Alexandria University- Faculty of Engineering

Computer and Systems Engineering Department

Pattern Recognition

Lab 2

Image Segmentation

Name	ID	Email
Ahmed Gamal Mahmoud	18010083	ahmed.gamal5551.ag@gmail.com
Marwan Mohamed Saad	18011736	marwangabal99@gmail.com
Shehab Mohamed Saad	18010857	shehab.mohamed2104@gmail.com

colab link

Downloading dataset and understanding the format:

- We uploaded the dataset on google drive mount the dataset folder to our colab
- Every image in the dataset is in the format of jpg with dimensions (321 * 481 * 3) or

  (481 * 321 * 3) so reading the image in RGB and show it using cv2.

- The ground truth from .mat files we get from 4 to 8 ground truths for each image

  And each one has a segmentation 2D array and boundaries 2D array.

  Functions:

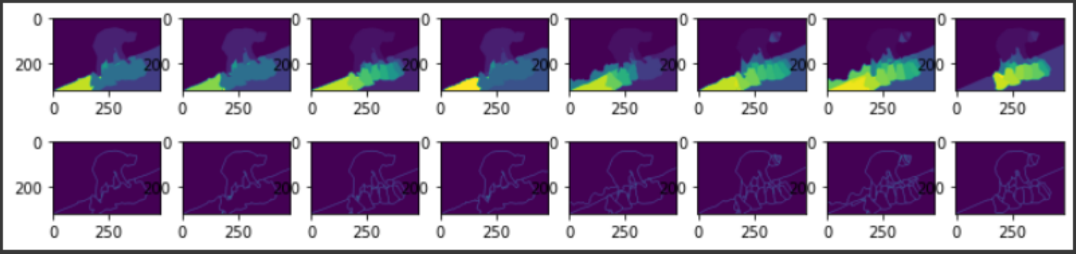
  - read_images(directory)
  - read_groundTruth(directory)

Visualization of images and ground truth:

- Each image is shown by cv2
- Each ground truth has two 2D arrays Segmentation and Boundaries, we show them using “image.fromarray()”

```
[37] view_image(trainImg[131])
```

```
[38] view_segmentation(groundTruthList[131])
```



Functions:

- view_image(image)
- view_segmentation(segment) ⇒ to show all segmentations and boundaries of the ground truth of image
- view_single_sigmentation(array)
- convert_to_seg(one_d_array, shape) ⇒ convert 2D array to 1D array with specific shape

Helpful Functions to Kmeans:

- save_clustered_colored_pic(centroids, labels, shape, filename) ⇒ save coloured image after Kmeans
- plot_clustered_colored_pic(centroids, labels, shape) ⇒ plot coloured image after Kmeans.

K-means clustering :

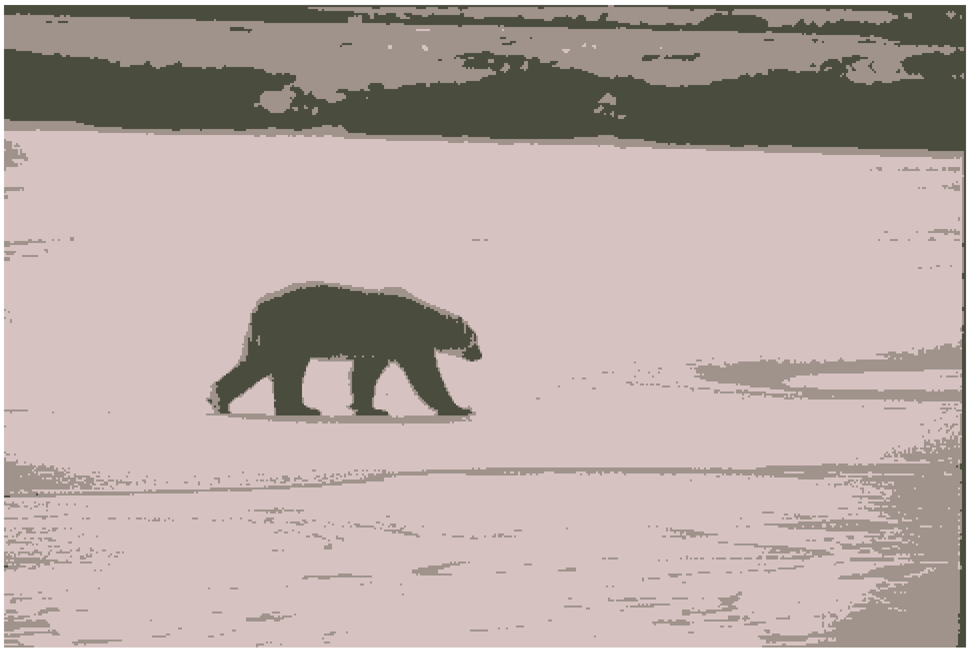
1. Functions :
    a. **assignToCluster(pixel, centroids)** -> this function is responsible for calculating the distance (euclidean distance) between the current pixel sent to the function and the centroids to return the index of the cluster that this pixel must be assigned to.

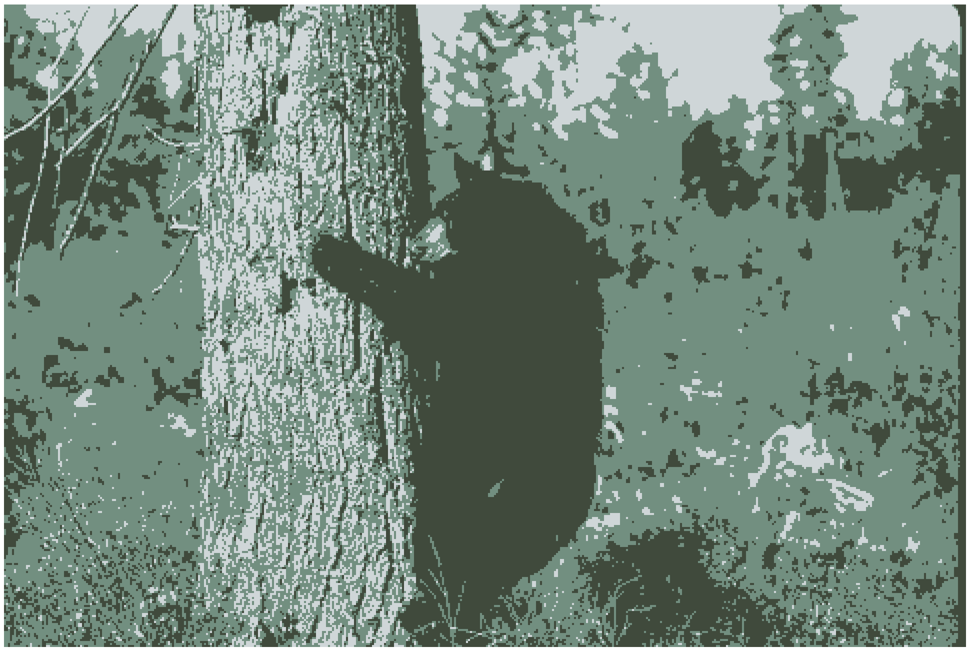
    b. **updateCentroids(clusters)** -> this function calculates the new centroid by looping on the list of pixels of each cluster and dividing their sum with the number of pixels in this cluster to be at this point the new centroid.

    c. **generateCentroids** -> this function is responsible for generating a random index to be accessed in the array for setting the starting centroids, this is used in random start condition as well.

    d. **kMeans** -> this function run the main algorithm and by the end of this function it returns the clusters with pixels assigned to each cluster and the finals centroids to each cluster

