# Project 1(part 3) - Markov Network Based Image Segmentation

## I. Introduction

Segmentation is a fundamental process in digital image processing, which is widely used in extensive area. Image segmentation usually consist of two parts, one part is to choose proper set of characteristics which could differentiate different-content regions; the other part is to choose the proper segmentation method to get a segmentation map. For this project, our goal is to segment the grayscale image, and the characteristics that we consider are spatial correlation and the similarity of the pixels within the same group. And the method we use is to maximize the condition probability of label assignments given the characteristics of this image, which equals to minimize the energy of the same image. We use EM algorithm to achieve this minimum value.

In our work, at first, we want to find the "good" value of K* and $\beta$*, we count the number of pixels for different grayscale values and get a histogram, then analyze the histogram to get an estimated value of K. Then we use K and values near K to do some tests. Based on the results of the tests, we choose the value which give the best result as our estimated optimal K*. And we analyze the graph of ratio of $E_R$ and $E_F$, since either of them becomes dominant will lead to bad segmentation, if $E_R$ becomes dominant, the image becomes too smooth, and if $E_F$ becomes dominant, there will be many small regions.

Then we evaluate the effect of K and $\beta$. Firstly, we fix the value of $\beta$, and we vary the value of K in the range $[\frac{K^*}{2}, 2K^*]$, and evaluate the segmentation results and performance by analyzing the graph of $P(Y \mid f)$, the graph of computation time and the graph of regions. Secondly, we fix the value of K, and vary the value of $\beta$ in the range $[\frac{\beta}{2}, \beta]$, then do the same evaluation methods for $\beta$.

However, when we implemented the algorithm according to our first report, we realized that we made a mistake. In the M-step of our EM algorithm, we used the Gaussian distribution to assign labels by mistake. We should use the Energy function to assign labels to each pixels.

The remaining sections of this report is organized as below: in part II, we illustrate how we select K* and $\beta$ *; in part III, we present our analysis of varying K; in part IV, we present our analysis of varying $\beta$.

## II. Analysis for $K^*$ and $\beta^*$

When analyzing the $K^*$, we use the histogram to plot the point number distribution according to the point grayscale and determine the $K^*$ value based on the histogram firstly. For example, we can count the number of local maximal values in the histogram, and assume that every local maximal value represents one class, so the number of local maximal value is an estimated value of $K^*$. After that, for some K value around the $K^*$ (such as $K^*-1$, $K^*$, and $K^*+1$), we run the algorithm for these K values and plot the segmented images for each K value, then we figured out one K value with the best segmentation result and choose that K as the new $K^*$. Here is an example:

Figure 1 is the histogram of the point number distribution according to the point grayscale. From the histogram, we can see that there are almost four local maximal values, so we choose $K^*=4$. Then we plot the segmented images for $K=3$, $K=4$ and $K=5$ (Figure 2) and found that the best segmentation result comes from $K=4$, so we still choose $K=4$ as our $K^*$.

