CMPT 414 Project Report

Improvement and Analysis on Image Segmentation Techniques

Haoming Chen 301215629

Xue Han 301188819

Introduction:

Motivations:

There are many image segmentation techniques to partition a digital image into segments available nowadays such as Thresholding, Clustering methods, Compression-based methods etc. But we found that some methods, especially K-Means clustering and Mean-Shift were not working perfectly in some circumstances of image segmentation process.

There are two main problems that affect the performance of these two image segmentation methods. One of the problem is that a general K-Mean algorithm cannot correctly and automatically generate the value of K, which determines how many clusters will be segmented for the image. Randomly picking it will get us a value that's either too big or too small. In this case, some images, that have a relatively small RGB scale, may be over segmented.

And another problem is, the pixels cannot be distributed into correct cluster. We have experimented that the K-Mean and Mean Shift by using only the RGB values of the pixels as the feature to calculate and allocate the means as well as to sort the pixels to clusters. But the result is not good as the pixels, which appear as noise points, was being considered as a meaningful one and sorted into a different cluster.

Our solutions:

We involved the spatial feature of each pixel in the process of segmentation to deal with the problem where pixels cannot be allocated into a right cluster. In Addition to that, we used a method to automatically generate the K value, which is chosen as the number of peaks of the image histogram. With this method, the K value is always accurate and automatically chosen.

Methodology:

1. **K-Means:**

K-Means clustering algorithm aggregate sample items into groups according to the distance from some “K-values”, which was randomly distributed, to every other sample items to segment images.

There are two characteristics for each cluster that was partitioned by the K-means algorithm, must have.

- The items in same group have the shortest lengths from the K-values in its own group other than the K-values that other clusters have.
- Every cluster have exactly one “K-values” in it.

Also, the K-mean algorithm can be briefly described with the following pseudo code:
01. Choose one of your data points at random as an initial centroid. [1]
02. **repeat**
03. Calculate D(x), the distance between your initial centroid and all other data points, x.
04. Choose your next centroid from the remaining data points with probability proportional to D(x)
05. **until** all centroids have been assigned.

To applying K-means clustering for image segmentation, we will need to do the steps below for the algorithm implementation.

- Initialization: K chosen, an initial set of K centroids will be selected in advance. These K centroids is a pixel with its RGB values.
- Every point of the data set (pixels) is assigned to its nearest centroid. During this process, the distance is determined by the data vectors that calculated by using centroid with any other pixels. *Each data vector is three dimensional (R, G, and B) and the values are bounded integers in the [0, 255] range to compute the distance with centroid.* [2]
- The centroids forward to the center of each cluster.
- Then Repeat 2 and 3 steps until no centroid was shifted in on iteration.

Until now, the image was segmented into K’s clusters with its centroid. While the RGB value of the pixels in each group would also be replaced by the value of centroid. Therefore, more K centroids we were generalized, the more different color blocks in an image will indicates. In other words, the image would be more precisely with large number of centroids using in image segmentation.

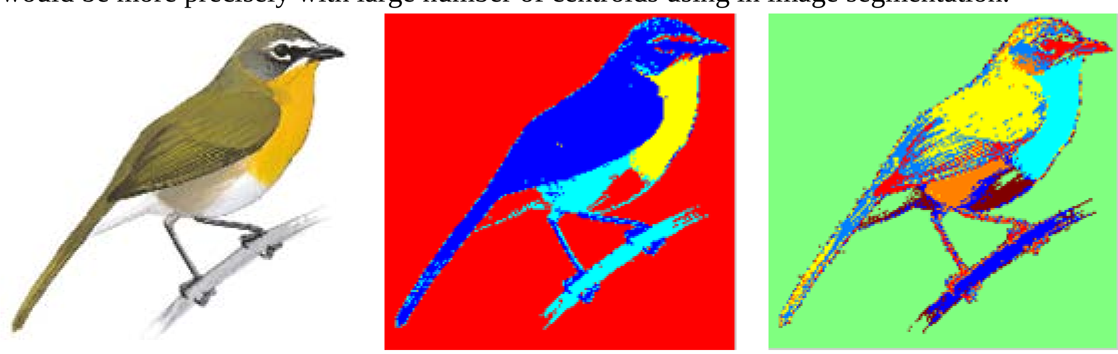


Figure 1:: Original Image; K = 4 segmented; K = 8 segmented

2. Mean Shift:

Mean shift algorithm is an iterative method start with an initial value x and dependent on a given "Kernel Functions"

$$K(x_i - x)$$

"*This function determines the weight of nearby points for re-estimation of the mean* [3]". Gaussian kernel function:

$$K(x_i - x) = e^{-c||x_i - x||^2}$$

is commonly used for computing the distance to the current value x. "*The weighted mean of the density in the window determined by*

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x)x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$ [3]".

