Zhixing Chang

Computer Vision

Ioana Flemming

4/7/2019

Segmentation via clustering

1. Task 3:

I have used color feature, color and position feature, and median filter to color feature. Median filter to color feature doesn’t have much difference with color feature. But color feature is really different from color and position feature. Color feature’s cluster is only group image by colors, so the segments are all based on colors, and there are some segmentations that are really spread out. Color and position feature’s cluster is depending on both color and position, so the image is clearly divided into several parts and all segmentations are together.

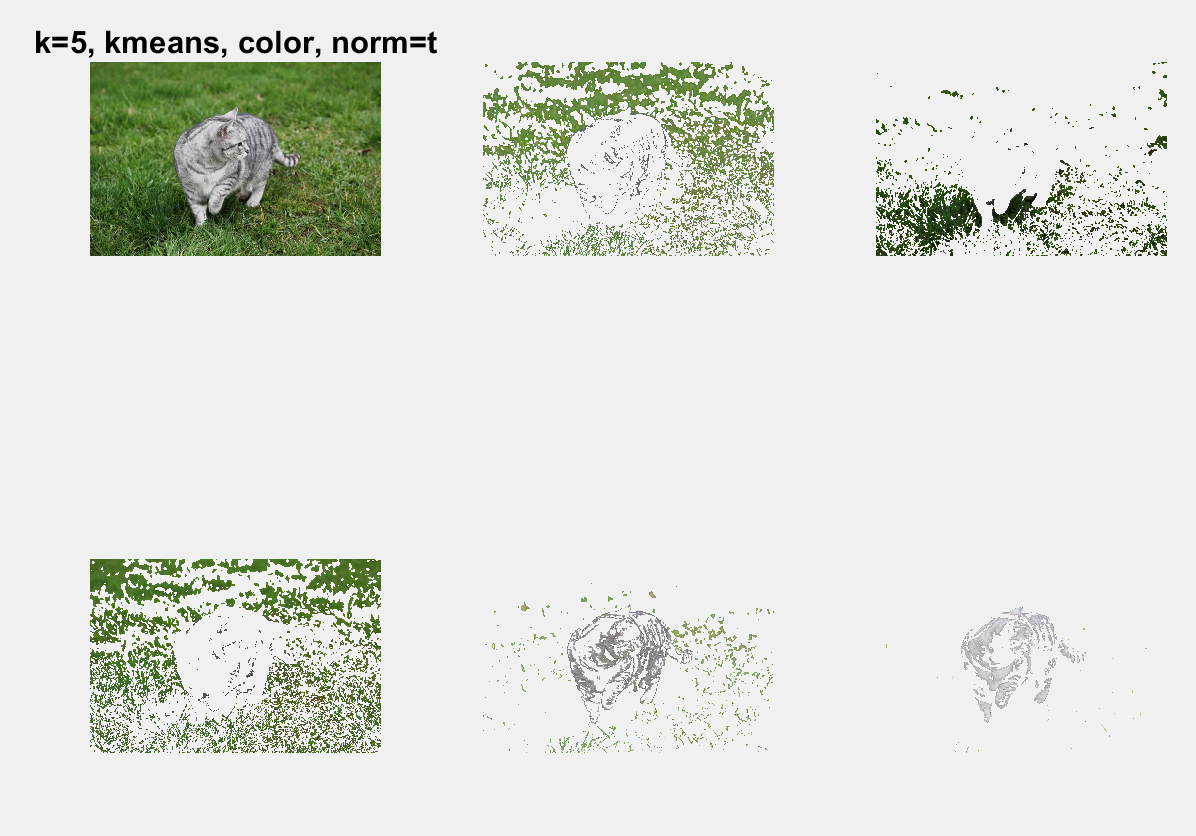
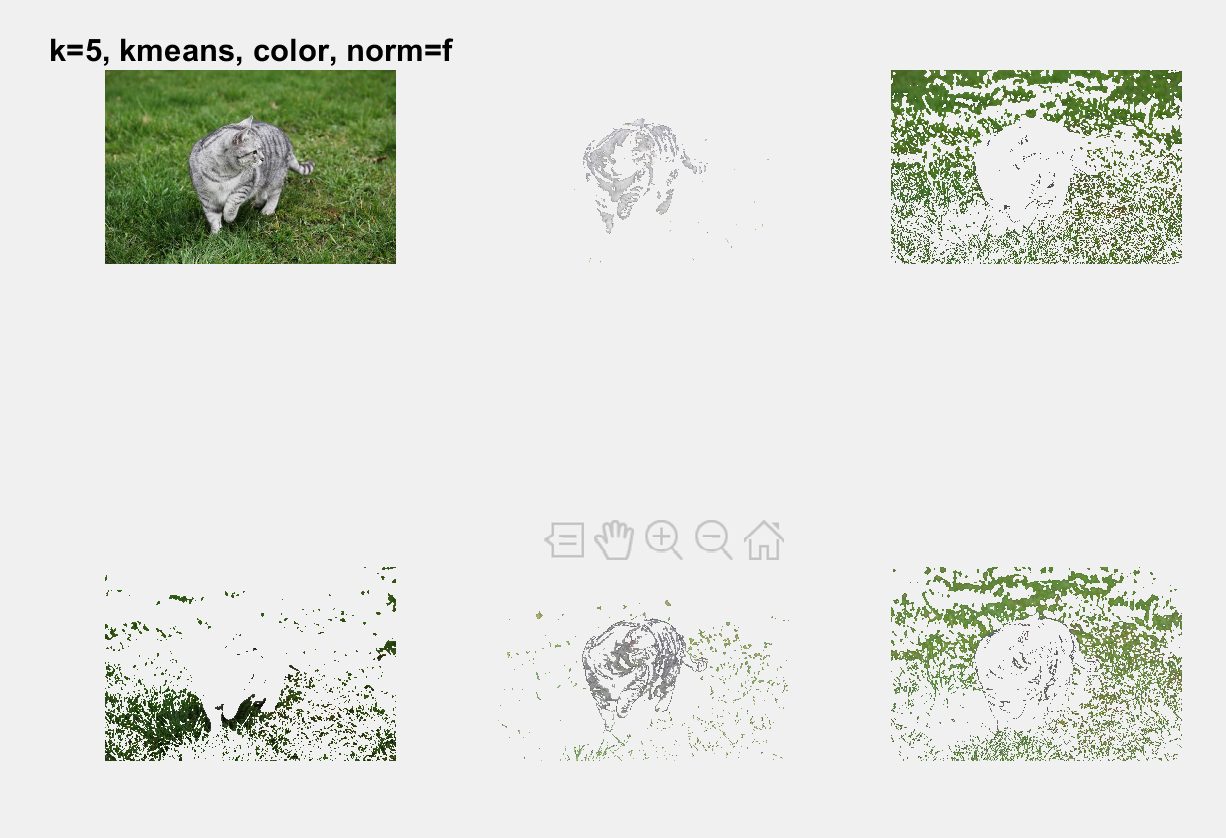
For color feature, if the color of background and cat is pretty close, we cannot get good segmentations, since similar color are in the same group.

