EE4212 Computer Vision

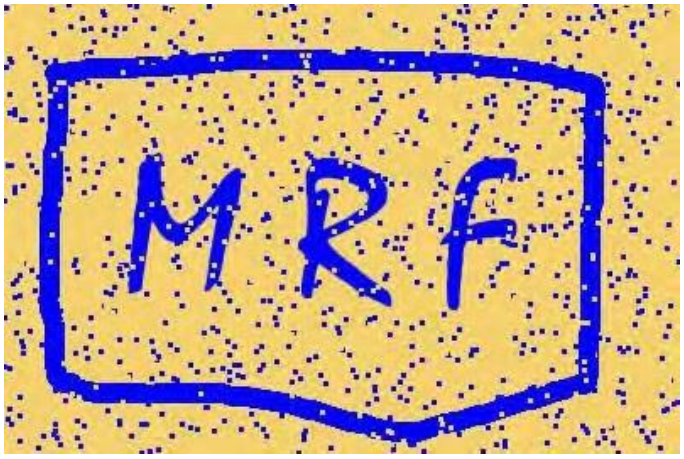
Assignment 2: Markov Random Field and Graphcuts

Name : Emerson Raja Jose Peeterson

Matric. No. : A0195021N/E0383686

Part 1 : Noise Cleaning

Remove pepper noise in MRF image below using binary graphcuts.



The code part1.m implements this algorithm. The results for different m_lambda are shown below.

m_lambda = 16

m_lambda = 20

m_lambda = 25

m_lambda = 30 (Best result)

m_lambda = 35

m_lambda = 40

The best m_lambda value lies between 30 and 35 without treating the blue colour MRF and the outline. A good result can be considered as m_lambda = 30 as shown above.

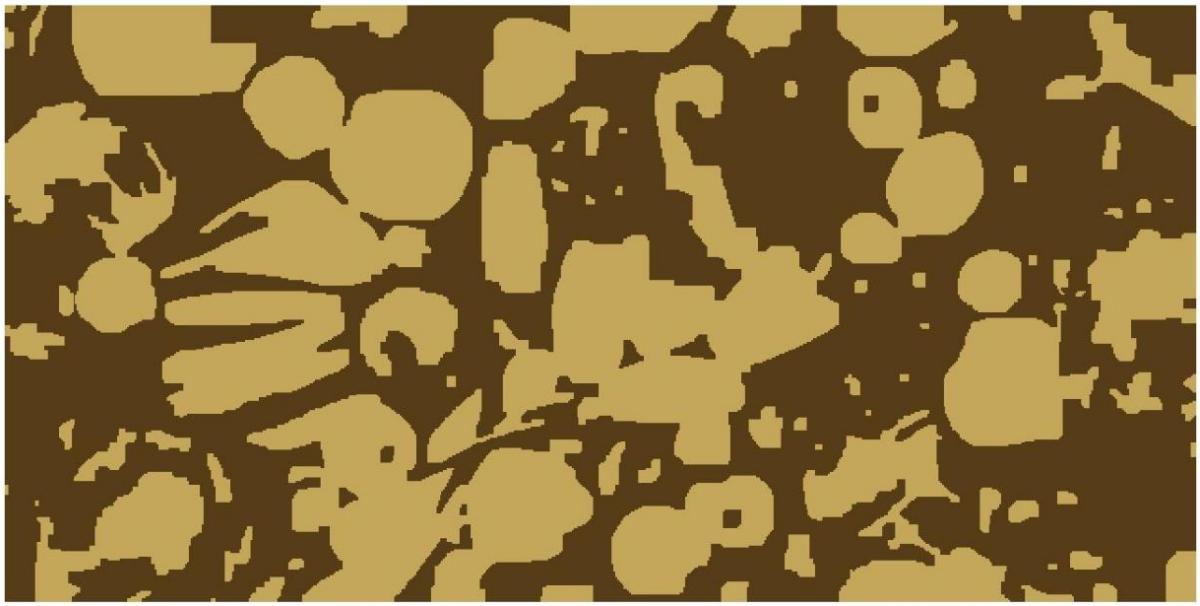
Part 2: Colour Segmentation

The code part2.m implements K-means clustering to obtain k number of segmented colours followed by building covariance matrix.

