CS5487 PROGRAMMING ASSIGNMENT 2: CLUSTERING	
WANG Yue	56359462

PART ONE: Clustering synthetic data

◆ (a) Implement 3 clustering algorithm

Implementation written in matlab is attached in the source code files.

File function:

Part_one (subfile)	This file is the code that implements part one question.
main_1b.m	Main function to call the algorithms on the three synthetic datasets.
main_1c.m	Main function to discuss how sensitive is mean-shift to the bandwidth parameter h.
Kmeans.m	The function of K-means algorithm for the 1(b) question.
EMgmm.m	The function of EM-GMM algorithm for the 1(b) question.
EMinitPara.m	Initialize the parameter for the model(Miu, Pi, Sigma) in EM-GMM algorithm.
Meanshift.m	The function of Mean-shift algorithm for the 1(b) question.
plot_clusters	Draw clustering figures of 3 algorithms.
Part_two (subfile)	This file is the code that implements part two question.
main_2a.m	Main function to use these three clustering algorithms to segment a few of the provided images.
main_2b.m	Main function to modify K-means and mean-shift implementations to allow different feature scaling.

◆ (b) Running algorithms on the three synthetic data

❖ Implementation and plot

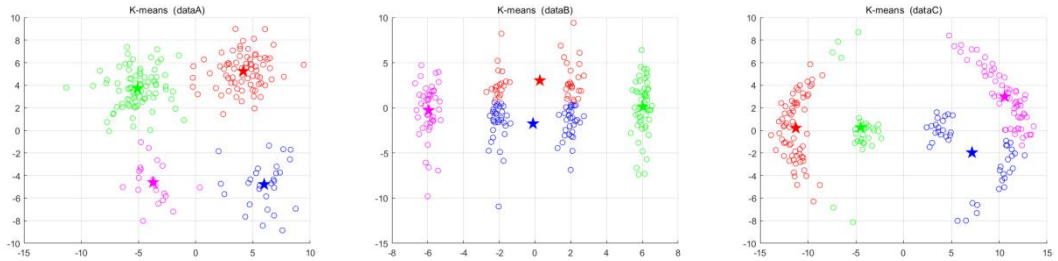


Figure group 1b.1 : K–means algorithm on three synthetic data (K=4)

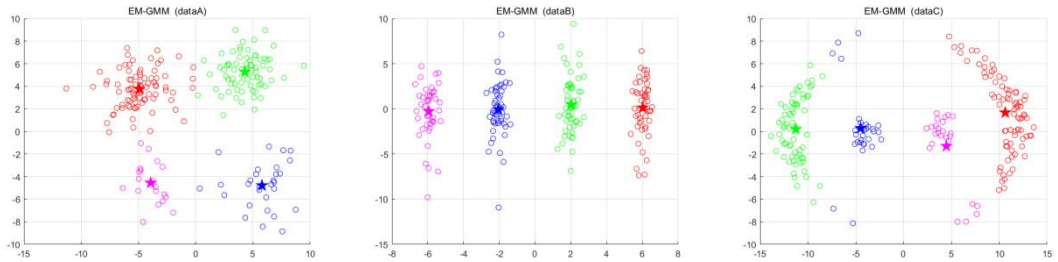


Figure group 1b.2 : EM–GMM algorithm on three synthetic data (K=4)

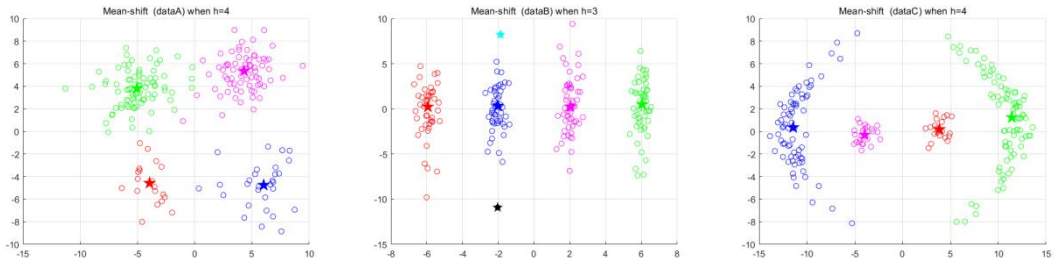


Figure group 1b.3 : Mean–shift algorithm on three synthetic data (h=4,3,4, respectively)

❖ Analysis

This experiment implement the three algorithms (K-means, EM-GMM, Mean-shift) on the three synthetic datasets. K-means and EM-GMM are parametric clustering and Mean-shift is non-parametric clustering method. In each figure, I have marked the center of each cluster with a pentagram, and different colors indicate different clusters.

Form figure group 1b.1 we can see that K-means algorithm can correctly cluster the dataA into four classes, but have some deviations on dataB and dataC. Form figure group 1b.2, we can see EM-GMM algorithm can correctly cluster the dataA and dataB into four classes, while for dataC, there are some marginal points cannot be assigned to the right cluster. For non-parametric algorithm Mean-shift, although it get rid of the choice of parameter K, it suffers from the adjustment on bandwidth h. In figure group 1b.3, the bandwidth h has already adjust to a appropriate value to cluster better. Unlike the previous two algorithms, Mean-shift performs very well on data C. However, there are two more clusters on data B than the standard classification, namely the cyan at the top and the black dot at the bottom.

By analyzing the performance of different algorithms on the same data set, we can find that, in terms of dataA, all three algorithms perform well. In terms of dataB, K-means is often blocked by the local optimal solution and unable to find the global optimal solution because the selection of the initial cluster center. Mean-shift has a few points as the extra clusters. The performance of EM-GMM on dataB is the best. In terms of dataC, only Mean-shift can make an accurate classification.

❖ Conclusion - advantages and limitations of the three algorithms

(i) K-means : It is easy to implement. But the result is greatly influenced by the initial cluster center generated randomly. If the center of the initial cluster is not good, it's easy for K-means to get stuck in a locally optimal solution. As for the data set with distribution which is relative uniform and does not cross each other, K-means performs well. However, the performance is not good when the densities between clusters are not equal like data C. besides, in K-means, cluster number is fixed and affiliation of data points is hard. One point belongs to one clusters only.

(ii) EM-GMM : Compared with K-means, it is more general and can form clusters of different sizes and shapes. Unlike K-means, GMM can output probability of cluster affiliation, and the initialization of estimated values potentially affect clustering result. Cluster number of GMM is also fixed.

(iii) Mean-shift : Different from the previous two algorithms, there is no need to set the number of

cluster classes. It can handle cluster classes of any shape. MS works well in our data A, B and C, and the representation is not by cluster center so it is flexible. From the mathematical formula, it make the every data points as one cluster center so that there are not cause error. As is shown in the figure, if there are some data not being together with a line or reunion distribution, the function also can take them in one cluster well. But the limitation is that the stagey of cluster prototyping will affect the performance of cluster and the speed is much slower than the other two.

❖ The effect of different initial centers on the results in K-means

In this experiment, the initial cluster center is set to different values to explore the effect of different initial centers on the results in K-means.

The initial clusters are center1=[-10,0; -2,0; 2,0; 5,0], center2=[-4,5; 4,5; -5,-5; 5,-5] and center = [-5,1; 0,0; 6,3; -5,3], respectively.