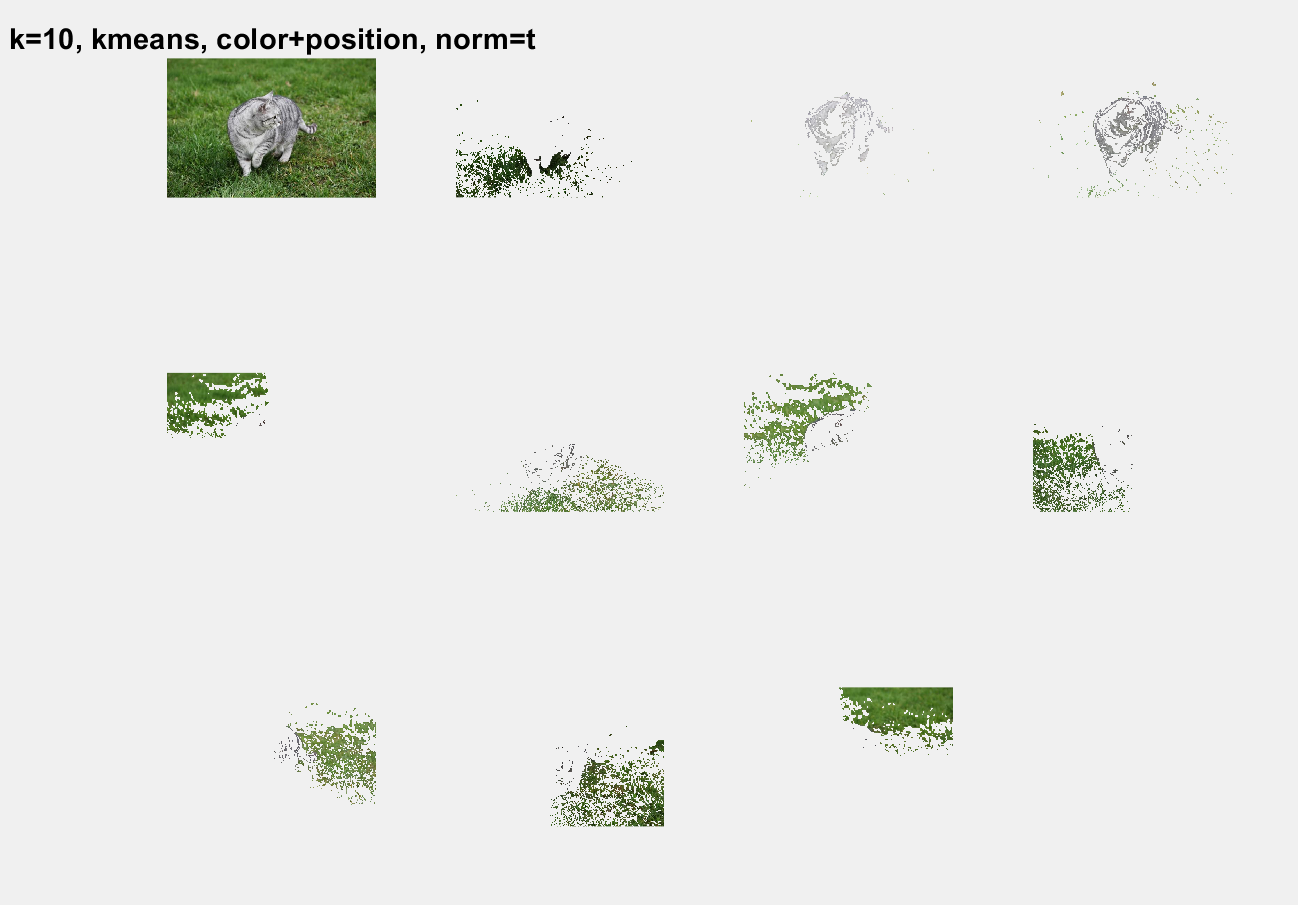
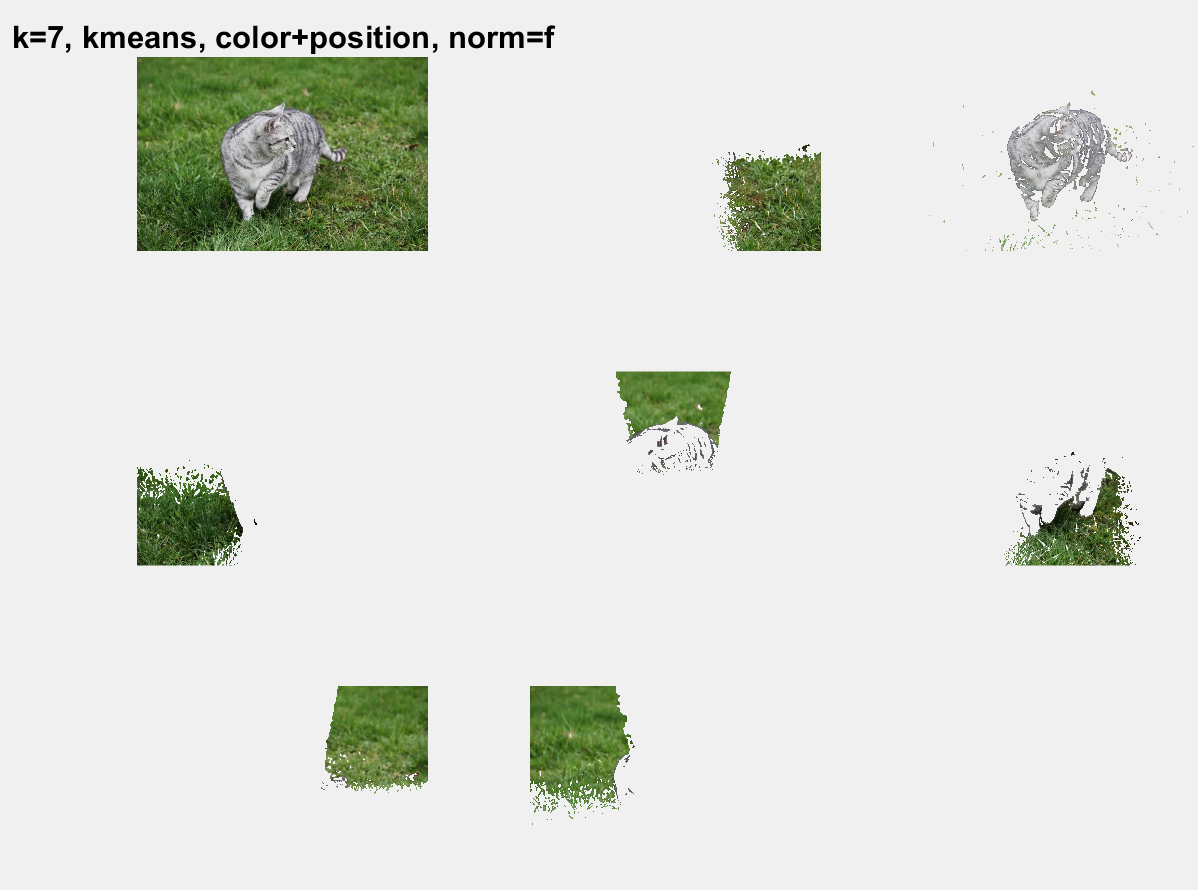
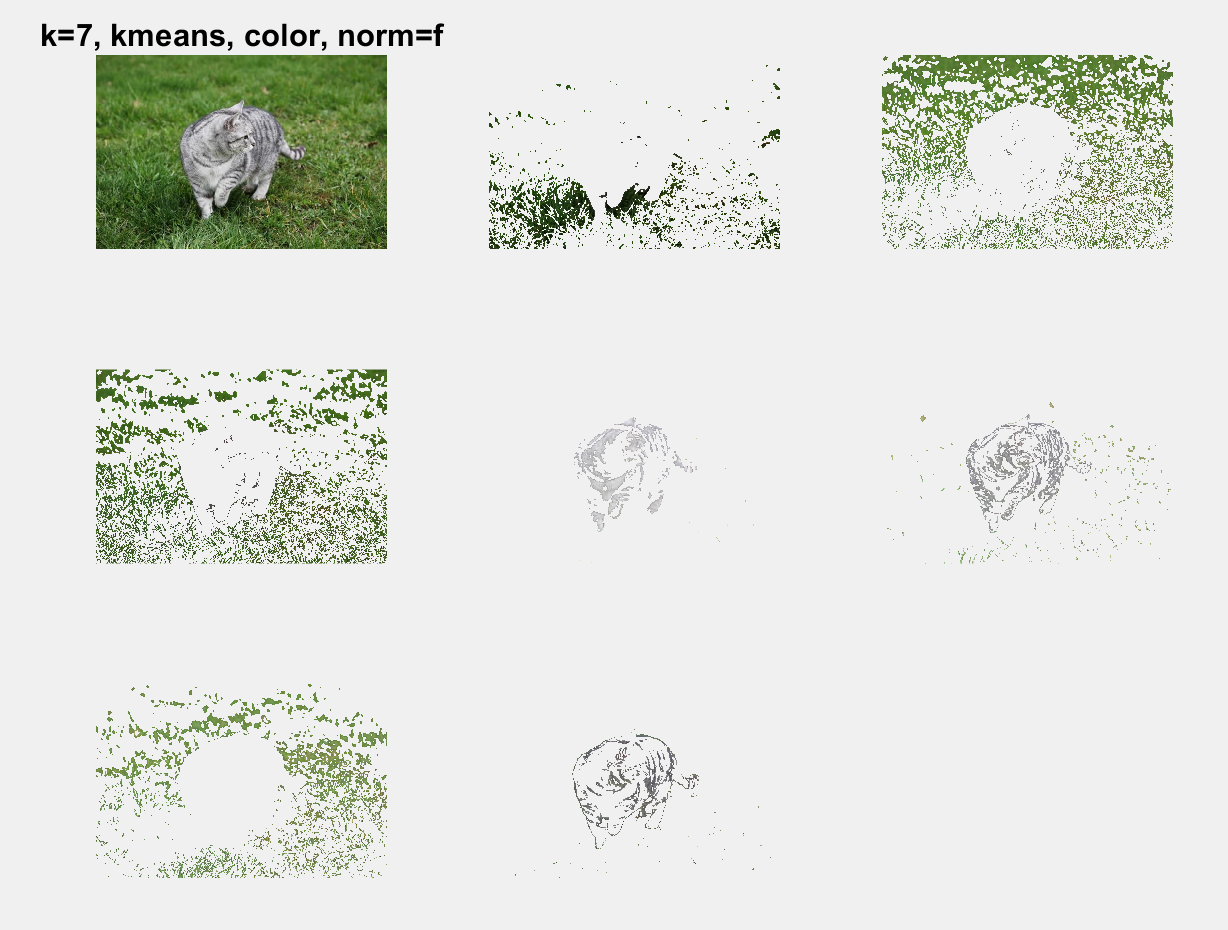