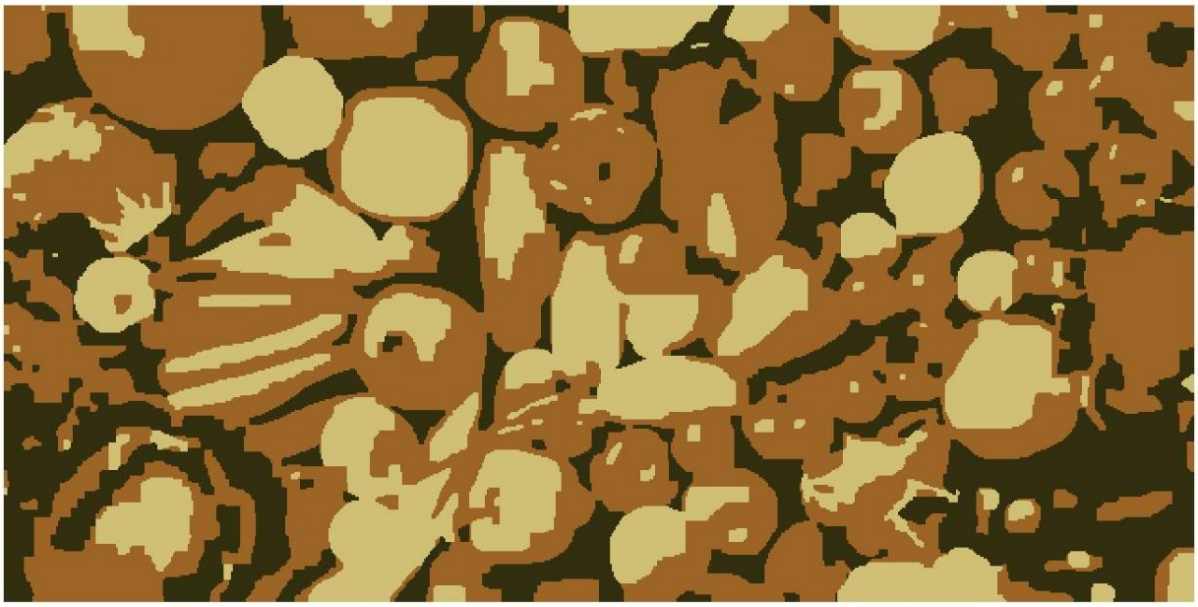
The results are shown below.