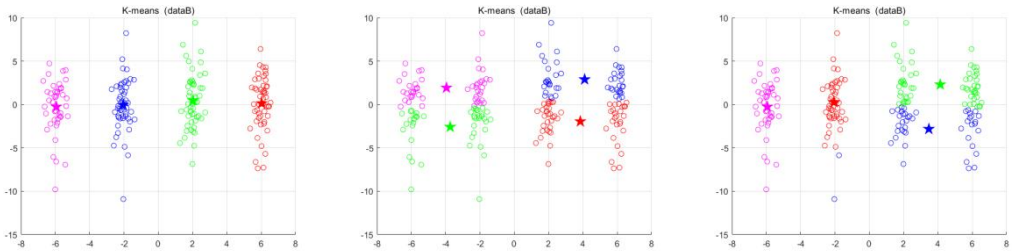
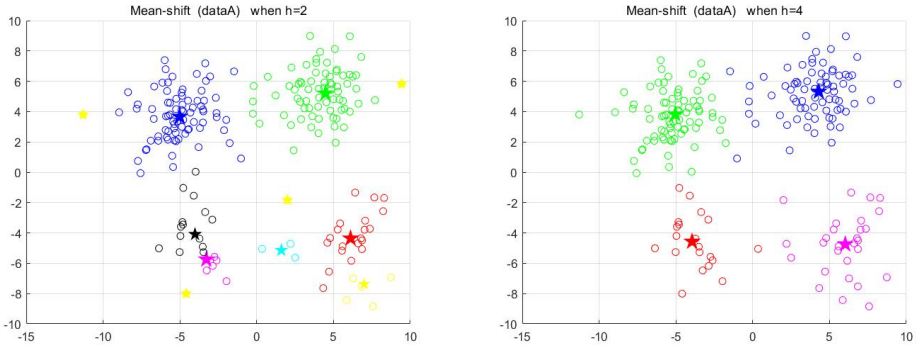


Figure group 1b.4: K–means algorithm on dataB when taking different initial cluster center

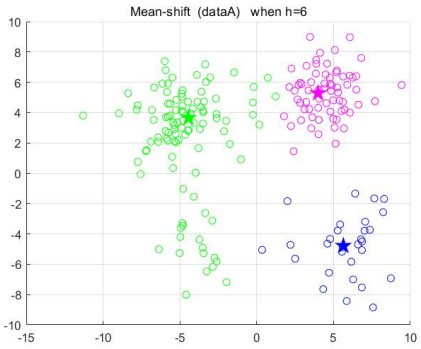
We can see from figure group 1b.4 that different initial cluster centers have great influence on the results of clustering.

◆ (c) How sensitive mean-shift is to the bandwidth parameter h

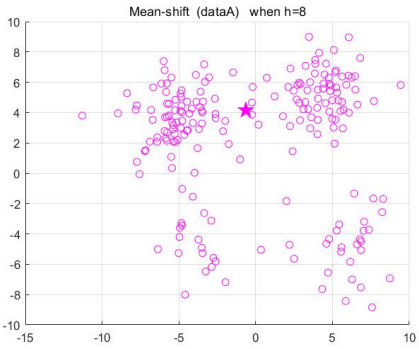
To explore how sensitive mean-shift is to the bandwidth parameter h, this experiment uses different h on three datasets.



(a) h=2 (b) h=4

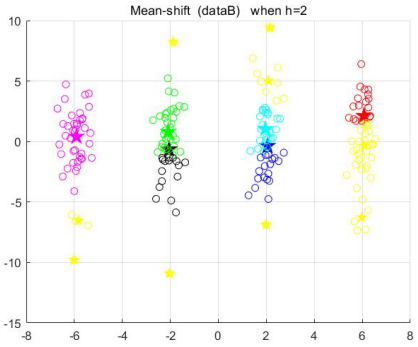


(c) h=6

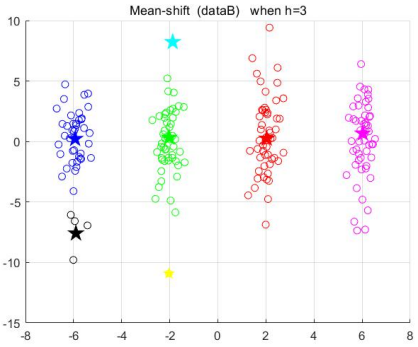


(d) h=8

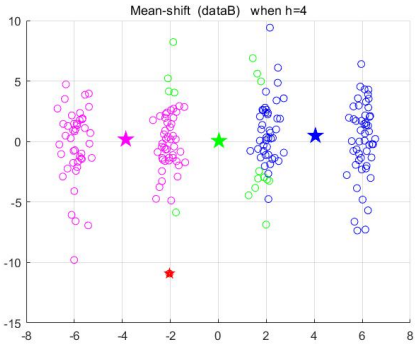
Figure group 1c.1: Mean–shift algorithm on dataA when taking different h (2/4/6/8)



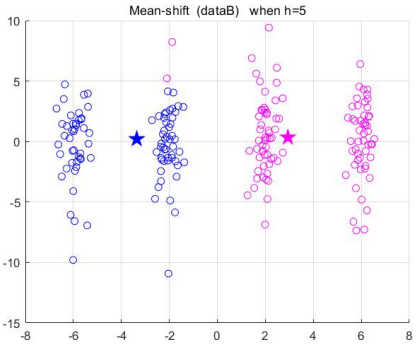
(a) h=2



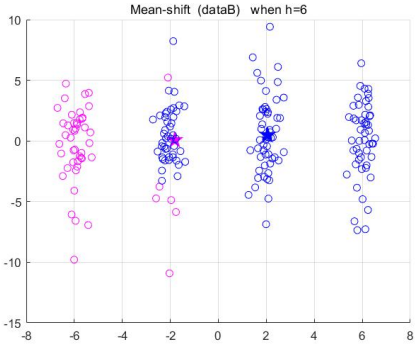
(b) h=3



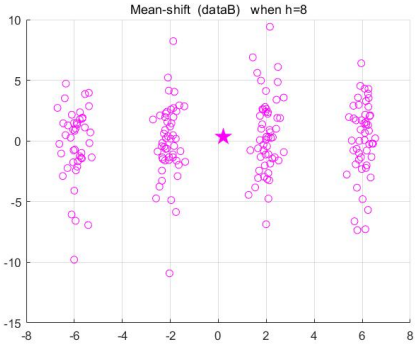
(c) h=4



(d) h=5

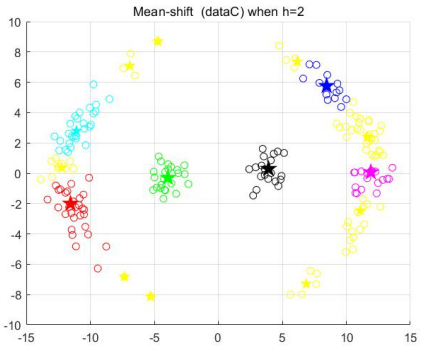


(e) h=6

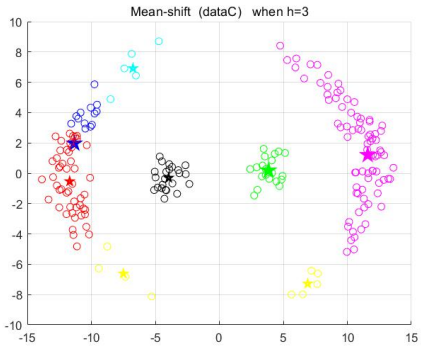


(f) h=8

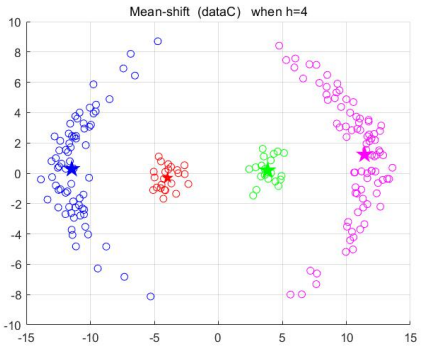
Figure group 1c.2: Mean–shift algorithm on dataB when taking different h (2/3/4/5/6/8)