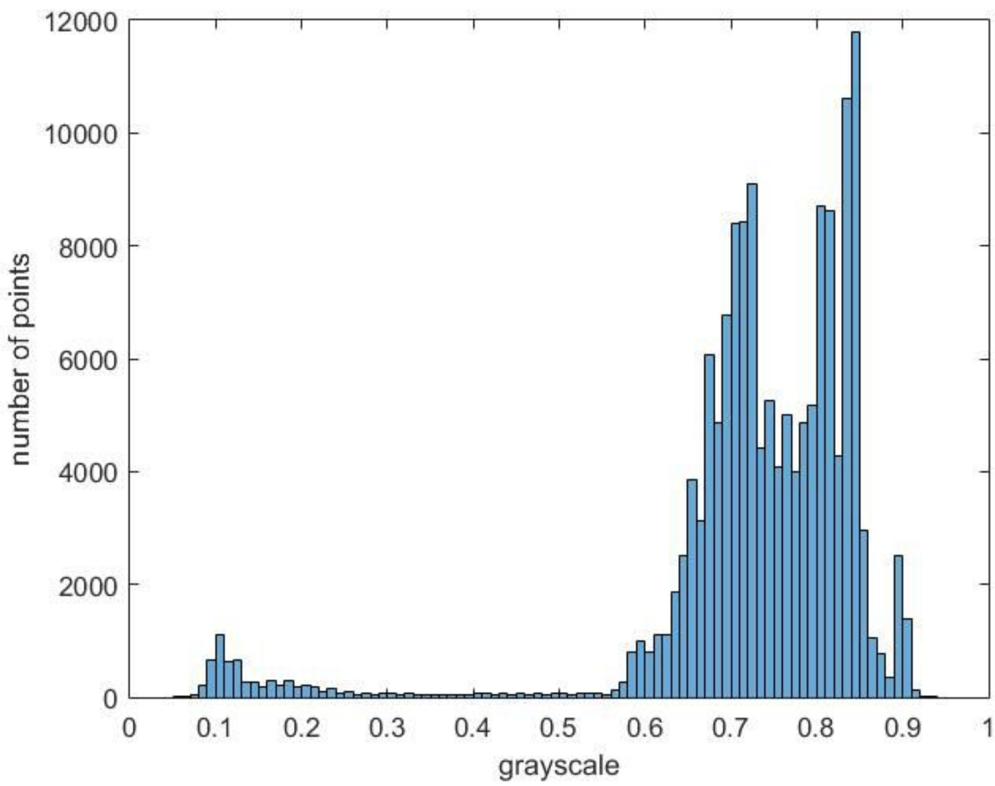


Figure 1. The point number distribution as a function of the point grayscale.

<table>
<tr><td></td><td><br></td></tr>
<tr><td>K=3</td><td>K=4</td></tr>
<tr><td><br></td><td></td></tr>
<tr><td>K=5</td><td></td></tr>
</table>

Figure 2. The segmented images for K values.

When analyzing the $\beta^*$, we consider the ratio of $E_F$ to $E_R$ ($E_F / E_R$) according to the $\beta$. When we segment the images, if $E_F / E_R \to \infty$, then $E_F$ is much greater than $E_R$, and the segmentation result will be dominated by the Gaussian distribution, which means that we segment the image mostly based on the point grayscale, so there will exist many noise points in our segmentation result image. However, if $E_F / E_R \to 0$, then $E_R$ is much greater than $E_F$, and the segmentation result will be dominated by the correlations between the points, so the whole image will be too smooth and different objects may be regarded as one object. Both these two situations come out bad segmentation results. In order to balance the affection between $E_F$ and $E_R$, we first plot the $E_F / E_R$ curve according to $\beta$ (Figure3), and we also plot different segmentation result images according to different $\beta$ values. From the $E_F / E_R$ curve and the segmented images, we find that the segmented images are appropriate for $\beta$ range [1-3.5]. Optimally, we choose $\beta = 2$ as our $\beta^*$.

