Junyi CHEN

CS5487

2017/11/04

CS5487 Programming Assignment 2: clustering

# PART 1 pOLYNOMINAL FUNCTION

## implementation

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

## Run the algorithms on the three synthetic datasets

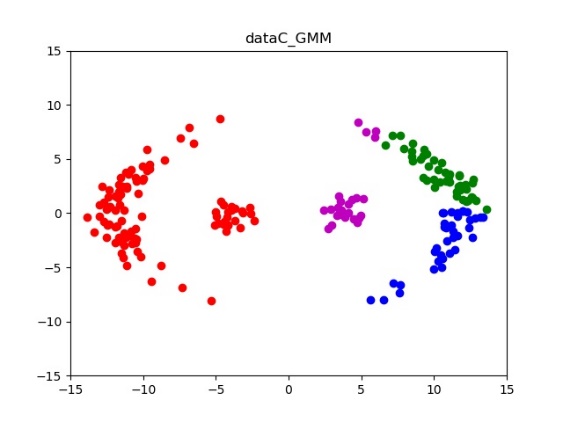
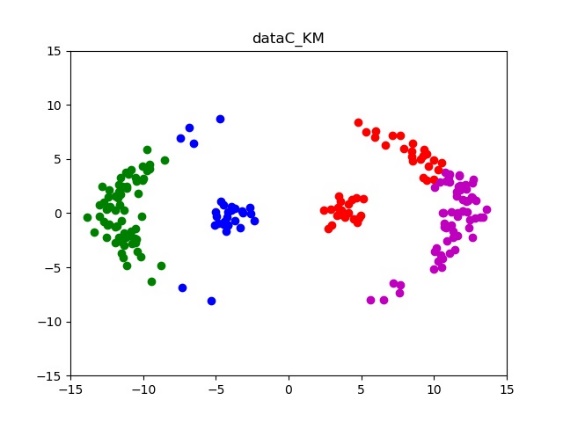
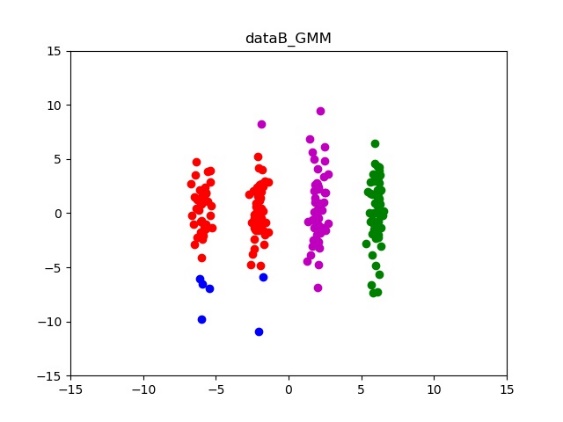
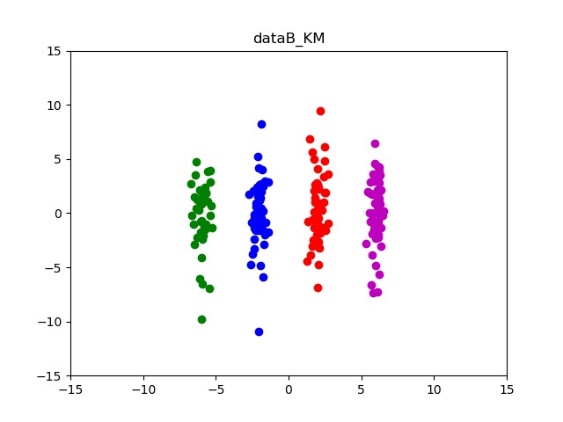
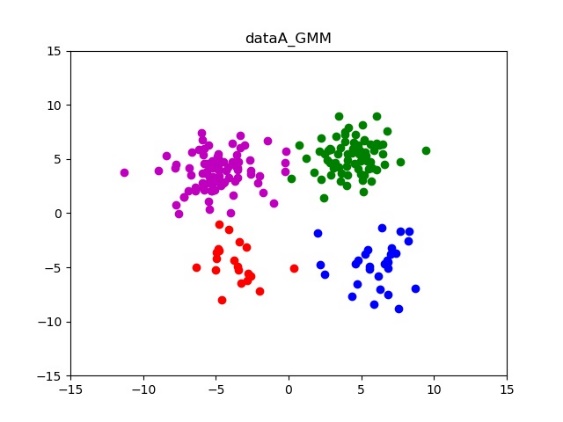
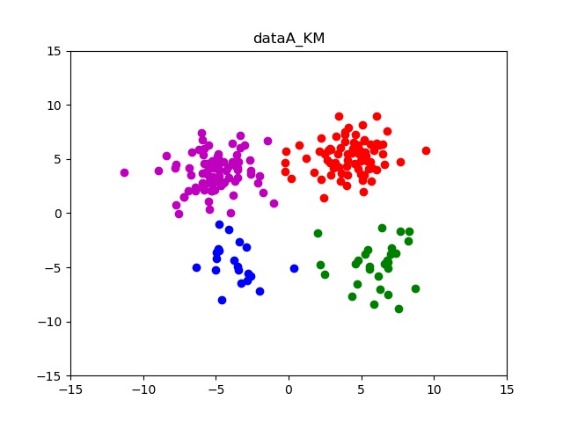


Figure 1: Predict of different clusters, the figures are K-means (left) and Gaussian mixture model (right)

In terms of K-means (KM) and Gaussian mixture model (GMM) the performance of data A is similar, both of them can correctly cluster four sets of data points. In terms of data B, K-means out puts better performance than Gaussian mixture model because GMM fail to discriminate the first and the second column wise data points. However, both KM and GMM do not work well in data C, KM cannot handle with points stay in the margin from the leftmost cluster. While GMM fail to separate the first and the second from left hand side.

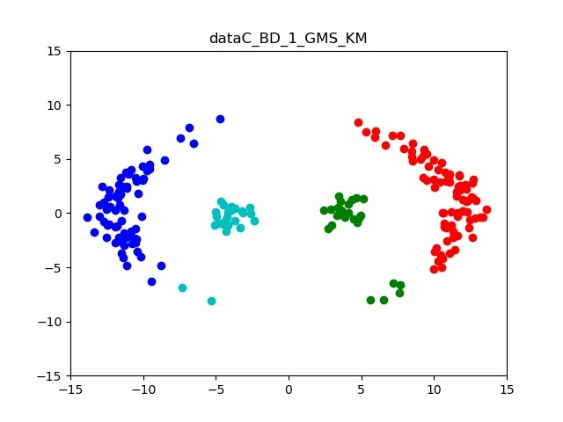
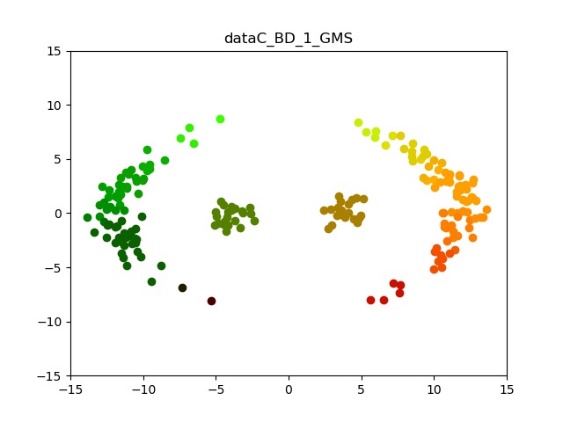
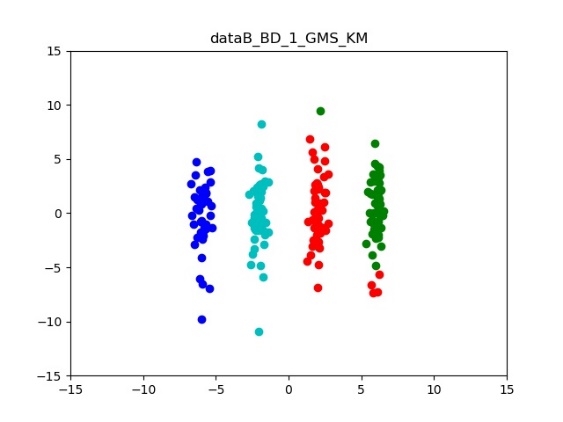
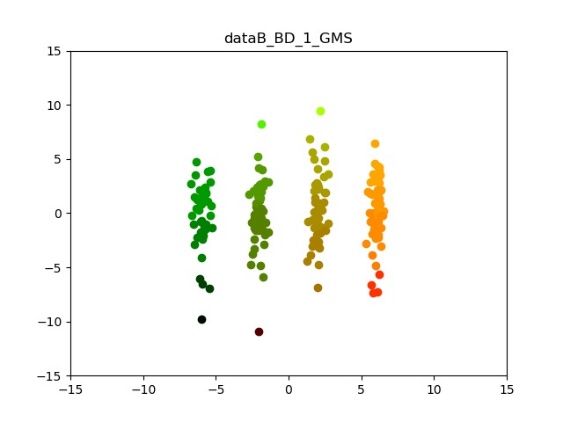
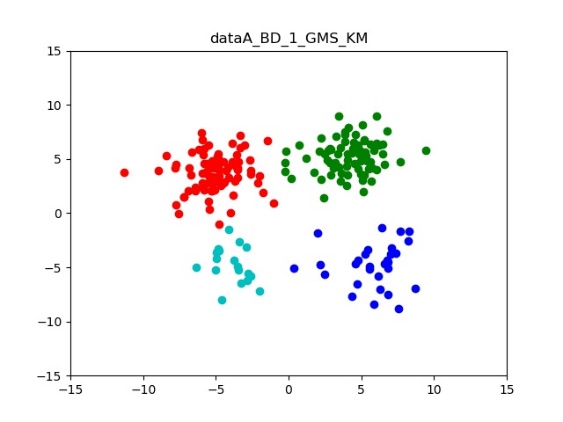
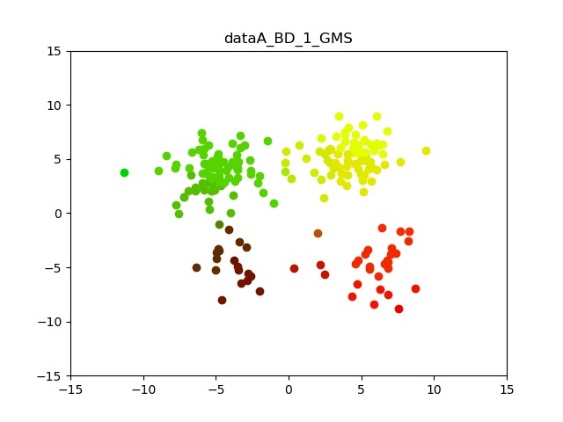