In this function, we can observe that the N(x) is the points near the estimated value x where the K(x) is not equal to 0.

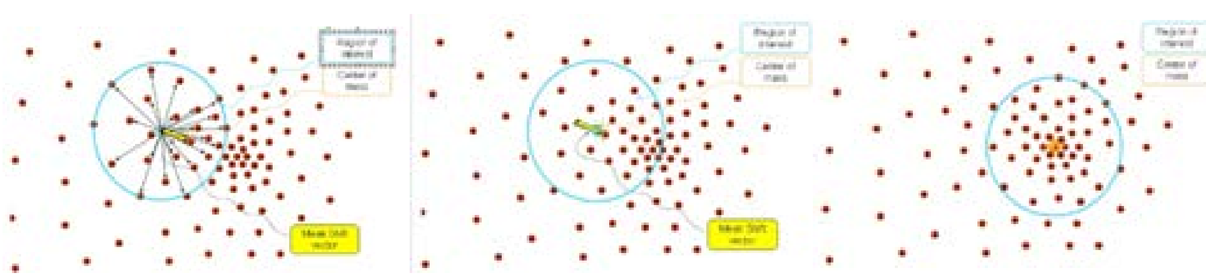
The m(x) -x is called mean shift in the current iteration calculus. Then the m(x) will be set to x. And a repeating calculation processed for each x, until m(x) converges.

To applying mean shift algorithm for image segmentation, we will need to do the steps below for the algorithm implementation.

1. Because each of our pixel in an image have its RGB values, then our calculation will progression on a 3-dimensional Coordinate system with R, G, and B axis.
2. Choose an initial value x.
3. Calculating the distances with current pixel value x and xi, which is positioned the neighborhood of x.
4. Then, we will have a mean shift vectors m(x) by using the function

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x)x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$

5. Repeat steps 3 and 4 until m(x) converges which means x is already moved to an area that have the highest pixel density.



6. The image is segmented in several clusters after applying the mean shift algorithm to it.

Comparison:

K-Means clustering algorithm is simple and fast when we are running it. Also this algorithm is easy to implement. But you need to choose K manually or randomly. Meanwhile, the K-Mean algorithm is rarely used for image segmentation because K-Means is sensitive to outliers and prone to local minima which means it can not produce a good segmented image for any image.
For Mean-shift algorithm, the computationally is very expensive. When the Mean-shift and K-Means working on a same image we provided. The Mean shift algorithm need more time for computation than K-Means algorithm. And the quality of the output image depends on the given window size. However, Mean shift also has many advantages. It can robust to outliers which is not like K-Means: sensitive to outliers. Moreover, the Mean-shift algorithm a more general used, application-independent tool for image segmentation. The quality of output image is often better than the K-Means algorithm.

Description of work:

Part 1:

<u>Initial comparison on K-means and Mean-Shift:</u>

Due to both K-means algorithm and mean shift algorithm are segmenting images by arrange pixels into clusters based on the initialized centroid. The parameters (e.g. RGB value) we have taken to decide which values will be used for the centroids to form pixels as cluster must be sufficient. For example, if we only take the intensity of pixels as parameters to do the calculation, the performance of the K-mean algorithm and mean shift algorithm for image segmentation will be bad (parameters not sufficient).
Therefore, because we are using RGB values as parameters to make series of iteration operations to clustering. The image would be partitioned as different color blocks. The problem will occur when some images have many pixels that have similar colors (RGB values). The different objects in an image that cannot be partitioned into same cluster but has the similar color will be regarded as the same objects, which is called "ambiguity “. The following figures shows both the shifting distances of means for each iteration, and the result of segmentation of images:

