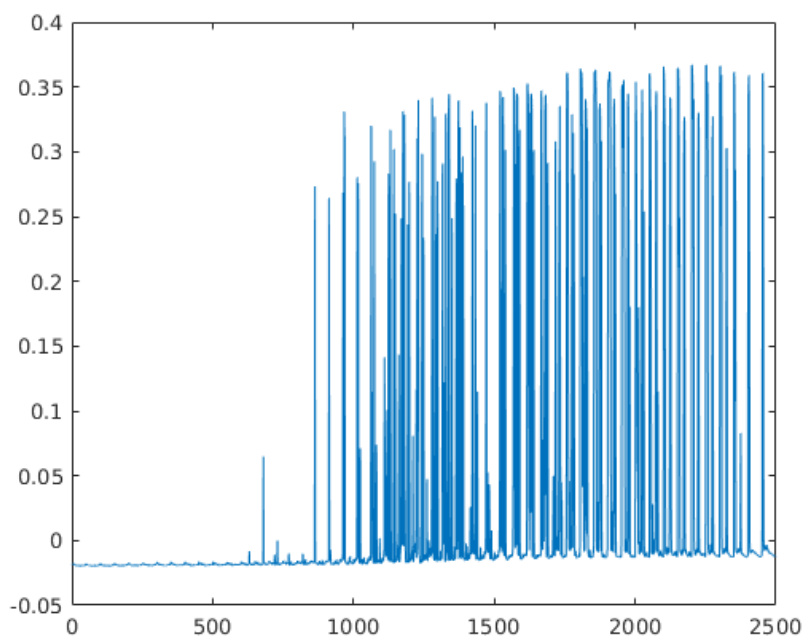


Figure 10: Distribution of values of 2nd eigen vector

The visual results also supports this observation. We can see that using mean gave us better segments as compared to using 0.



Figure 11: Segmenting eigen vector around 0



Figure 12: Segmenting eigen vector using the mean of its values

4. Keeping  $\sigma_X$  fixed at 1, I have compared the results of N-cut algorithm for different values of  $\sigma_I$ :

Figure 13: Results with  $\sigma_I = 1$  and  $\sigma_X = 1$ Figure 14: Results with  $\sigma_I = 2$  and  $\sigma_X = 1$



Figure 15: Results with  $\sigma_I = 5$  and  $\sigma_X = 1$ Figure 16: Results with  $\sigma_I = 10$  and  $\sigma_X = 1$ Figure 17: Results with  $\sigma_I = 15$  and  $\sigma_X = 1$ 

From all the observation, with  $\sigma_I = 5$  and  $\sigma_X = 1$  the segmentation results are the best as the blue dogs got completely separated by this choice. The green dogs can be segmented by again applying N-cut to the first segment.

**Part (c): Applying 2-cuts more than once**

For this part, I am using the same image which was used in the previous part. After applying 2-cut on the original image the results are the following:



Figure 18: Results after first segmentation

Results after second segmentation with different parameters:

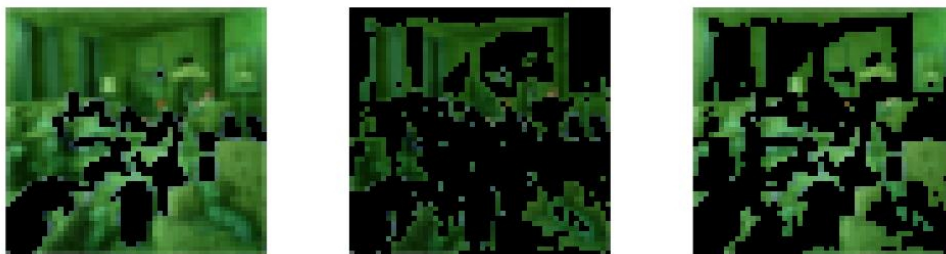


Figure 19: Results with  $\sigma_I = 20$  and  $\sigma_X = 30$  (Best result)