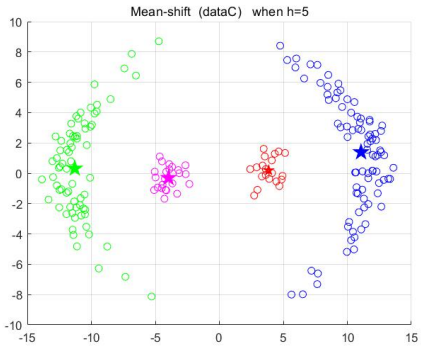
(a) h=2



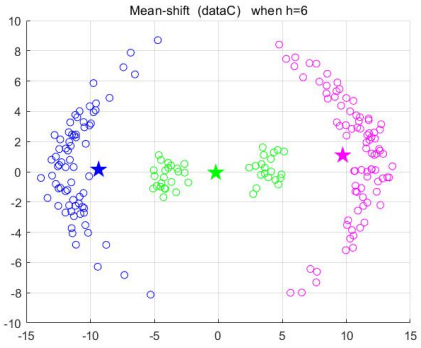
(b) h=3



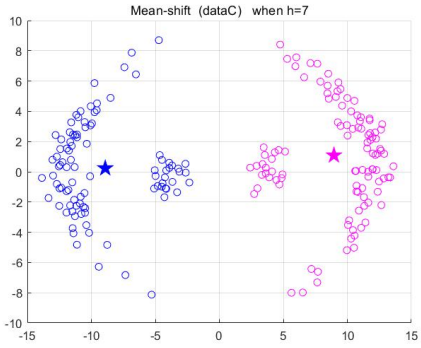
(c) h=4



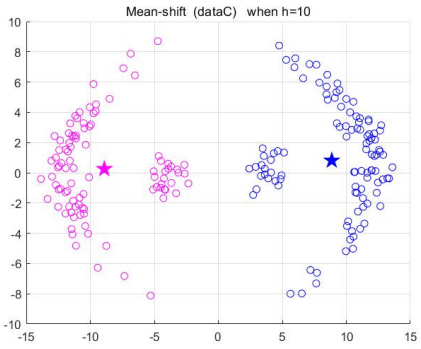
(d) h=5



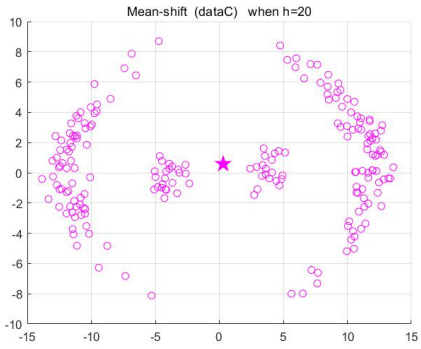
(e) h=6



(f) h=7



(g) h=10



(h) h=20

Figure group 1c.3: Mean–shift algorithm on dataC when taking different h (2/3/4/5/6/8/10/20)

❖ Analysis

Figure group 1c.1-3 are experiments results using different bandwidth h. Different clusters are shown in different colors. When the number of clusters is greater than 6, the remaining clusters after 6 are shown in yellow.

In figure group 1c.1, when h is 2, more than 7 kinds of clusters were produced, that is not consistent with the true clustering. When h is 4, it can correctly cluster the dataA into four classes. But as h grows up to 6 and 8, it produced fewer and fewer clusters, especially when h is 8, it can only produce one cluster.

In figure group 1c.2 and 1c.3, we can see that the greater the value of A, the less the number of clusters. The optimal value of h on dataB in this experiment is 3, and the optimal value of h on dataC in this experiment is 4.

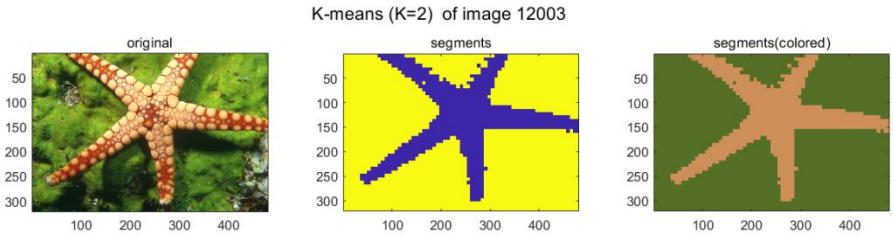
❖ Conclusion

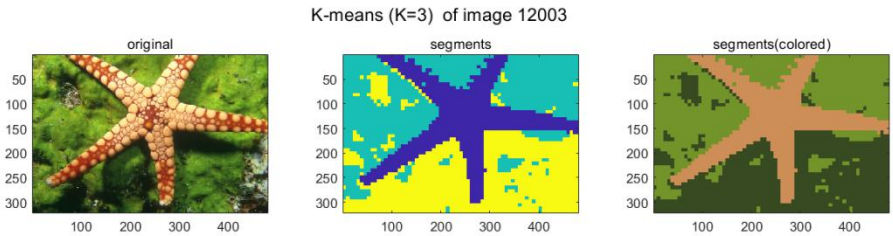
In summary, when the bandwidth increases, number of clusters will decrease, vice versa. What's more, there are different optimal h values for different data. A more appropriate h can help better clustering the data. This is also the limitation of this method.

PART TWO: A real world clustering problem-image segmentation

◆ (a) Segmentation Examples

In this experiment, three clustering methods were used to segment image 12003 and image 101087.





