Yue Pan
yuepan@student.ethz.ch

Assignment 5

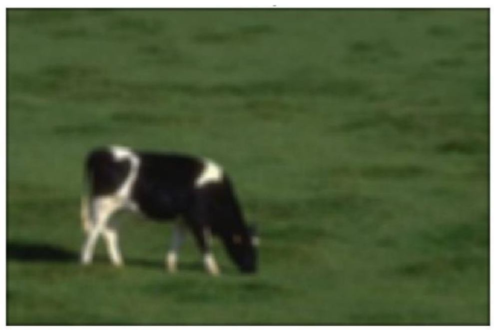
Task 1: Image Preprocessing

For this task, please refer to *exercise5.m*.

Before implementing image segmentation, it is recommended to do some preprocessing. Firstly, smooth the original image (Fig.1(a)) using a 5x5 Gaussian filter with $\sigma = 5$ to get rid of noise and make the color value in the image more uniform, as shown in Fig.1(b). Secondly, matlab functions makecform and applycform are adopted to transform the filtered image into the L*a*b* color space, as shown in Fig.1(c).

The L*a*b* color space expresses color as 3 values: L* for the lightness from black to white, a* from green to red, and b* from blue to yellow. For L*a*b*, the same amount of numerical change in these channels have roughly the same effect on visually perceived change. For image segmentation, L*a*b*'s advantage over RGB is that L*a*b* color space is more similar to human vision, which means it would have better performance on certain applications like image segmentation. For preprocessing result on zebra dataset, please refer to Fig.4(a),(b).


(a) original RGB image


(b) filtered RGB image


(c) L*A*B* image

Figure 1: Image preprocessing on cow dataset

Task 2: Mean-Shift Segmentation

For this task, please refer to *meanshiftSeg.m* and *find_peak.m*.

As for the mean-shift based image segmentation, we firstly create the density function X in L*a*b* color space, which is a $L \times 3$ matrix. L is the number of pixels in the image.

After that, the *find_peak* function is used to find the density peaks of X from every data point (which is corresponding to a certain pixel in the image). In this process, a window radius r should be determined. Then the window would gradually drift to the mass center of the points within it all the way to a density peak. The procedure would be iterated until the shifting distance of the window's mass center is smaller than a termination threshold.

Since there are a lot amount of distance computation during the mean-shift process, which is quite time-consuming, an efficient space indexing structure called k-d tree is adopted here. For all the points in the density function X, the k-d tree is constructed at the beginning using the *createns(X,'nsmethod','kdtree')* function. Then this kd-tree is also sent into the *find_peak* function to speed up the distance calculation.

For each data point, after finding its peak, the peak is compared to all the unique peaks that are already found. If the distance between them is smaller than a half of window radius, the latest found peak would not be saved (two peaks are merged together).

Finally, by setting the termination threshold as 1 and varying the window radius r from 5 to 30, we got different results shown in Fig 2. It can be noticed that the larger r is, the less peaks (segments) would be found. This may due to the fact that the merging threshold would increase as r increases.

The same experiment is also done on zebra dataset. You can refer to Fig.4 (c)(d) for details.