Figure 2 Predict of mean shift clustering with bandwidth equals to 1, color is displayed according to the value of local peak (left) and plots of local peak values clustered by K-means(right) to limit the cluster number of mean shift

Figure 2 are cluster result of mean shift clustering (MS). Because MS calculate the value of local peak, not represented by cluster center. To visualize the clustering result, the data point in figure 2 are colored by the value of local peak. Hence, points converge to the same peak will have similar color in figure 2. We can see that MS is working good on data A for the colors in 4 clusters are well separated. In terms of data B and C, colors of the 4 clusters can basically be discriminated. Anyway, from the plot of data B of MS, data points from the bottom seems to form their own cluster like GMM. Plots form right hand side is picture we apply KM on the output local peaks of MS. The reason why we do the second clustering is to limit the number of clusters. Qualitatively, MS is not as good as KM in data A and B but it works well in data C.

In summary, distance based KM works good in data A and B, and it is easy to implement. But the performance is not good using KM when the densities between clusters are not equal like data C. Also, cluster number is fixed and affiliation of data points is hard. One point belongs to one clusters only. GMM can output probability of cluster affiliation, but the cluster may not be as precise as KM and the initialization of estimated values potentially affect clustering result. Cluster number of GMM is also fixed. MS works well in our data A, B and C, and the representation is not by cluster center so it is flexible. But the limitation is that the stagey of cluster prototyping will affect the performance of cluster and the speed of GMM is much slower than KM and GMM.

## SENSITIVITY OF MEAN SHIFT BANDWIDTH

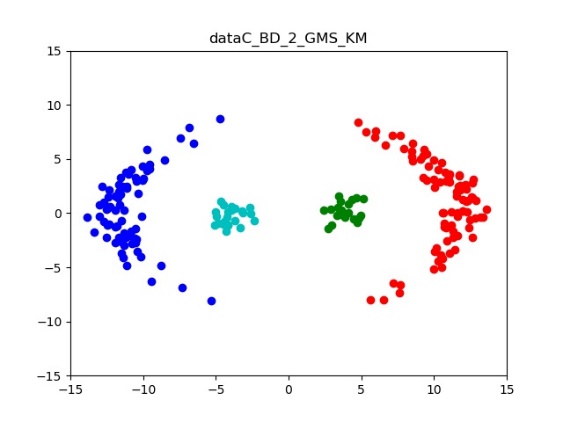
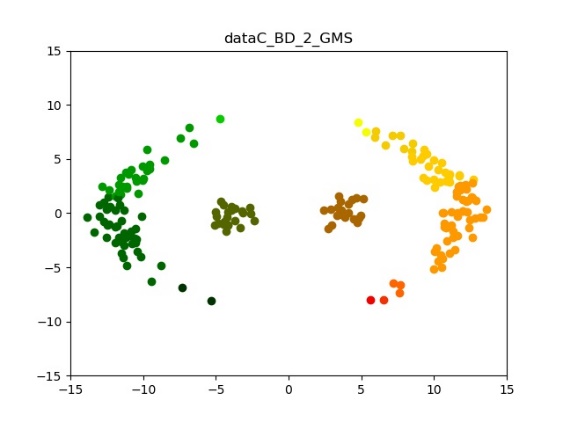
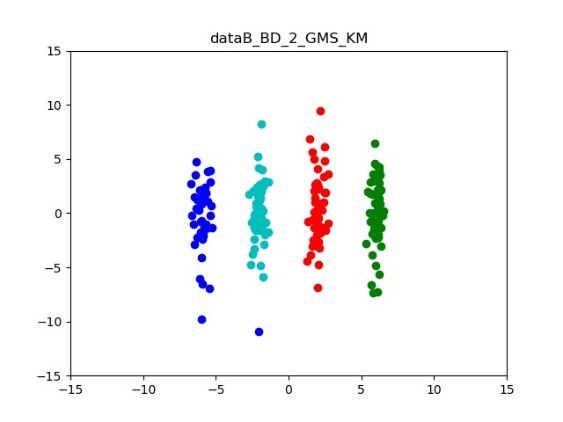
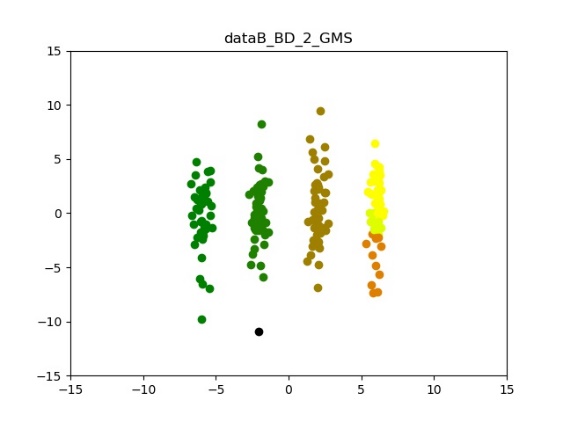
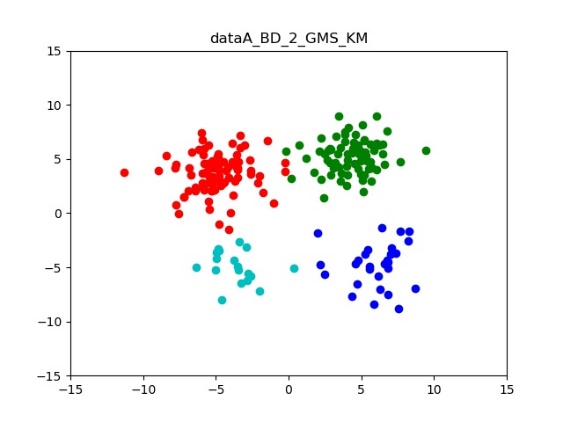
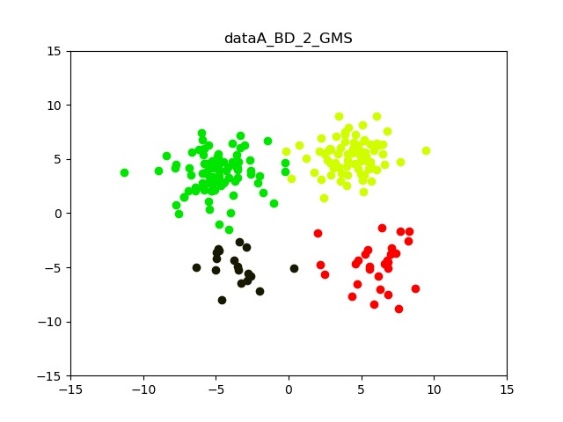


Figure 3 Predict of mean shift clustering with bandwidth equals to 2, color is displayed according to the value of local peak (left) and plots of local peak values clustered by K-means(right)

To analysis the sensitivity of MS bandwidth. We tested different bandwidth values on three datasets. By increasing the bandwidth to 2 in Figure 3, the clustering performance is better than 1. Using KM to limit the cluster value to 4, data C is clustered without fault. But when the bandwidth increases to 10 shown in Figure 4, data point on data A and data B cannot be distinguished while data C can be divided into 2 clusters. When the bandwidth increases to 20, all data set are clustered in one single cluster and failed to form valid clusters.

In summary, when the bandwidth increases, number of clusters will decrease, vice versa. In the presented problem, we have prior know ledge of cluster number so this prior knowledge can help us to generate good result using MS algorithm. And one optimum we tested is bandwidth equals to 2 and using KM to limit the number of local peaks.

