Computer Vision Exercise 5: Image Segmentation

Soomin Lee (leesoo@student.ethz.ch)

December 1, 2020

1. Image Preprocessing

Before performing image segmentation algorithms on an image, we should first smooth the image so that the color becomes uniform, which will lead to better results. Therefore, a 5×5 Gaussian filter with $\sigma = 5.0$ is applied to the image. After smoothing, the image is transformed to the feature space we are going to use - L*a*b* color space. In L*a*b* color space, L* represents the lightness of color, and a* and b* channels are chromatic components. It is designed to be perceptually uniform compared to RGB color space, meaning that a change of the same amount in a color value should produce a change of about the same visual importance. I believe this property is the reason why it is better to do segmentation in the L*a*b* color space than in the RGB color space. The original image, the smoothed image, and the converted image of a cow are shown in Figure 1.

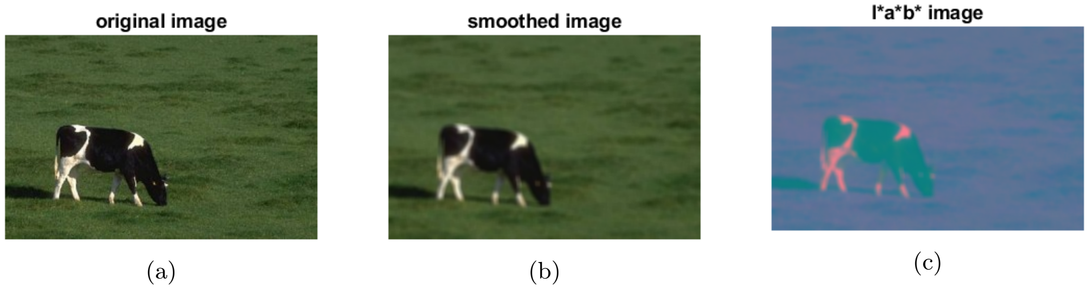


Figure 1: Original image (a), image smoothed with a Gaussian filter (b), and image converted to the L*a*b* color space (c).

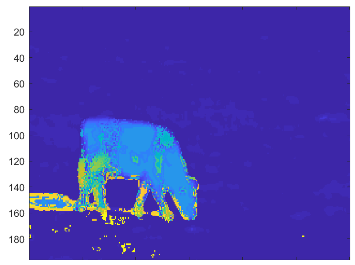
Remark When we transform the image into the L*a*b* color space using the functions `makecform` and `applycform` from Matlab, they convet the image to the L*a*b* values in `uint8` format. This gives a better visualization of the images with `imshow`, so for all the figures in the report are visualized in this way. However, when calculating the parameter values in EM segmentation, the function `rgb2lab` was used which transforms the values into the actual standard range so one can get a better intuition of each value.

2. Mean-Shift Segmentation

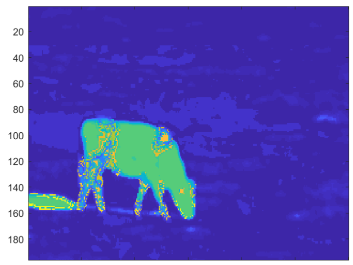
To perform the mean-shift algorithm, first the density map is obtained from the image by resizing the preprocessed image into a $N \times 3$ matrix where N is the total number of pixels and the 3 columns correspond to the L*a*b* color space values.

Then for each pixel, we find the peak it belongs to in the density map (given a function that finds the peak). If the peak has a distance bigger than $r/2$ from all the previously found peaks, the peak is added to the list of peaks. Otherwise, the newly found peak is merged to the one that is less than $r/2$ away. The index of the peak where each pixel belongs to is saved in the process.

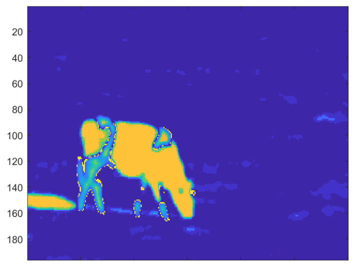
The function that finds the peak for each pixel is implemented separately. Each loop in the function computes the mean of all pixels that lie within a spherical window of radius r, and then shift the center of the window



(a) Segmented image when $r = 5$.