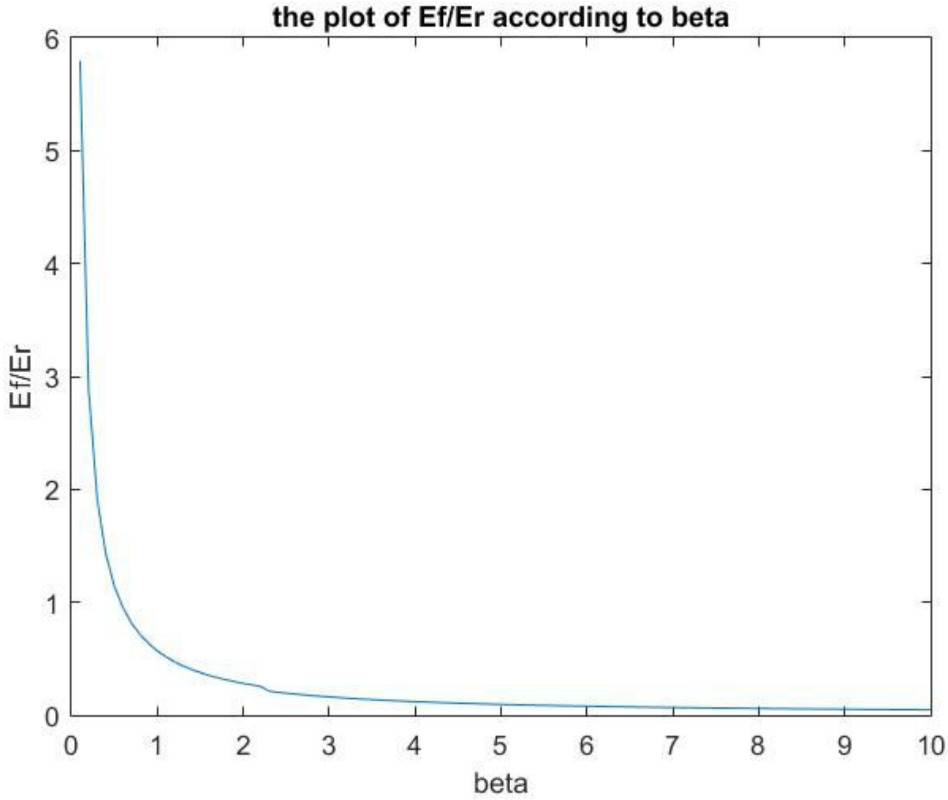



Figure 3. The $E_F / E_R$ curve as a function of varying $\beta$

| | |
|---|---|
|  |  |
| $\beta$=0.5 | $\beta$=1 |

| | |
|---|---|
|   |   |
| $\beta$=2 | $\beta$=4 |
|   | |
| $\beta$=8 | |

Figure 4. The segmented images for $\beta$ values.

# III.Analysis for varying $K*$

When analyzing the varying $K*$, we fix the $\beta$ value and plot the Energy, size of regions and time complexity as a function of varying $K*$.

From the plot of the Energy of the varying $K*$ (Figure 5), we can figure out that the Energy decreases slightly as K increases, which means that the selection of K does not influence the Energy much.

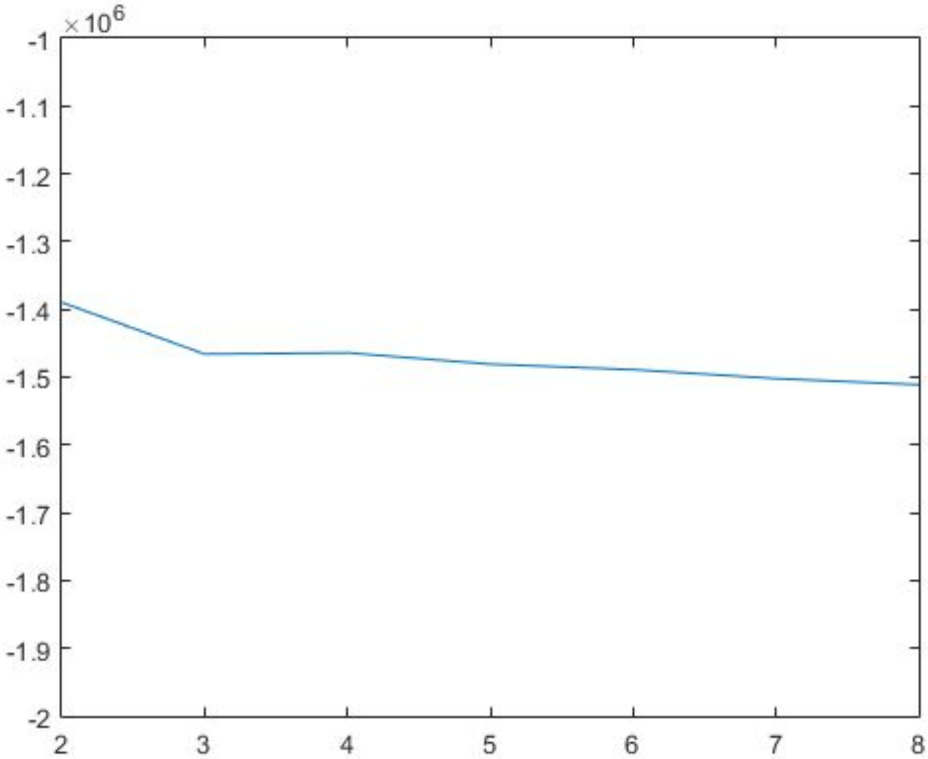



Figure 5. The Energy of the varying $K^*$

From the plot of the size of regions of the varying $K^*$ (Figure 6), we can find that as K increases, some classes always have almost same percentage, so we can infer that there are some parts in the image are distinct from others, so it can not be re-clustered with other pixels into a new class , and it will always be an independent class.