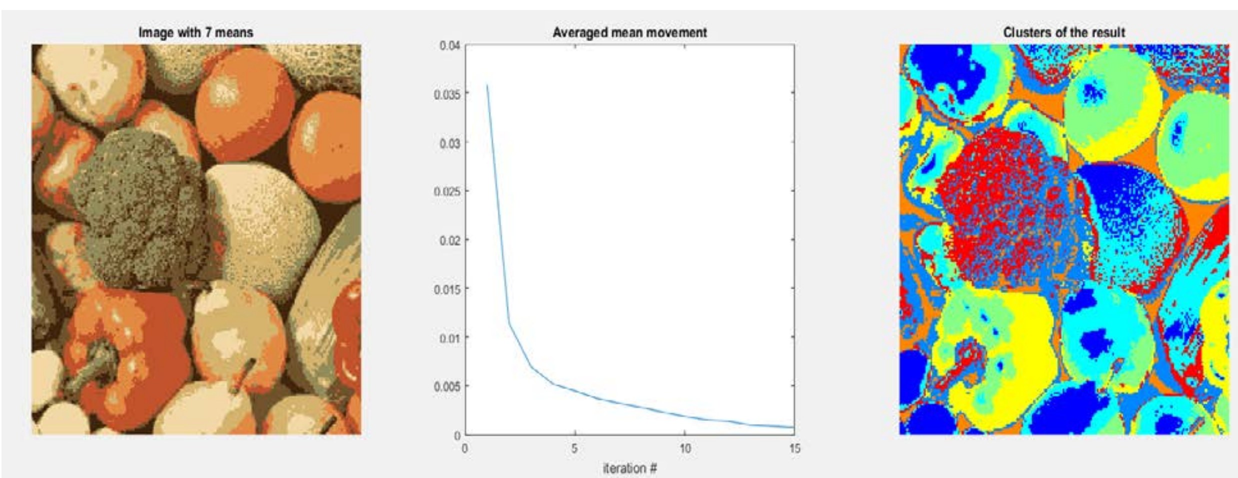


Figure 2: K-Mean with RGB

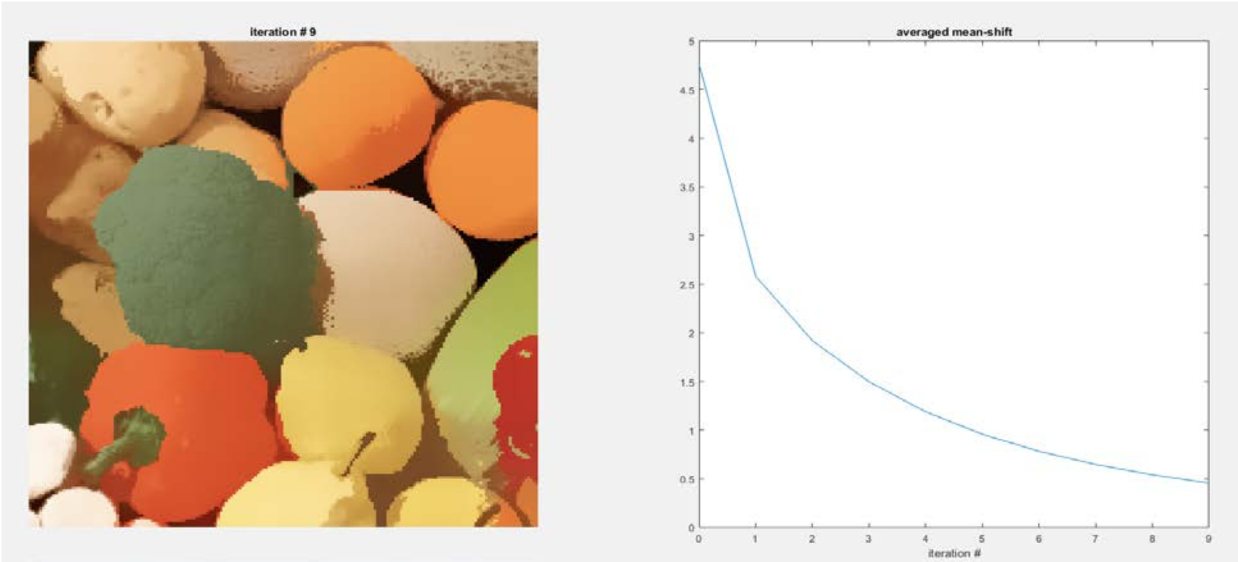


Figure 3: Mean Shift Algorithm with RGB

<u>Improvement on the algorithm with an addition feature – spatial correlation of pixels:</u>

The solution is to apply the spatial location as parameters to *calculate D(x)*, the distance between the initial centroid and all other data points of K-mean or mean shift algorithm.
For example, the vector that was calculated is the RGB vector. And these RGB vectors may be the artifacts that calculated by these pixels that belong to different objects but with similar RGB values. However, if we involve in a spatial location as a parameter, the different objects in image will be distributed into different clusters even if they have the exactly same color because of the difference of its space. Precisely, the spatial location of each pixel was defined as parameters x and y, which is a "Cartesian coordinate system"[4]. Therefore, we increase 2 dimensions for calculating the vectors with centroid and other pixels to avoid the interference brought by the color similarity.