2. Different values for K: [link of all pics]
   a. K = 3:



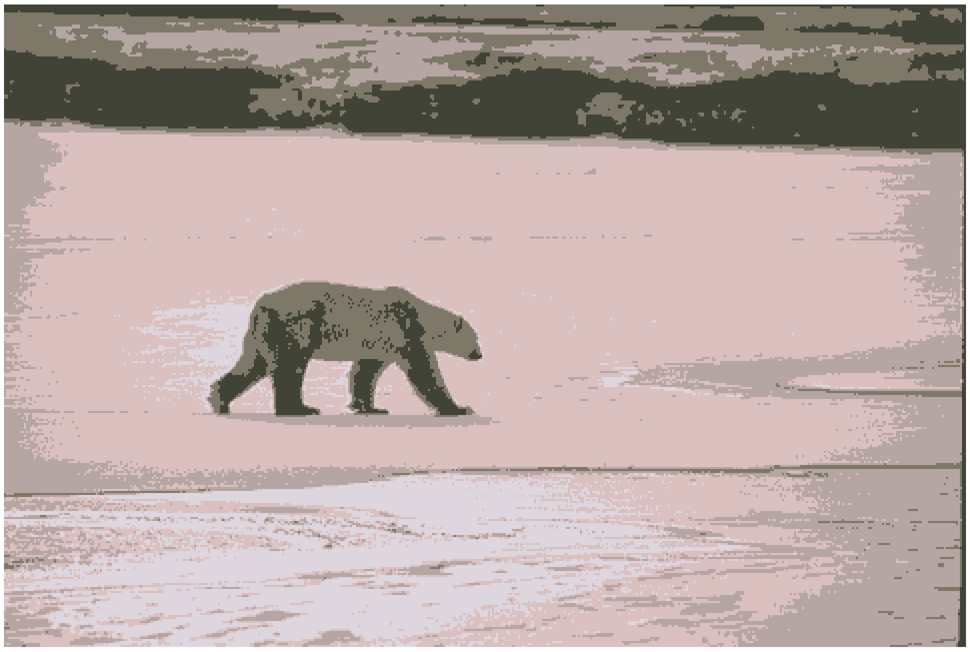


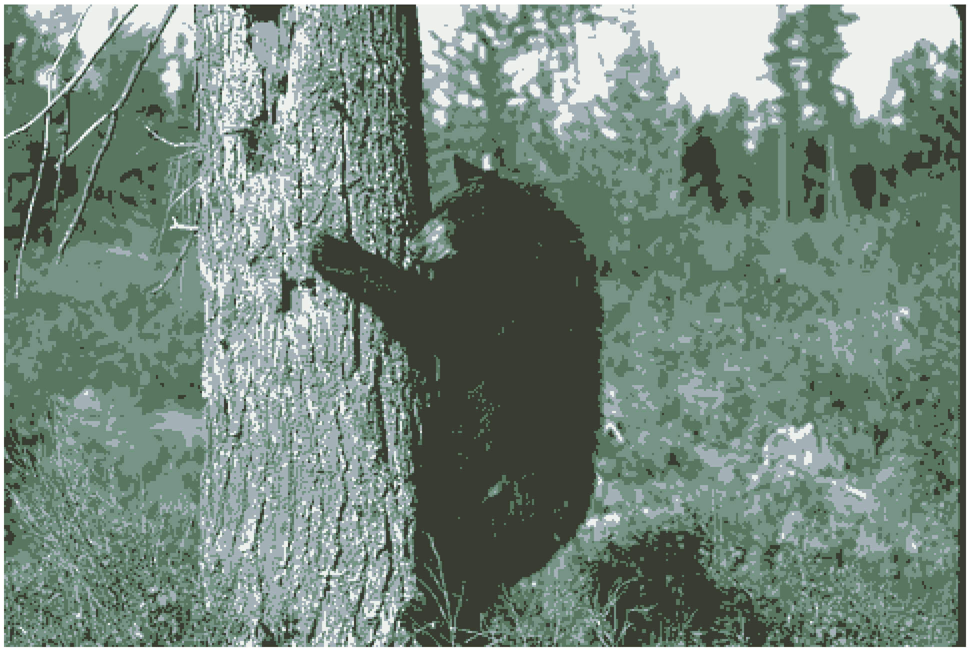
```
F_measure : [0.055978936610462694, 0.03942545073257343,
0.0024435587988665274, 0.07264711357669414, 0.0017976218195010804]
```

```
Avg : 0.034458536307619574
```

```
cond_entropy : [0.331913440770064, 0.3521251358642472,
0.4351051417421084, 0.41285169373298247, 0.43748263922243386]

Avg :0.393895610266367
```

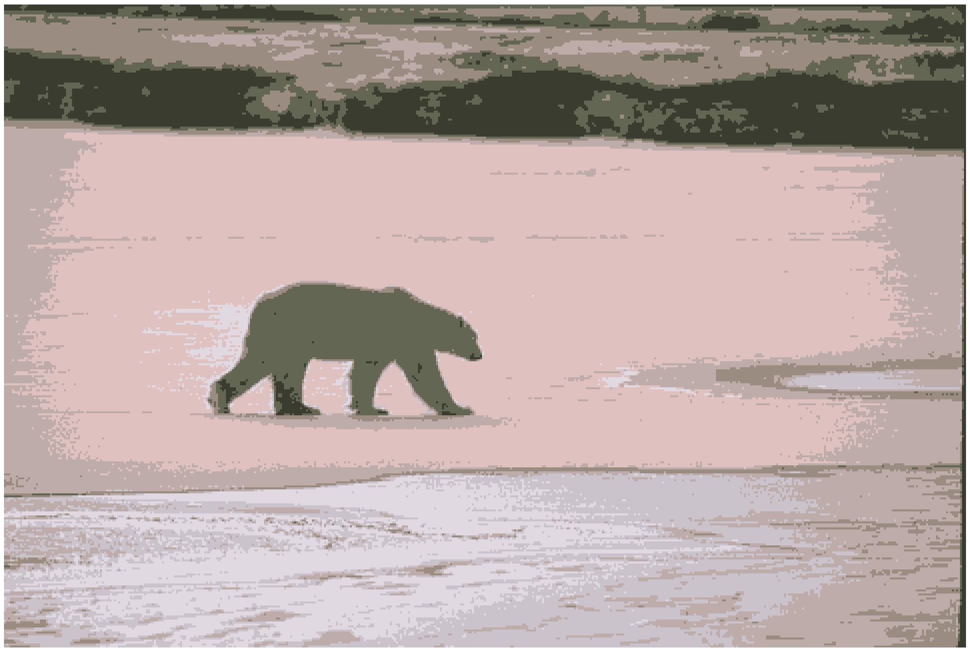
b. K = 5:

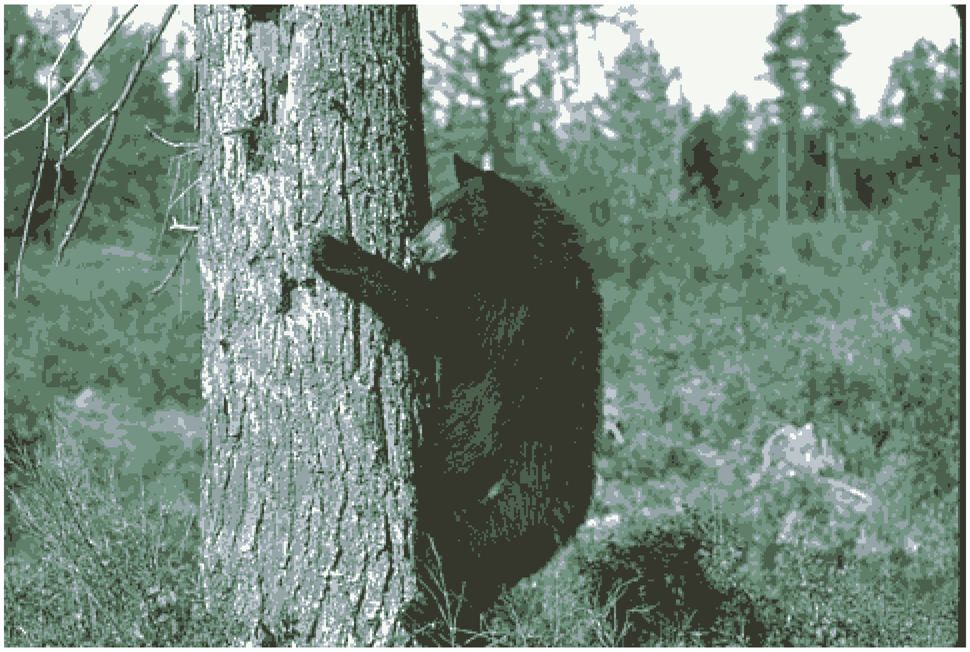




```
F_measure : [0.015023909252915652, 0.010478642890284173,
0.1075699677562719, 0.16568837573164644, 0.0042403097293638]
Avg : 0.0606002410720964


cond_entropy : [0.2816467279171293, 0.3043151414472042,
0.38293073104214737, 0.3601868993641797, 0.3829453945735835]
Avg : 0.3424049788688488
```

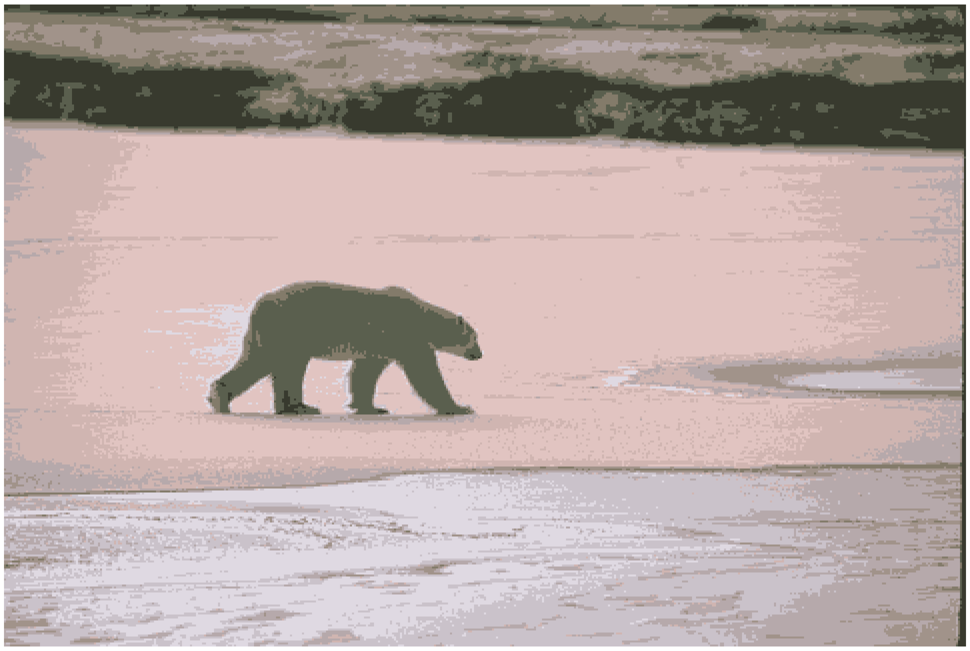
c. K = 7:

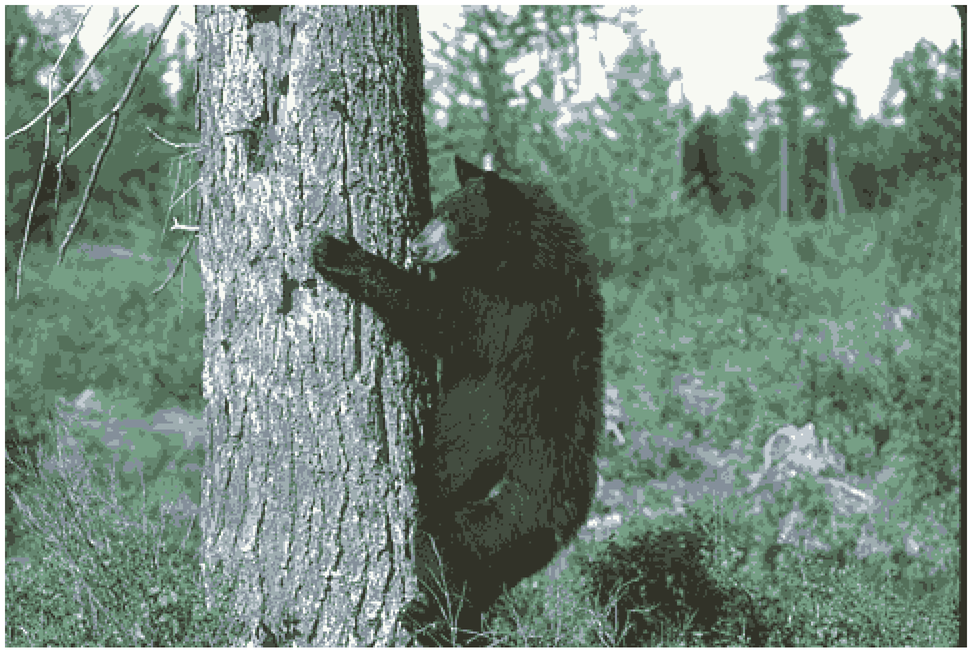




```
F_measure : [0.08550478960057355, 0.005729573578674560 5,
0.02457871722961395 3, 0.05255476140044795,
0.003763303077490467 6]

Avg : 0.03442622897736009


cond_entropy : [0.1770049881712876, 0.19678643536588725,
0.27510449382796837, 0.2548346286083342, 0.2723112446757974]
Avg : 0.23520835812985497
```

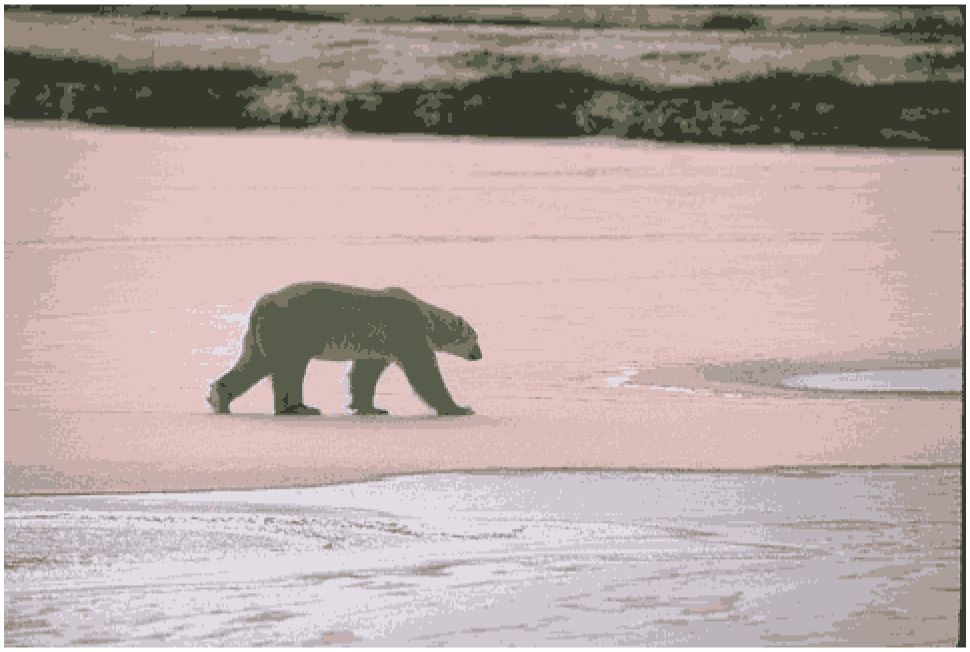
d. K = 9



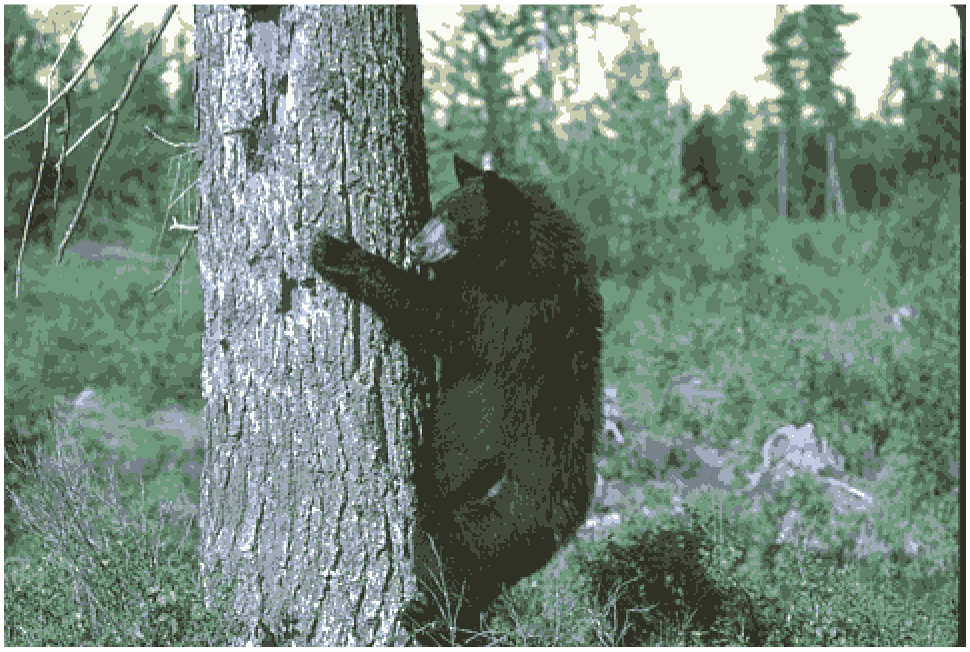


```
F_measure : [0.03009238657495649, 0.006874172489071815,
0.00924496447275736, 0.05029791293691038, 0.00551625015274811]

Avg : 0.020405137325288834

cond_entropy : [0.15251851351092788, 0.17167529176737492,
0.24856193173348012, 0.22827399282643263, 0.24316333385043035]

Avg : 0.20883861273772916

F_measure : [0.1391445877503677, 0.018408138702360872,
0.180074895795193, 0.17970258297894665, 0.028463308856557547]

Avg : 0.10915870281668516


cond_entropy : [0.4646057174607904, 0.8391907450409402,
0.412572461634212, 0.41303652916958095, 0.6307371573471492]

Avg : 0.5520285221305345
```

e. K = 11



```
F_measure : [0.10993520484306529, 0.004448728180146626,
0.022344768872259443, 0.01100262632267883,
0.018857465006236755]

Avg : 0.03331775864487739


cond_entropy : [0.12870283013549752, 0.14807460503000783,
0.22443654999691612, 0.204400741566176, 0.2185521517236978]

Avg : 0.18483337569045905
```



```
F_measure : [0.18495673222611994, 0.048097505030044216,
0.17448322891263005, 0.17477964431167836, 0.14555607888196812]

Avg : 0.14557463787248812

cond_entropy : [0.4493864674345097, 0.8129523940127532,
0.39821866489645696, 0.3991972810556999, 0.613133631629338]

Avg : 0.5345776878057517
```

3.Evaluation:

- Functions:
  - F_measure for each image with all ground truths using ⇒
    - `calc_F_measure(true, pred)`
  - Conditional_entropy for each image with all ground truths using ⇒

        - calc_entropy(true, pred)
        - calculate(matrix) using it inside calc_entropy to calculate it for only one ground truth
- Examples on images (like above)
- Average of data set:
    - F_measure K=3 : 0.14896335403379882
    - Conditional_entropy k=3 :0.5558454686749122

    - F_measure K=5 : 0.107965610545471
    - Conditional_entropy k=5 :0.47229517195741605

    - F_measure K=7 : 0.09554371541713111
    - Conditional_entropy k=7 : 0.42081322565779333