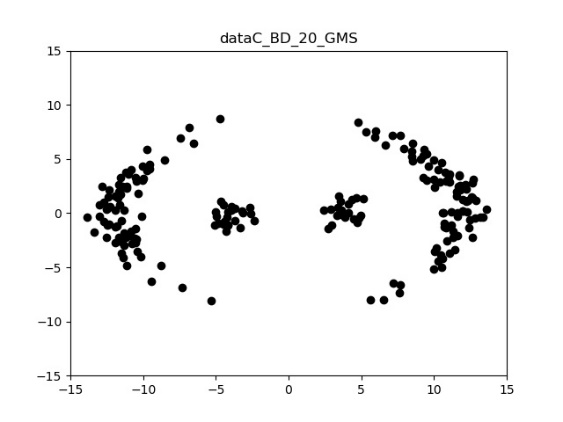
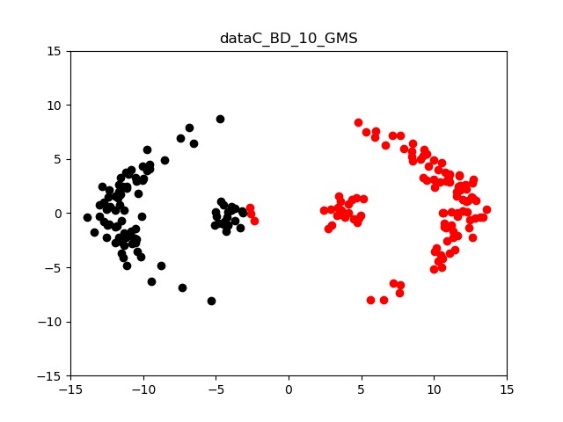
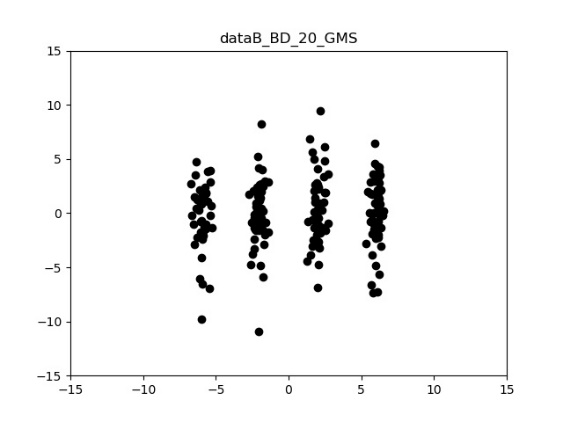
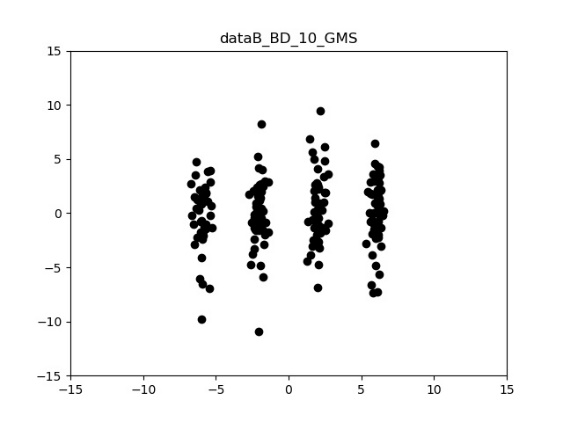
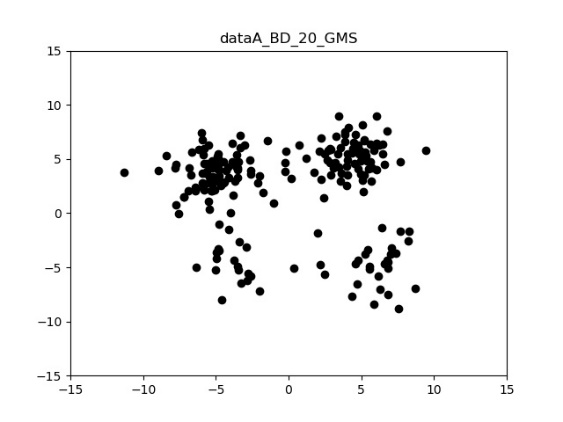
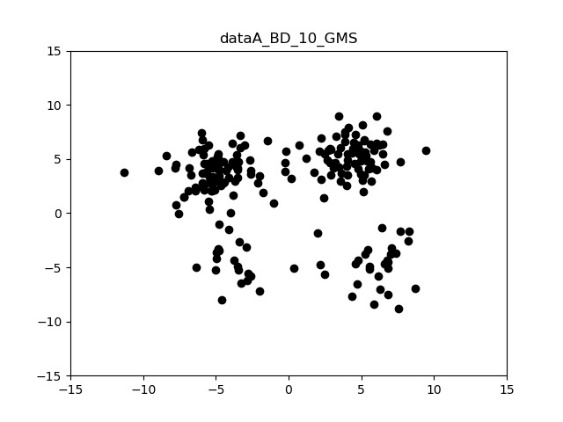
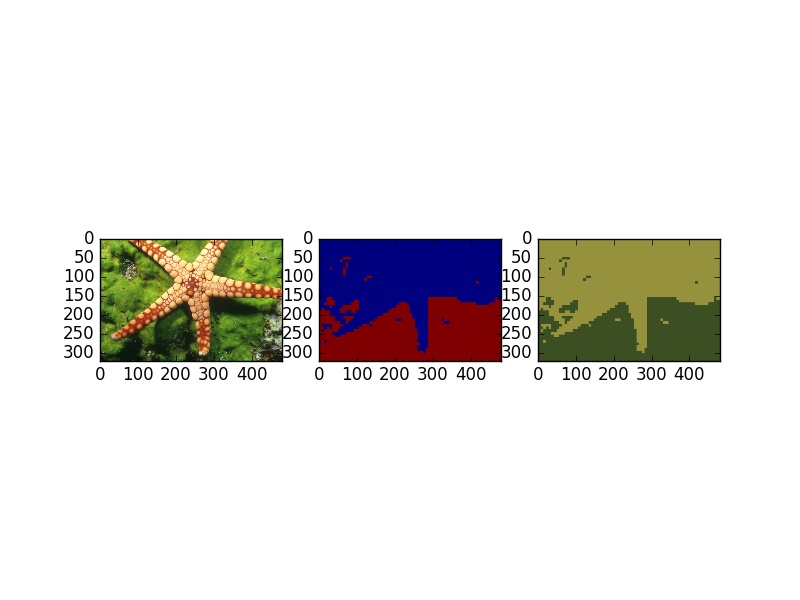
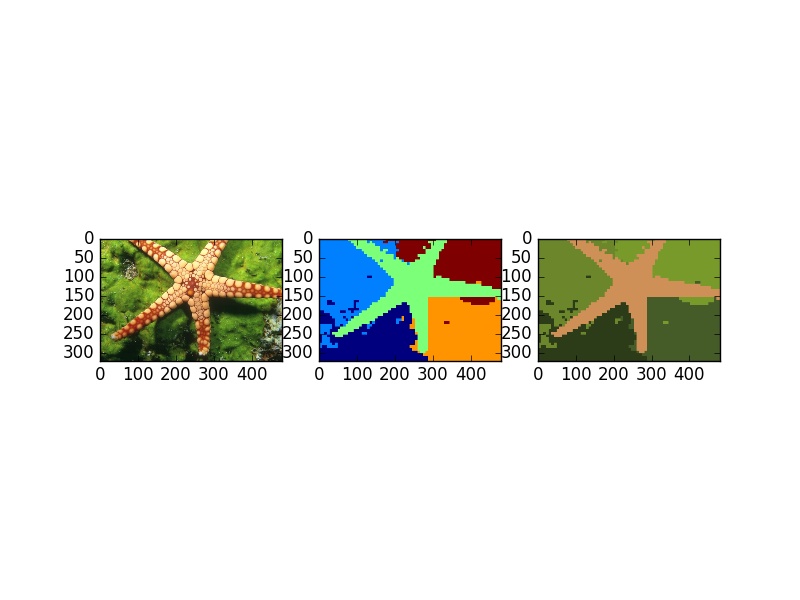


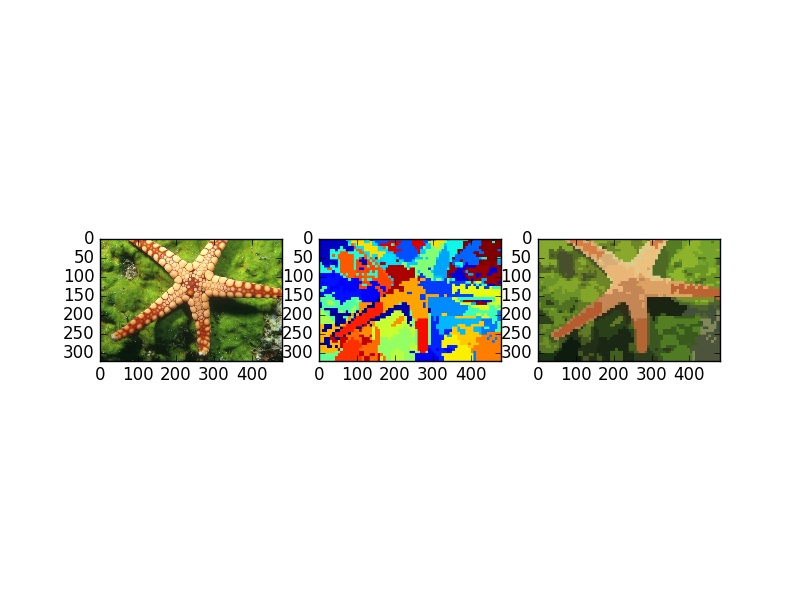
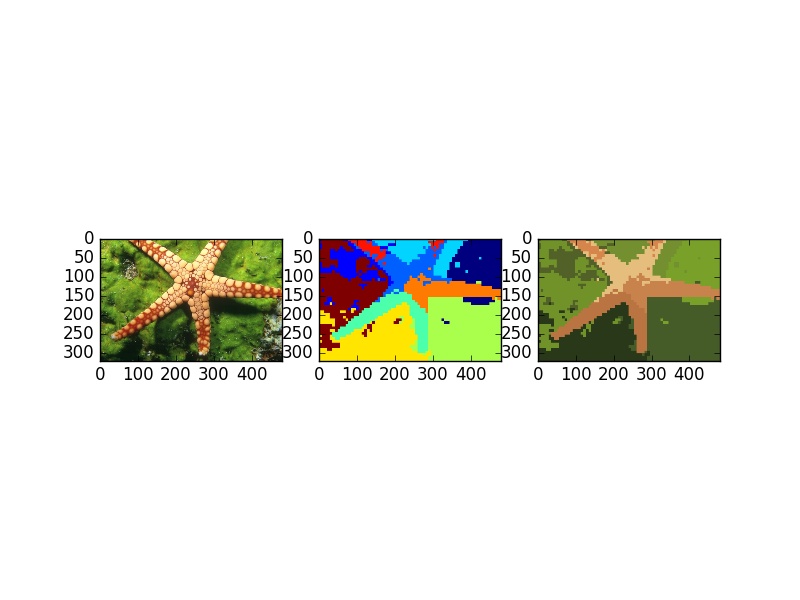
Figure 4 Predict of different bandwidths, the figures are bandwidth = 10 (left) and bandwidth = 20 (right)