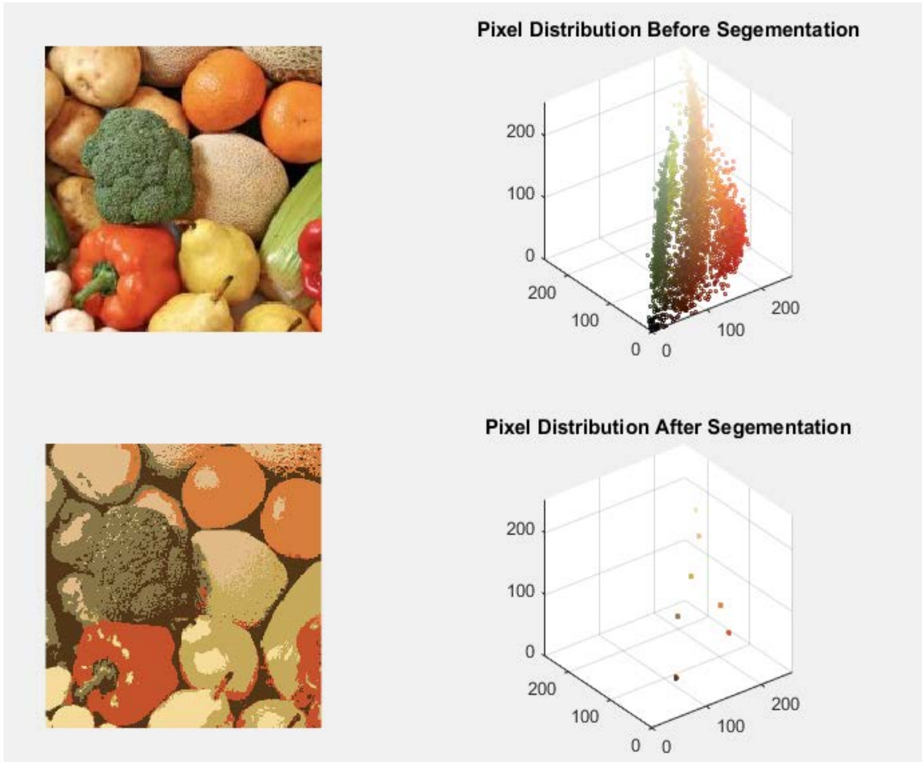


Figure 4: K-means with Color and Spatial Feature

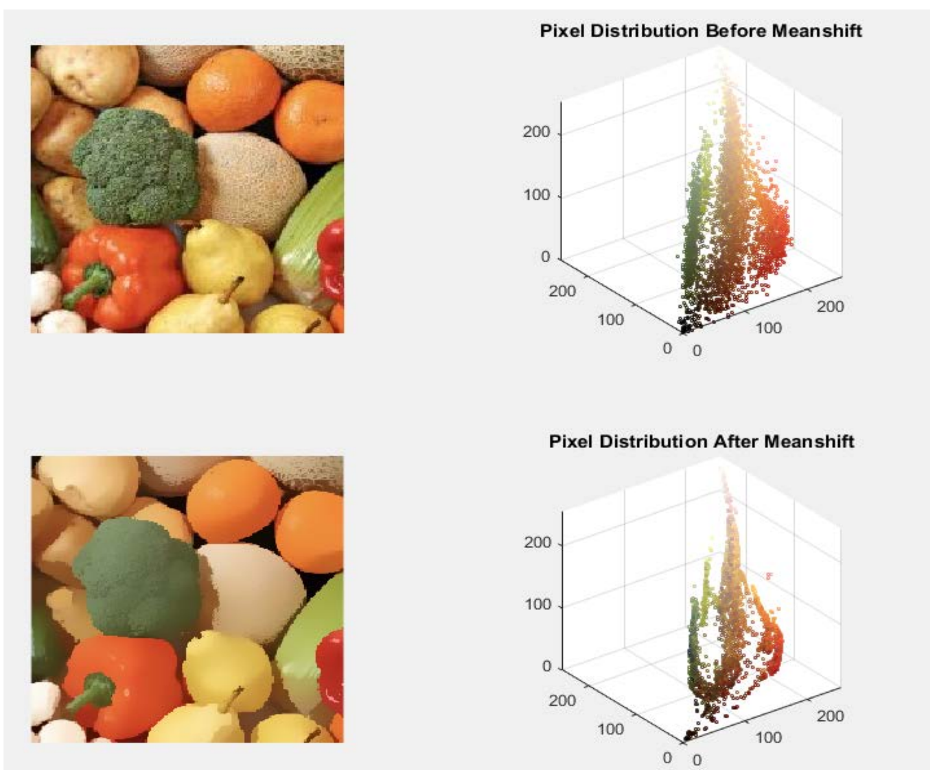


Figure 5: Mean-Shift with Color and Spatial Feature

Figure 6: An overall comparison of four different methods

Part 2:

The difficulty in find the correct K value for K-means algorithm:

The number of K values was generated randomly for any type of images segmentation even if not compatible. For example, an image with only colors red and green that was given an oversized K's value (e.g. K = 10) for segmenting will not work well with clusters.
For example,

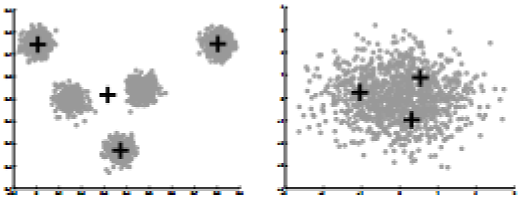


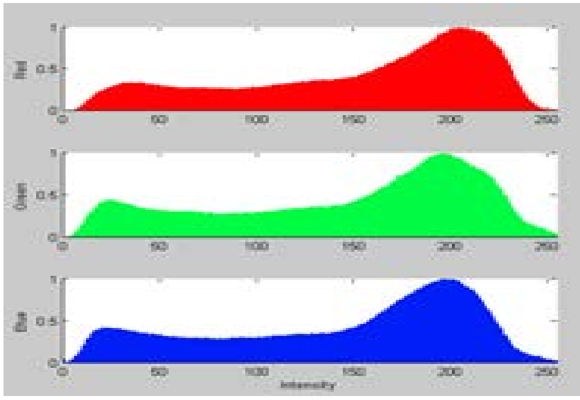
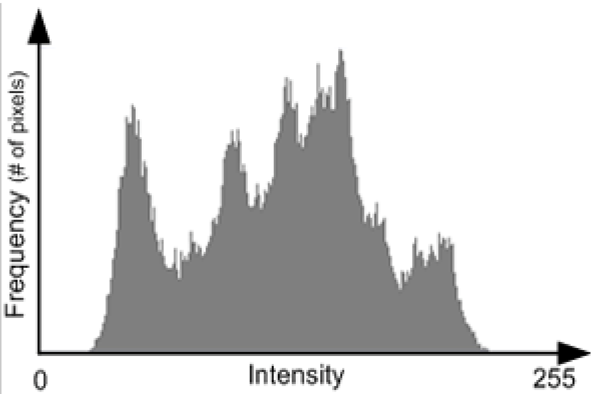
Figure 7: Possible false detection

"Two clustering where k was improperly chosen. Dark crosses are k-means centers. On the left, there are too few centers; five should be used. On the right, too many centers are used; one center is sufficient for representing the data. In general, one center should be used to represent one Gaussian cluster." [7]

Following is the comparison between the K-means and Mean-shift, both with 2 conditions: using only Color feature vs. using Color as well as Spatial features:

Solution to find K automatically:

Because a color image has 3 color based histograms which are red, green, and blue channel. It is hard to find/distinguish some valuable data in these histograms to help us to choose the K-value. Therefore, we transfer the image from a color image to a grayscale image to reduce the amount of data.

Then, we make a gray-level histogram for the image. After that, we can easily find that there are some "peaks" on the histogram, which are the local maxima of the gray-level intensity frequency. The amount of number of peaks will be used to decide how many K-values we will use in the image segmentation process. *Because the peaks mean that the pixel count with those tone values were high* [8] which means there are many pixels that has same RGB values refer to the original color image.