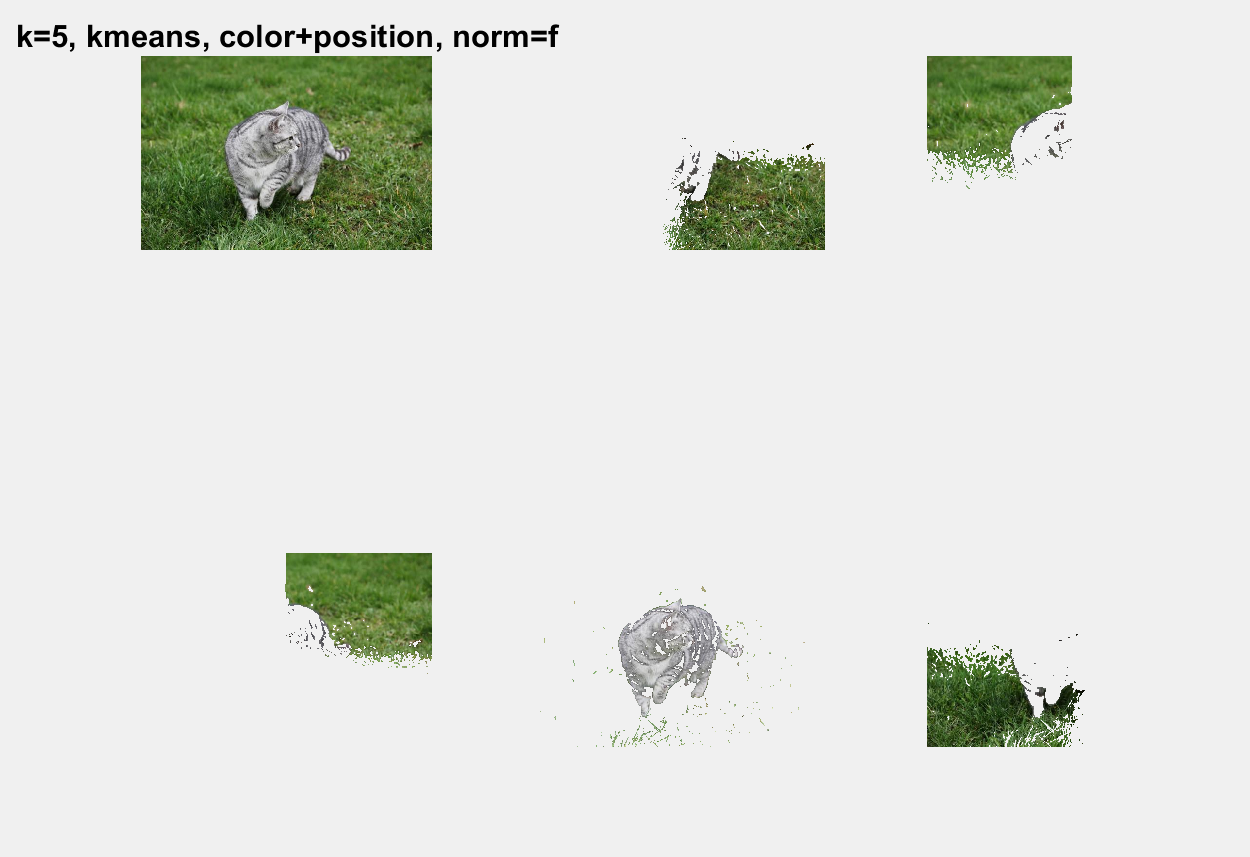
For color and position feature, if the number of group is too small like 2, the image will be divided into half and will not give a good segmentations.