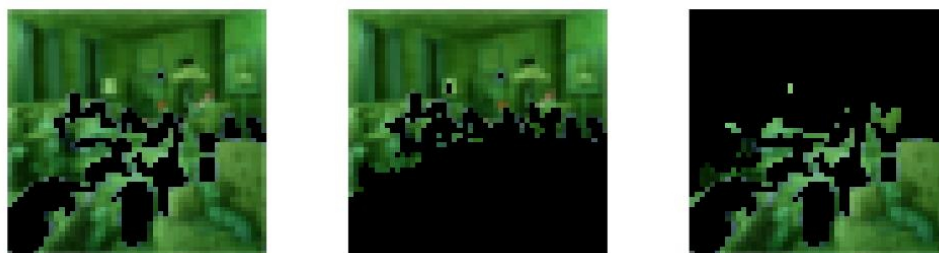


Figure 20: Results with  $\sigma_I = 10$  and  $\sigma_X = 1$

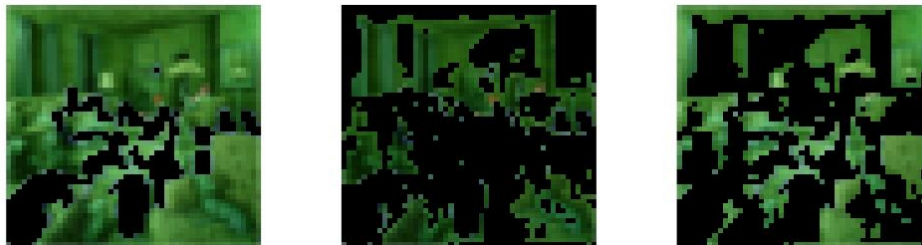


Figure 21: Results with  $\sigma_I = 50$  and  $\sigma_X = 100$

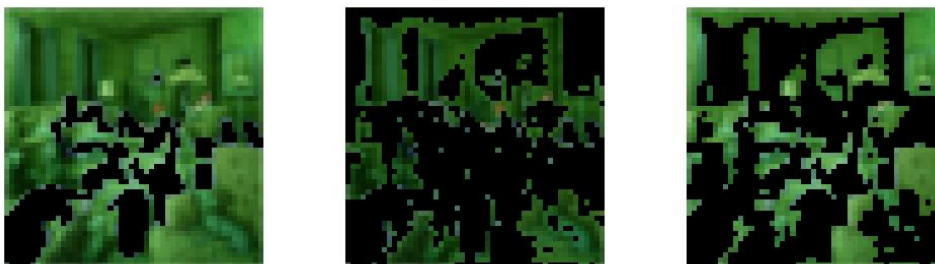


Figure 22: Results with  $\sigma_I = 50$  and  $\sigma_X = 25$

### Observations:

1. On applying 2-cut again to the first segment, I got the following results with different parameters. Among all of them  $\sigma_I = 20$  and  $\sigma_X = 30$  gave me the best results. As can be seen in the figure, we successfully segmented out green dogs from the image which are very difficult to segment out even by a human being in such a low resolution image.
2. The above results shows that too small value of  $\sigma_X$  compared to  $\sigma_I$  is a poor choice. Taking equal value for both of these parameters and taking  $\sigma_I$  to be smaller than  $\sigma_X$  gives almost the same results.

## Problem 2

### Part (1)

I chose **Efficient Graph-Based segmentation** by P. Felzenszwalb et.al. as another algorithm for image segmentation. The algorithm is based on representing an image as a graph, where pixels are represented by vertices and an edge exists between every pair of pixels with dissimilarity between them as its weight. The algorithm segments the graph into distinct components such that the elements within a cluster are similar and elements in different components are dissimilar.

### Basic algorithm steps

1. Put each vertex in its own component and sort all the edges by their weight
2. For each edge
  - (a) If the edge is between vertices belonging to two different components and the edge weight is less than the threshold  $k$  then the components are merged otherwise nothing is done.
3. Step 2 is repeated multiple times until desired clusters are obtained.