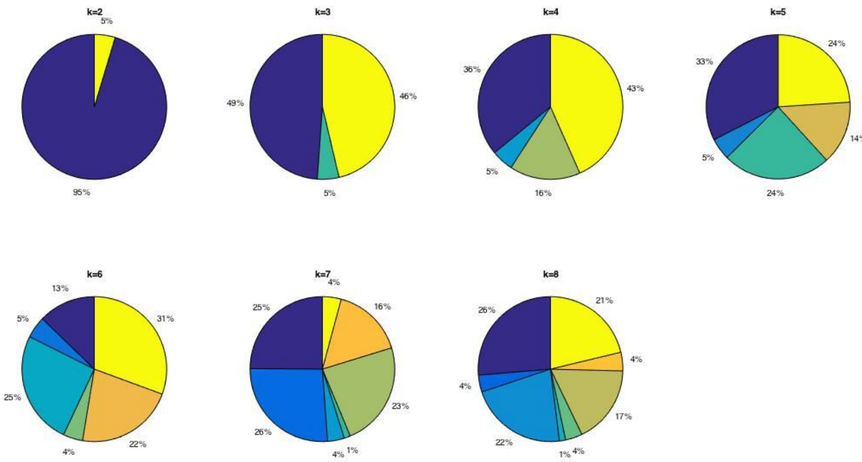



Figure 6. The size of regions of the varying $K^*$

From the plot of the time complexity of the varying $K*$ (Figure 7),we can find that as $K*$ increases, the time complexity increases linearly based on $K*$.

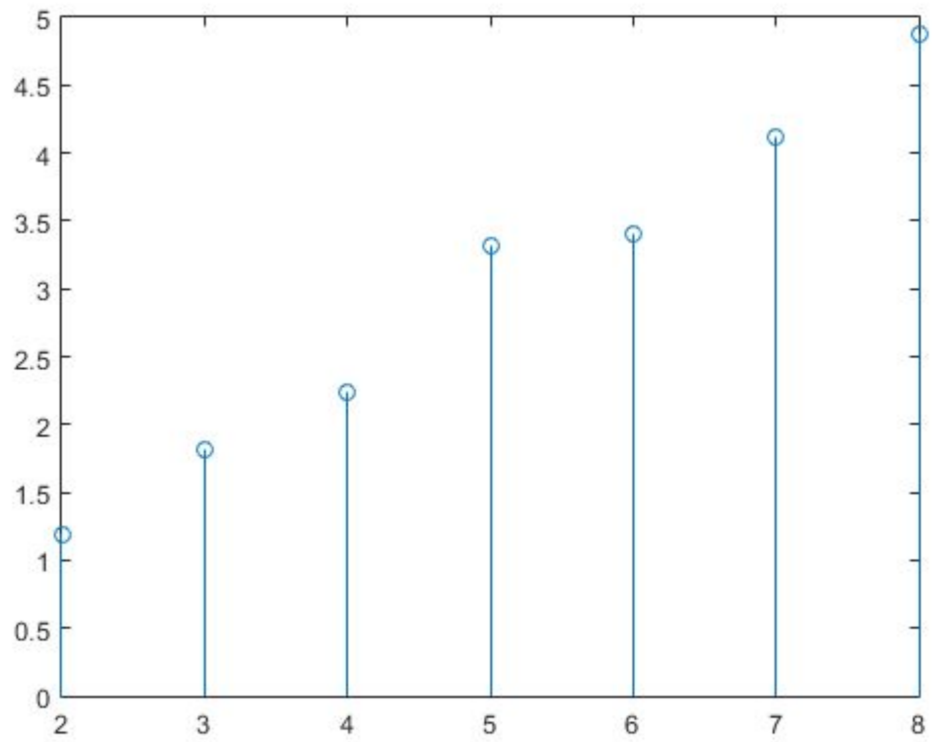



Figure 7. The time complexity of the varying $K*$

## IV. Analysis for varying $\beta*$

When analyzing the varying $\beta*$, we fix the $K$ value and plot the Energy, size of regions and time complexity as a function of varying $\beta*$.

