

Erin Fago  
ML HW 4: Image Segmentation Using K-means Algorithm  
Due: Monday, April 20, 2020

## **Report**

### Method:

For each of the two images, for each value of  $k$  (2,3,5,10), I ran my script. “The initial choice of  $k$  centers has significant effects on the clustering results, [making it] necessary to run K-means algorithm multiple times for each  $K$  value to find a satisfactory clustering result (or segmentation effect).” My script runs each  $k$ -value 15 times and selects the clustering with the lowest mean squared error from the 15 trials: selecting the most satisfactory clustering result (or segmentation effect). Within these 15 trials, we utilize the `kmeans` functions; from `kmeanstrain` we get an array of  $k$  center values and then from `kmeansfwd` we get an array of number for each pixel, this number represents which center the pixel should use. Based on the information from the trial with the lowest mean squared error, we can then create a new image based on the new clusterings. Finally, we display the lowest error and the original image compared to our new image!

---

### Original images:

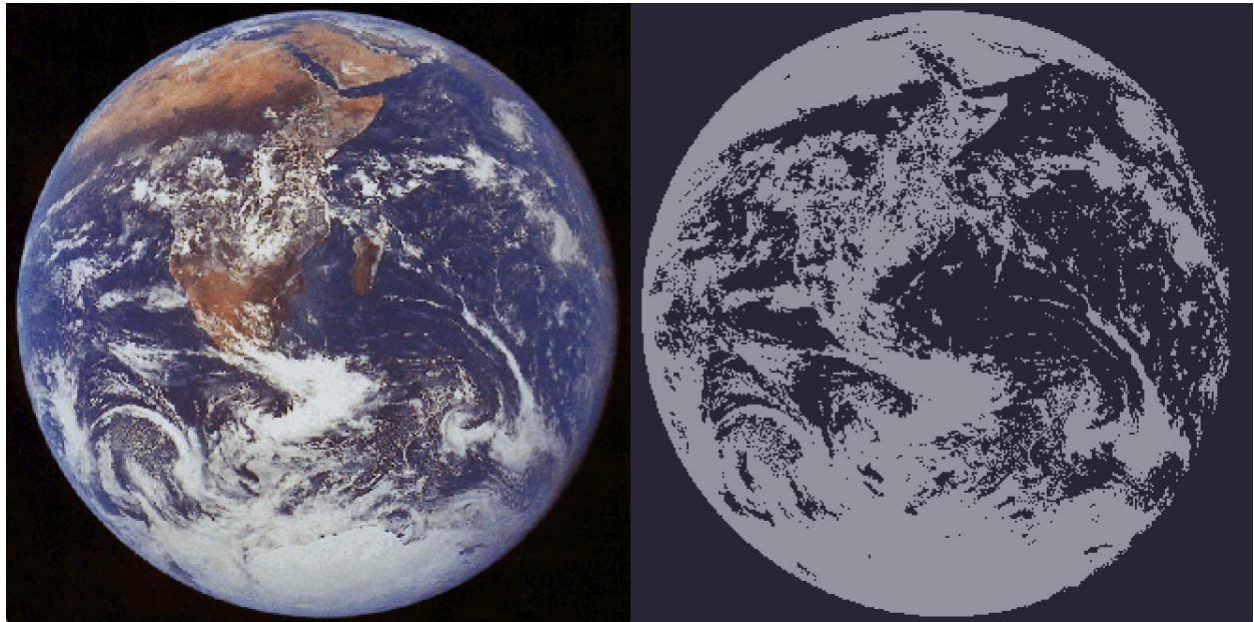


Segmentation images with various  $K$  values (i.e., 2, 3, 5, 10)

(note: the original image is on the left and the new image is on the right with its  $k$  value below it)