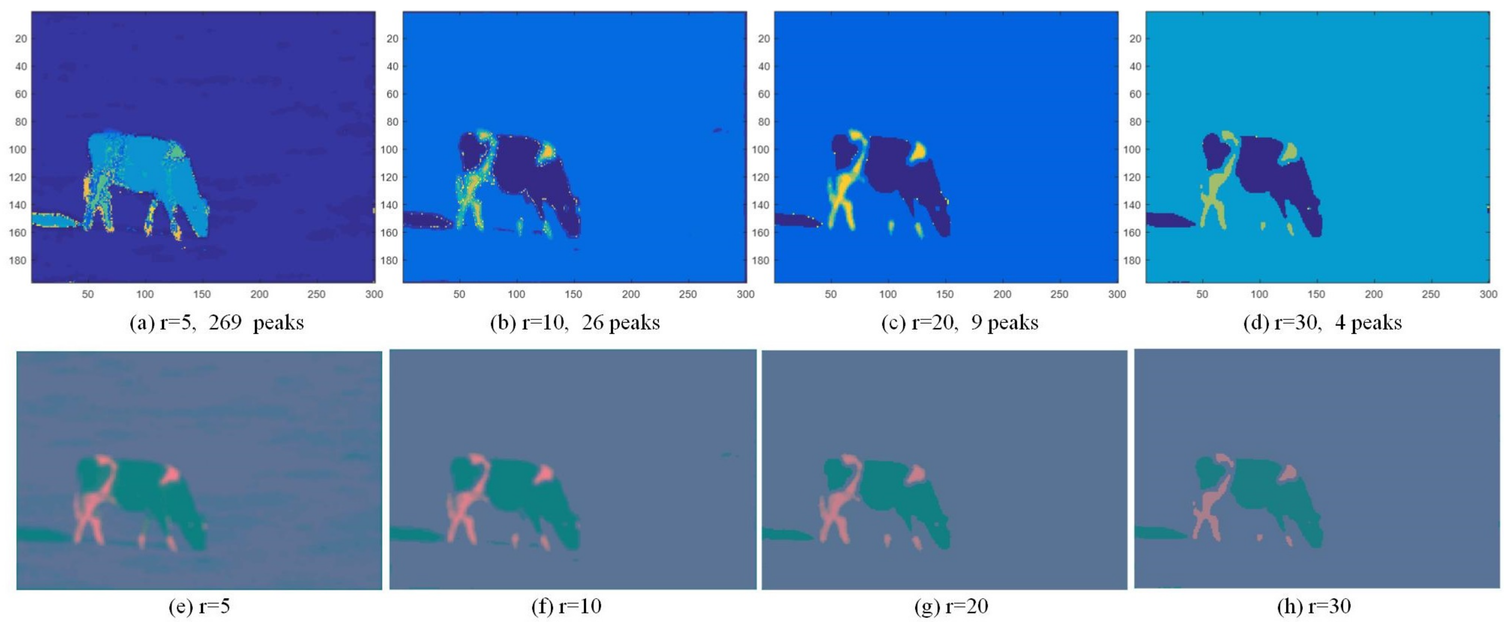


Figure 2: Image segmentation using Mean-shift on cow dataset w.r.t different shift window radius. For (a)-(d), segments are shown in their label space. For (e)-(h) segments are shown in L*a*b space.

Task 3: EM Segmentation

For this task, please refer to *EM.m* ,*generate_mu.m*, *generate_cov*, *expectation.m* and *maximization.m*.

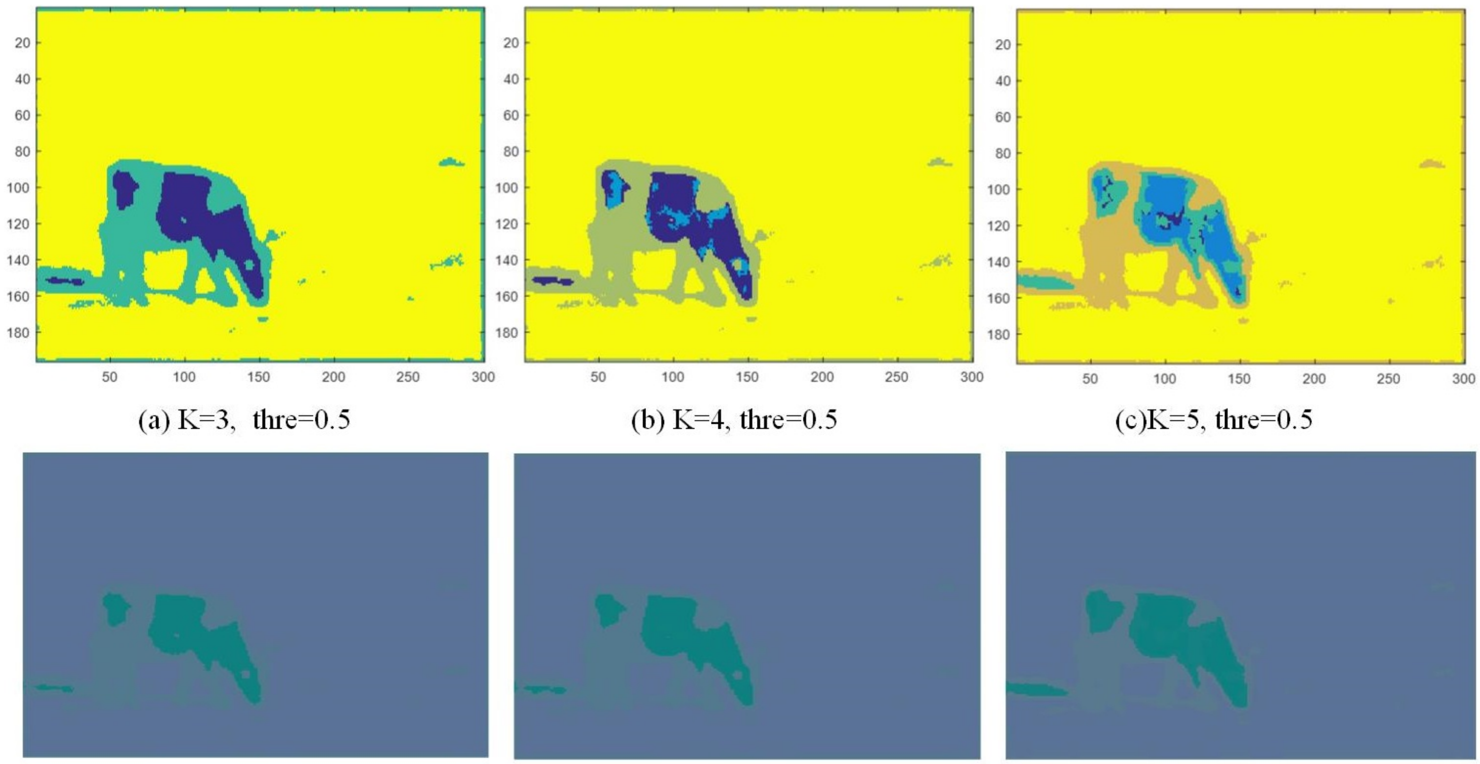
Expectation Maximization (EM) algorithm can also be used to realize image segmentation. In this case, the whole image is assumed as a kind of Gaussian Mixture Model (GMM). However, we don't know the parameters Θ (the weight α, the mean μ and the covariance matrix Σ) of each Gaussian kernel. There are 3 steps that are essential in EM algorithm:

The first is the parameter initialization. The weight α is initialized as $1/K$, in which K is the number of Gaussian kernel. All the covariance matrices Σ are initialized as a diagonal matrix with elements corresponding to the range of the L*, a* and b* values. The Gaussian kernels' center μ are initialized as spreading equally in the L*, a* and b* space.

The second one is the expectation. In this step, given the initial Θ, every pixel's probability in each Gaussian distribution can be estimated.

The third one is the likelihood maximization. A MLE (Maximum Likelihood Estimation) is done given the probability estimated in the expectation step to get the new GMM parameters.

The expectation and maximization steps are conducted iteratively. The final convergence would be reached when the distance between the GMM parameters of two adjacent iterations (for example μ) is smaller than a fixed threshold (set as 0.5 in my case).



(a) K=3, thre=0.5 (b) K=4, thre=0.5 (c)K=5, thre=0.5

(d) K=3, thre=0.5 (e) K=4, thre=0.5 (f)K=5, thre=0.5

Figure 3: Image segmentation using EM on cow dataset w.r.t different K value. For (a)-(c), segments are shown in their label space. For (d)-(f) segments are shown in L*a*b space.

The final result for different Gaussian kernel number K is shown in Fig.3. The corresponding Θ results are listed as following for K ranging from 3 to 5 and termination threshold set as 0.5. From the result of Θ, it's easy to notice that the increasing of K acts like further dividing a segment into more segments. Besides, there's always a gaussian kernel with a dominant weight (corresponding to the background of the image) which is larger than 0.8.

Result of Θ when $K = 3$:

$$\mu_1 = \begin{bmatrix} 14.839 & 128.673 & 128.476 \end{bmatrix} \tag{1}$$

$$\mu_2 = \begin{bmatrix} 82.143 & 121.482 & 141.661 \end{bmatrix} \tag{2}$$

$$\mu_3 = \begin{bmatrix} 89.026 & 114.442 & 149.098 \end{bmatrix} \tag{3}$$

$$\alpha_1 = 0.042,\ \alpha_2 = 0.124,\ \alpha_3 = 0.834 \tag{4}$$

$$\Sigma_1 = \begin{bmatrix} 13.092 & 2.905 & -1.206 \\ 2.905 & 2.293 & -0.793 \\ -1.206 & -0.793 & 2.682 \end{bmatrix} \tag{5}$$

$$\Sigma_2 = 1000 \times \begin{bmatrix} 2.347 & 0.074 & 0.055 \\ 0.074 & 0.023 & -0.019 \\ 0.055 & -0.019 & 0.032 \end{bmatrix} \tag{6}$$

$$\Sigma_3 = \begin{bmatrix} 58.295 & 0.022 & 0.796 \\ 0.022 & 0.875 & -0.188 \\ 0.796 & -0.188 & 1.586 \end{bmatrix} \tag{7}$$

