Data Mining

**Assignment 5**

horizontal line

Image Segmentation

Team Members:

Yassmin Barakat 86

Youssef Zook 90

## **Lab Objective:**

* Learn how to apply image segmentation
* Use clustering techniques to differentiate between image objects

## **Problem Statement**

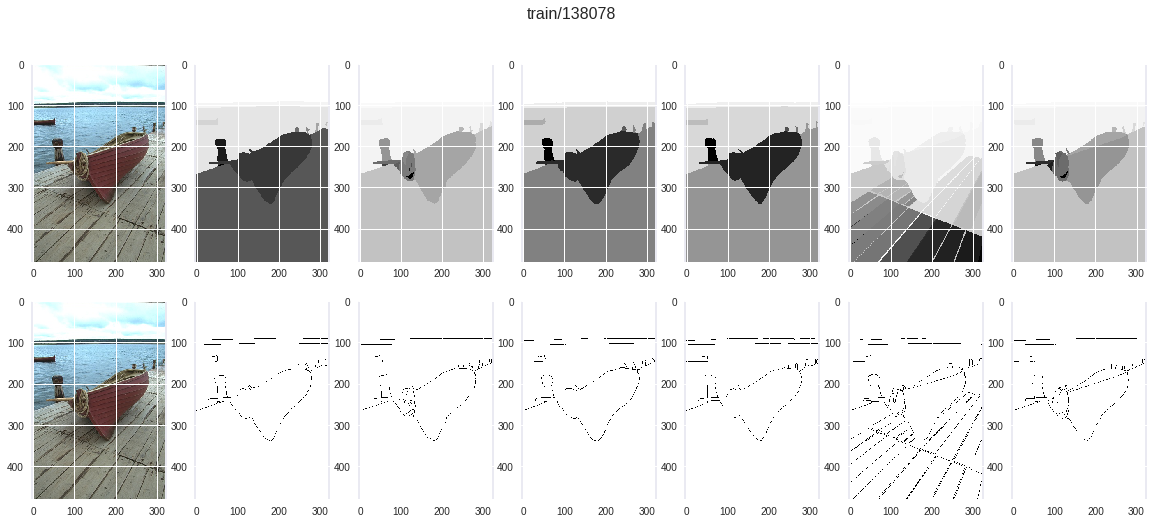
We intend to perform image segmentation. Image segmentation means that we can group similar pixels together and give these grouped pixels the same label. The grouping problem is a clustering problem. We want to study the use of K-means and Normalized - Cut methods on the Berkeley Segmentation Benchmark. Below we will show the needed steps to achieve the goal of the assignment.

[**source code**](https://colab.research.google.com/drive/1hLitxroXFXz0fy11Dl6Sa-drxx3GP1TD)

### **Download the dataset and understand the format**

* The jpg images is in images folder and groundTruth is in the groundTruth folder
* Mat Images type is dictionary
* Mat images keys are ['\_\_header\_\_', '\_\_version\_\_', '\_\_globals\_\_', 'groundTruth']
* Type for Mat Images [gorundTruth] is ‘numpy.ndarray’
* Mat images [groundTruth] shape is (1,6)
* Each mat image has k images for different ks. in the following example in visualization : k = 6, we will plot the k images as columns