### Parameters to the algorithm

$\sigma$  - For Gaussian smoothing to reduce the noise before applying the algorithm

$k$  - threshold value for clustering

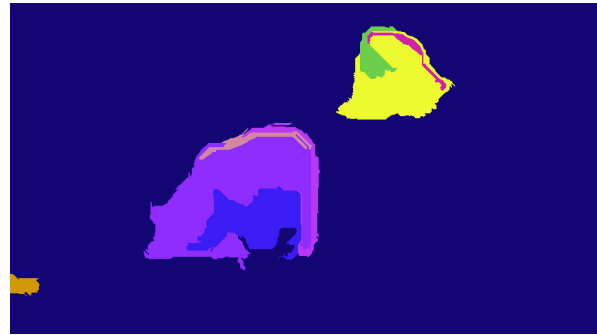
$min$  - minimum number of vertices in any cluster

### Observations

1. On using the algorithm with the images used in the first part of the problem 1, I got the following results. The algorithm properly clustered both the original and synthetic images successfully. Parameters used are  $\sigma = 5$ ,  $k = 100$ ,  $min = 900$  for clustering the first image and  $\sigma = 5$ ,  $k = 200$ ,  $min = 5000$  for the second one. The change in parameter values shows that the algorithm is very sensitive to its parameters.
2. Clustering results with image used in part (b). Parameters used:  $\sigma = 2$ ,  $k = 100$ ,  $min = 500$ . As we can see from the results in Figure 25 that all the dogs are successfully segmented by the new algorithm.

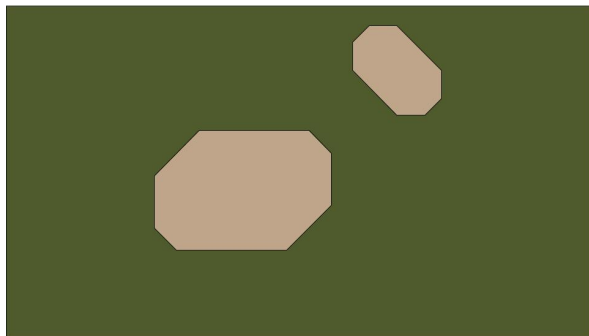


(a) Image used in part (a)

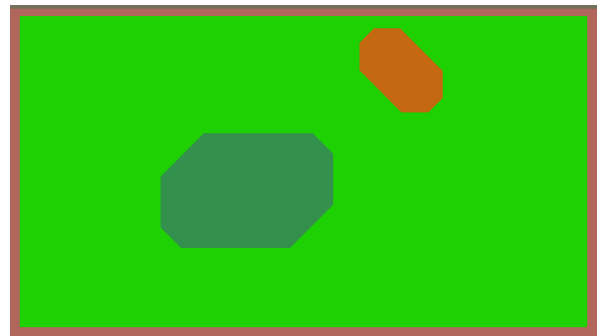


(b) Clustering output (# Clusters = 8)

Figure 23: Clustering Original image used in part (a)



(a) Image used in part (a)



(b) Clustering output (# Clusters = 5)

Figure 24: Clustering synthetic image used in part (a)



(a) Image used in part (b)



(b) Clustering output (# Clusters = 81)

Figure 25: Clustering complex image used in part (b)

3. The algorithm is very sensitive to its parameters. I first checked how the algorithm's output varies as we increase the value of  $\sigma$ . We can clearly see from the results below



that the large value of  $\sigma$  blurs the boundary between segments and thus makes the algorithm cluster them as a single segment. For the image below, the palm tree got confused with ocean on increasing  $\sigma$ .