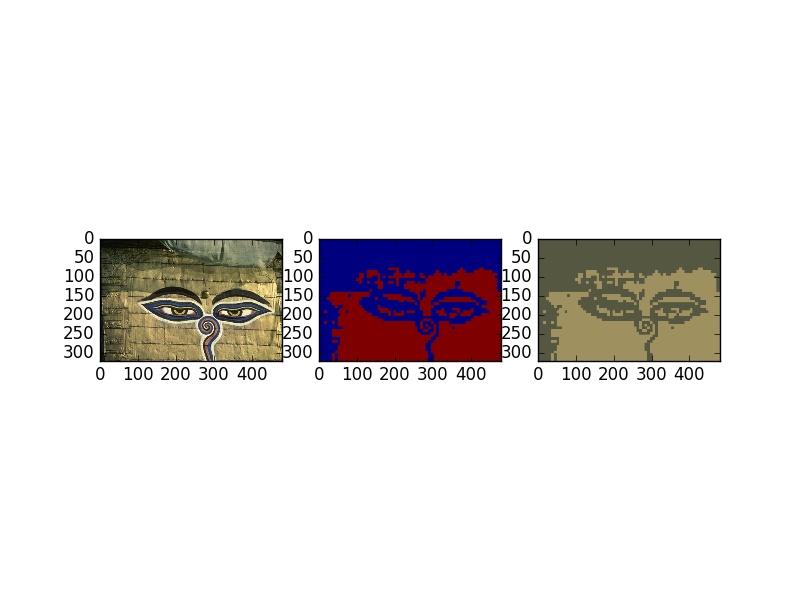
# Part 2 A REAL-WORLD clustering problem – image segmentation

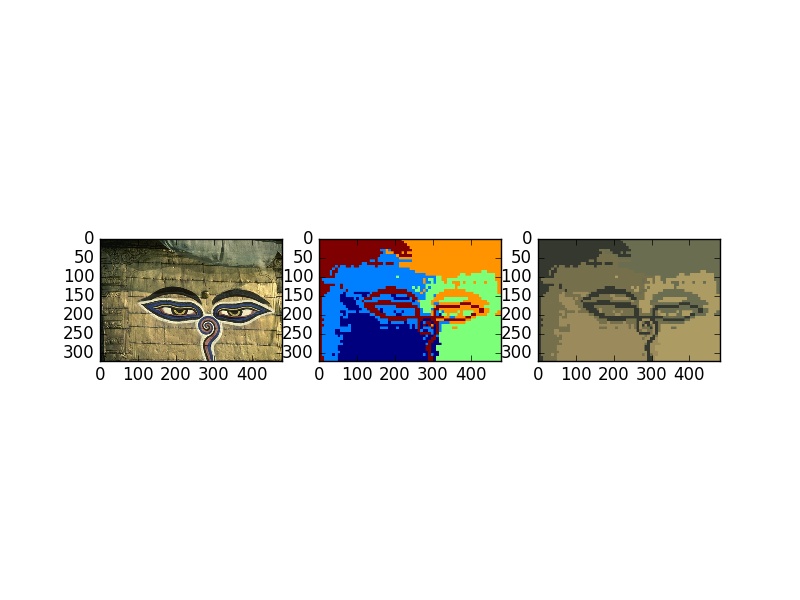
## result comparison, hpyer-parameters and properties

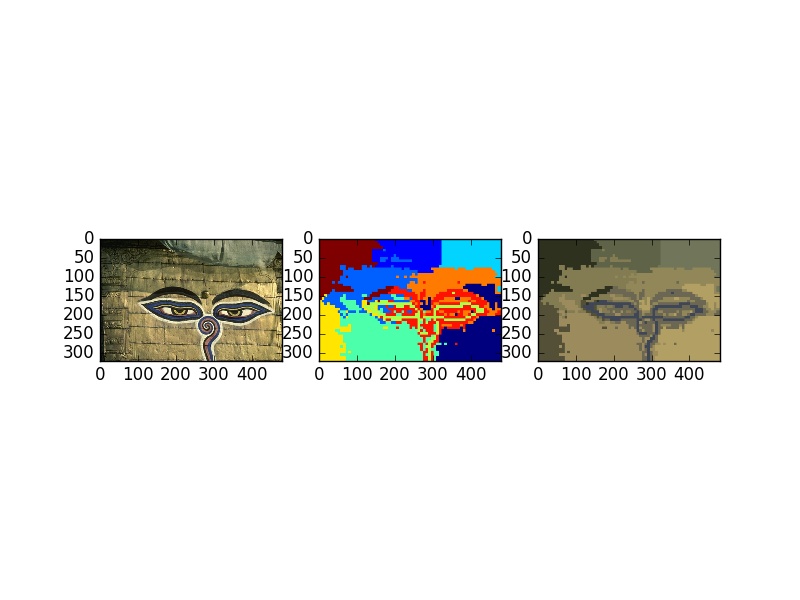












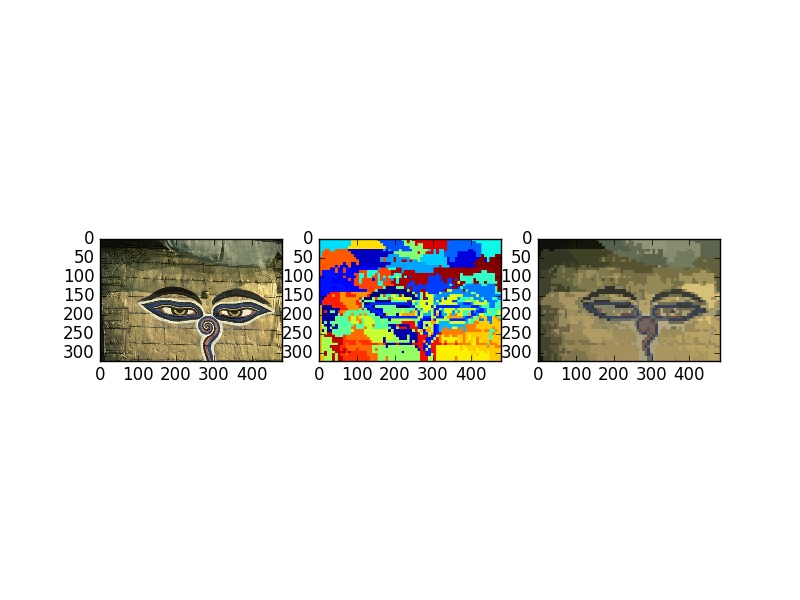
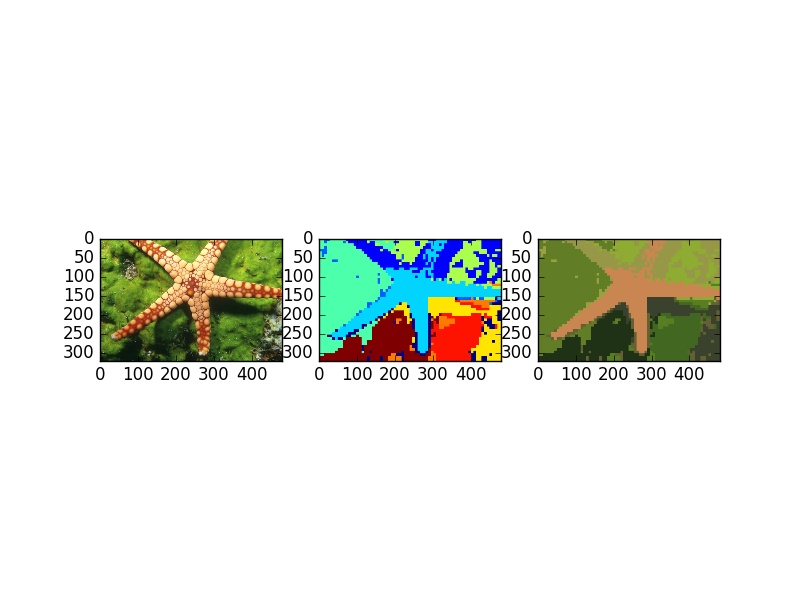
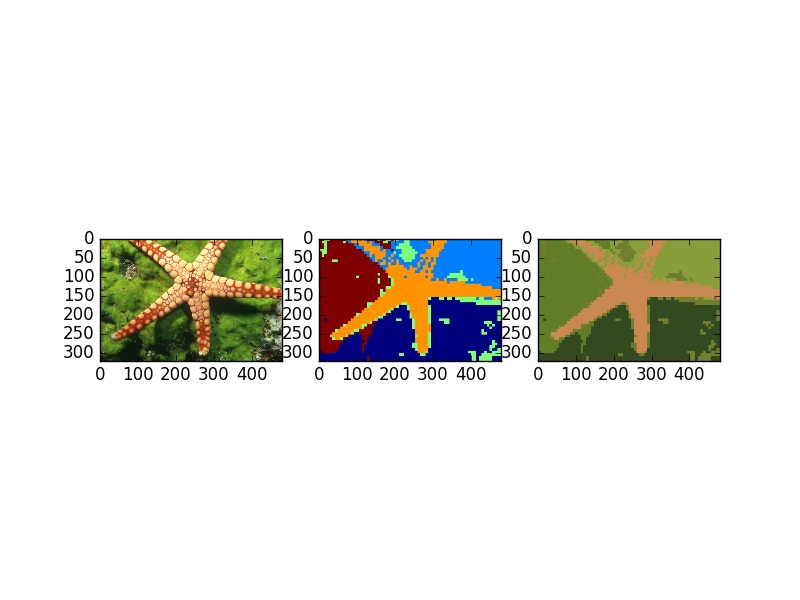
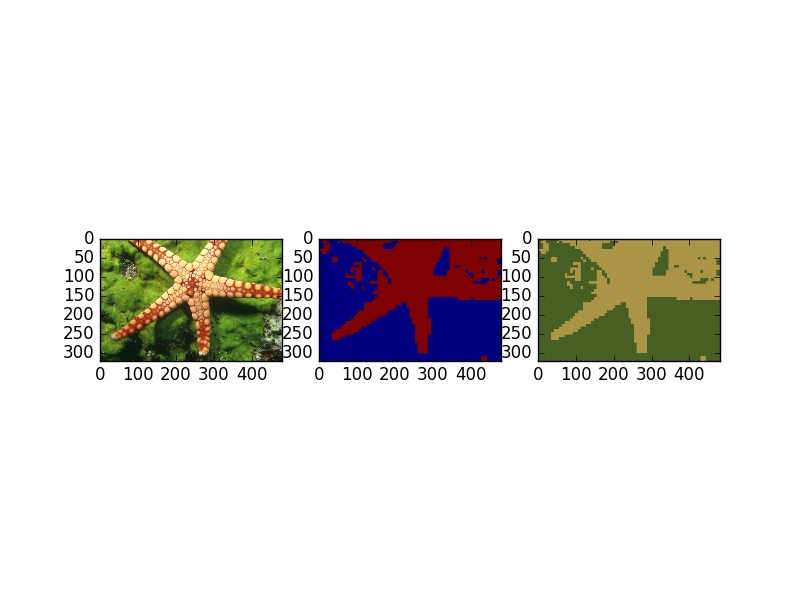
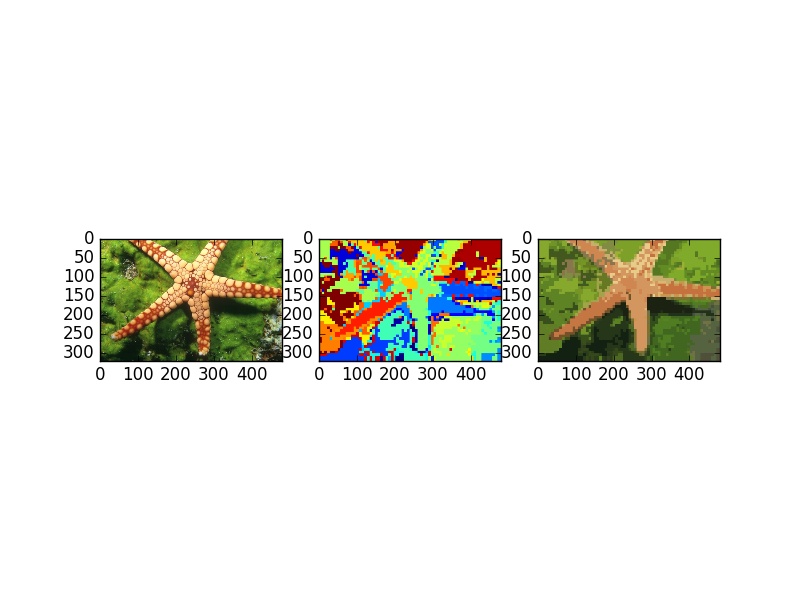