Result of Θ when $K = 4$:

$$\mu_1 = \begin{bmatrix} 15.043 & 128.154 & 128.702 \end{bmatrix} \tag{8}$$

$$\mu_2 = \begin{bmatrix} 15.384 & 129.703 & 128.263 \end{bmatrix} \tag{9}$$

$$\mu_3 = \begin{bmatrix} 83.237 & 121.450 & 141.782 \end{bmatrix} \tag{10}$$

$$\mu_4 = \begin{bmatrix} 89.011 & 114.444 & 149.098 \end{bmatrix} \tag{11}$$

$$\alpha_1 = 0.032,\ \alpha_2 = 0.012,\ \alpha_3 = 0.121,\ \alpha_4 = 0.836 \tag{12}$$

$$\Sigma_1 = \begin{bmatrix} 17.938 & 2.630 & -0.800 \\ 2.630 & 2.204 & -1.428 \\ -0.800 & -1.428 & 2.722 \end{bmatrix} \tag{13}$$

$$\Sigma_2 = \begin{bmatrix} 9.423 & 2.934 & -0.692 \\ 2.934 & 2.072 & 0.209 \\ -0.692 & 0.209 & 3.356 \end{bmatrix} \tag{14}$$

$$\Sigma_3 = 1000 \times \begin{bmatrix} 2.347 & 0.081 & 0.045 \\ 0.081 & 0.022 & -0.018 \\ 0.045 & -0.018 & 0.030 \end{bmatrix} \tag{15}$$

$$\Sigma_4 = \begin{bmatrix} 58.547 & 0.009 & 0.818 \\ 0.009 & 0.880 & -0.190 \\ 0.818 & -0.190 & 1.593 \end{bmatrix} \tag{16}$$

Result of Θ when $K = 5$:

$$\mu_1 = \begin{bmatrix} 17.26 & 129.92 & 126.26 \end{bmatrix} \tag{17}$$

$$\mu_2 = \begin{bmatrix} 12.55 & 128.13 & 128.50 \end{bmatrix} \tag{18}$$

$$\mu_3 = \begin{bmatrix} 23.31 & 128.09 & 130.84 \end{bmatrix} \tag{19}$$

$$\mu_4 = \begin{bmatrix} 89.40 & 120.27 & 142.92 \end{bmatrix} \tag{20}$$

$$\mu_5 = \begin{bmatrix} 89.05 & 114.44 & 149.10 \end{bmatrix} \tag{21}$$

$$\alpha_1 = 0.002,\ \alpha_2 = 0.024,\ \alpha_3 = 0.031,\ \alpha_4 = 0.110,\ \alpha_5 = 0.833 \tag{22}$$

$$\Sigma_1 = \begin{bmatrix} 2.073 & 0.175 & 0.098 \\ 0.175 & 3.172 & 3.285 \\ 0.098 & 3.284 & 3.665 \end{bmatrix} \tag{23}$$