From the plot of the Energy of the varying $\beta*$ (Figure 8), we can find that the Energy decreases linearly as $\beta*$ increases.

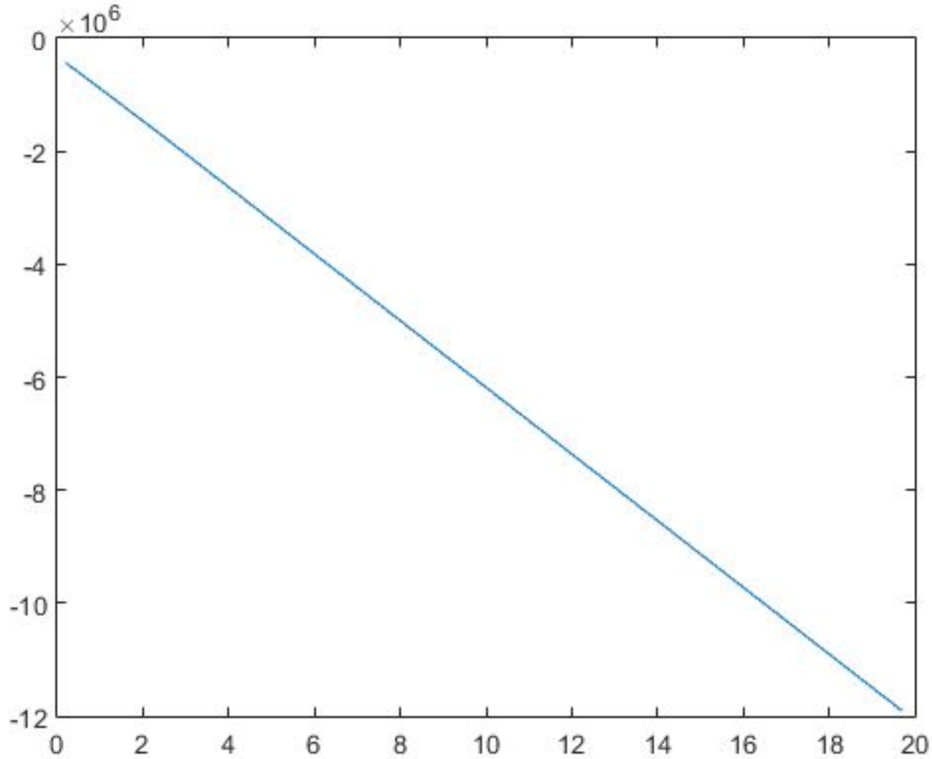



Figure 8. The Energy of the varying $\beta *$

From the plot of the size of regions of the varying $\beta *$ (Figure 9), we can find that as $\beta *$ increases, more points are clustered into one class.

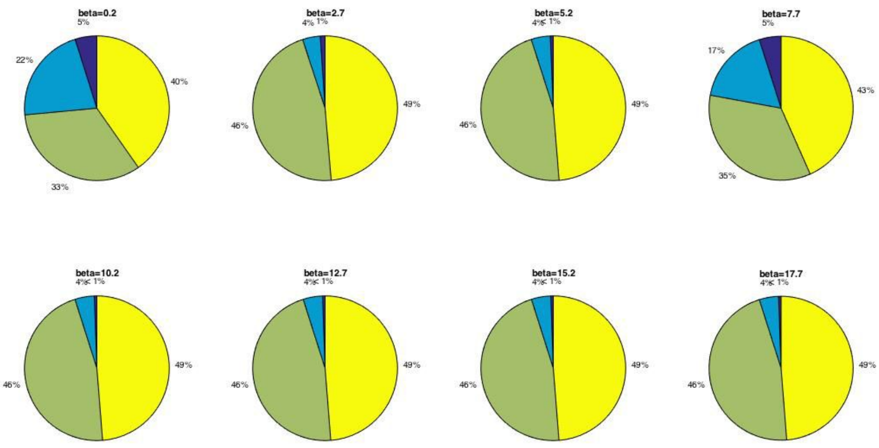



Figure 9. The size of regions of the varying $\beta *$

From the plot of the time complexity of the varying $\beta*$ (Figure 10),we can find that the selection of $\beta*$ does not influence the time complexity.