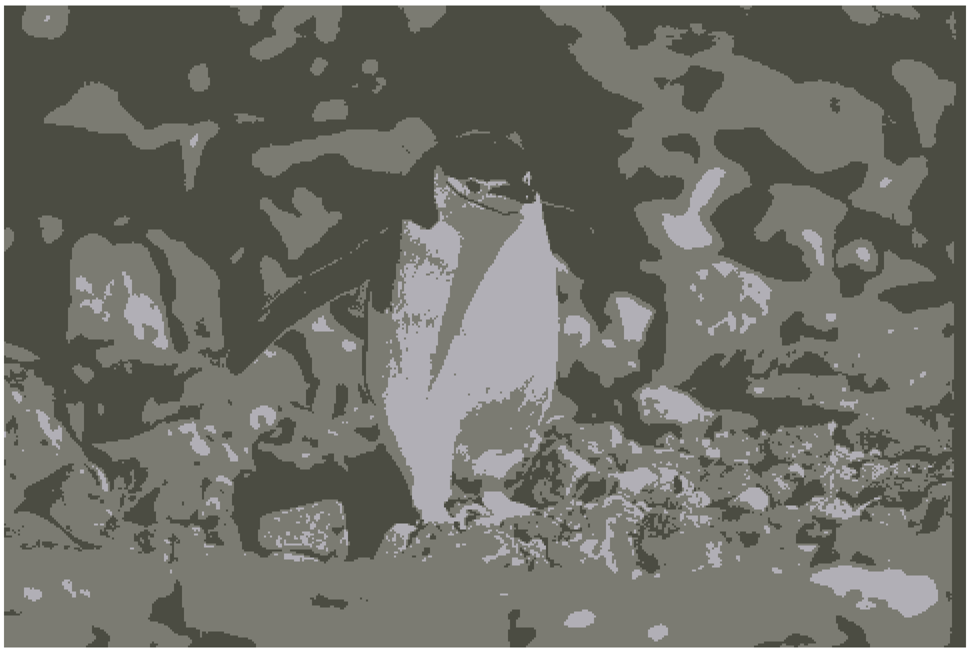
    - F_measure K=9 : 0.1649822116470378
    - Conditional_entropy k=9 : 0.3931504672347952

    - F_measure K=11 : 0.07699169876861764
    - Conditional_entropy k=11 : 0.18347021804290442

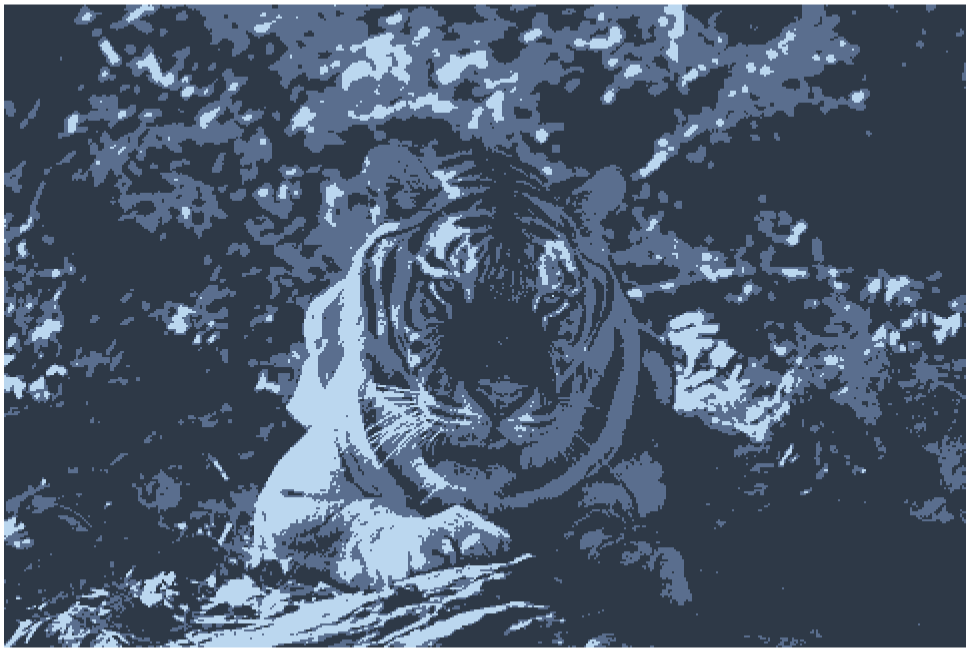
4. Good and bad results

Good results With k = 3

F-measure : 0.4758089660107171

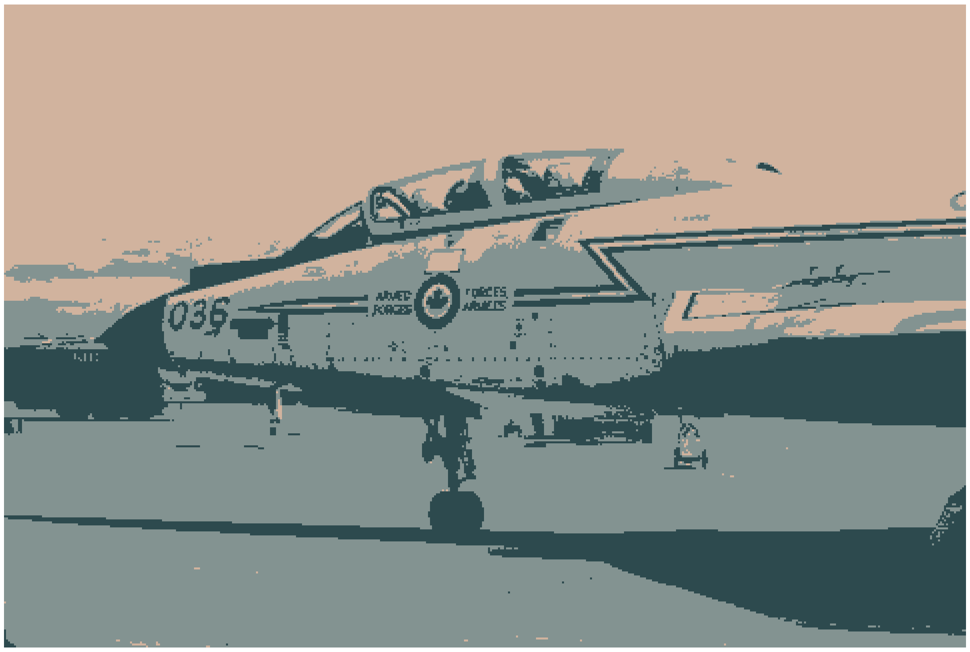


`F-measure : 0.523036143428053`

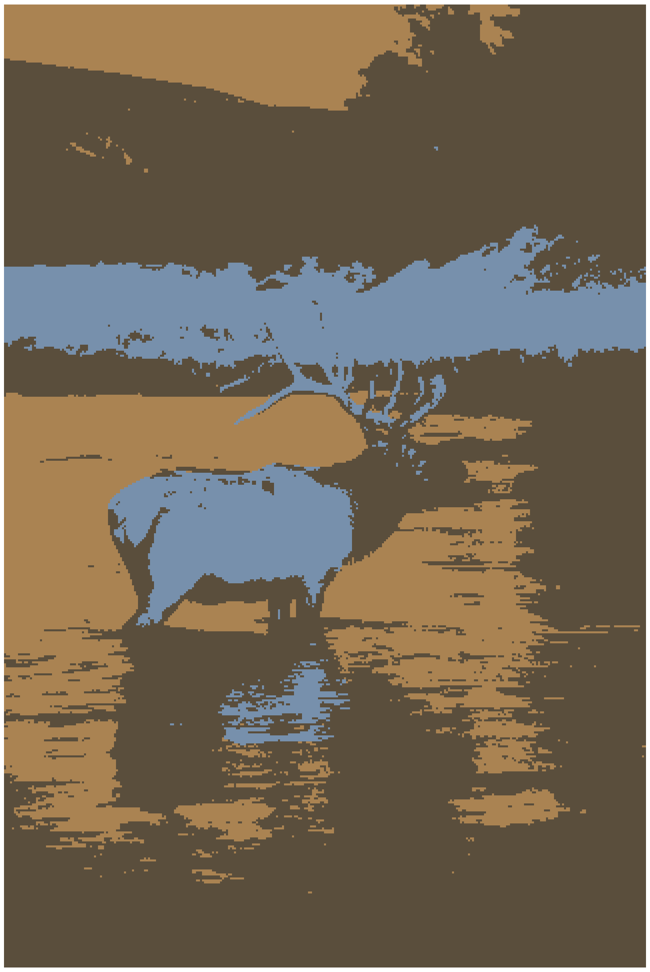


Bad results :