When the cat’s color is clearly different from background, both feature can do a pretty good job.

The Normalization I used is the same as the lecture, using (mean-point)/std. The result shows that normalization is not really helpful for clustering.

1. Good:



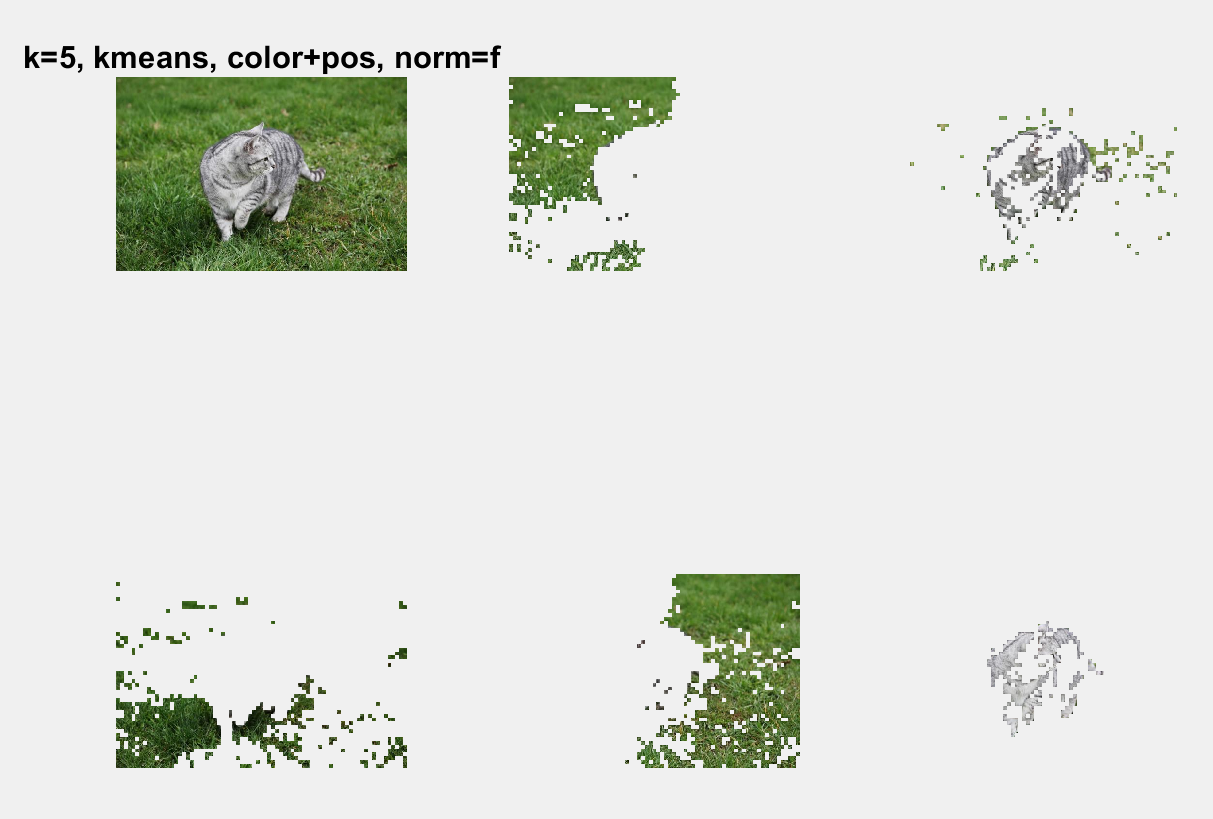
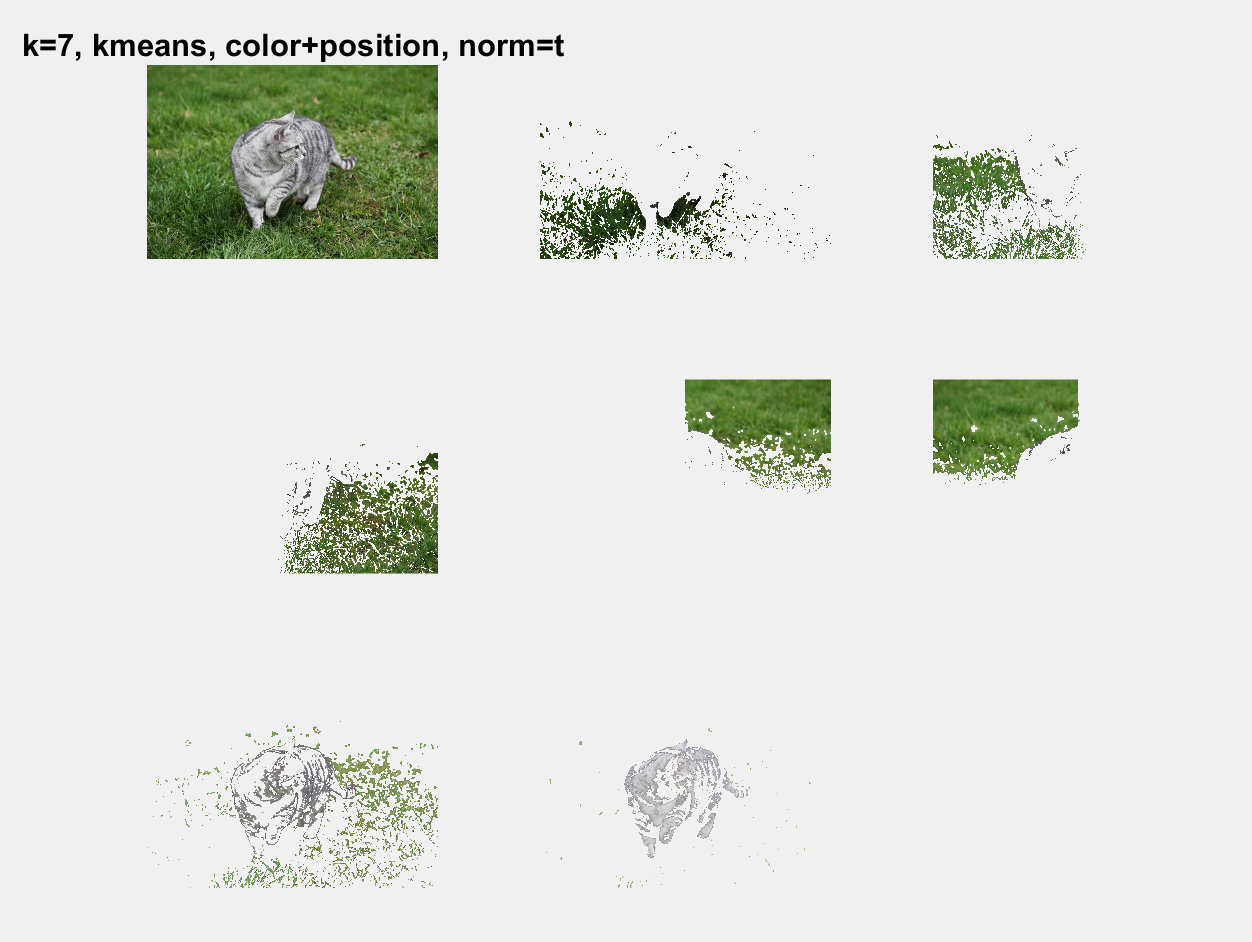