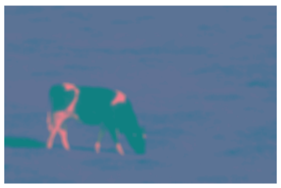


(b) Segmented image when $r = 10$.



(c) Segmented image when $r = 20$.

(d) Validation image when $r = 5$.


(e) Validation image when $r = 10$.

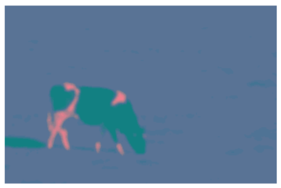

(f) Validation image when $r = 20$.

Figure 2: Results using Mean-Shift segmentation method with different radius r.

to the mean. The function repeatedly performs this procedure until the mean converges, or the difference is under the predefined threshold.

The results of image segmentation using this method with different radius r of the spherical image is shown in Figure 2. The first row illustrates the result by coloring pixels with the same color that are associated with the same peak. The second row shows the validation of each segmentation by reconstructing the image in $L^*a^*b^*$ color space using the peaks. One can compare the images with Figure 1c. As the radius grows, the number of peaks decreases as more peaks are merged together. Therefore, the pixels are divided into a fewer groups and the validation image looks smoother. It loses the details compared to a smaller r but simplifies the image which can make it easier to segment objects from the image.

The advantage of using the mean-shift method is that it is a model-free method and one doesn't have to assume any prior shape of data clusters. However, it is computationally expensive as it has to find a peak for every pixel.

3. EM Segmentation

One can perform image segmentation in a probabilistic way using EM algorithm. Given the number of segments, the algorithm models each segment as a Gaussian. Then it iterates through expectation and maximization steps. In expectation step, the probability that the data point is in segment $k = 1, ..., K$ is computed given the parameters from the previous maximization step. In maximization step, the parameters are updated given the probability from the previous expectation step. The equations needed for the implementation can be found on the exercise slides and the code. The iterations are done until the maximum component of the difference in the mean μ is smaller than a certain threshold.

The initial values of the mean μ are set in a way that they are uniformly distributed - each dimension was divided into $K+1$ segments, and K points in between the maximum and minimum value are set as the initial mean value for each cluster. Then the covariance matrices are set as diagonal matrices - each component corresponding to the range of the L^*, a^*, and b^* values respectively. The weight α is initialized as $1/K$ for each cluster, giving them uniform weights.

The results for $K = 3, 4, 5$ are shown in Figure 3. The figures are generated in the same fashion as Figure 2. Since we are directly setting the number of clusters in this case, it is obvious that pixels are divided into more groups as K grows. Thus, compared to Figure 2, it has the opposite tendency as the parameter changes from the left to the right. Also, the segmented images obtained by EM algorithm tend to have smoother clustering due to the underlying Gaussian model of the EM algorithm, while there is no model assumption in mean-shift algorithm. Lastly, the final values of the parameters are as follows:

$K = 3$:

$$\begin{aligned}
&\mu_1 = \begin{bmatrix}17.2834 & -5.6343 & 9.5592\end{bmatrix}, \mu_2 = \begin{bmatrix}34.8692 & -15.3895 & 21.2536\end{bmatrix}, \mu_3 = \begin{bmatrix}55.3380 & -3.5457 & 12.4170\end{bmatrix}\\
&\Sigma_1 = \begin{bmatrix}130.8014 & -63.9929 & 96.0689\\ -63.9929 & 43.1628 & -55.8174\\ 96.0689 & -55.8174 & 80.8977\end{bmatrix}, \Sigma_2 = \begin{bmatrix}8.5838 & 0.0666 & 0.0436\\ 0.0666 & 0.8435 & -0.2729\\ 0.0436 & -0.2729 & 1.5286\end{bmatrix},\\
&\Sigma_3 = \begin{bmatrix}346.6425 & 20.7445 & 17.5509\\ 20.7445 & 10.9516 & -8.3884\\ 17.5509 & -8.3884 & 24.5501\end{bmatrix}\\
&\alpha_1 = 0.1119,\ \alpha_2 = 0.8527,\ \alpha_3 = 0.0354
\end{aligned} \tag{1}$$