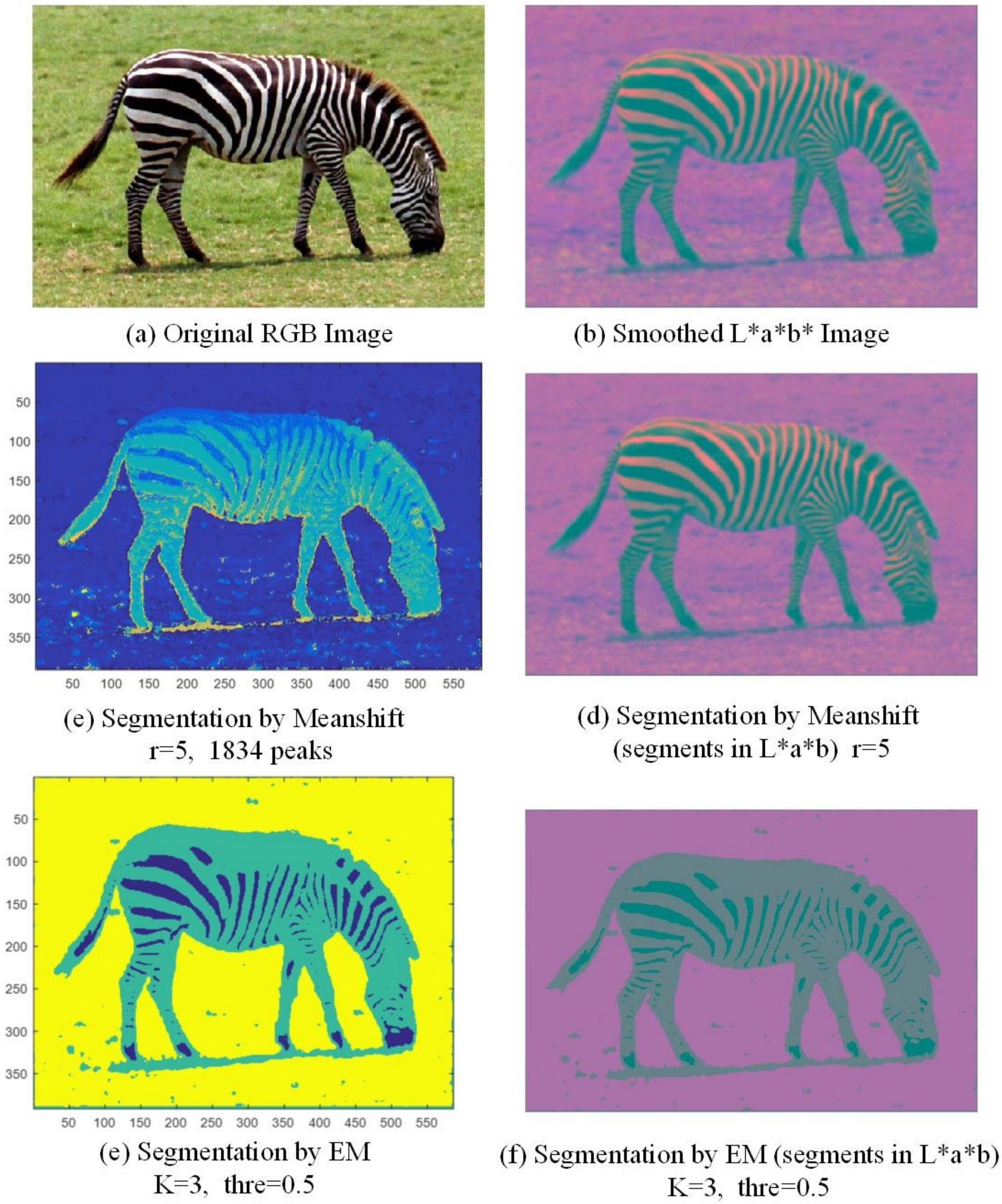
$$\Sigma_2 = \begin{bmatrix} 1.676 & 0.252 & -0.479 \\ 0.252 & 0.583 & -0.394 \\ -0.479 & -0.394 & 1.346 \end{bmatrix} \tag{24}$$

$$\Sigma_3 = \begin{bmatrix} 76.348 & 0.671 & 12.619 \\ 0.671 & 8.095 & -5.172 \\ 12.619 & -5.172 & 9.757 \end{bmatrix} \tag{25}$$

$$\Sigma_4 = 1000 \times \begin{bmatrix} 2.191 & 0.128 & -0.013 \\ 0.128 & 0.020 & -0.013 \\ -0.013 & -0.013 & 0.023 \end{bmatrix} \tag{26}$$

$$\Sigma_5 = \begin{bmatrix} 57.907 & 0.093 & 0.759 \\ 0.093 & 0.871 & -0.188 \\ 0.759 & -0.188 & 1.577 \end{bmatrix} \tag{27}$$

The same experiment is also done on zebra dataset. You can refer to Fig.4 (e)(f) for details (On next page).



(a) Original RGB Image

(b) Smoothed L*a*b* Image

(e) Segmentation by Meanshift
r=5, 1834 peaks

(d) Segmentation by Meanshift
(segments in L*a*b) r=5

(e) Segmentation by EM
K=3, thre=0.5

(f) Segmentation by EM (segments in L*a*b)
K=3, thre=0.5

Figure 4: Image segmentation experiment results on zebra dataset.