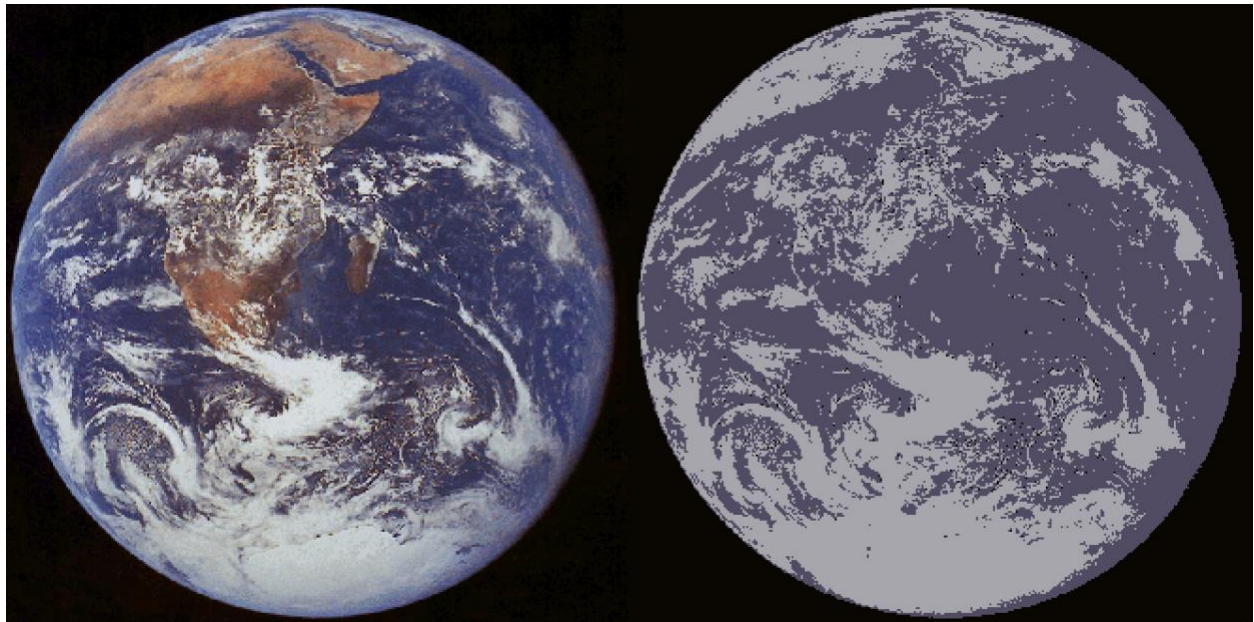
-----

**Earth.gif:**



OG

$K = 2$



OG

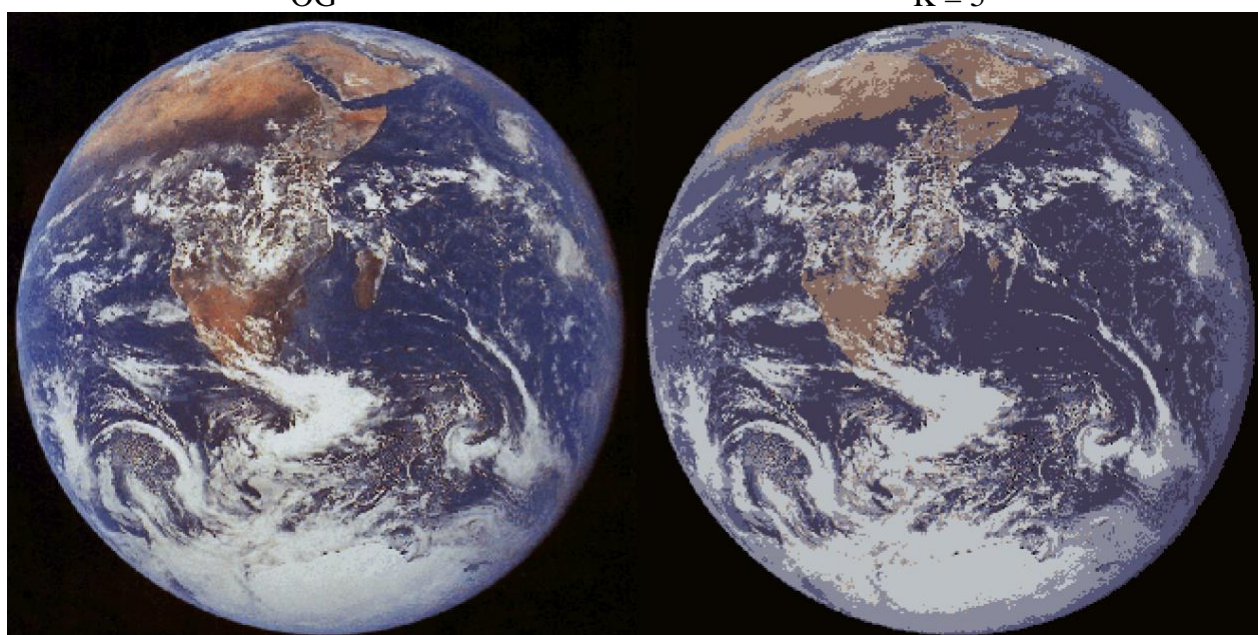
$K = 3$





OG

$K = 5$



OG

$K = 10$

Looking at the average mean2 errors and how the clustering for each kvalue was selected:

```
[Erins-MacBook-Pro-2:FAG0hw4ML erinfago$ python3 seg.py Earth.gif 2
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
curr error: 0.39654867907278507
lowest error: 0.39354567305137267

[Erins-MacBook-Pro-2:FAG0hw4ML erinfago$ python3 seg.py Earth.gif 3
curr error: 0.565307473721949
curr error: 0.5636482227318557
curr error: 0.553568877672877
curr error: 0.5653686905176156
curr error: 0.5636482227318557
curr error: 0.5636482227318557
curr error: 0.6625636522250938
curr error: 0.5653686905176156
curr error: 0.6560582524444443
curr error: 0.5636482227318557
curr error: 0.5653686905176156
curr error: 0.553568877672877
curr error: 0.5653686905176156
curr error: 0.656006868766404
curr error: 0.5653686905176156
lowest error: 0.553568877672877

[Erins-MacBook-Pro-2:FAG0hw4ML erinfago$ python3 seg.py Earth.gif 5
curr error: 1.057488774441157