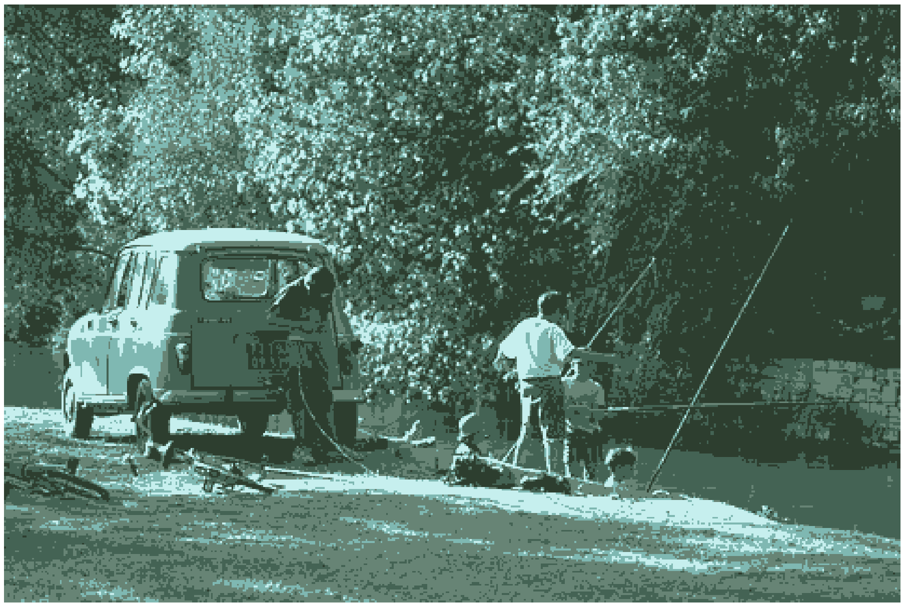
F-measure : `0.0006416522926151907`



F-measure : 0.0028738596714602545

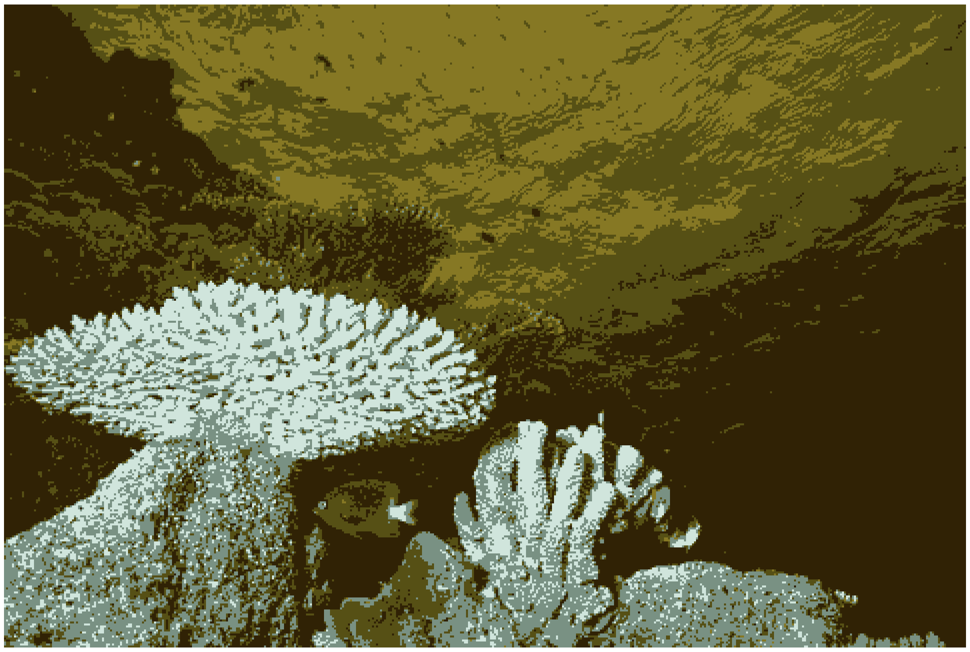


```
With k = 5

Best Results :

    F_measure : 0.2827962457307048
```



Bad results:

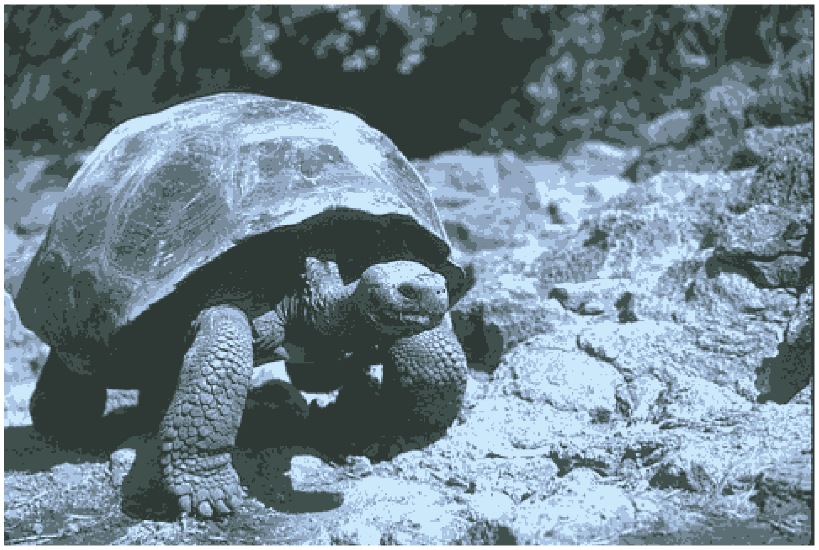
```
F_measure : 0.005530685157229402
```



With k = 7

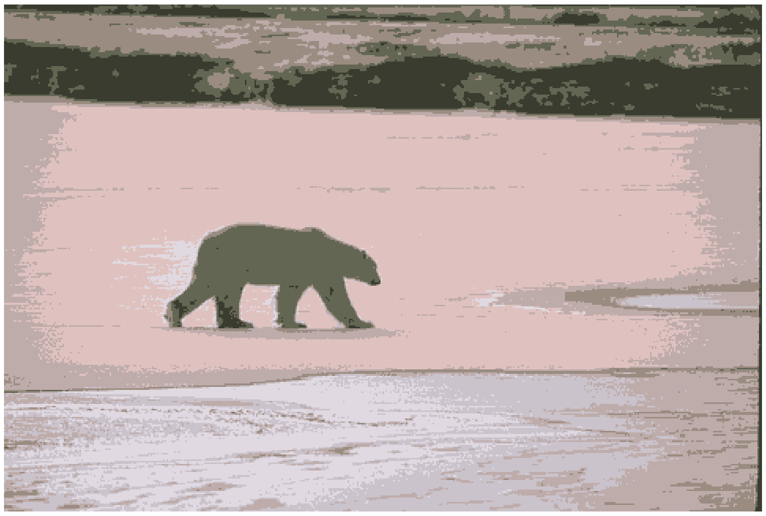
Best Results :

```
F_measure : 0.22725002307335948
```



Bad results :

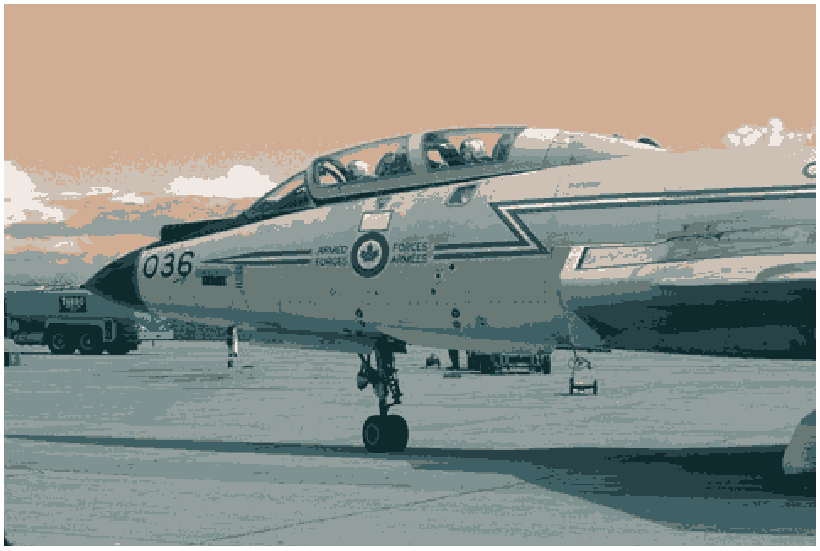
F_measure : 0.03442622897736009



With k = 9

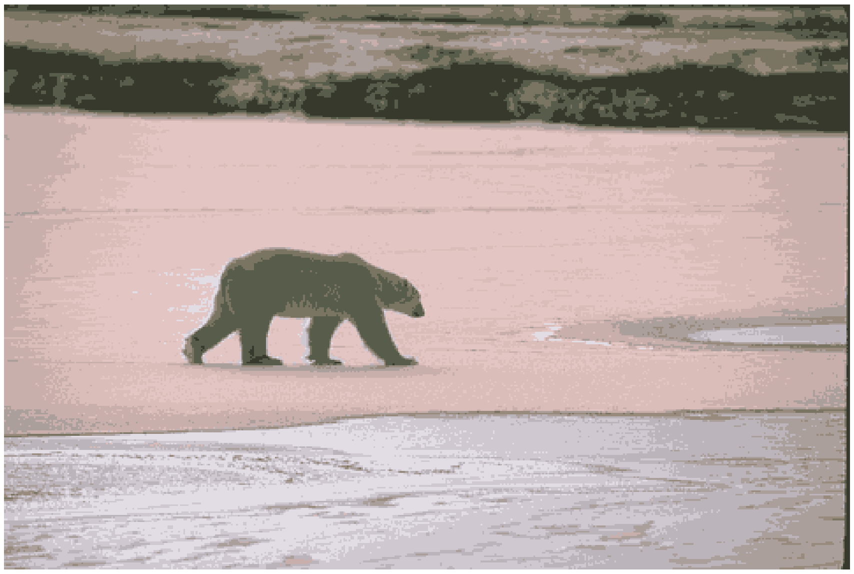
Best Results :

F_measure : 0.3599914411009442



Bad results :