Figure 5 Image segmentation using K-means (k= {2,5,10,50})





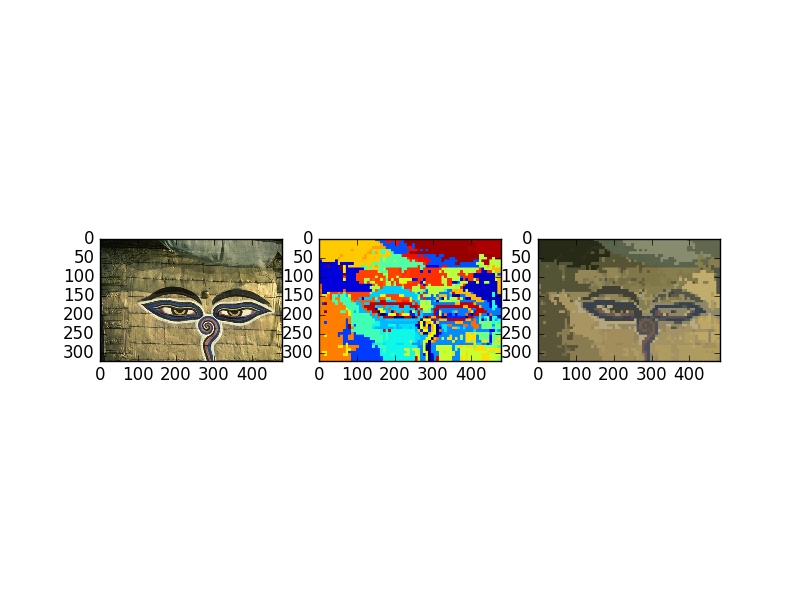
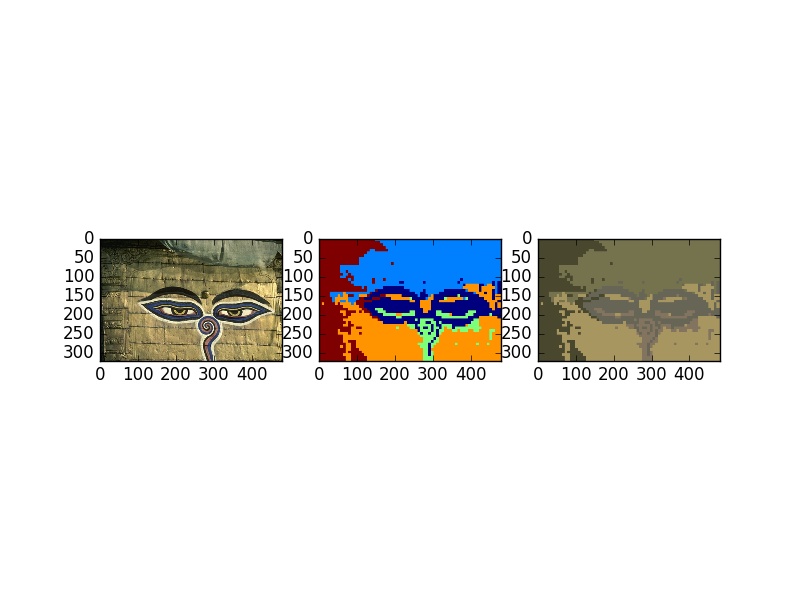
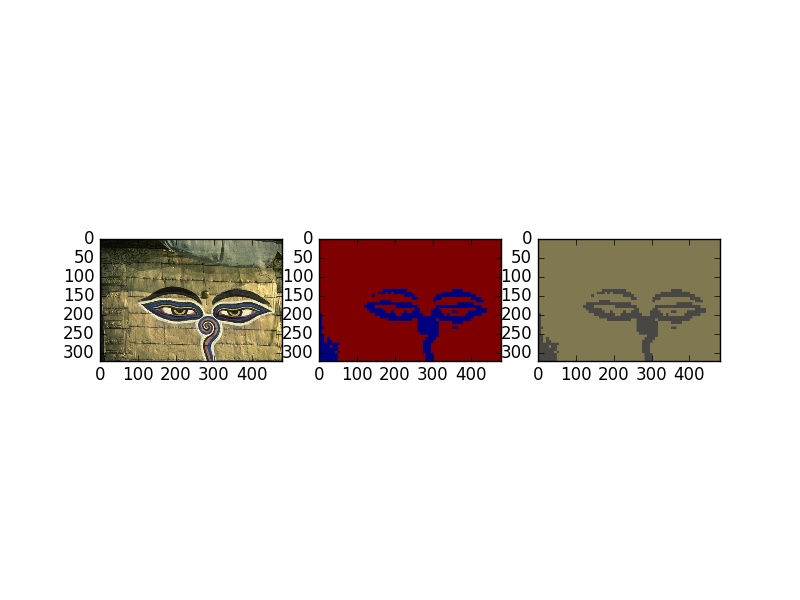
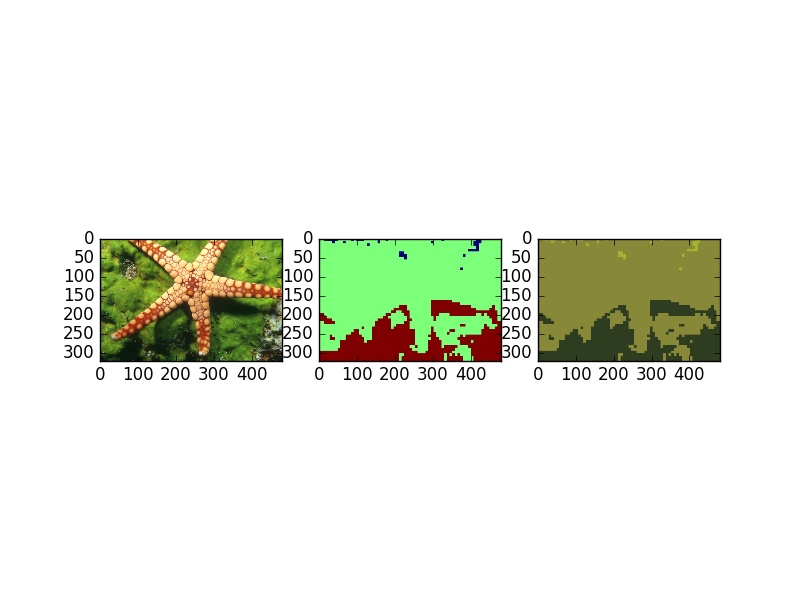
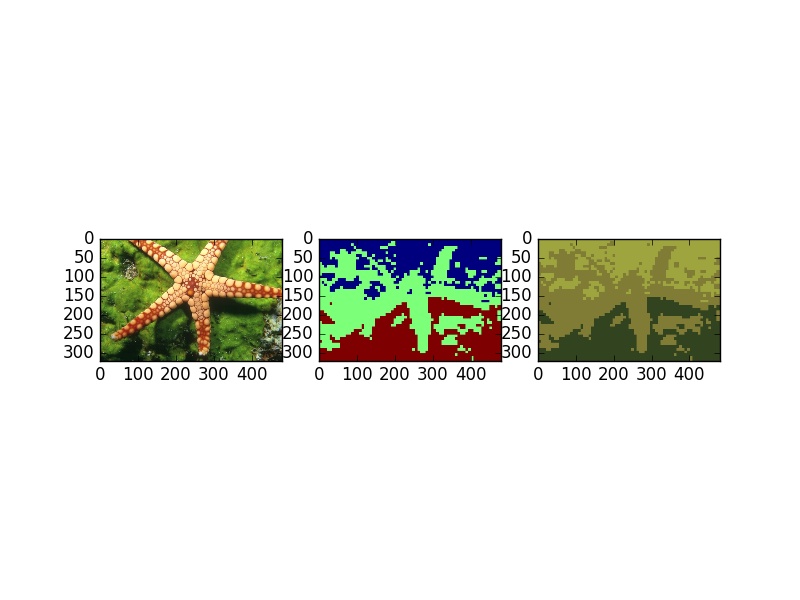
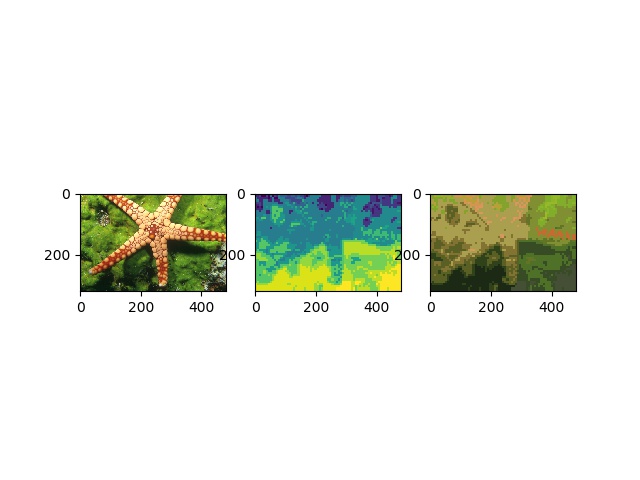
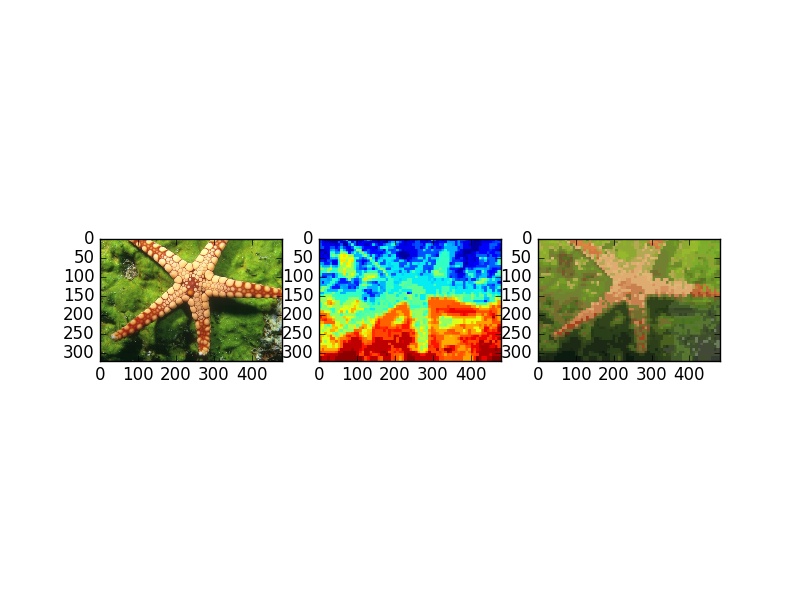