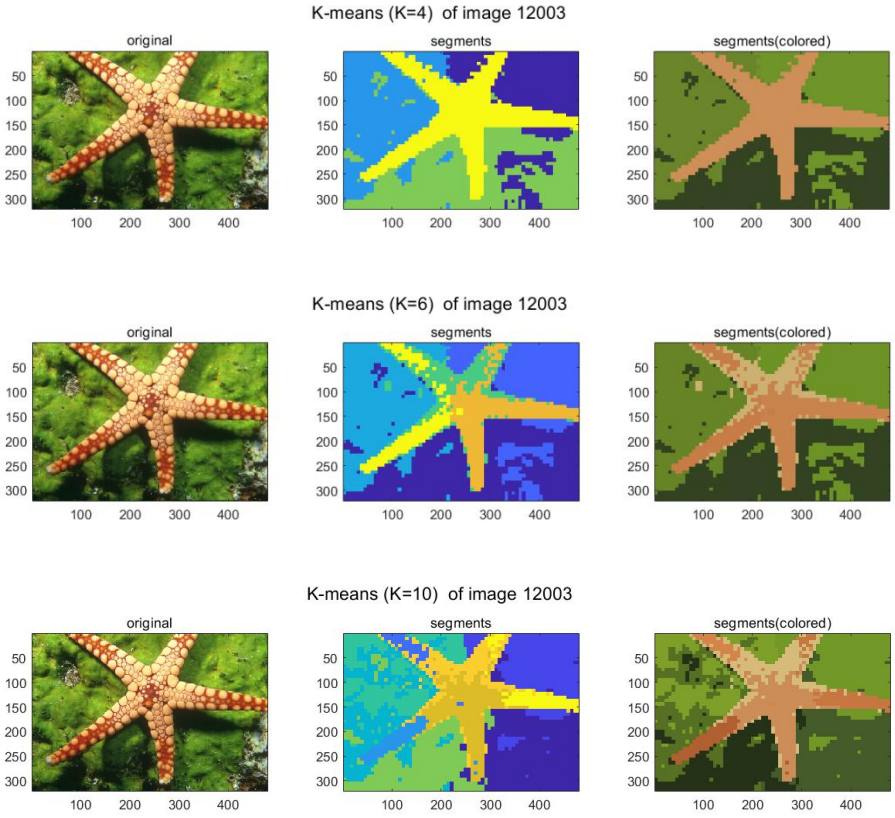
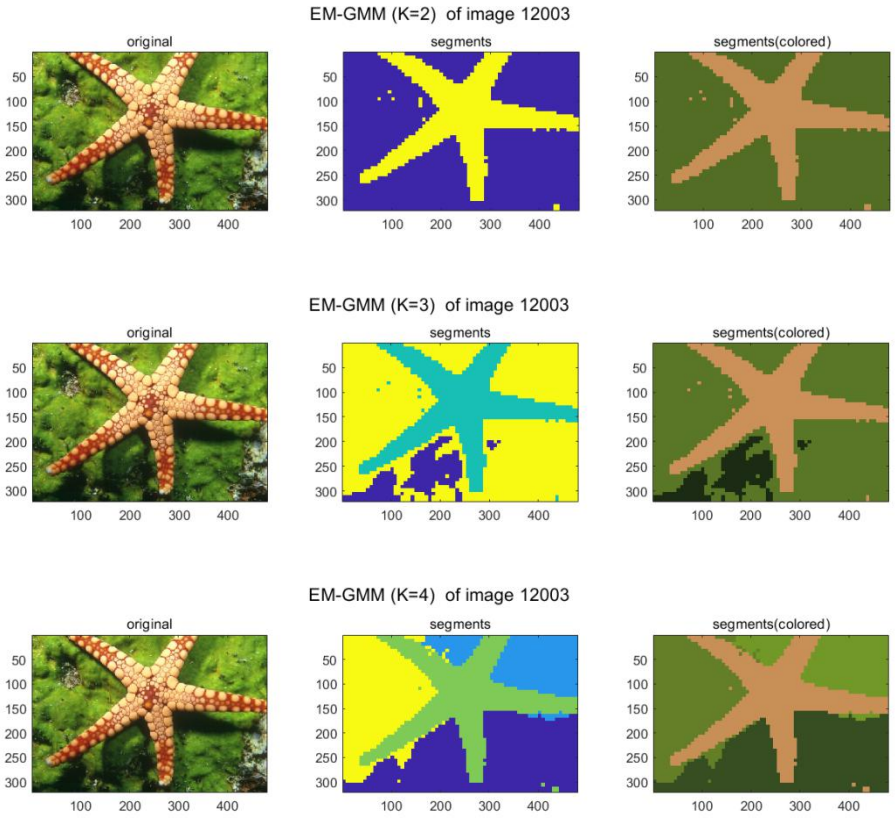


Figure group 2a.1: Image segmentation using K–means algorithm (image 12003)



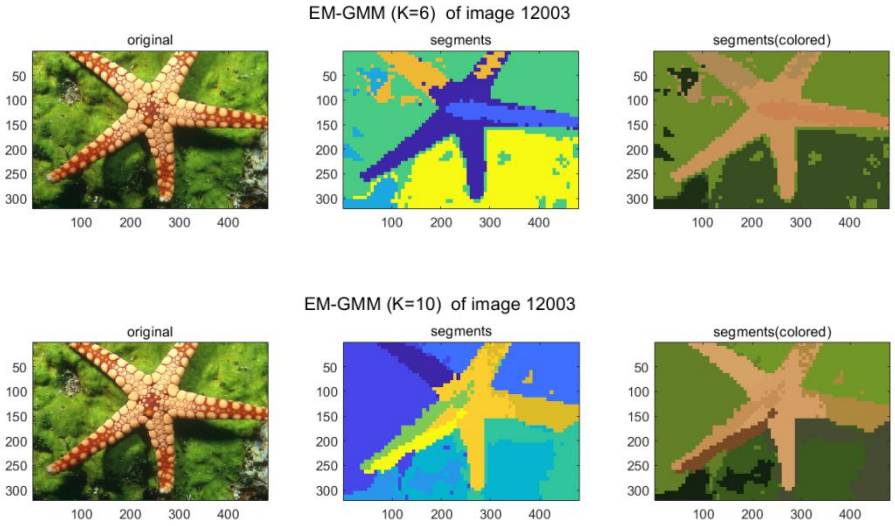
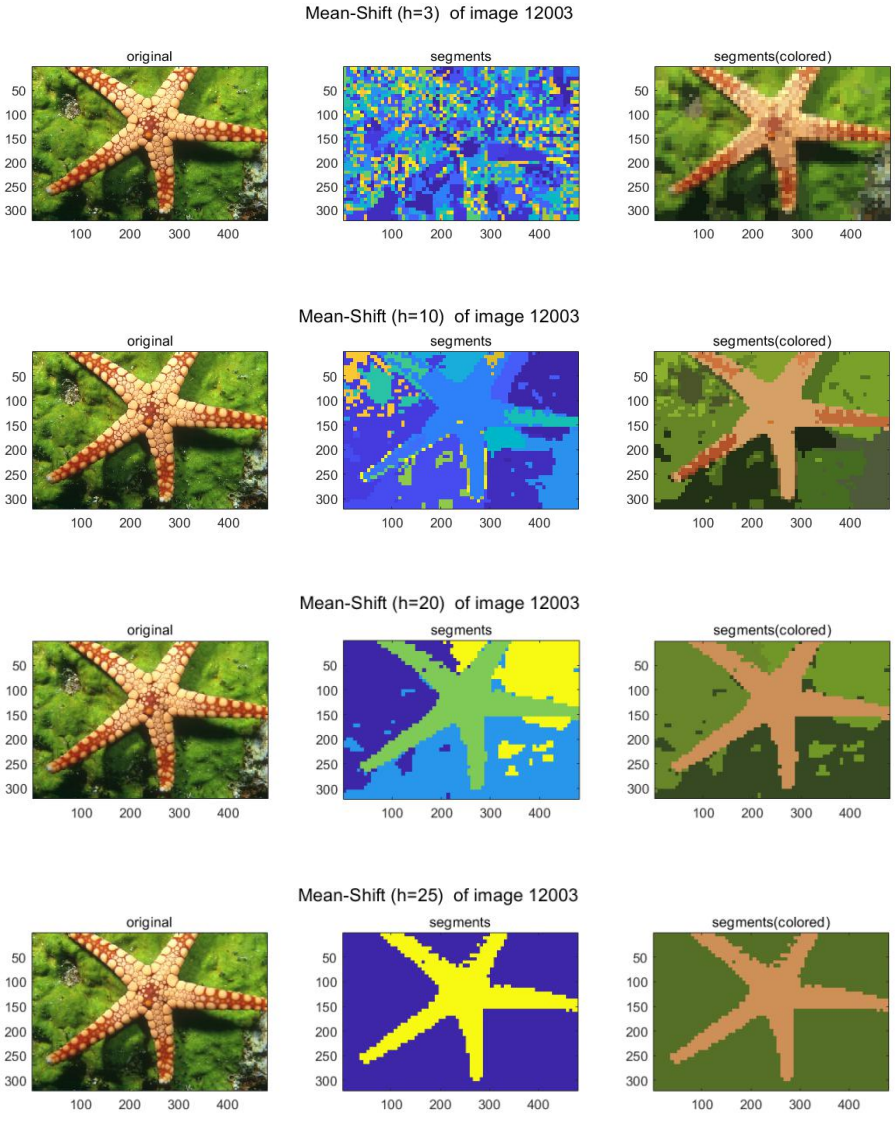