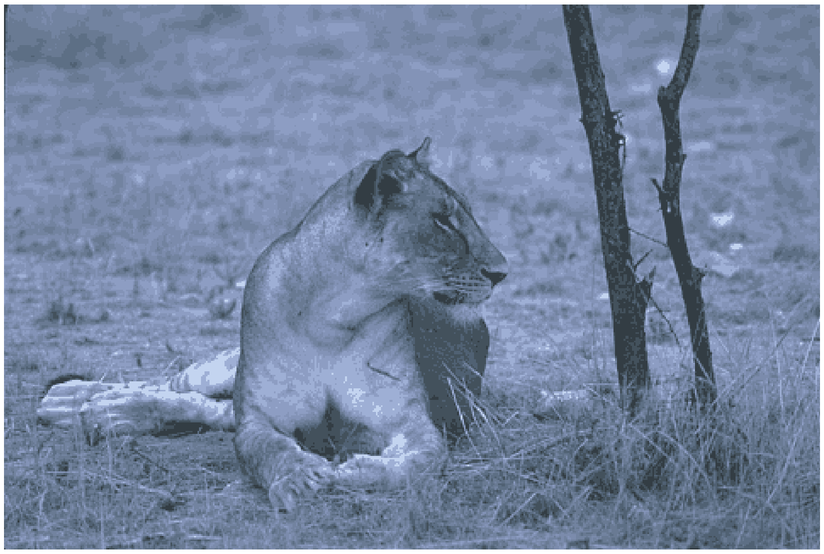
F_measure : 0.020405137325288834



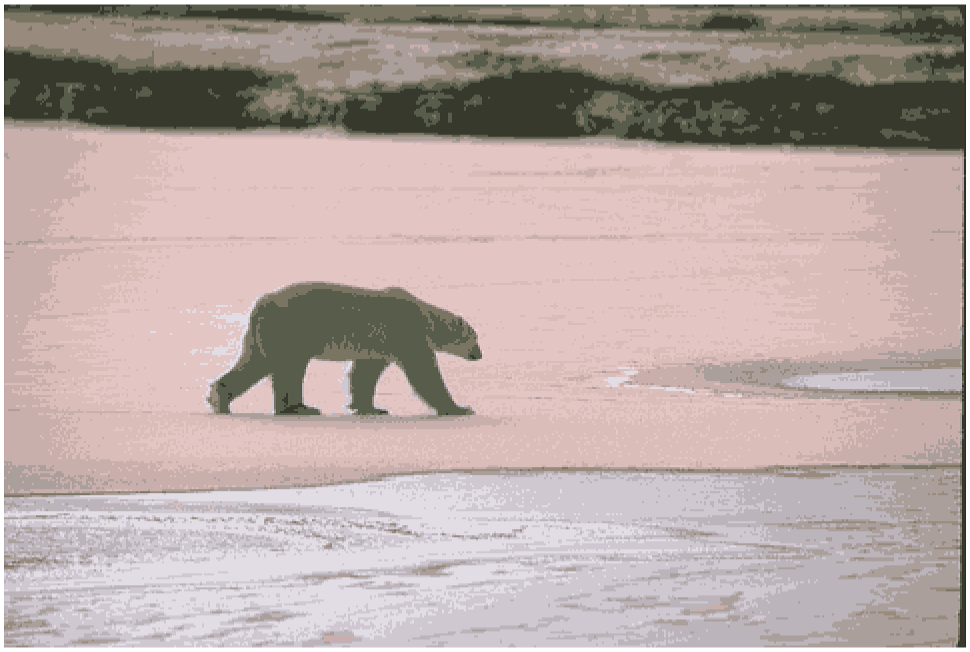
With k = 11

Best Results :

F_measure : 0.19391086543907352



```
Bad results:

        F_measure : 0.03331775864487739
```

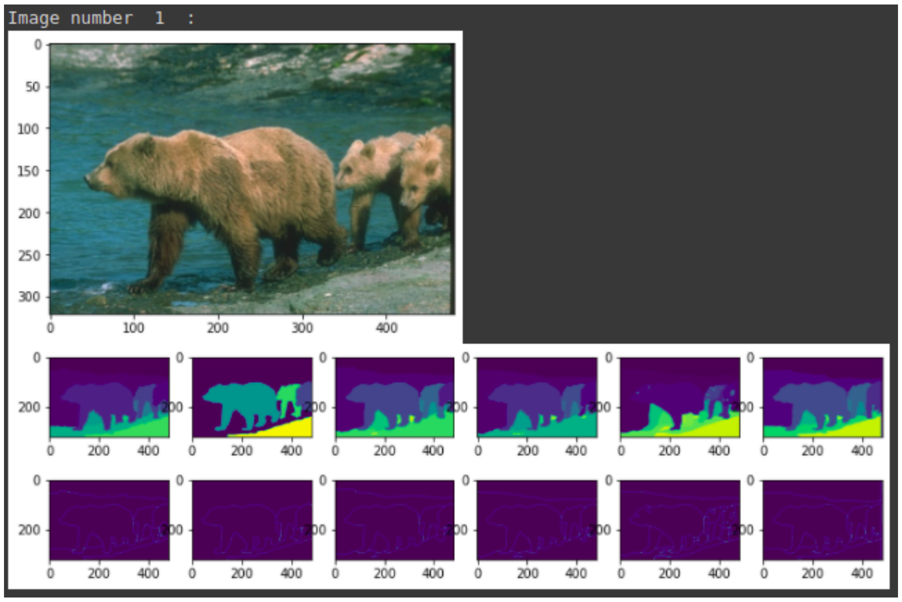


Big Picture:

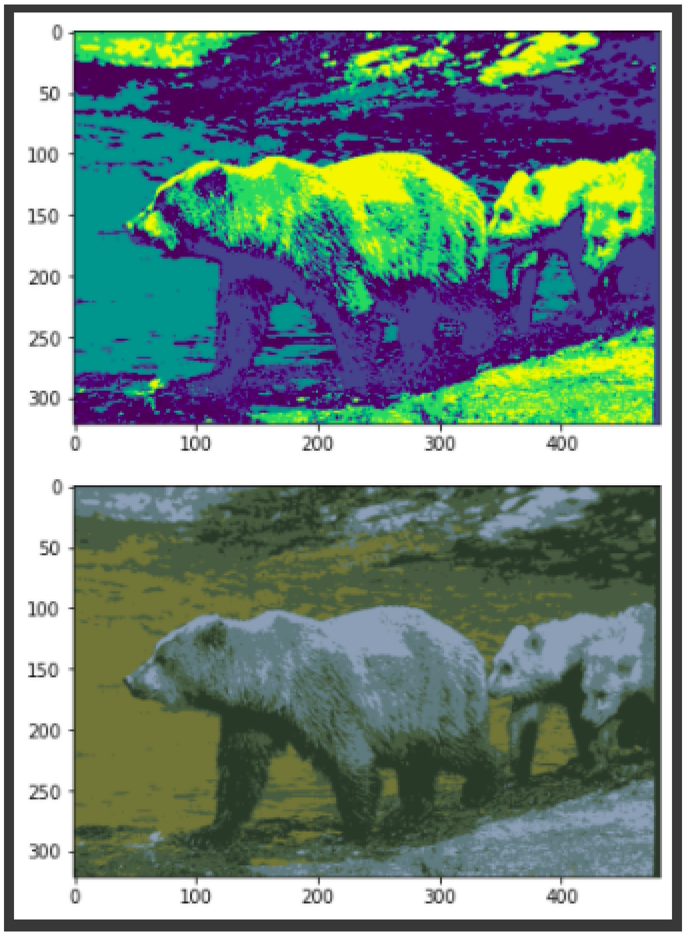
1. First Five pictures of the training set were selected along side it's ground truth segmentation(s)
2. Reshaping the images and calling the implemented k means and built-in spectral clustering
3. Converting the labels into segmentation and visualize it

A. K Means segmentation with original ground truth
    a. Comment :
        i. K means clustering had some regions correct and mixed another regions with different clusters but overall it was close