curr error: 0.7343587346742099
curr error: 0.8657530321989386
curr error: 1.037453445330298
curr error: 0.8845576105150208
curr error: 0.839943451431039
curr error: 1.115819308172217
curr error: 0.7823922441080797
curr error: 0.7524016392882577
curr error: 0.9268512248614239
curr error: 1.145693396597438
curr error: 0.7694063249619456
curr error: 0.8476355586149322
curr error: 0.945371596055363
curr error: 0.9835097243900258
lowest error: 0.7343587346742099

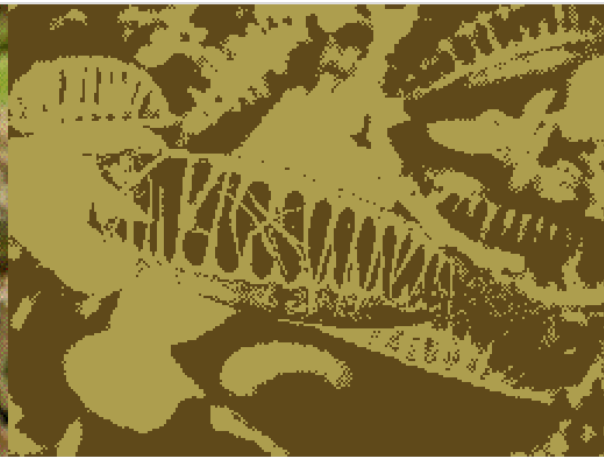
[Erins-MacBook-Pro-2:FAG0hw4ML erinfago$ python3 seg.py Earth.gif 10
curr error: 1.5190313560707578
curr error: 1.4676282668313283
curr error: 1.7643177010548046
curr error: 1.6765728315589756
curr error: 1.4874137635496727
curr error: 1.4299336025522449
curr error: 1.4344220274203014
curr error: 1.4789752148933095
curr error: 1.9299786768523306
curr error: 1.4186330065043276
curr error: 1.8537665714193237
curr error: 1.4914701749578256
curr error: 1.6222897389126916
curr error: 1.4951914456359114
curr error: 1.6216499042438188
lowest error: 1.4186330065043276
```

-----

venus\_fly\_trap.gif:



OG

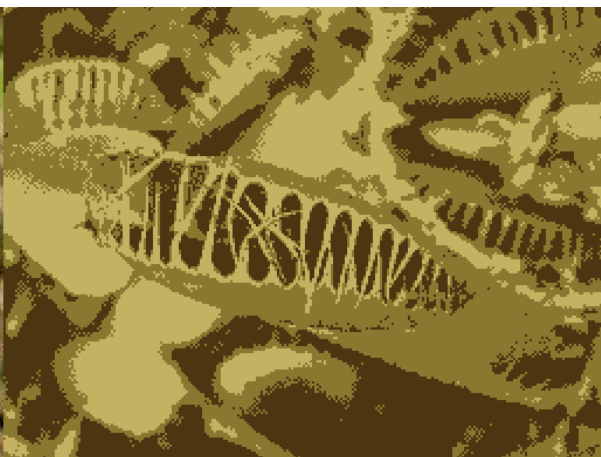


K = 2





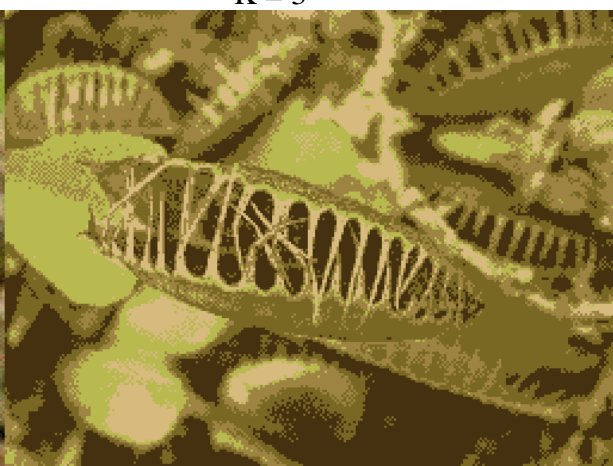
OG



K = 3



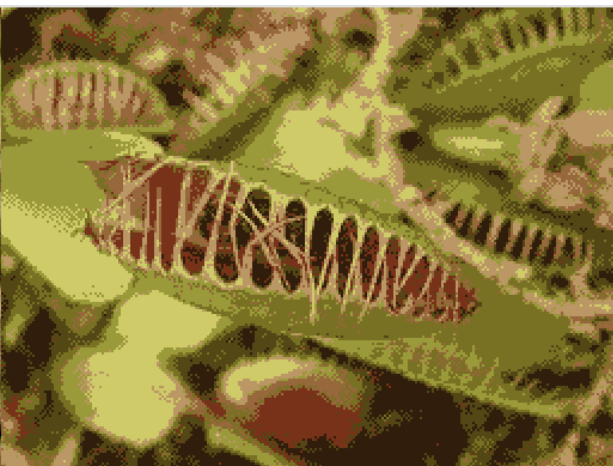
OG



K = 5



OG



K = 10

Looking at the average mean2 errors and how the clustering for each kvalue was selected:

```
[Erins-MacBook-Pro-2:FAG0hw4ML erinfago$ python3 seg.py venus_fly_trap.gif 2 [Erins-MacBook-Pro-2:FAG0hw4ML erinfago$ python3 seg.py venus_fly_trap.gif 3
curr error: 0.43096475605975015 curr error: 0.6072246426340681
curr error: 0.42679734953118753 curr error: 0.71585965608080586
curr error: 0.43096475605975015 curr error: 0.6017077611654669
curr error: 0.42679734953118753 curr error: 0.6146988008157613
curr error: 0.43096475605975015 curr error: 0.8104487058803282
curr error: 0.42679734953118753 curr error: 0.6017077611654669
curr error: 0.42679734953118753 curr error: 0.5954570133005157
curr error: 0.43096475605975015 curr error: 0.6017077611654669
curr error: 0.42679734953118753 curr error: 0.678088930017325
curr error: 0.42679734953118753 curr error: 0.6939577682655864
curr error: 0.42679734953118753 curr error: 0.949361925660612
curr error: 0.426728574467785 curr error: 0.6045115466112583
curr error: 0.42619410900822746 curr error: 0.5954570133005157
curr error: 0.42679734953118753 curr error: 1.1289323008382641
curr error: 0.43096475605975015 curr error: 0.601162112000145
lowest error: 0.42619410900822746 lowest error: 0.5954570133005157
[Erins-MacBook-Pro-2:FAG0hw4ML erinfago$ python3 seg.py venus_fly_trap.gif 5 [Erins-MacBook-Pro-2:FAG0hw4ML erinfago$ python3 seg.py venus_fly_trap.gif 10
curr error: 0.9202885505718448 curr error: 1.8817052816812492
curr error: 1.1820101533761668 curr error: 1.672195531115281
curr error: 1.1370875324343697 curr error: 2.1492372989737216
curr error: 1.7135692297726446 curr error: 2.622540663733071
curr error: 1.0314988175050974 curr error: 1.8093400073434378
curr error: 0.9598897858149432 curr error: 1.7613095769369036
curr error: 1.6560384980581653 curr error: 2.399485787035055
curr error: 1.0942415716077099 curr error: 2.885042726420367
curr error: 0.9786375079096705 curr error: 1.779876859842691
curr error: 0.9319739465539353 curr error: 2.647836834226711
curr error: 1.156770329471842 curr error: 1.8798224841789535
curr error: 1.4896029933498818 curr error: 2.585297307198062
curr error: 1.1436199298006882 curr error: 2.294942774696482
curr error: 0.894219468690431 curr error: 2.2626909350034574
curr error: 1.4534511495998081 curr error: 1.9625148203727858
lowest error: 0.894219468690431 lowest error: 1.672195531115281
```

---

## Conclusions:

After looking at the various segmentation images created by the various k values, we see that as we increase the value of k, the closer the image begins to look like the original. Since we were only tasked with exploring k values of 2, 3, 5, and 10, we see that k = 10 gives us the segmentation that is closest to the original image. I predict that there is a threshold; although between k = 2 and k = 10 there is a significantly noticeable difference, once you hit a threshold, the image will not appear much visibly different than the previous k value. We could demonstrate this by using higher values of k and comparing them to one another and the original image.