Figure group 2a.2: Image segmentation using EM–GMM algorithm (image 12003)



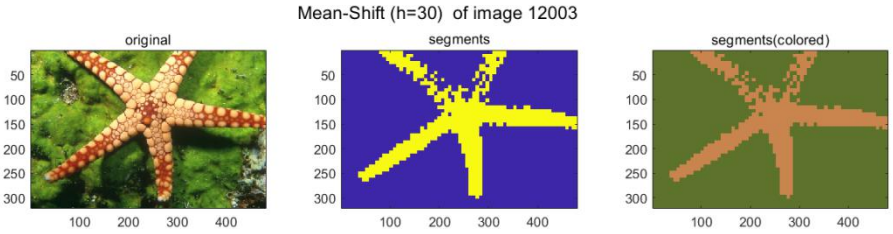


Figure group 2a.3: Image segmentation using Mean–shift algorithm (image 12003) (h=3/10/20/25/30)

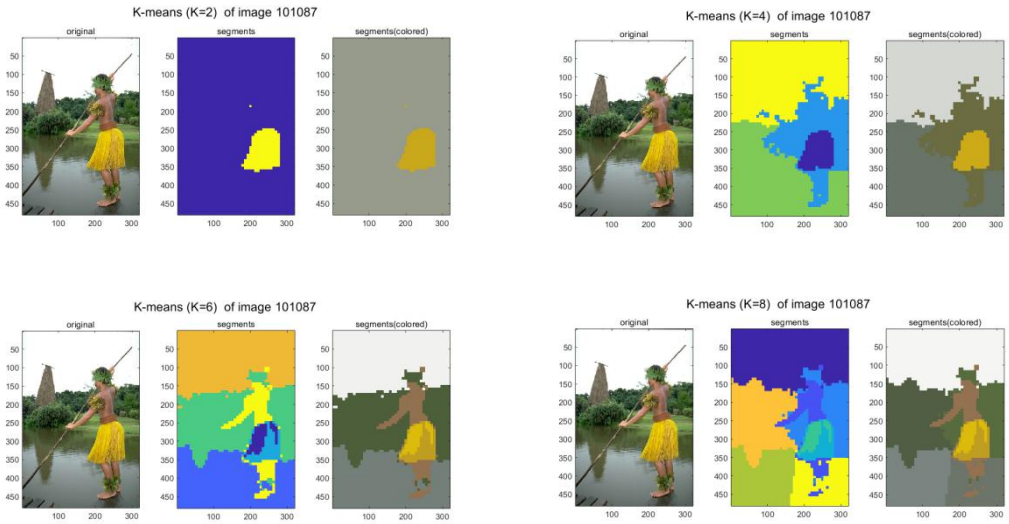
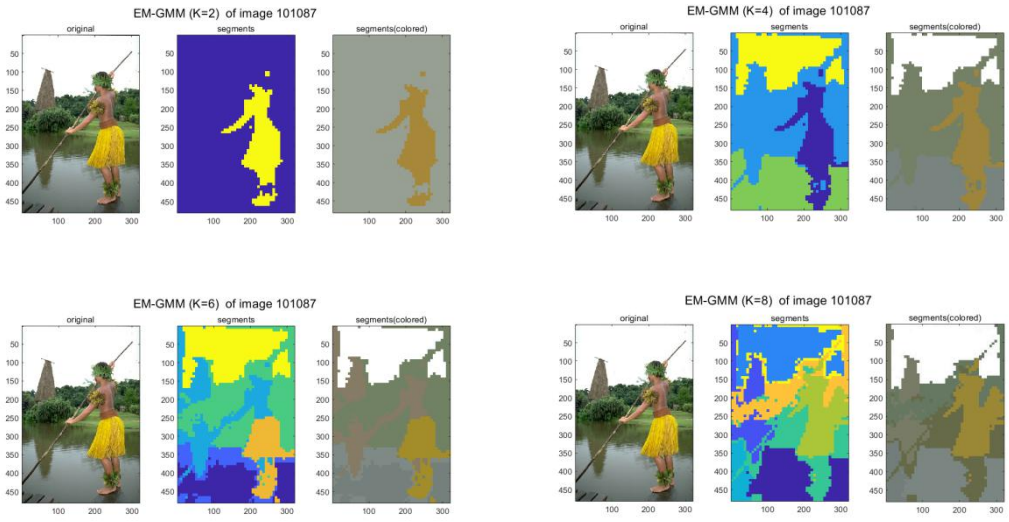
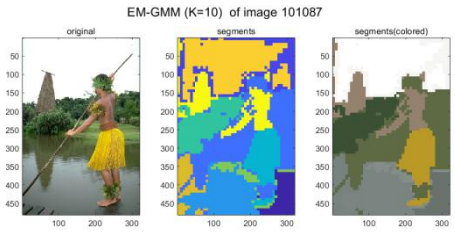


Figure group 2a.4: Image segmentation using K–means algorithm (image 101087) (K=2/4/6/8)





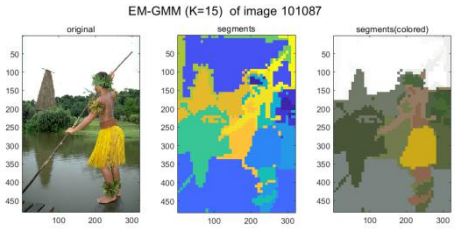
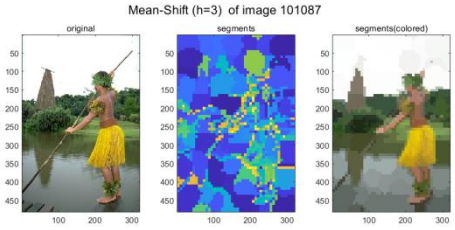
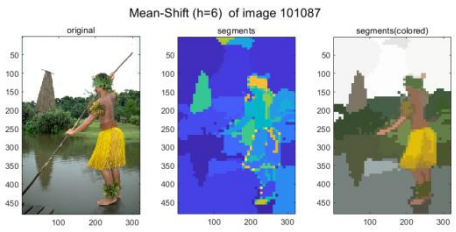
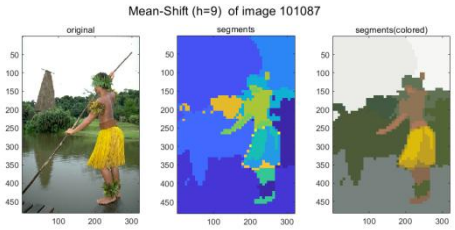
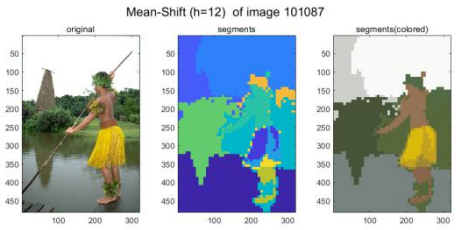


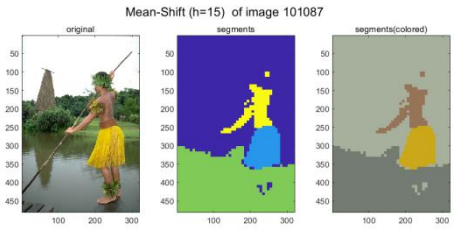
Figure group 2a.5: Image segmentation using EM–GMM algorithm (image 101087) (K=2/4/6/8/10/15)











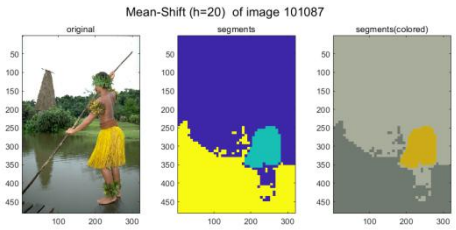


Figure group 2a.6: Image segmentation using Mean–shift algorithm (image 101087) (h=3/6/9/12/15/20)

❖ Analysis

Figure group 2a.1-3 and group 2a.4-6 the segmentation results of three algorithms using image 12003 and image 101087 respectively.