Original image

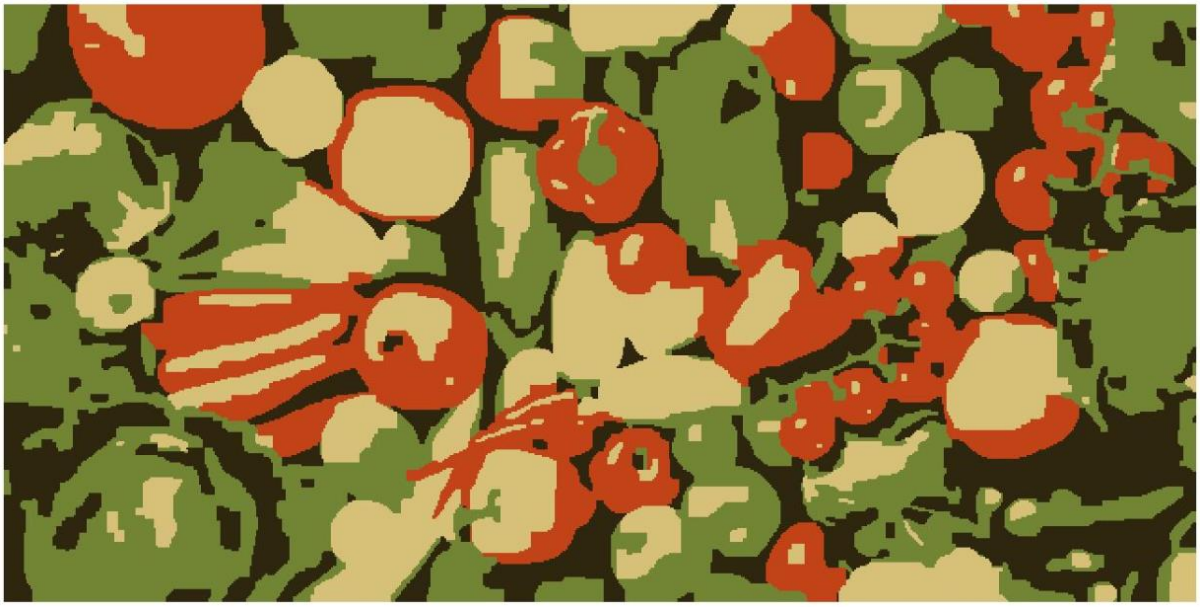
Segmentation results for k = 2



Segmentation results for k = 3



Segmentation results for k = 4



Segmentation results for k = 8

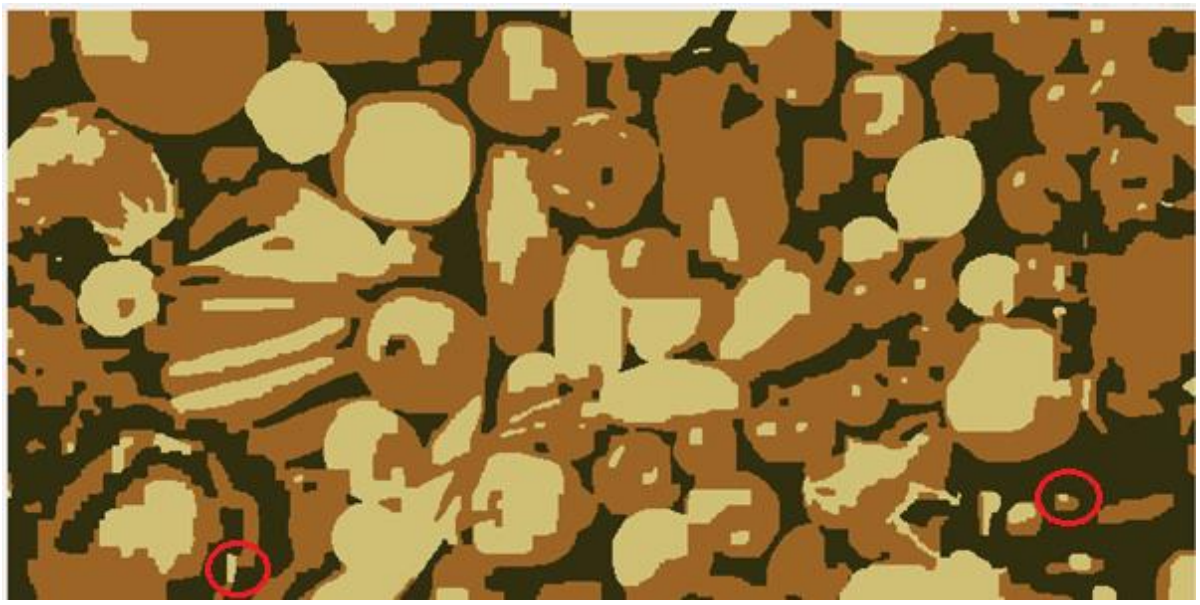
Segmentation results for k = 15

Segmentation results for k = 25

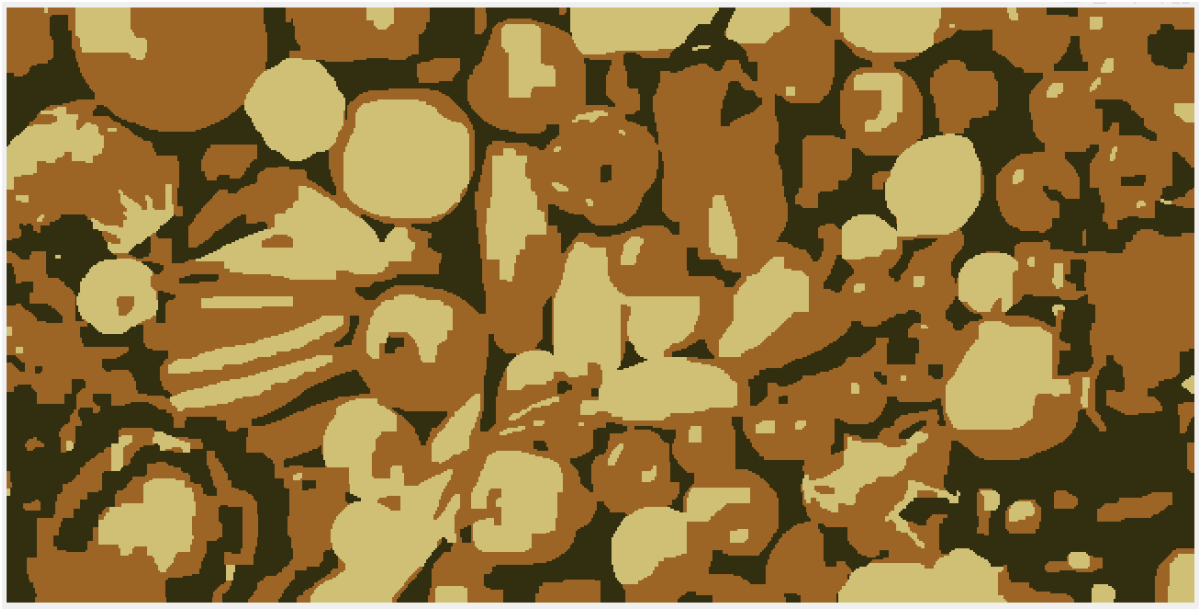
Drawbacks of the algorithm

1) Time consumption is high as the k centers become larger even though sparse matrix was used. The problem is computationally difficult (NP-hard);
2) K-means can converge to a “local minimum” not always the global minimum. It only detects spherical clusters only.
3) The problem with K-means clustering is that it is highly sensitive to initial choice of random k center points. For example, k_clusters = 3 when run twice gives a slightly different result as shown by red circles below. It is also sensitive to outliers.

Run 1



Run 2



4) Even with efficient graph cuts, an MRF formulation has too many nodes for interactive results.

Suggestions to improve the algorithm

1) Use alpha-expansion moves to speed up.

2) Use mean-shift which is Model-free, does not assume any prior shape (spherical etc.) on data clusters. It is just a single parameter (window size h). h has a physical meaning (unlike k-means). It finds variable number of modes and it is robust to outliers

2) We can use K-means++ algorithm to prevent arbitrarily bad local minima because it picks new centers with probability proportional to SSD of pixels.

4) We can use tricks like Superpixels to group together similar-looking pixels for efficiency of further processing. It is computationally cheap, local oversegmentation.