Before all size=0.5

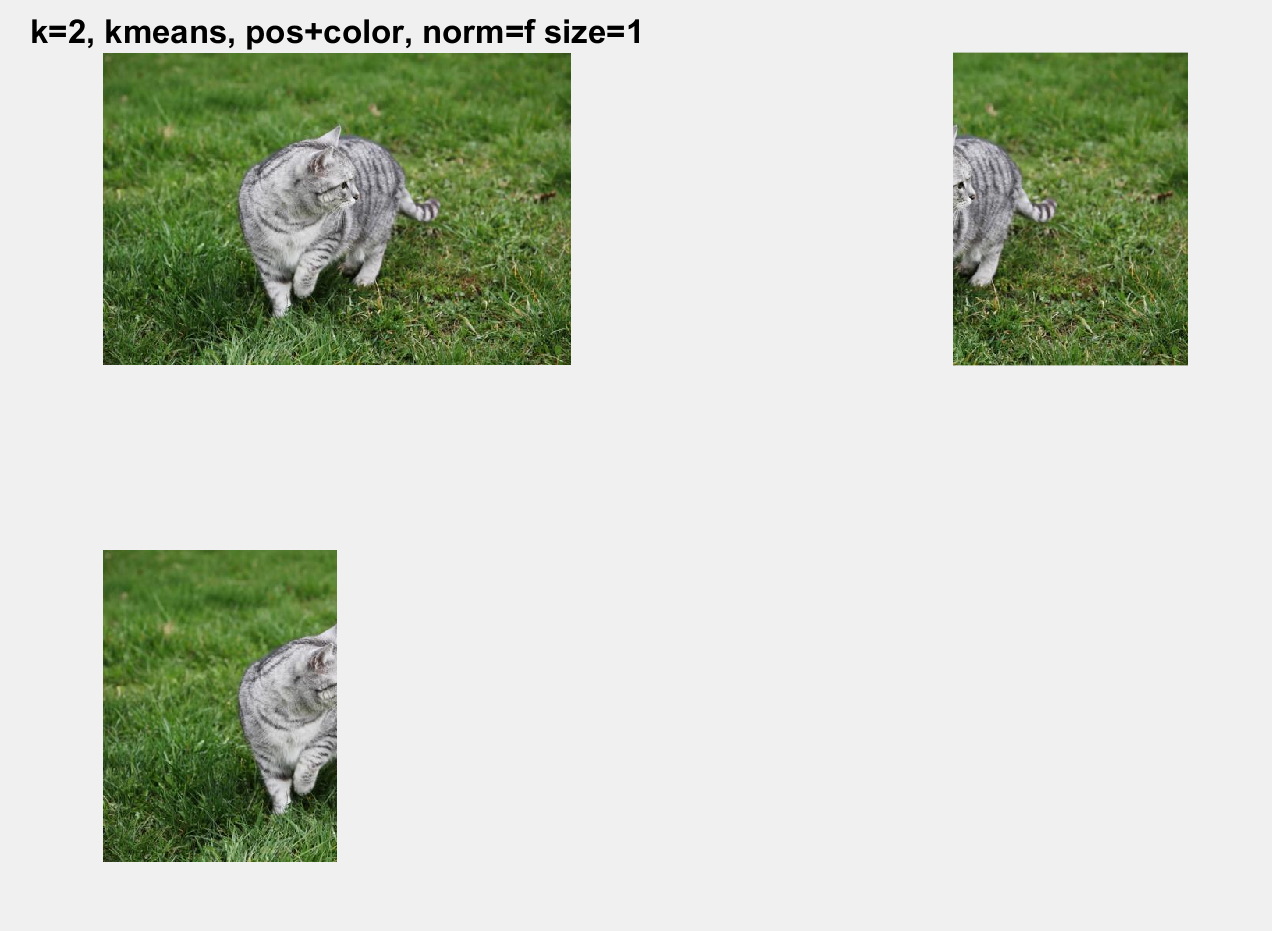
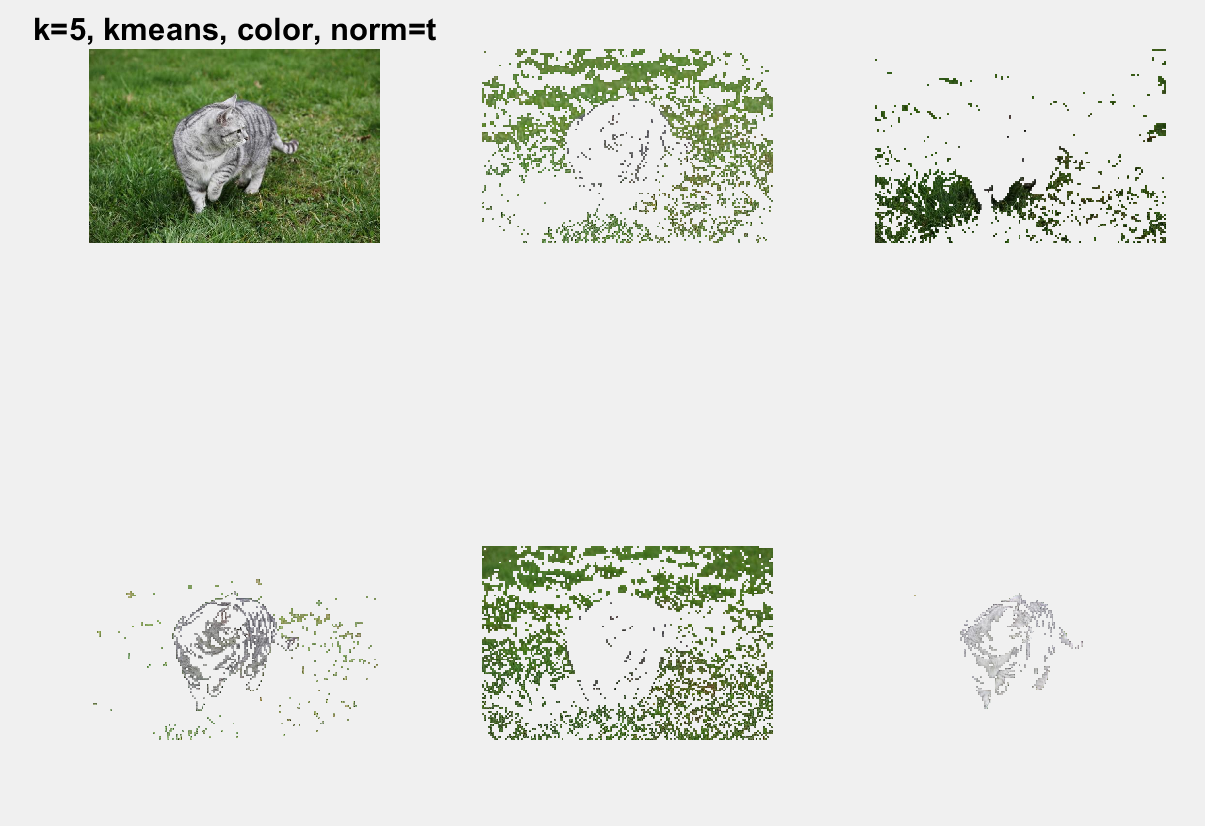
size=0.25

Bad:

Size=0.5 size=0.125



Size=0.25



1. With a good feature transformation, the segmentations can be clear and complete. I have talked about how to get good and bad segmentations from color feature or color and position feature in a). If we use filter, the feature may give a much clear segmentations compare to color feature.

Feature normalization didn’t provide a better segmentation, however, with a good normalization, the result can turn better.

If we have more number of clusters, after selecting proper clusters, we can get a really good segmentations. If we have nXm clusters for a nXm size image, it means we are separating each pixel to a group, and we can get the whole cat by selecting all proper clusters(pixels). However, with more cluster, it takes more time for user to select. Therefore, certain range of cluster is best for segmentations.

We have two clustering methods, kmeans and hac. HAC can get better result since it is choosing the best pairs on every loop and kmeans just randomly select n centers and get result based on these n centers.

The resize affect quality of the final segmentation. If we are resizing image too small, the segmentation we got is not clear but still accurate.

1. Feature transform and feature normalization doesn’t affect speed too much.

When we have more cluster, we should get result much faster but we need to take time selecting more clusters and combine them.

HAC is a really slow method, and Kmean is much faster.