In figure 2a.1, when k is small(k=2), the edge of the starfish is serrated in the segmentation results, which means it is not very accurate. With the gradual increase of k, the details of the segmentation result become more and more and the edge serrations have been reduced. Similarly, in figure 2a.4, when k is 2, K-means can only segment the yellow skirt with the most obvious color, but cannot recognize the whole person.

In figure 2a.2 and figure 2a.5, we can see that unlike K-means, when k is small, EM-GMM can still segment the image well. For example, when k is 2, it can more completely segment the outline of starfish, unlike K-means which produces edge serrations. And in figure 2a.5, EM-GMM can produce a

more complete portrait. With the gradual increase of k, the details of the segmentation result become more and more.

Figure 2a.3 and figure 2a.6 are the results of using Mean-shift. In this algorithm, the only parameter is bandwidth h. The larger h is, the more detail the segmentation results show. However, it is worth noting that when h is too large, the edge of the segmentation result will seem to be corroded. For example, the starfish, when h is 30, the edge of the starfish is eroded by the background cluster.

❖ Conclusion

(i) Qualitatively, which algorithm gives better results?

Qualitatively, from the segmentation result above, EM-GMM is a better model than K-means and Mean-shift. When the cluster number is small (k=2 or h=20), K-means and Mean-shift failed to preserve boundary details between the starfish and the background. As far as the second set of images, EM-GMM can separate the person in a grass skirt and the background, while K-means and Mean-shift link the part of the person with the background.

(ii) How do the results change with different K and h?

In terms of the change of K value in K-means and EM-GMM, the image will be closer to the original picture accompany with the increase of K values but it lost the property of abstraction of segmentation. On the other hand, when the K value is small, the image is abstract and can represent the segments of the original picture.

In terms of the change of h in Mean-shift, larger h values make the result figure more abstract and are divided into less clusters. Small h values produce a blurry version of the original one and contain more details about original figure.

(iii) Which is less sensitive to changes in the parameters?

As can be seen in the figures above, the EM-GMM algorithm is least sensitive to parameter changes. For EM-GMM, even when cluster number k is very small, the segmentation still can represent the shape of original figures.

Mean-shift is more sensitive. The number of clusters is not determined in advance in Mean-shift algorithm and the optimal h may vary from picture to picture. Sometimes, when the bandwidth setting is nor appropriated, the algorithm may output singular matrix when updating estimated parameters and fail to get image segmentation. When bandwidth h is very small or large, the segmentation totally loses the properties of original pictures.

(iv) Some interesting properties or limitations observed about the clustering algorithms.

For Mean-shift algorithm, It doesn't rely assume shape on clusters and has only one parameter choice (bandwidth). Therefore, it is a generic technique for clustering and fit into many different models. The main limitation lies on the selection of window size. But Mean-shift is a much slower algorithm compared with K-means.

For K-means algorithm, despite of its efficiency and simplicity, it lacks consistency, which means clustering results mainly rely on Initialization sets.

For EM-GMM algorithm, When the number of clusters is small, the algorithm has a significant advantage over the other two algorithms. But it needs more computation time and could easily fall into the local maximum.

◆ (b) Allowing different Scaling of the Features

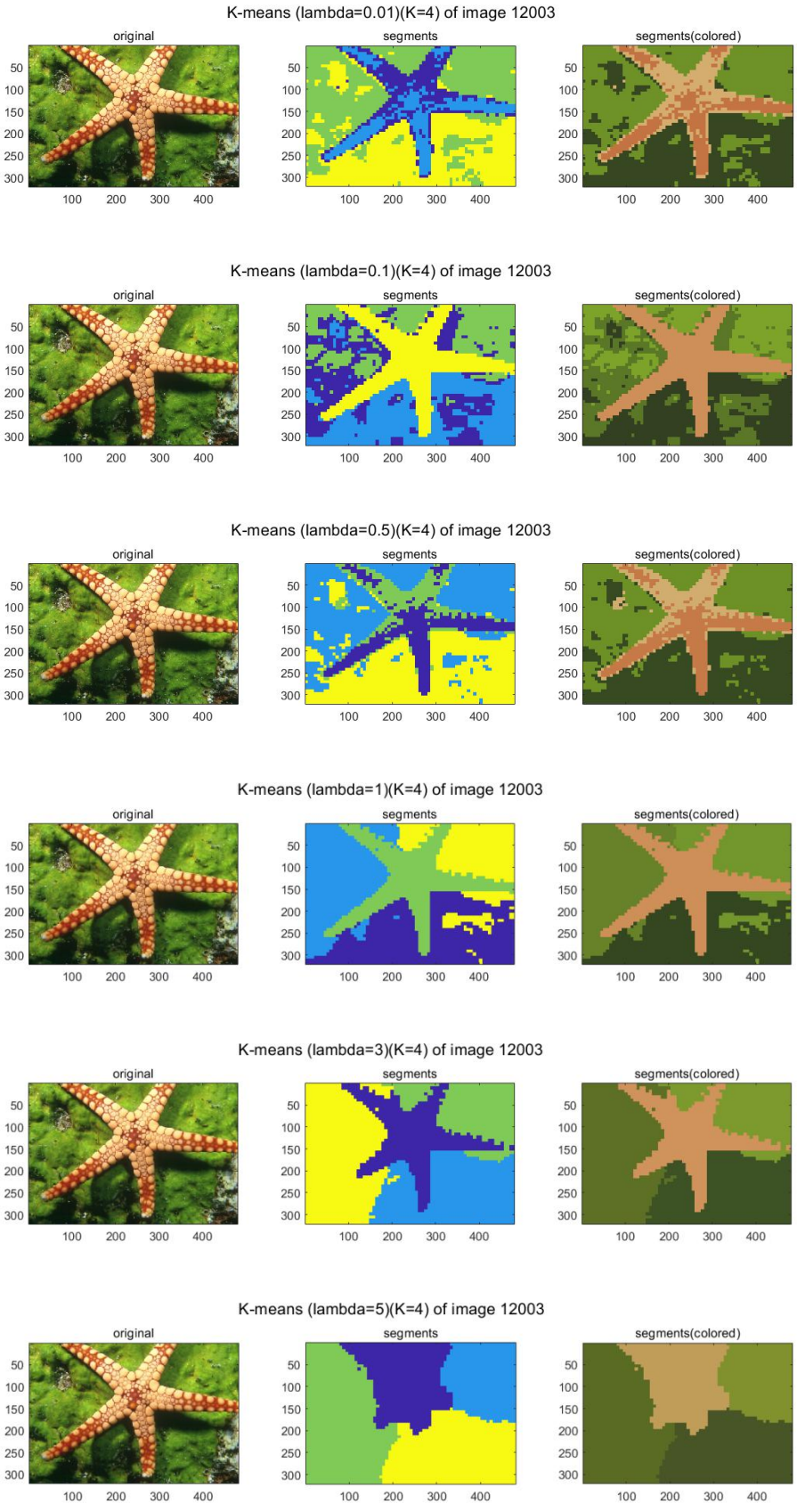


Figure group 2b.1: K–means with different scaling values(image 12003)