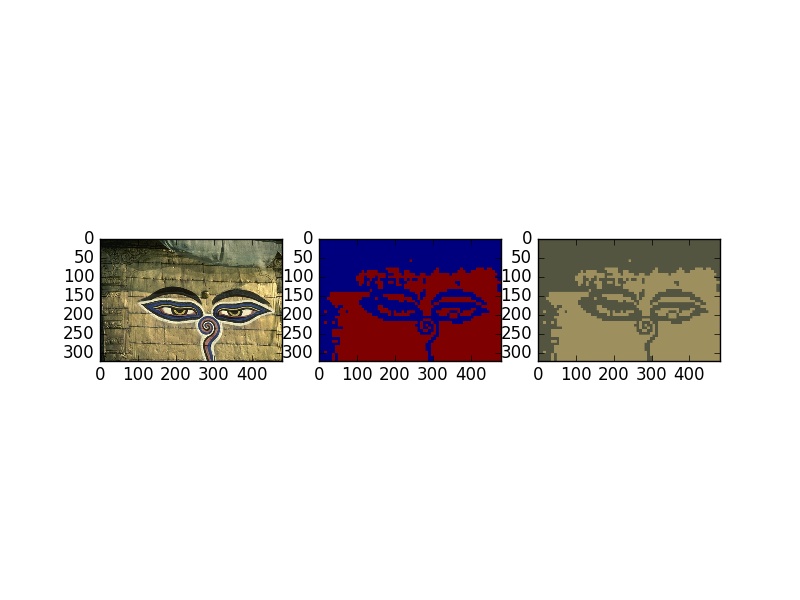
Figure 6 Image segmentation using Gaussian mixture model (k= {2,5,10,50})