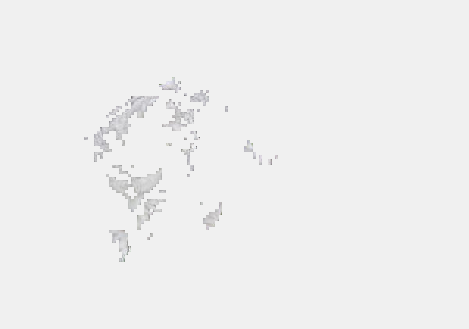
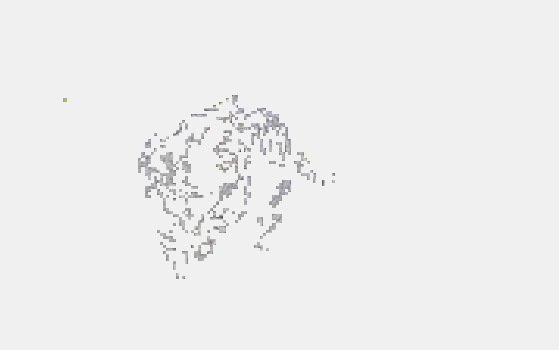
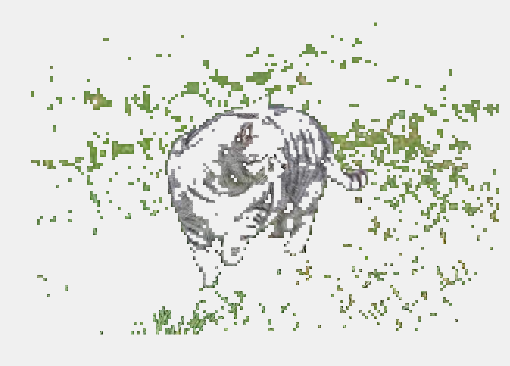
If we resize image really small, the speed will be faster.

1. The image property affects the segmentations we get. If the image is not clear, we cannot get clear segmentations. If the image is too large, it will take a long time to get result. If the image is combined with all kinds of colors and the cat’s color is pretty close to background, we cannot get good segmentations. Based on these, the image should be clear, and the color of object should be pretty separate from color of background.



k=5, method=’hac’, colorfeature, norm=true, resize=0.25, using grey-cat-grass.jpg

used





K=5 method=’kmeans’, position color feature, norm=false, resize=0.25, using grey-cat-grass.jpg

Used





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Feature Trans | Feature Norm | Clustering Method | Number of clusters | Max pixels | Mean accuracy |
| Color | T | HAC | 5 | 2500 | 0.8625 |
| Color | T | HAC | 10 | 2500 | 0.8873 |
| Position+Color | T | HAC | 10 | 2500 | 0.9119 |
| Position+Color | F | HAC | 10 | 2500 | 0.8919 |
| Position+Color | F | HAC | 10 | 1000 | 0.8878 |
| Position+Color | F | HAC | 5 | 2500 | 0.8671 |
| Color | F | HAC | 10 | 2500 | 0.8859 |
| Color | F | HAC | 10 | 1000 | 0.8775 |
| Color | T | Kmeans | 10 | 50000 | 0.8924 |
| Color | T | Kmeans | 5 | 50000 | 0.8636 |
| Color | F | Kmeans | 5 | 50000 | 0.8691 |
| Color | F | Kmeans | 10 | 50000 | 0.8708 |
| Color | T | Kmeans | 5 | 10000 | 0.8450 |
| Color | T | Kmeans | 10 | 10000 | 0.8921 |
| Color | F | Kmeans | 5 | 10000 | 0.8703 |
| Color | F | Kmeans | 10 | 10000 | 0.8963 |
| Position+Color | T | Kmeans | 10 | 50000 | 0.9112 |
| Position+Color | T | Kmeans | 5 | 50000 | 0.8746 |
| Position+Color | F | Kmeans | 5 | 50000 | 0.8861 |
| Position+Color | F | Kmeans | 10 | 50000 | 0.9099 |
| Position+Color | T | Kmeans | 5 | 10000 | 0.8779 |
| Position+Color | T | Kmeans | 10 | 10000 | 0.9080 |
| Position+Color | F | Kmeans | 5 | 10000 | 0.8718 |
| Position+Color | F | Kmeans | 10 | 10000 | 0.9098 |

Best: pos+color, norm=T, HAC, k=10, maxpixel=2500 =>0.9119

pos+color, norm=T, Kmeans, k=10, maxpixel=50000 => 0.9112

|  |  |  |
| --- | --- | --- |
|  | Kmeans with maxpixel=50000 | HAC with maxpixel=10000 |
| black-white-kittens2.jpg | 0.8997 | 0.9141 |
| black\_kitten.jpg | 0.9603 | 0.9740 |
| black\_kitten\_star.jpg | 0.9908 | 0.9880 |
| cat-jumping-running-grass.jpg | 0.9521 | 0.9777 |
| cat\_bed.jpg | 0.9417 | 0.9519 |

1. Smaller size has lower accuracy.

HAC is better than Kmeans. Since HAC is using smaller sizer, but its accuracy is similar or larger than Kmeans’ accuracy.

Position+Color feature is better than color feature.

More cluster is better.

Normalization is not stable, it depends on other conditions. Sometimes it is better and sometimes it leads to smaller accuracy.

1. Yes, the color of object and background is not really different, or the background is messy, the segmentations are poor.
2. The size is affecting speed, and HAC is SOOOOOOOOOOOOOO slow. But HAC is somehow almost “perfect” algorithm. When I set max pixel to 10000, it took about 1.5 hours to run. And for runcomputesegmentation, I set size to 0.25 and it took about 4 hours to finish program.