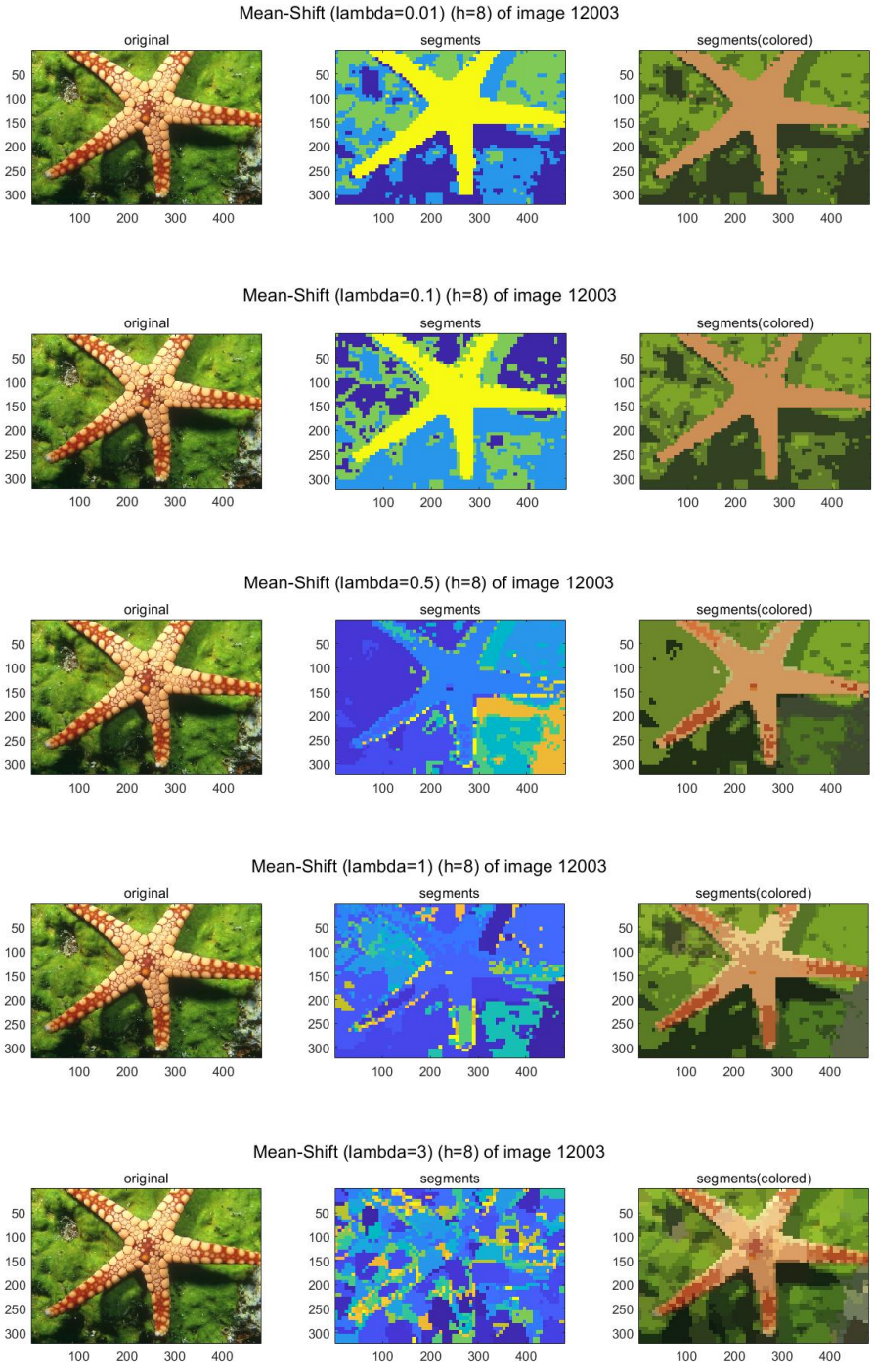
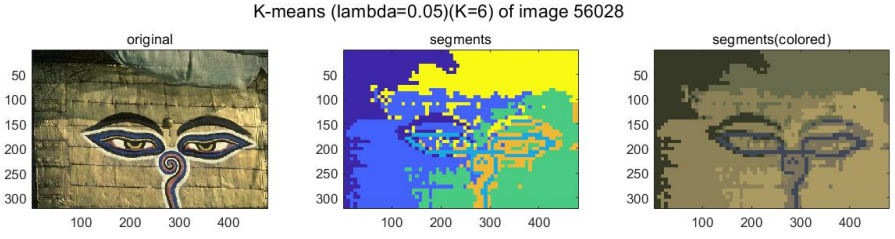


Figure group 2b.2: Mean–shift with different scaling values(image 12003)