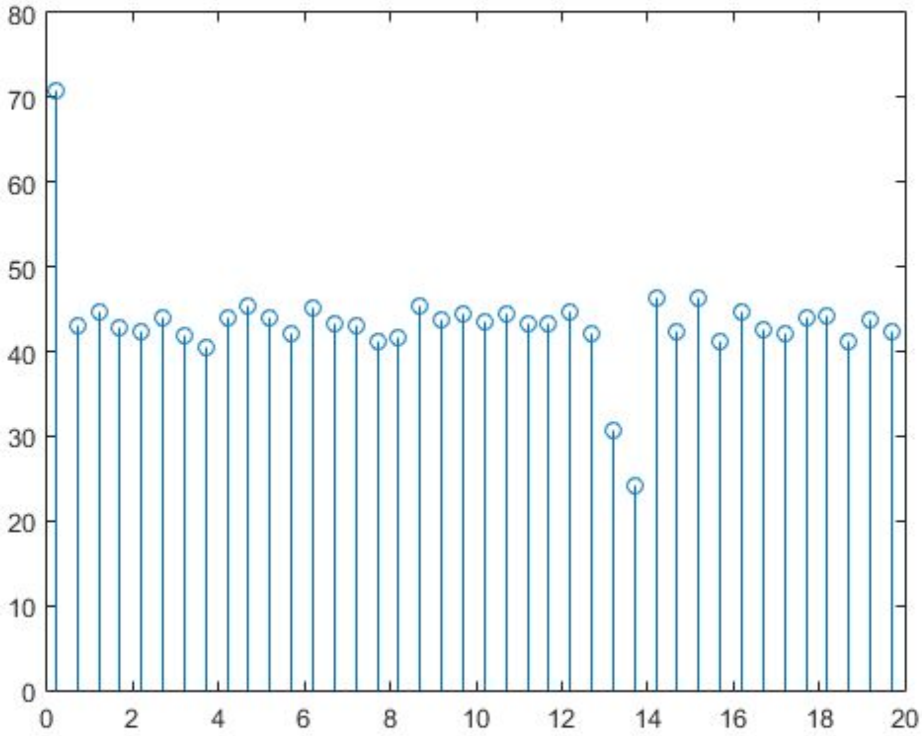


Figure 10. The time complexity of the varying $\beta*$

## V. The segmentation results of all the images with appropriate $K* = 4$ and $\beta* = 2$

| | |
|---|---|
|   |   |
| **Mean representation of Image** | **Sigma representation of Image** |

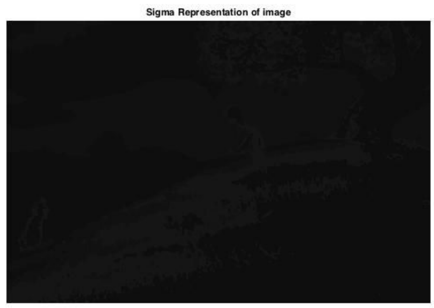
| | |
|---|---|
|  |  |
| **Mean representation of Image** | **Sigma representation of Image** |
|  |  |
| **Mean representation of Image** | **Sigma representation of Image** |
|  |  |
| **Mean representation of Image** | **Sigma representation of Image** |