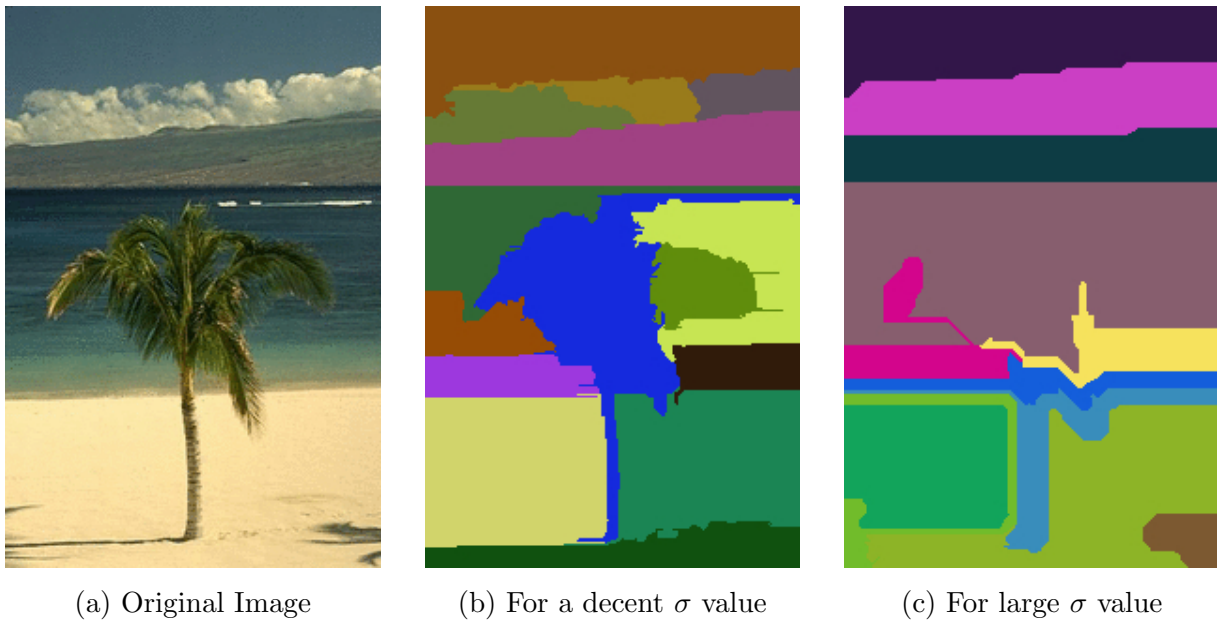


Figure 26: Clustering results with change in  $\sigma$

- The results below shows how algorithm behaves as we change the value of the parameter  $k$ . We can see that as we increase  $k$  the clustering becomes more and more aggressive.

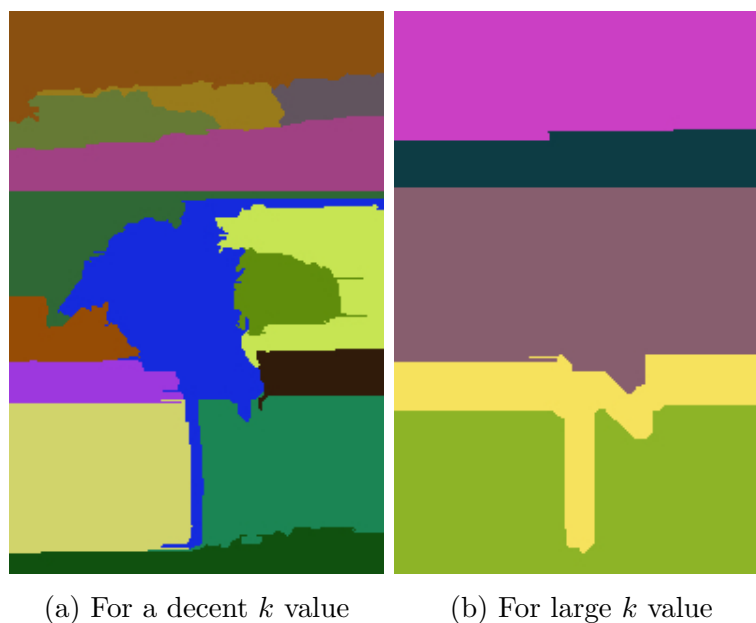


Figure 27: Clustering results with change in  $k$



5. The results below shows how algorithm behaves as we change the value of the parameter  $min$ . We can see that as we decrease the value of  $min$  more smaller clusters start getting identified by the algorithm.



(a) For a decent  $min$  value

(b) For a smaller  $min$  value

Figure 28: Clustering results with change in  $min$

**Comparison with N-cut** Compared to N-cut algorithm, ‘Efficient Graph-Based segmentation’ is very very fast. Even for quite high resolution images the segmentation was done within few seconds. On the other hand, N-cut algorithm’s space requirement is so huge that it cannot be used for high resolution images. It requests for space in thousands of gigabytes for such images. In terms of performance also the new algorithm is superior to N-cut. It can compute more complex segments as compared to N-cut.