### **Visualize the image and the ground truth segmentation**



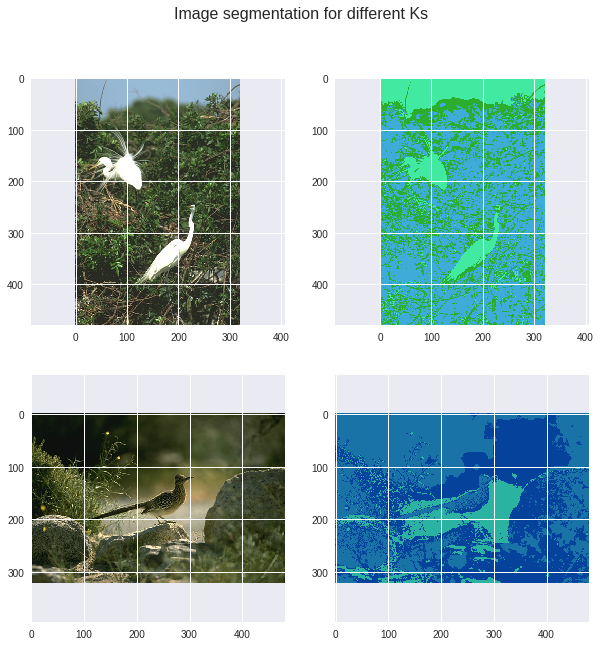
### **Segmentation using K-means**

* We will change the K of the K-means algorithm between {3,5,7,9,11} clusters. You will produce different segmentations and save them as colored images. Every color represents a certain group (cluster) of pixels.
* We will evaluate the result segmentation using F-measure, Conditional Entropy. for image I with M available ground-truth segmentations. For a clustering of Kclusters we will report our measures M times and the average of the M trials as well then Report average per dataset as well

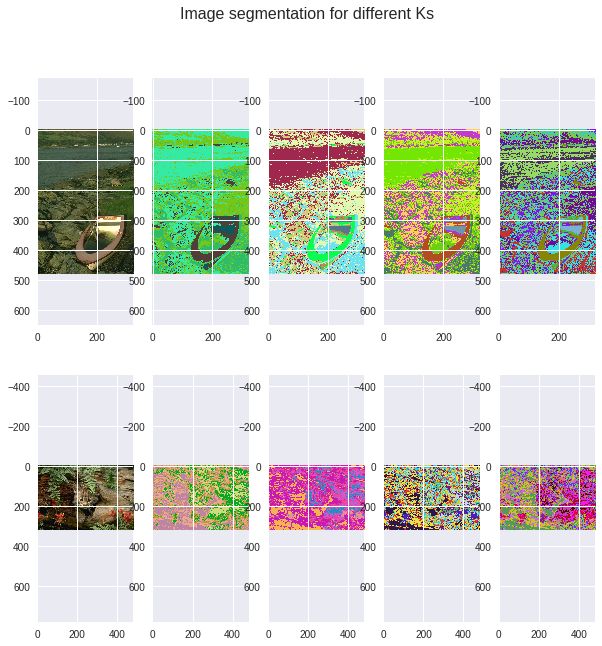
|  |  |  |
| --- | --- | --- |
| K | F-measure | Conditional Entropy |
| 3 | 0.5336 | 1.2686 |
| 5 | 0.4542 | 1.1506 |
| 7 | 0.3942 | 1.0892 |
| 9 | 0.3498 | 1.0455 |
| 11 | 0.3164 | 1.0102 |

We selected bad and good results based on f-measure, as if it greater than 0.5 it will be good and otherwise.

* Good results (for K = 3)



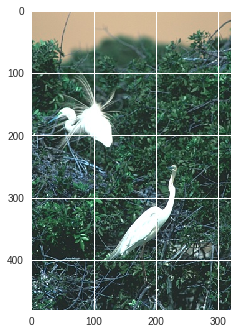
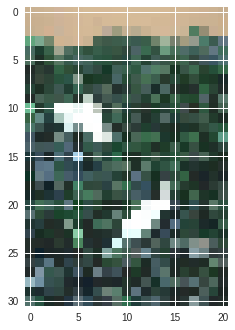
* Bad results (for K = 5, 7, 9, 11)



### **Normalized-Cut Segmentation**

* Kernel used is RBF with gamma = {1,,10} and k={3,5,7,9,11}
* To construct similarity graph - as matrix, we will compare each pixel in the photo with other pixels. If the image dimensions = 481\*321 then the similarity matrix dimensions = 154401\*154401 and this will need about 192 GB of RAM which is not practical. So, we resized our photos dimensions by dividing each of Height and Width by 4 and this needs about 12GB of RAM which is more practical.