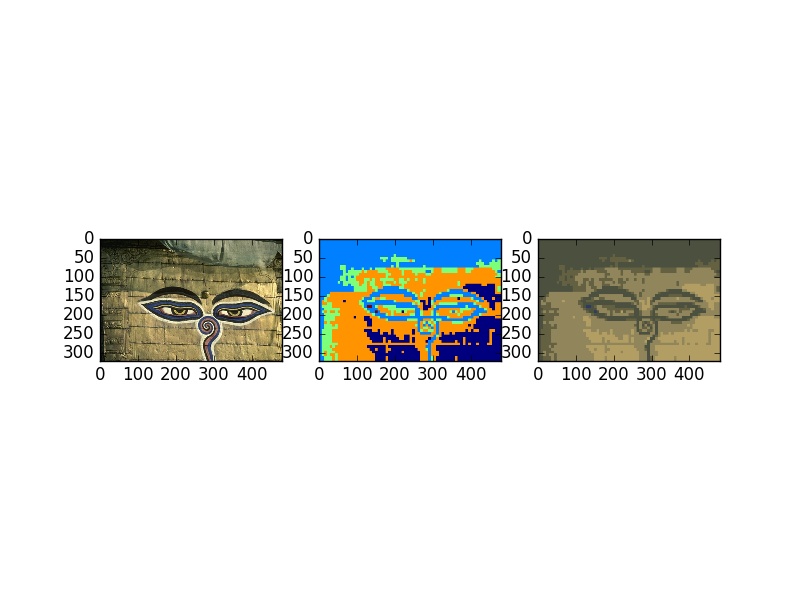


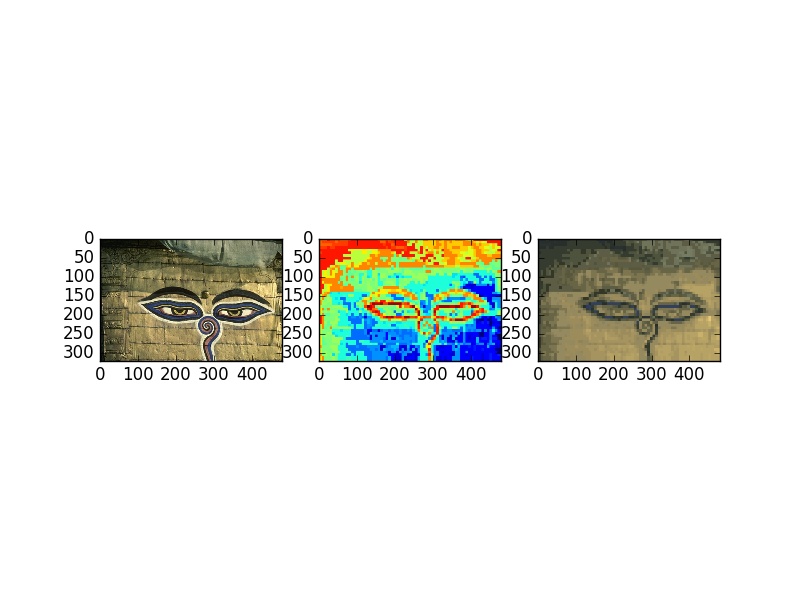












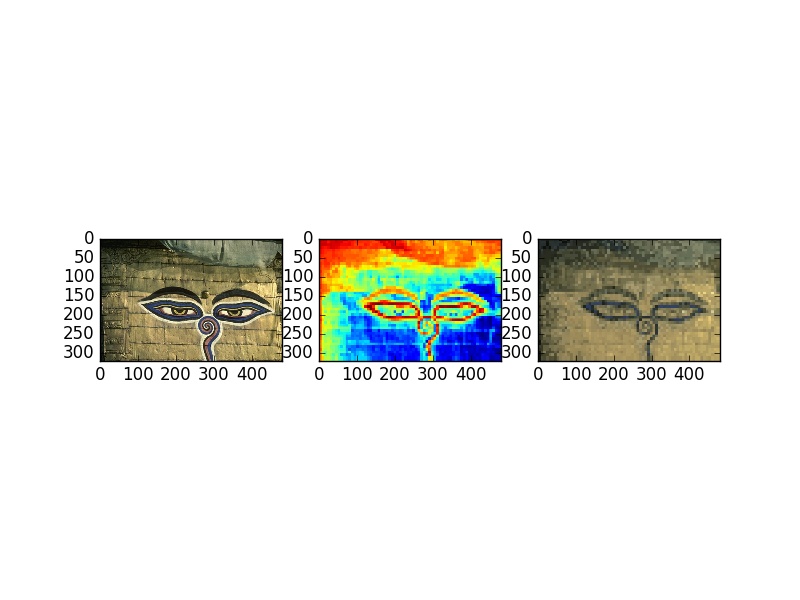


Figure 7 Image segmentation using mean shift model, h = {3.5,4,4.25,4.5} (fist set of pictures), h = {10,6,5,4} (second set of pictures)

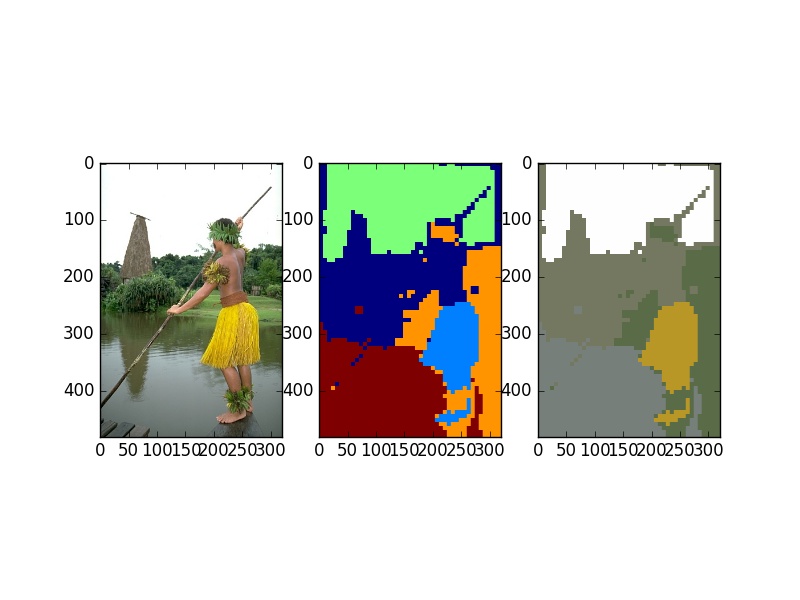
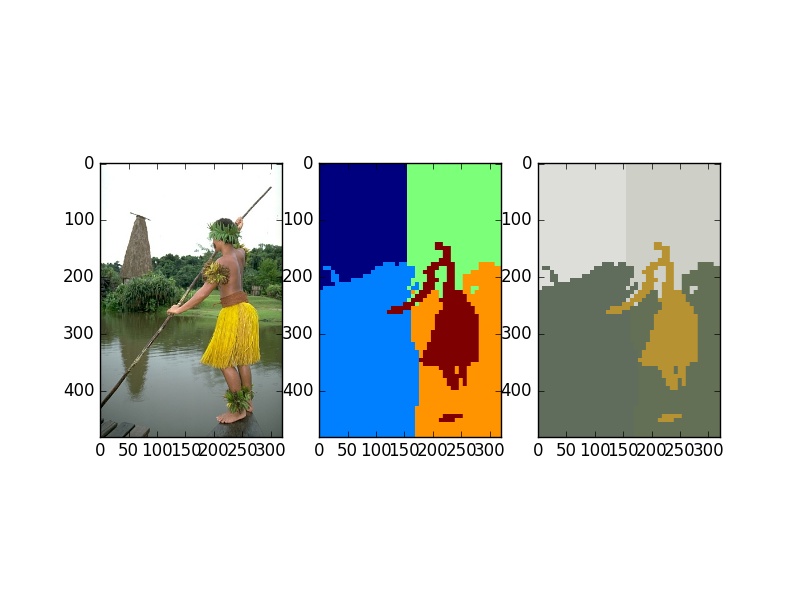
Figure 5-7 are two sets of pictures segmented by different clustering algorithms. Upper each consecutive 4 pictures are the same original picture processed by different hyperparameters. Qualitatively, GMM is a better model than KM and MS. When the cluster number is small (2 clusters) KM and MS failed to preserve boundary details in the upper part of the image between the ‘sea star’ and the background. As far as the second set of images, GMM can separate the totem and the background while KM and MS link the totem with the dark part of the background.

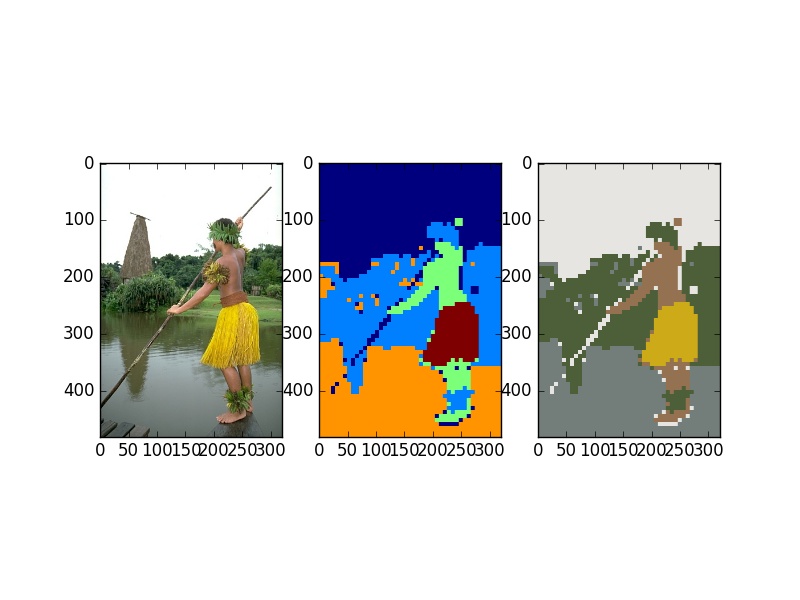
In terms of the change of K value in KM and GMM, the image will be closer to the original picture accompany with the increase of K values but it lost the property of abstraction of segmentation. On the other hand, when the K value is small, the image is abstract and can represent the segments of the original picture. Contrary, larger h values make the output picture more abstract and are divided into less clusters. Small h values produce a blurry version of the original one and may considered to be an anti-noise processing.

Consider the sensitivity of hyperparameters, MS is more sensitive. The number of clusters is not controllable in MS algorithm and the same set of parameter configuration cannot be transfer to another image. Sometimes, when the bandwidth setting is nor appropriated, the algorithm may output singular matrix when updating estimated parameters and fail to get image segmentation.

Another observation is that MS is a much slower algorithm compared with KM and GMM. In our version of implementation, training 1 picture using MS usually needs more than 2 minutes while KM and GMM only needs a few seconds.

