Assignment 5

Use K-means clustering for image compression.

(a) Brief summary about the images:

k = 2:

k = 5:

k = 10:

k = 15:

k = 20:

Original:

While k increases, the images I got is more and more clear. Actually, the image I got with k = 20 is quite similar with the original image. The information I lost during compression is mostly some unimportant details, like the leaves behind the koala and the spindrift behind the penguin.

(b) Answers for questions posed:

(1) Is there a tradeoff image quality and degree of compression? What would be a good value of K for each of the two images?

Yes. While k increases, the image quality increases and degree of compression decreases. The higher the degree of compression is, the more information will lose during compression and the less the image quality is.

For Koala.jpg, I would choose k = 10. When k = 5, the koala's face is still blur. When k = 10, we could see nearly all details in the image. When k = 15, we could not see much improvement comparing to k = 10. Hence, I would choose k = 10 for Koala.jpg.

For Penguins.jpg, I would choose k = 20 or even more. Actually, when k = 10, we could find all colors in the image. However, comparing to Koala.jpg, there are gradient color region on penguins' neck. Hence, when we choose k = 20 which is the maximum k in our task, we still find the penguin's neck on the left blur. Hence, I would suggest to use k = 20 or try some bigger k.

(c) Working note:

(1) I use k-means++ algorithm to optimize the beginning centers.

I will randomly choose the first center. Then I will always find next center which has the maximum distance from the nearest existing center. This algorithm will help me spread the centers which might reduce the following calculation in clustering.

(2) Actually, my algorithm runs quite slow. And it takes more time to make Koala.jpg converge than Penguin.jpg because Penguin.jpg has more separately color which makes clustering easier.

Below is my running log:

```
Img: Koala.jpg
k = 2
Start with centers:
[61, 54, 38]
[255, 255, 255]
Processing: 35 times ********************************** 100%
Time cost: 11 min 23.118 sec,
Output: k=2_compressed_Koala.jpg
k = 5
Start with centers:
```

[236, 201, 181]
[0, 0, 0]
[149, 93, 60]
[124, 147, 161]
[55, 55, 65]
Processing: 109 times
****
********** 100%
Time cost: 84 min 27.459 sec,
Output: k=5_compressed_Koala.jpg
k = 10
Start with centers:
[255, 254, 255]
[0, 0, 0]
[183, 118, 80]
[163, 181, 193]
[64, 77, 83]
[241, 170, 138]
[112, 137, 134]
[75, 32, 15]
[226, 223, 192]
[130, 74, 47]
Processing: 64 times ********** 100%
Time cost: 99 min 55.562 sec,
Output: k=10_compressed_Koala.jpg
k = 15
Start with centers:
[46, 32, 19]
[255, 255, 255]
[206, 137, 96]
[91, 110, 124]
[173, 190, 198]
[130, 70, 46]
[255, 212, 178]
[156, 135, 152]
[136, 138, 73]
[67, 67, 75]
[188, 197, 134]
[221, 209, 231]
[241, 170, 138]
[0, 0, 0]
[200, 159, 165]
Processing: 353 times
****
****
****
********** 100%
Time cost: 790 min 50.456 sec,

```
Output: k=15_compressed_Koala.jpg
k = 20
Start with centers:
[87, 64, 50]
[255, 255, 255]
[223, 150, 118]
[0, 0, 0]
[112, 137, 134]
[178, 197, 203]
[163, 105, 68]
[255, 212, 178]
[162, 169, 92]
[178, 137, 167]
[62, 23, 8]
[84, 92, 103]
[203, 209, 149]
[228, 209, 231]
[35, 44, 51]
[111, 109, 61]
[172, 123, 118]
[131, 100, 105]
[215, 176, 181]
[178, 172, 136]
Processing:                                          180                                          times
**************************************************************************************************
******************************************************************************** 100%
Time cost: 552 min 11.571 sec,
Output: k=20_compressed_Koala.jpg
Img: Penguins.jpg
k = 2
Start with centers:
[0, 13, 47]
[255, 255, 255]
Processing: 11 times ********** 100%
Time cost: 3 min 59.388 sec,
Output: k=2_compressed_Penguins.jpg
k = 5
Start with centers:
[119, 179, 215]
[0, 0, 0]
[255, 98, 1]
[255, 248, 109]
[95, 144, 52]
Processing: 23 times ********************** 100%
Time cost: 19 min 8.282 sec,
Output: k=5_compressed_Penguins.jpg
k = 10
Start with centers:
```

```
[144, 191, 237]
[0, 0, 0]
[255, 101, 0]
[0, 102, 160]
[255, 249, 106]
[99, 131, 34]
[255, 255, 255]
[145, 15, 0]
[182, 148, 123]
[206, 206, 0]
Processing: 66 times ********************************************************************* 100%
Time cost: 107 min 24.955 sec,
Output: k=10_compressed_Penguins.jpg
k = 15
Start with centers:
[185, 205, 242]
[0, 0, 0]
[255, 98, 1]
[1, 119, 167]
[254, 249, 85]
[99, 131, 34]
[136, 106, 158]
[145, 15, 0]
[1, 7, 119]
[86, 154, 251]
[219, 136, 96]
[206, 206, 0]
[1, 94, 63]
[255, 253, 186]
[160, 198, 149]
Processing:                                          100                                          times
**************************************************************************************************
*** 100%
Time cost: 241 min 42.812 sec,
Output: k=15_compressed_Penguins.jpg
k = 20
Start with centers:
[164, 172, 183]
[0, 0, 0]
[255, 72, 0]
[0, 102, 160]
[255, 247, 48]
[99, 131, 34]
[255, 255, 255]
[131, 14, 4]
[102, 51, 110]
[94, 96, 219]
[219, 136, 96]
```

```
[254, 224, 152]
[4, 6, 117]
[200, 169, 0]
[95, 206, 251]
[0, 87, 55]
[73, 165, 154]
[210, 182, 255]
[171, 85, 36]
[135, 128, 110]
Processing:                                    129                                    times
********************************************************************************************
****************************** 100%
Time cost: 403 min 25.998 sec,
Output: k=20_compressed_Penguins.jpg
Press and key to continue...

Process finished with exit code 0
```

I listed the following possible direction to optimize with the help of TAs and Google:

1. Only select data points as the core center. If so, each time, we always try to calculate the distance between every data point. We can build data structure to remember the results after first calculating the distance. Then, we can use space to get a tradeoff from run time.
2. Simplify the calculation of distance. Maybe we can use Sum(|x1 – x2|) instead of Sum((x1 – x2)^2) for distance. From this, we can reduce a lot of calculation.