However, due to some limitations in time and resources we resized it more by dividing the h and w of the original image by 16. This will produce bad similarity matrices but they are just used as a proof of concept for our work as the high required resources is not available for us. The following images show the difference of the quality before and after resizing images:

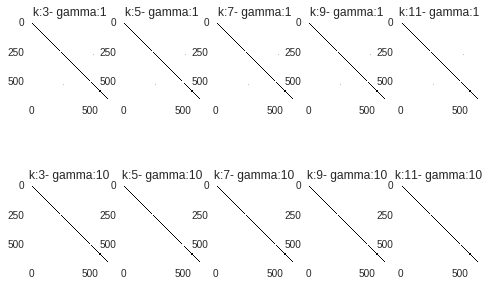
* Code:

model = SpectralClustering(n\_clusters=ks[i], affinity='rbf', gamma=gamma, random\_state=0, n\_jobs=-1).fit(X.reshape(w\*h,d))

similarity\_matrix = model.affinity\_matrix\_

SpectralClustering is used from SKLearn library. Parameters passes are the gamma from {1, 10} and number of clusters for differens ks and using rbf kernel

* Output similarity matrices:

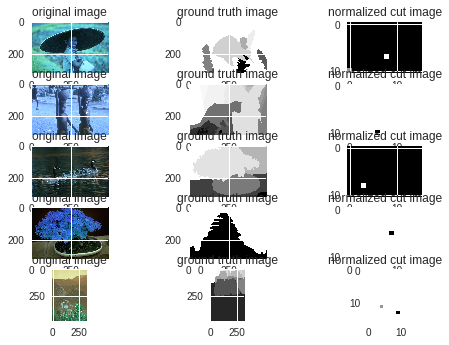


|  |  |  |
| --- | --- | --- |
| K | F-measure | Conditional Entropy |
| 3 | 0.00123 | 0.1 |
| 5 | 0.00122 | 0.2 |
| 7 | 0.00082 | 0.1 |
| 9 | 0.00082 | 0.05 |
| 11 | 0.000821 | 0.0 |

Note: the entropy is not good because as mentioned above the image is resized to a very small dimensions which give bad results, however we print this just as a proof of concept to our work.

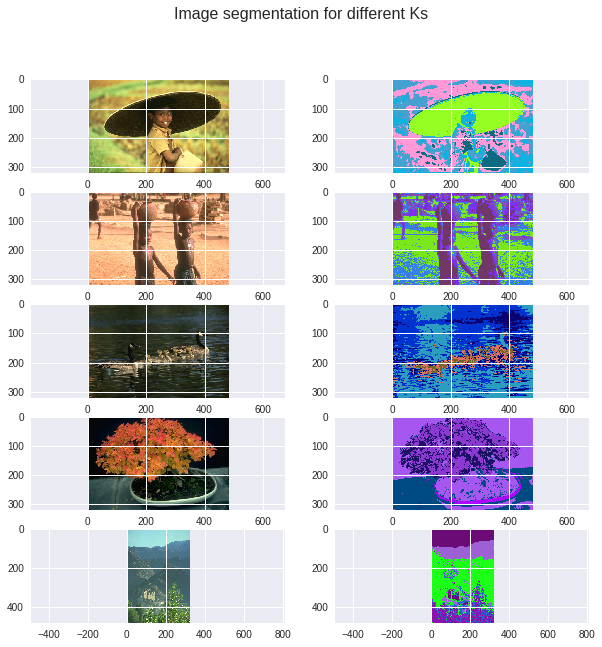
### **Big Picture**

* Normalized Cut



**F-meaure** = 0.0075

* K-Means



for K = 5 F\_measure = 0.0117

for K = 5 cond\_enntropy = 0.0253

* F-measure is better in k-Means than normalized cut
* In K-means, k=5 segmentation differs a lot, as f\_meassure is more less than 0.5