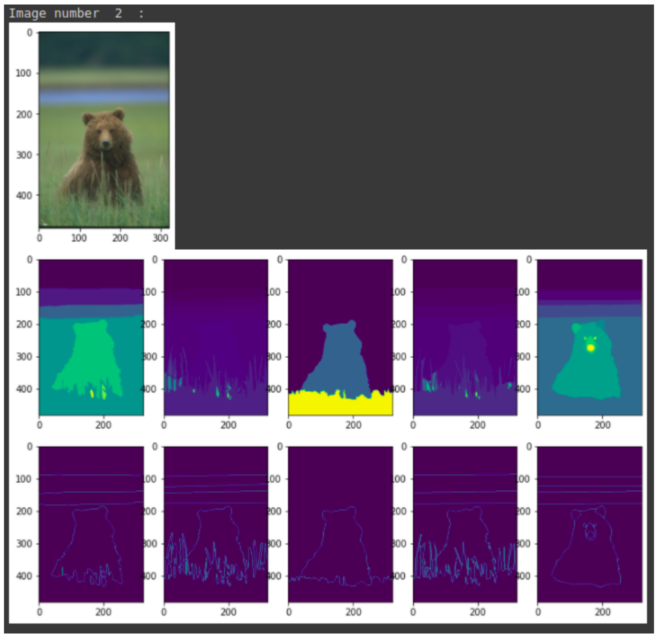
- Image 1 with its ground truth



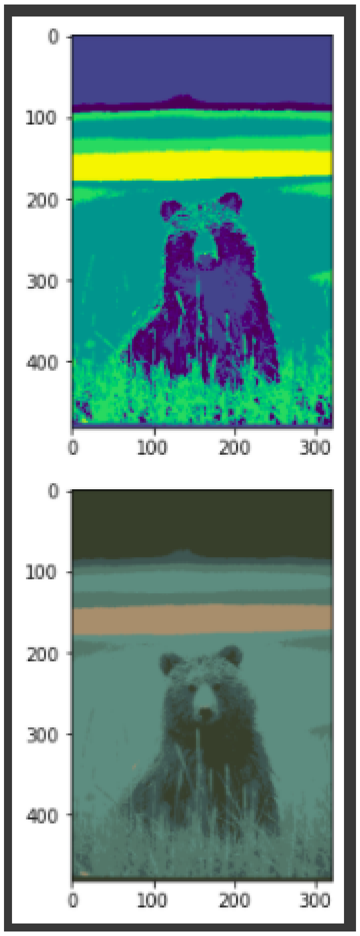
➢ Image 1 Segmentation



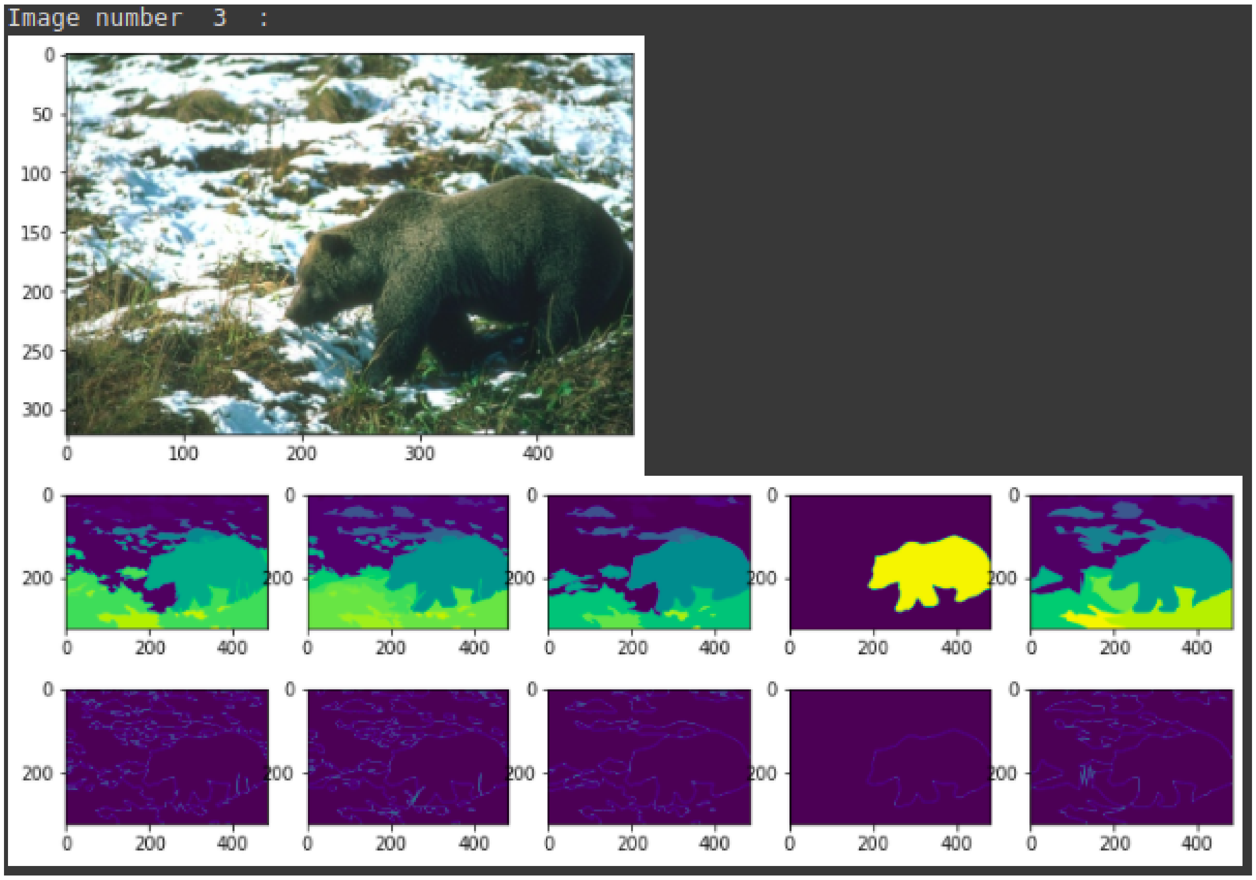
- Image 2 with it’s ground truth



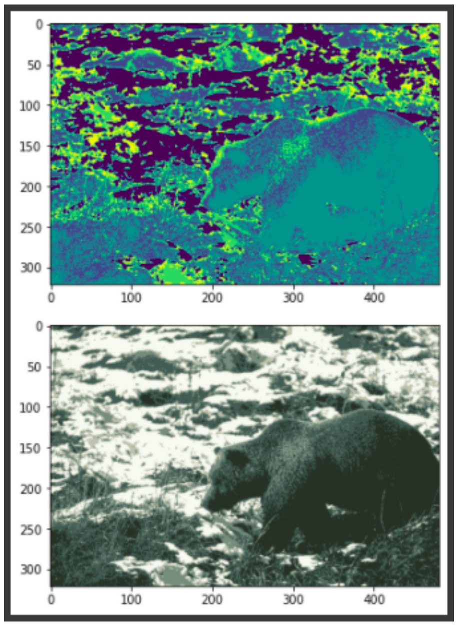
➢ Image 2 Segmentation



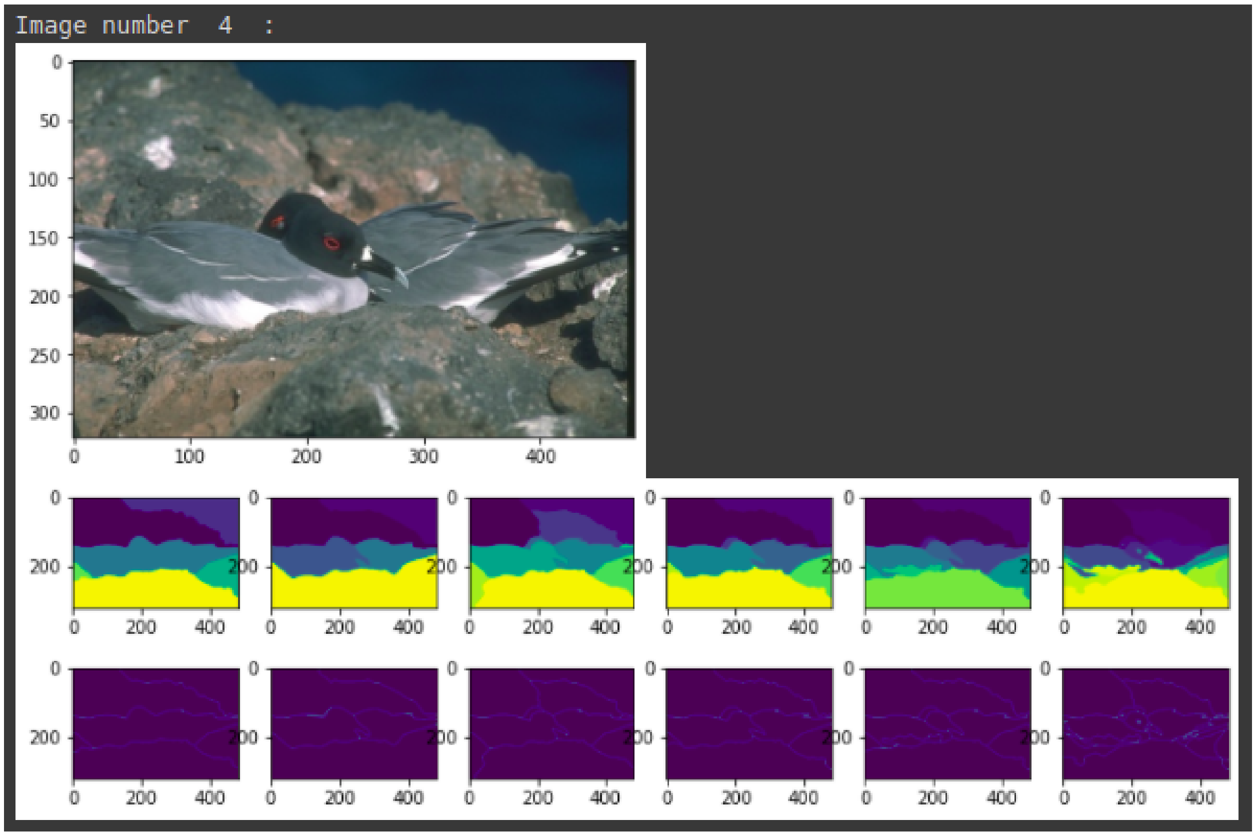
- Image 3 with it's ground truth


```
Image number  3  :
```