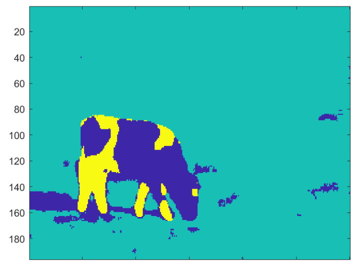
$K = 4$:

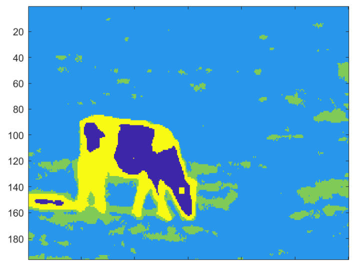
$$\begin{aligned}
&\mu_1 = \begin{bmatrix}6.1984 & 0.4924 & 0.8118\end{bmatrix}, \mu_2 = \begin{bmatrix}34.9000 & -15.3982 & 21.2477\end{bmatrix},\\
&\mu_3 = \begin{bmatrix}27.5043 & -11.4730 & 17.6522\end{bmatrix}, \mu_4 = \begin{bmatrix}49.7324 & -2.5836 & 10.8431\end{bmatrix}\\
&\Sigma_1 = \begin{bmatrix}3.4656 & 1.2052 & 0.4547\\ 1.2052 & 3.1101 & -1.5357\\ 0.4547 & -1.5357 & 3.6253\end{bmatrix}, \Sigma_2 = \begin{bmatrix}8.4041 & 0.0774 & 0.0462\\ 0.0774 & 0.8210 & -0.2776\\ 0.0462 & -0.2776 & 1.5034\end{bmatrix},\\
&\Sigma_3 = \begin{bmatrix}63.3246 & -15.4232 & 33.6356\\ -15.4232 & 14.8819 & -17.1935\\ 33.6356 & -17.1935 & 30.1123\end{bmatrix}, \Sigma_4 = \begin{bmatrix}535.8247 & -4.1318 & 60.7440\\ -4.1318 & 8.9148 & -7.8989\\ 60.7440 & -7.8989 & 25.8682\end{bmatrix}\\
&\alpha_1 = 0.0469,\ \alpha_2 = 0.8449,\ \alpha_3 = 0.0689,\ \alpha_4 = 0.0383
\end{aligned} \tag{2}$$

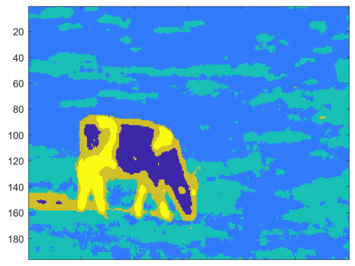
$K = 5$:

$$\begin{aligned}
&\mu_1 = \begin{bmatrix}6.0371 & 0.5385 & 0.7030\end{bmatrix}, \mu_2 = \begin{bmatrix}33.4476 & -15.4921 & 21.9453\end{bmatrix},\\
&\mu_3 = \begin{bmatrix}36.2327 & -15.2619 & 20.5320\end{bmatrix}, \mu_4 = \begin{bmatrix}25.3522 & -7.8478 & 13.3877\end{bmatrix},\\
&\mu_5 = \begin{bmatrix}62.6642 & -2.5461 & 12.0841\end{bmatrix}\\
&\Sigma_1 = \begin{bmatrix}2.7821 & 1.1925 & 0.1265\\ 1.1925 & 2.8026 & -1.2973\\ 0.1265 & -1.2973 & 3.2216\end{bmatrix}, \Sigma_2 = \begin{bmatrix}11.3036 & -0.2627 & 2.1070\\ -0.2627 & 1.2565 & -0.2451\\ 2.1070 & -0.2451 & 1.6594\end{bmatrix},\\
&\Sigma_3 = \begin{bmatrix}2.7938 & 0.0126 & 0.0150\\ 0.0126 & 0.5234 & -0.0844\\ 0.0150 & -0.0844 & 0.4058\end{bmatrix}, \Sigma_4 = \begin{bmatrix}102.8443 & -23.6233 & 45.1288\\ -23.6233 & 23.0210 & -25.5137\\ 45.1288 & -25.5137 & 39.0804\end{bmatrix},\\
&\Sigma_5 = \begin{bmatrix}262.6097 & 6.9683 & 22.6037\\ 6.9683 & 5.9006 & -4.3464\\ 22.6037 & -4.3464 & 21.8951\end{bmatrix}\\
&\alpha_1 = 0.0447,\ \alpha_2 = 0.4543,\ \alpha_3 = 0.4138,\ \alpha_4 = 0.0615,\ \alpha_5 = 0.0256
\end{aligned} \tag{3}$$

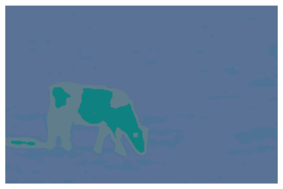
Note that the values of $L*$, $a*$, and $b*$ are in range of 0 to 100, -128 to 128, -128 to 128 respectively. One can also infer which mean values correspond to which region by comparing the images and values. For instance, the dark part of the cow's pattern is likely to be the cluster centered around μ_1 for all cases since the lightness value should be low. Then this hypothesis aligns well with the images since in Figure 3b and 3c, the darkest region are almost the same just like the first component value of $\mu_1^{k=4}$ and $\mu_1^{k=5}$, while in


(a) Segmented image when $K = 3$.


(b) Segmented image when $K = 4$.


(c) Segmented image when $K = 5$.

(d) Validation image when $K = 3$.


(e) Validation image when $K = 4$.

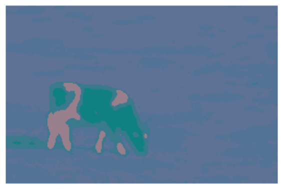
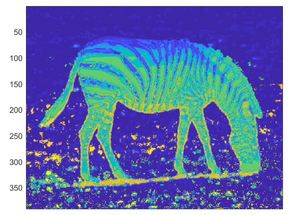
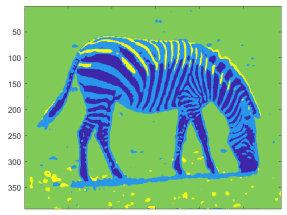

(f) Validation image when $K = 5$.

Figure 3: Results using EM segmentation method with different number of segments K.


(a) Segmented image using mean-shift.

(b) Validation image using mean-shift.


(c) Segmented image using EM.

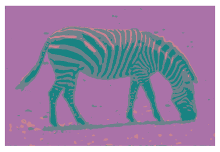

(d) Validation image using EM.

Figure 4: Image segmentation on zebra image. Radius $r = 10$ is used for mean-shift, and the number of clusters $K = 4$ is used for EM.

Figure 3a the darkest region needs to include a bit brighter region as well so the first component of $\mu_1^{k=3}$ is larger than the others. Likewise, one can observe how the cluster that most of the background belongs to in Figure 3a separates into two in Figure 3c by looking at the parameter values as well.

The advantage of the EM segmentation is that it enables soft or probabilistic cluster assignments to data points, and that it can predict novel data points since it is a generative model. However, one might have difficulties initializing the parameters and setting an appropriate value for the number of segments K. It will converge to a local optimum so the result depends on the initialization.

4. Zebra example

The image segmentation is performed on the zebra image in the same way as before. The radius r is set as 10 for the mean-shift algorithm, and the number of cluster K is set as 4 for the EM algorithm. The result can be interpreted in the same way as before as well, and both algorithms enabled us to segment the background and the zebra, and the pattern of the animal on its back. The resulting images are shown in Figure 4.