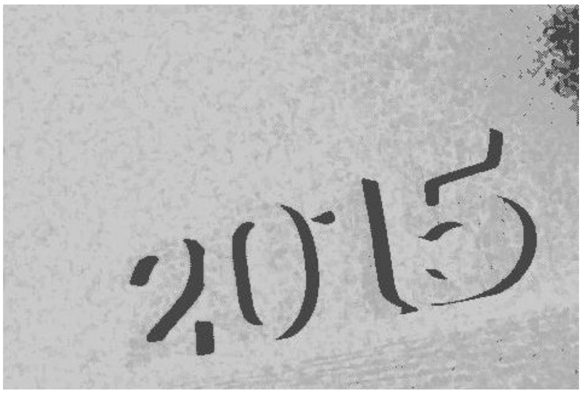
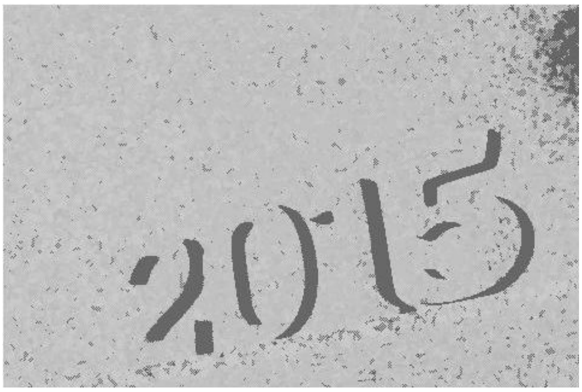
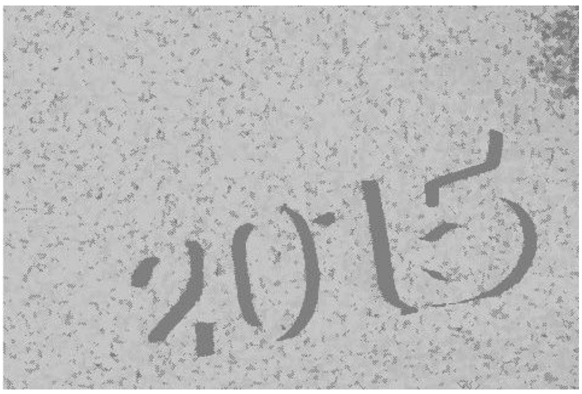
**Figure 11. The segmentation results of the images**

## VI. Analysis of Initialization process based on our experiment

We investigate how the initialization process will effect the outcome of our experiment. We employee two ways for initialization: a randomized initialization and the initialization via k-means clustering and compare the result of partitioning.

For the random initialization, we randomly assign the label to each pixels. For the k-means, we cluster the pixels as one dimensional data points, and use the build-in functions from Matlab to return the label of each pixels. We compare the result for different beta both in term of energy function as well as partitioning result.

We conduct this test with #01 test picture, and set k = 4. The result is shown as below:

| Random Initialization | K-means |
| --- | --- |
| <br>beta = 0.25 | <br>beta = 0.25 |
| <br>beta = 0.5 | <br>beta = 0.5 |
| <br>beta = 1 | <br>beta = 1 |

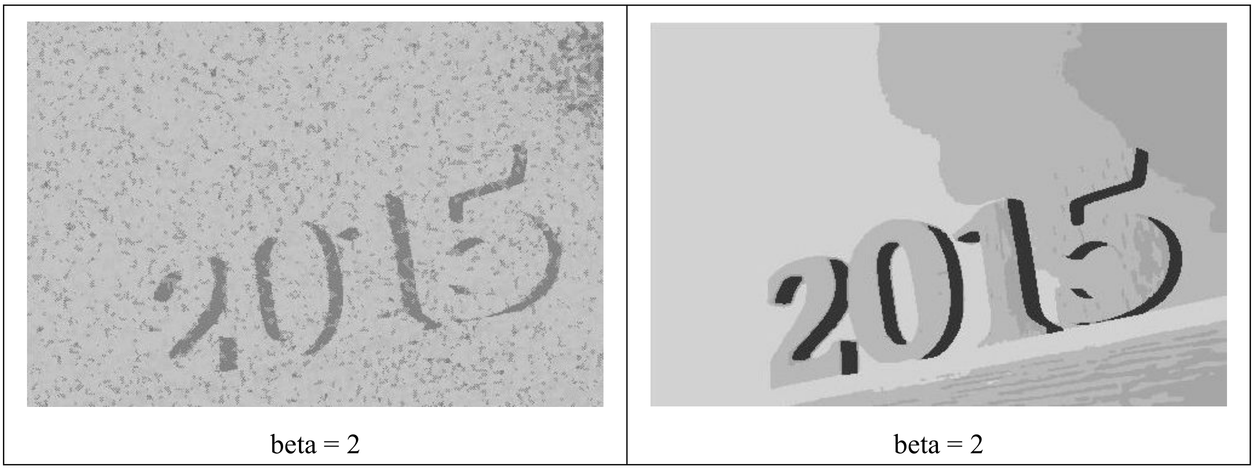



Figure 12. The different segmentation images based on Random Initialization(left) and K-means(right)