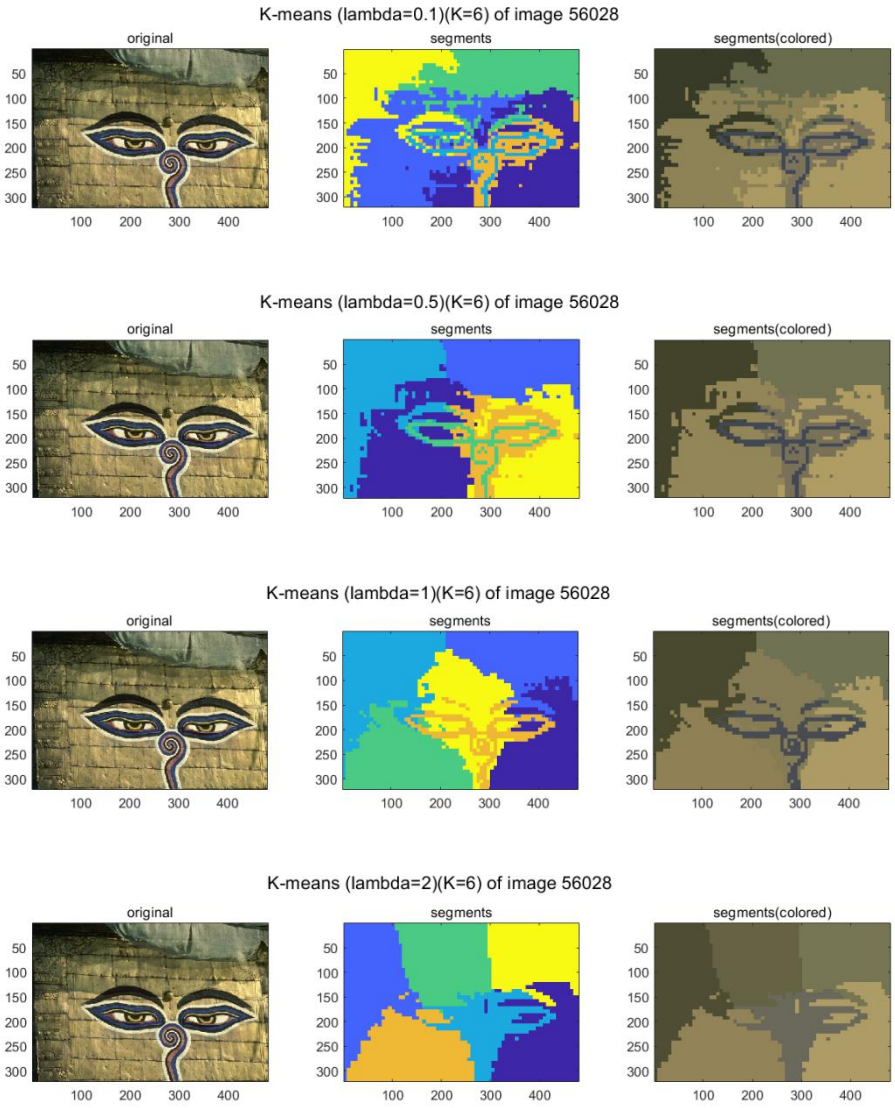
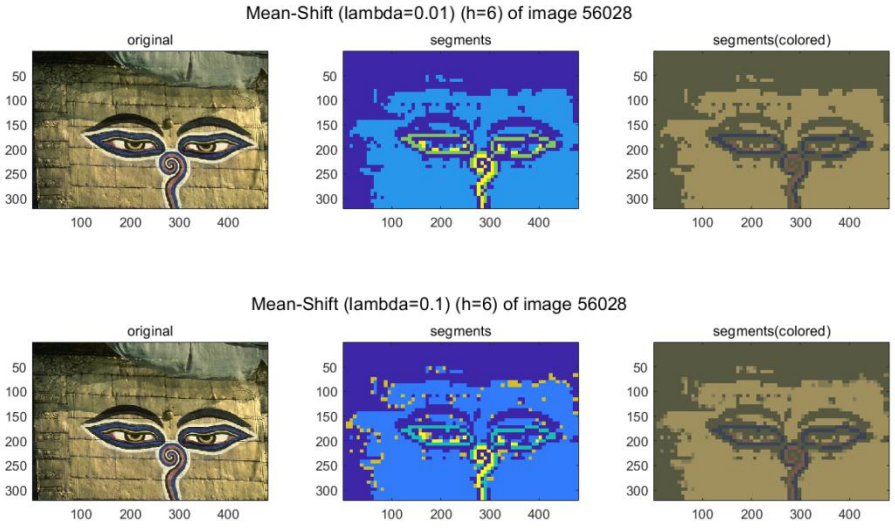


Figure group 2b.3: K–means with different scaling values(image 56028)



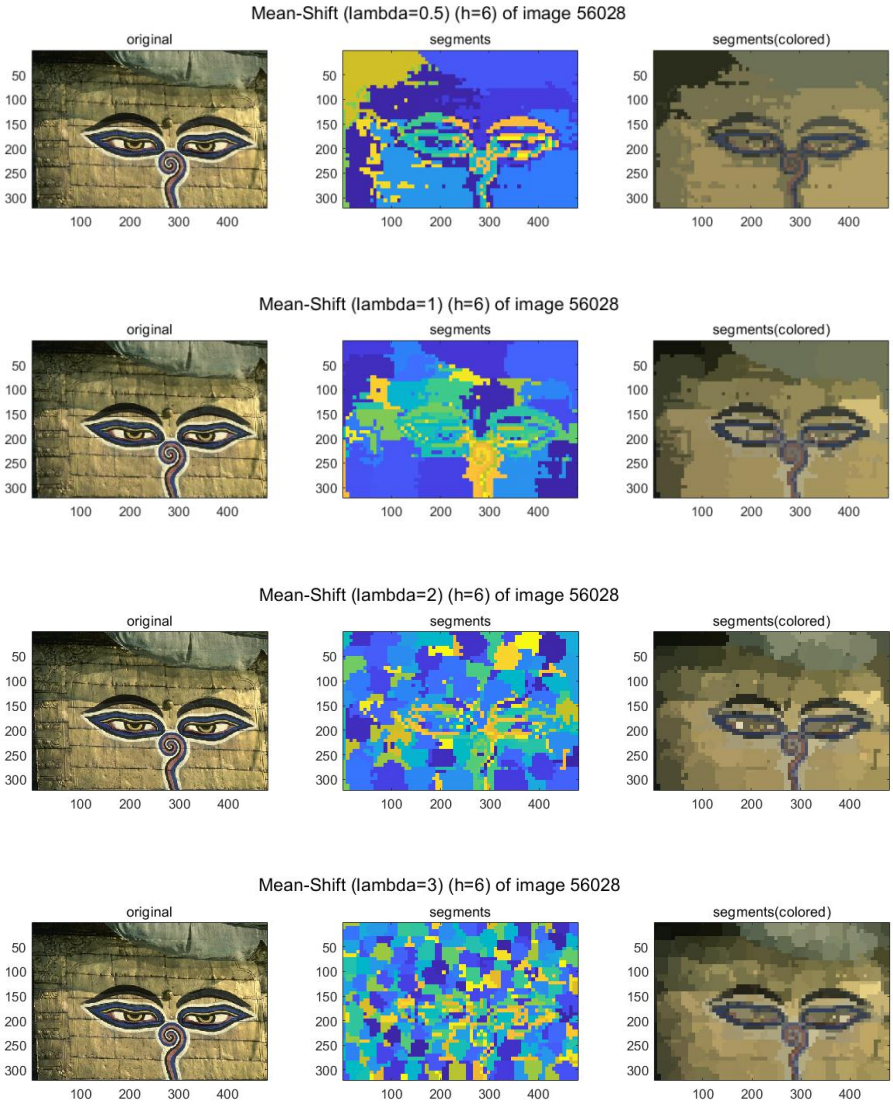


Figure group 2b.4: Mean–shift with different scaling values(image 56028)

❖ Analysis

In this experiment, lambda λ is defined as the the ratio of pixel location and chrominance values, it controls the weighting between the feature types. For algorithm Mean-shift, the bigger lambda is, the smaller bandwidth for the pixel locations (h_p) is.

In figure group 2b.1 and 2b.3, by comparing the segment result of lambda=1 with other results, we can find that the greater the value of lambda, the more likely those belonging to the same cluster are to lump together. For example, when lambda is 5, each cluster is like a lump, and even the most basic segmentation cannot be completed. On the other hand, when lambda is small, for example, lambda is 0.01, we can see that even if pixels of the same color are far apart, they can still be grouped together. From this perspective, the image segmentation performance is improved compared to that without scaling.

In figure group 2b.2 and 2b.4, lambda controls the ratio of the two bandwidths(the bandwidth for the pixel locations h_p and the bandwidth for the color features h_c). When decreasing the weights

on chrominance values, outputs tend to be more sensitive on pixel locations, which leads to the clustering of nearby data points. In other words, when raise the weights of chrominance values, result tends to group the data points with the same color. When the location of pixels is scaled up (which means increasing lambda, and it also means using a smaller bandwidth of the pixel locations h_p compared with the color bandwidth) image will outputs more clusters and the segmentation is more sensitive to the pixel location, which means pixels nearby tend to group together. If the location is scaled down (which means decreasing lambda, and it also means using a larger bandwidth compared with the color bandwidth), image segmentation is more sensitive to pixel color. When the location bandwidth is larger, these regions with similar colors will come together regardless of the difference of pixel location.

❖ Conclusion

In conclusion, by adjusting the feature scaling to the appropriate value, we can get a better segmentation result.