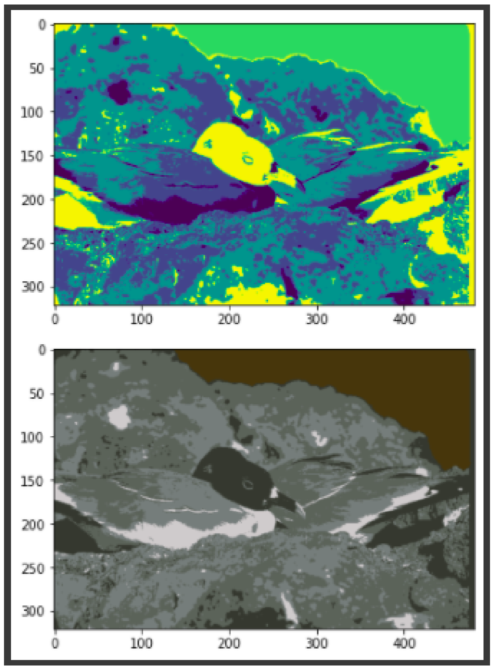
➢ Image 3 Segmentation



- Image 4 with it’s ground truth

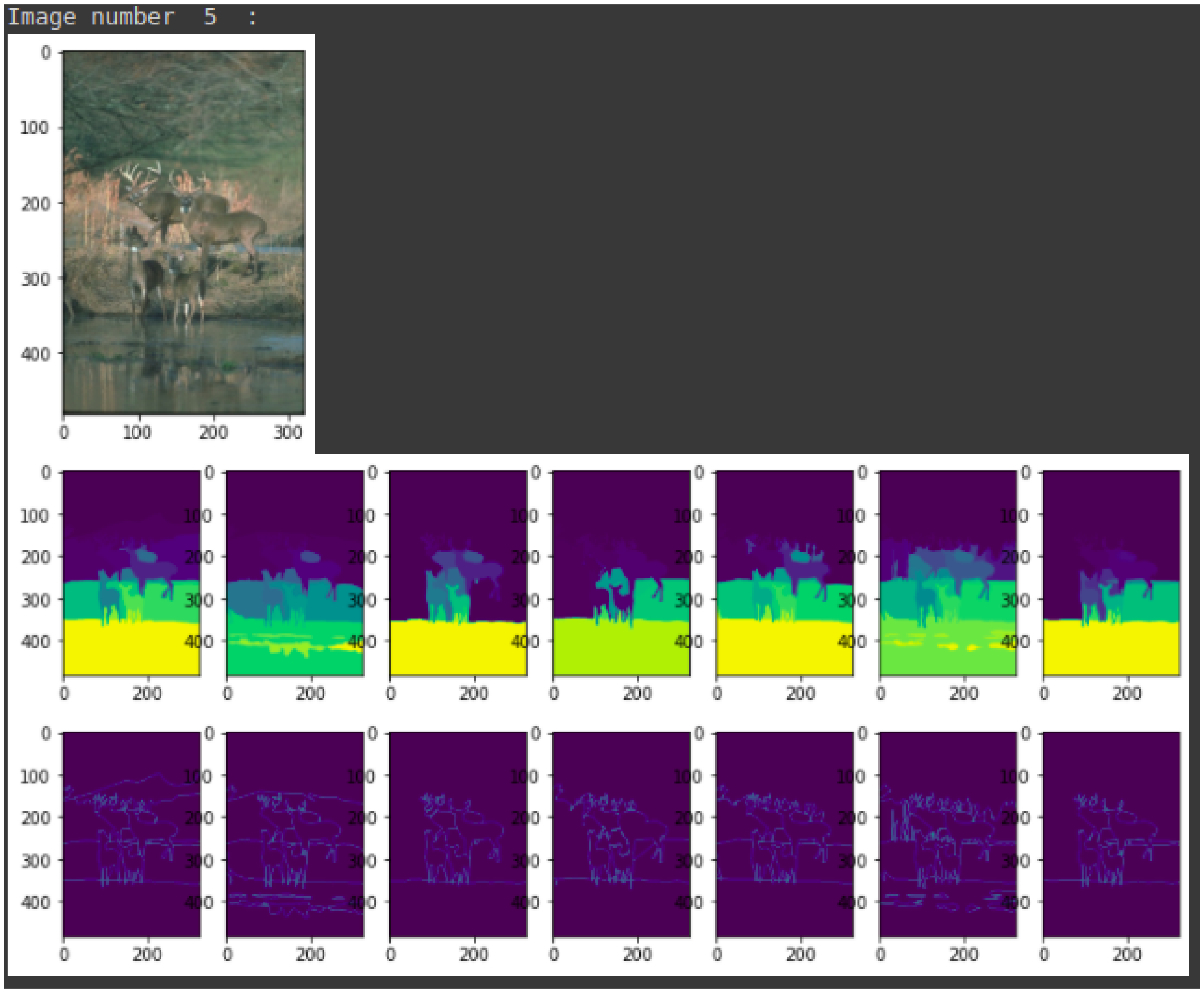


➢ Image 4 Segmentation

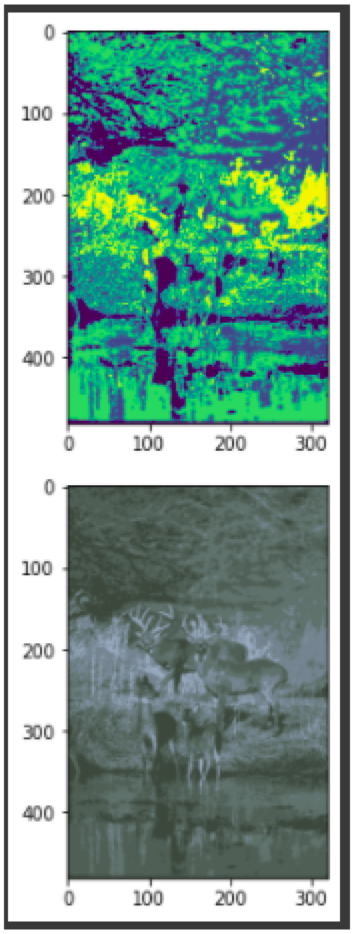


- Image 5 with it’s ground truth

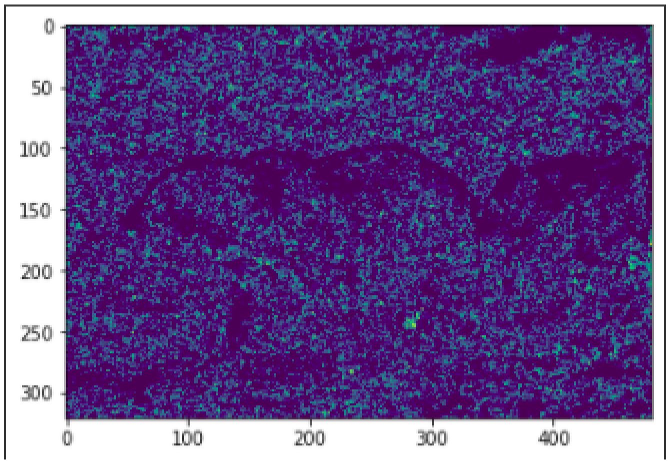
```
Image number  5  :
```



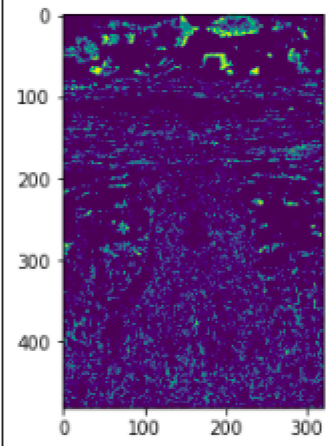
➢ Image 5 Segmentation



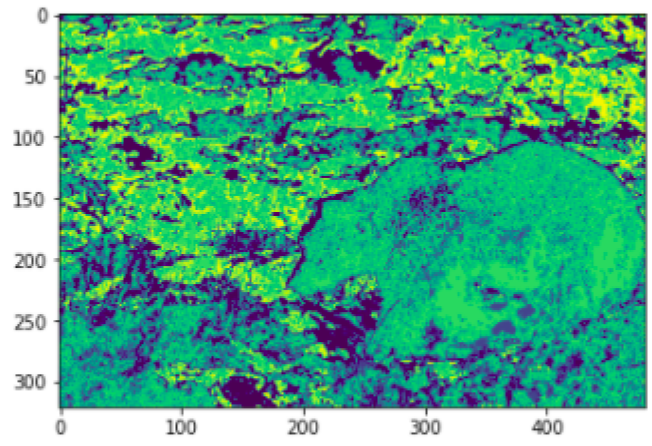
B. Spectral Clustering Segmentation with original ground truth
    a. Comment :
        i. N-cut didn't classify any object correct and had everything mixed together
    - Image 1 Segmentation



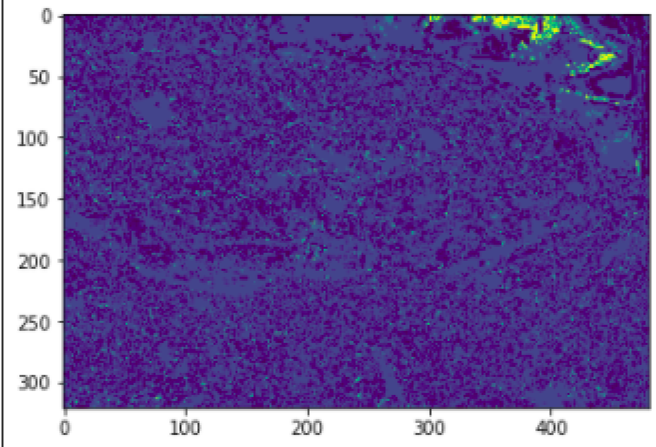
- Image 2 Segmentation



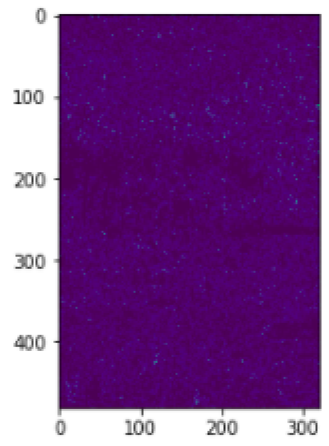
- Image 3 Segmentation



- Image 4 Segmentation



- Image 5 Segmentation



C. Comparing Kmeans & spectral clustering segmentations
    a. Comment :
        i. K means is much better than normalized cut and the classification in the N-cut were so far behind and mixed objects together