We can see that when beta is small, the result is somewhat similar. But when beta becomes large, the noise in the partition increase greatly which is against out assumption about how the beta will affect the partitioning. We then look into the changes of the energy function for different beta, which is shown as below:

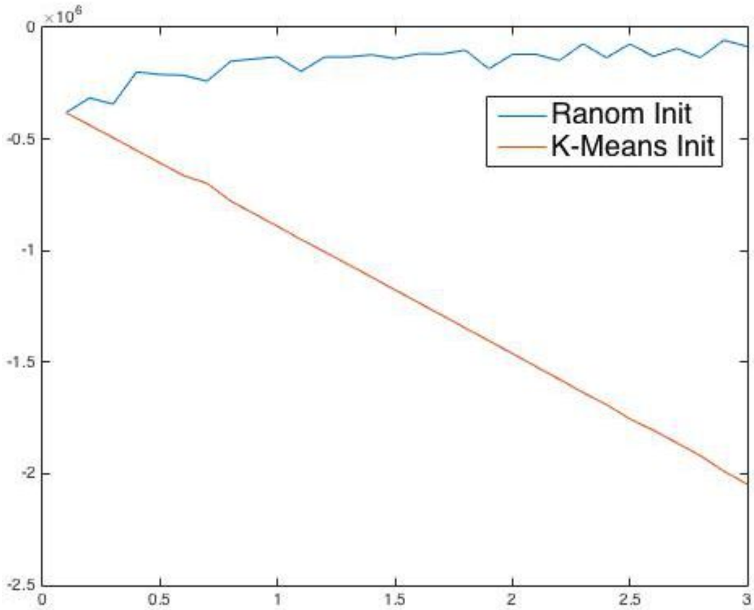



Figure 13. The Energy change as a function of $\beta$ for Random Initialization and K-means

We see that the energy function gets by the k-means clustering algorithm go down strictly, while for the random initialization, it is quite unstable. A possible explanation for this is that using the random initialization method, the minimization of the energy function jump to a local minimum very quickly, especially when beta is large. Because it is possible that most of pixels are not labeled with the same labels with their neighbor, convergence of this could be problematic for a large beta.

# VII. Conclusions

In this report, we implement this algorithm under Matlab. We test different initialization strategy: random label assignment and K-Means, and compare the effect of different initialization methods. We find $K*$ through analyzing the histogram of pixels' grayscale distribution and observing test results based on it. Meanwhile, the $\beta*$ is determined by analyzing the ratio of $E_r$ and $E_F$ and observing the test results based on several $\beta$ values. From the experiment , our $K*$ and $\beta*$ yield a good performance, which shows that our method for selecting $K*$ and $\beta*$ is feasible. Furthermore, we do test on varying $K$ and $\beta$. And we got the trend of Energy as K and $\beta$ varying. From the trend, we find that K does not influence the Energy much, but $\beta$ has impact on the Energy, since the Energy will be dominated by $\beta$ mostly as $\beta$ increases. Then we test and analyze the algorithm performance by comparing the computational time, and we find $\beta$ has little influence on computational time, but increasing the $K$ will linearly increase the computational time, since when increasing K values, the time to assign labels to each pixels will increase linearly. Then we analyze some other characteristics of the image, such as the size of region, which reveals the distribution of pixels in different labels. And we find that the K and $\beta$ influence the distribution of size of regions of different classes, since the correlations between points are changed.

Our project help us reveal the impact of $K$ and $\beta$ on both of the energy and the segmentation results, and how to select a "good" $K$ and $\beta$. And our result provide a reference for other application which use MRF to segment image. Also, it can be used as a base for further research and improvement.

# VIII. Reference

Deng, Huawu, and David A. Clausi. "Unsupervised image segmentation using a simple MRF model with a new implementation scheme." *Pattern recognition*37.12 (2004): 2323-2335.