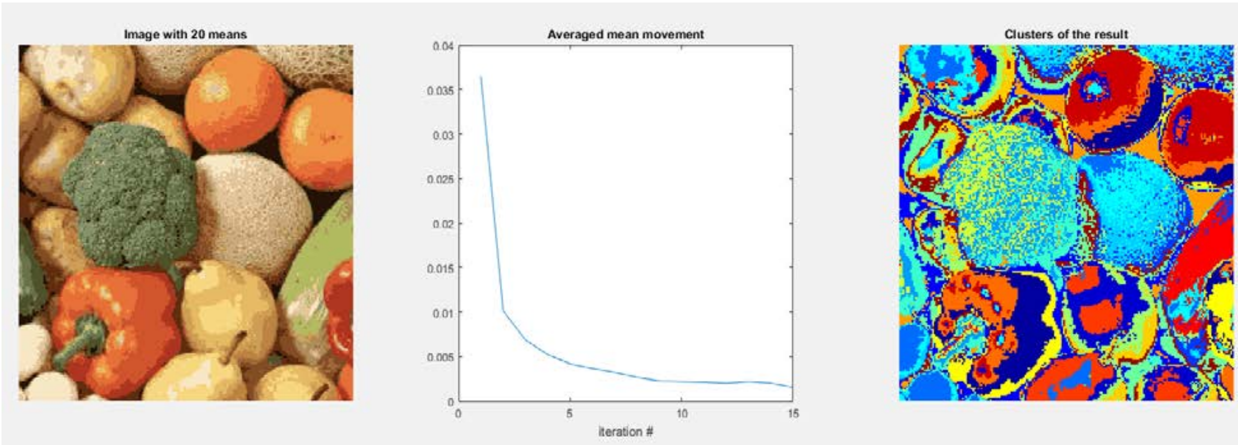


Figure 8: Manually setting value of K results in over segmentation

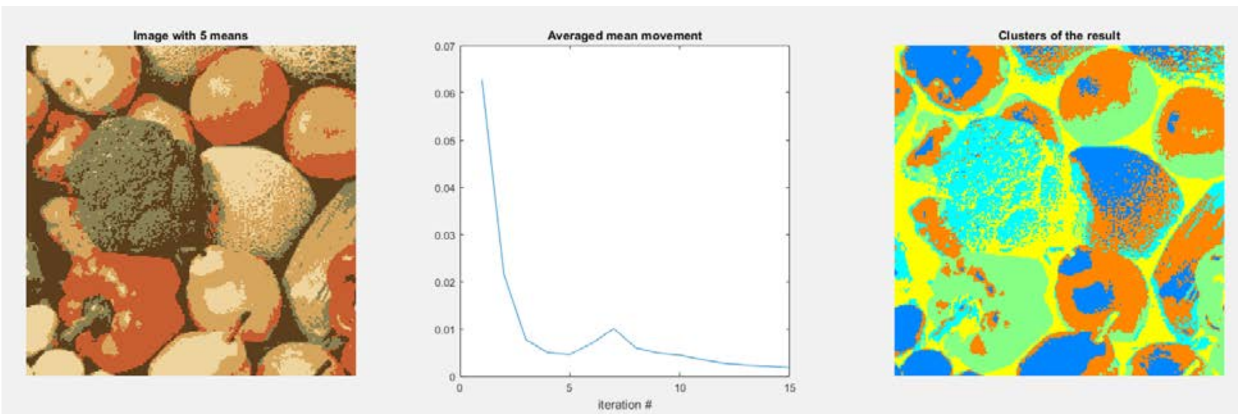


Figure 9: Automatically generate number of Ks depend on the gray-level histogram

Conclusion:

By importing new parameters (x, y to define the pixel position) to the process of image segmentation, the performance of K-mean algorithm and mean shift algorithm for image segmentation was significantly increased. Theoretically, the ambiguity caused by the color similarity can be mostly removed.

In addition, because the number of K-value can be determined correctly but not randomly generalized. The number of clusters will be suitable for any images for the segmentation. How many pixels an image has, how much centroid will be generated. There will be no shortage or excess clusters a segmented image would have.

Reference:

1. https://www.quora.com/What-is-a-good-way-to-choose-initial-points-of-k-clusters-in-k-means-clustering.

2. http://softwareengineering.stackexchange.com/questions/314222/how-to-implement-k-means-algorithm-on-rgb-images
3. https://en.wikipedia.org/wiki/Mean_shift
4. https://en.wikipedia.org/wiki/Cartesian_coordinate_system
5. https://en.wikipedia.org/wiki/Watershed_(image_processing)
6. https://www.hindawi.com/journals/ddns/2008/384346/
7. https://papers.nips.cc/paper/2526-learning-the-k-in-k-means.pdf
8. http://www.scantips.com/simple1b.html
9. Lecture 13: k-means and mean-shift clustering by Professor Fei-Fei Li, Stanford Vision Lab