## Feature scaling





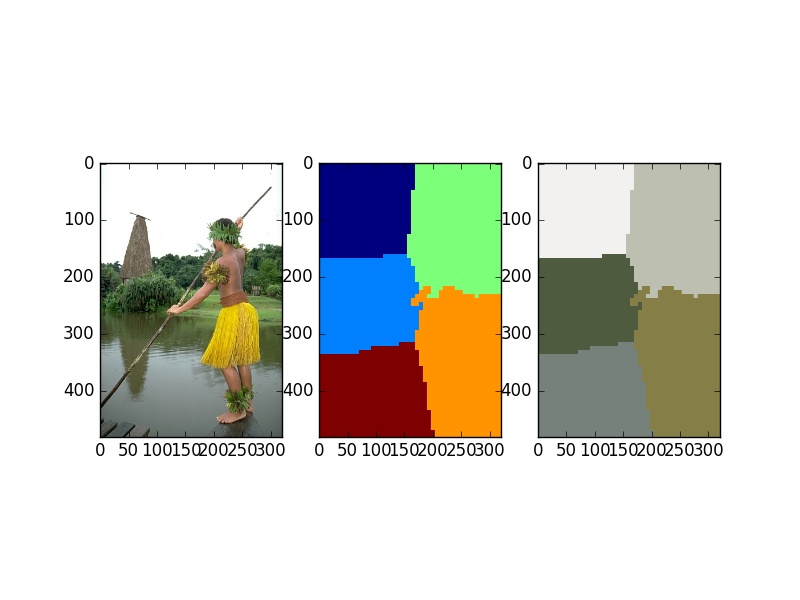
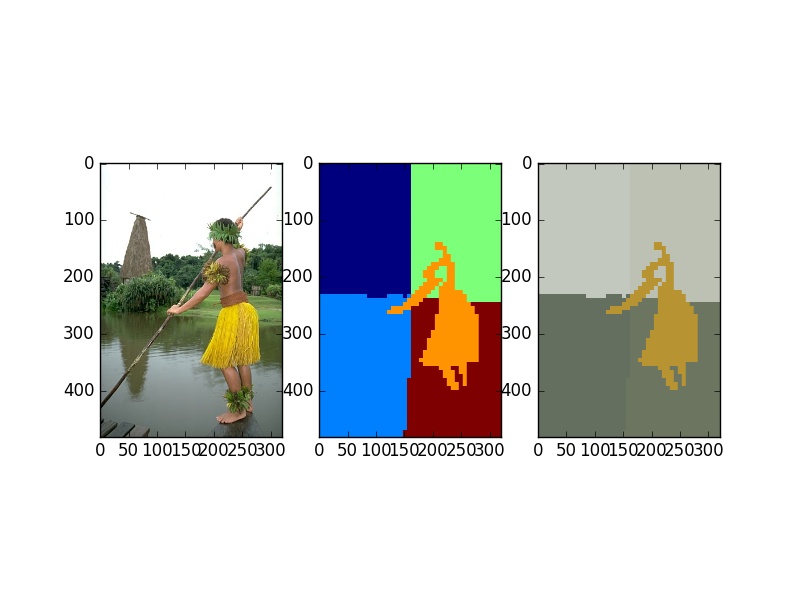
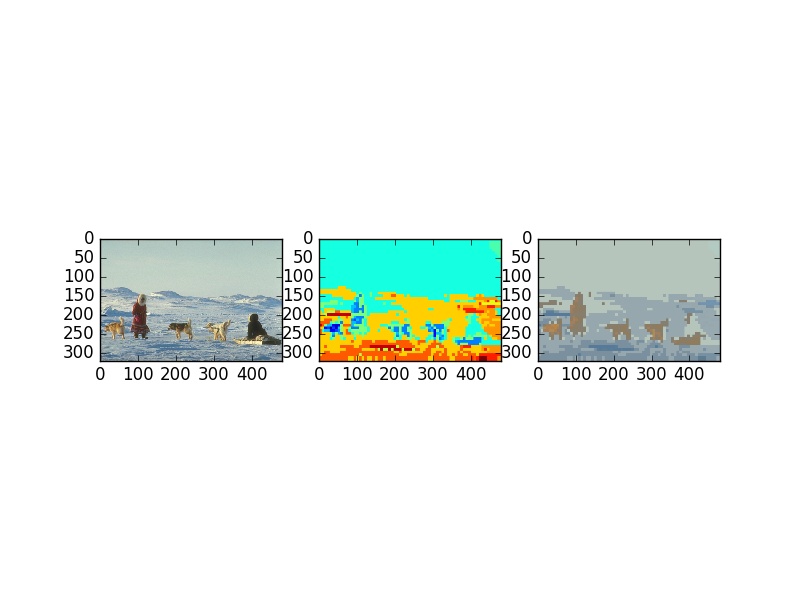
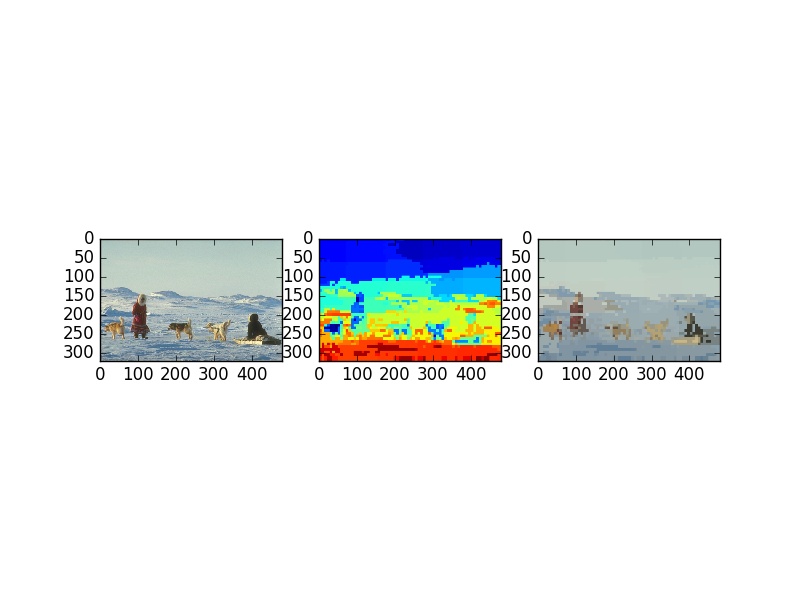


Figure 8 Image segmentation using 5-clusters K-means (KM), Gaussian mixture model (GMM) and scaling features. The first picture is segmented by KM, the second is GMM while the rest pictures are KM clustering pixel locations weighted by {0.1, 5,10}

Qualitatively, if no weighting or scaling of location features. GMM is a superior method to KM according to the first and the second picture. Because GMM can successfully segment the sky, river, mountain and the man which are elements that KM cannot distinguish. However, when feature scaling (by modifying the pixel distance metric) is introducing to the KM algorithm, KM is a better option than GMM because the outline of the man is much clearer than GMM and the skirt and feet are well separated.

By the comparison between different weights of location, the segmentation performance is also affected. From the last picture in Figure 8, we set the weight of location to 10 which means the location of the pixel contribute 10 times greater than pixel color. High weighted pixel location will output a segmentation mainly according the coordinate of pixel.





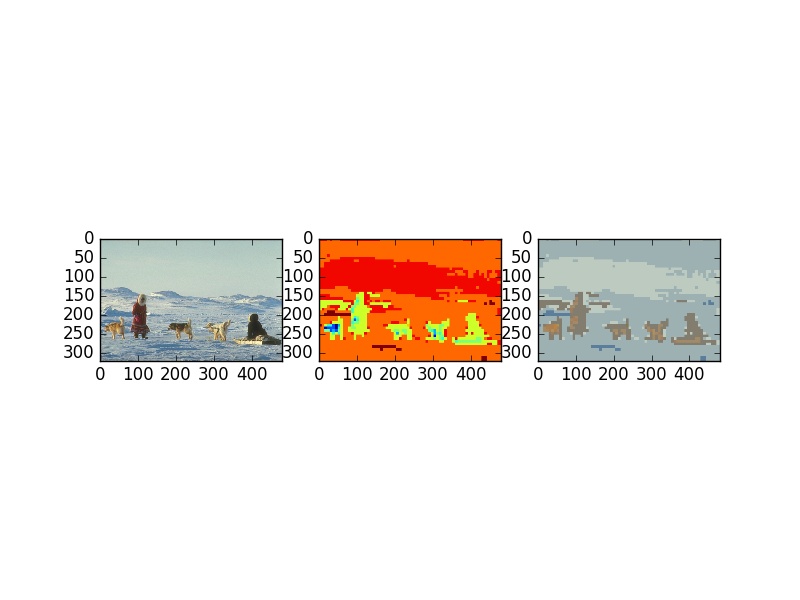
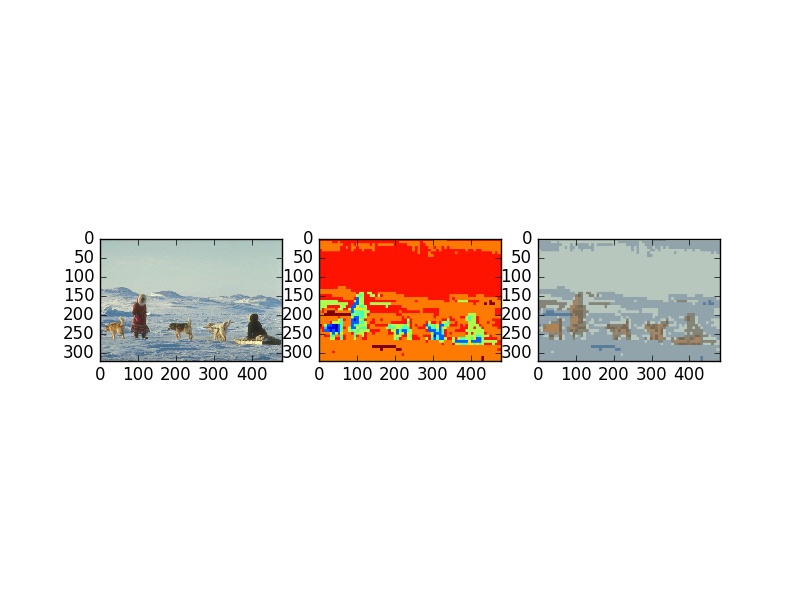


Figure Image segmentation using MS with h = 4 (1st picture) and the MS with location scaling hp = {3,6,7} (2nd- 4th picture)

This set of pictures in figure 9 is displayed to indicate the weighting effect of MS clustering. When using isotropic covariance matrix (MS without feature scaling) the man on the right cannot be segmented correctly. By applying feature scaling, MS can distinguish the animals and the background even with smaller number of clusters and large bandwidth.

When the location of pixels is scaled up (using a smaller bandwidth compared with the color bandwidth) image will outputs more clusters and the segmentation is more sensitive to the pixel location: pixels nearby tend to group together. If the location is scaled down (using a larger bandwidth compared with the color bandwidth), image segmentation is more sensitive to pixel color. This comparison can be illustrated in figure 9, when the location bandwidth is small, the sky and the snow on the ground have similar color and different coordinate are separated into two discriminative clusters. When the location bandwidth is large, these regions with similar colors will come together regardless of the difference of pixel location.