Computer Vision Assignment 5: Image Segmentation

Charlotte Moraldo

1. Image Preprocessing

Before applying image segmentation, it is necessary to smooth the image so that the color becomes more uniform. We do so by applying a 5x5 Gaussian filter with $\sigma = 5$ (figure 2). Then, we use the matlab functions `makecform` and `applycform` in order to transform the smoothed image in the L*a*b* color space (figure 3).

The L*a*b* color space expresses color as 3 numerical values: L* for lightness, a* for the green-red color components, and b* for the blue-yellow color components. In the case of image segmentation, using L*a*b* makes more sense as using RGB because clusters with a similar color but different lightness will have closer values, and will more likely be segmented together. This corresponds more closely to the color difference perceived by the human eye., which is what's desired in the case of image segmentation.

Figure 1: Original image

Figure 2: Smoothed image

Figure 3: L*a*b image

2. Mean-Shift Segmentation

After preprocessing the image, we can apply mean-shift segmentation. We begin by creating the density function X in the L*a*b* space. X is a 3xL matrix, where L is the number of pixels and $n = 3$ is the dimension of the chosen color space. In order to create this function, we take the pixel value of each dimesion of the L*a*b* space, put them into a vector and finally concatenate the 3 resulting vectors into a matrix X.

Then, the function `find_peak` is called in order to find the mode of X for each pixel. We compute the distance of this pixel to all the other ones in the L*a*b* space, then keep all those for which this distance is smaller than a chosen radius r. Then, we shift the window to the mean of these selected points. This process is iterated until the shift between the pixel and the new center of mass is smaller than a certain threshold.

The pictures below are examples of the results of mean-shift segmentation. For each of them, I varied the size of the radius ($r = 5$, $r = 15$, and $r = 31$), resulting in a different number of peaks (258, 15, and 3 respectively). The picture on the left shows each cluster in a different colour, while the one on the right shows the reconstruction by assigning the peak colours to each colour, in the L*a*b* space.

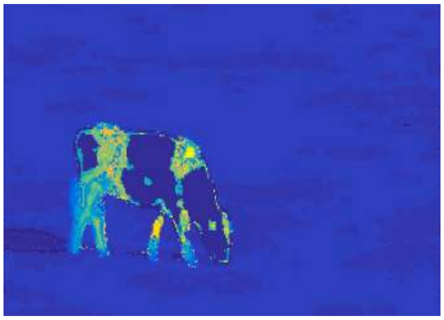

Figure 4: $r = 5$, 258 peaks

Figure 5: $r = 5$, 258 peaks

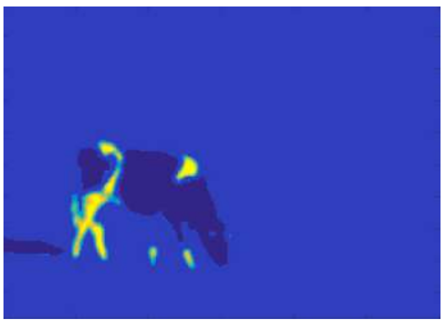

Figure 6: $r = 15$, 15 peaks

Figure 7: $r = 15$, 15 peaks

Figure 8: $r = 31$, 3 peaks

Figure 9: $r = 31$, 3 peaks

3. EM Segmentation

We begin by initializing the prior probability α, the covariance Σ, and the mean μ. For α, we set each weight equal to $1/K$, where K is the number of segments. Σ is a 3x3 diagonal matrix where the diagonal corresponds to the range of the L*, a*, and b* values. Finally, μ is is a 3x1 vector representing a point in the L*a*b* space, which we initialize by spreading them equally in the L*a*b* space.
After this initialization step, we alternate between maximization and expectation, and iterate until convergence. In the expectation step, we compute for each pixel its probability γ_{lk} to be in each segment from α, Σ or μ. In the maximization, we update α, Σ and μ such that they maximize the log likelihood under the current probabilities. Convergence is reached when the norm between the current μ and the updated one is under a predefined threshold.

The results presented below show a much better segmentation than with the mean-shift, even with fewer clusters. We can especially see this at the legs of the cow or at the areas of grass in the background, which weren't visible with mean-shift but are now much more present.

K = 3 segments

For 3 segments, we set the convergence threshold at 0.3 it was reached after 12 iterations, generating the following segmentation and reconstruction:

Figure 10: EM segmentation with K = 4

Figure 11: Reconstruction after segmentation

The values obtained for α, Σ and μ are as follows:

$$\Sigma_1 = \begin{bmatrix} 707.8874 & -139.5714 & 212.5892 \\ -139.5714 & 36.6201 & -48.4543 \\ 212.5892 & -48.4543 & 72.0556 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 53.3432 & 0.0217 & 0.5167 \\ 0.0217 & 0.7879 & -0.1588 \\ 0.5167 & -0.1588 & 1.5289 \end{bmatrix}$$

$$\Sigma_3 = 10^3 \cdot \begin{bmatrix} 2.4250 & 0.0977 & -0.0015 \\ 0.0977 & 0.0153 & -0.0120 \\ -0.0015 & -0.0120 & 0.0295 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 47.7056 & 89.3955 & 130.2091 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} 121.4757 & 114.4474 & 124.4317 \end{bmatrix}$$

$$\mu_3 = \begin{bmatrix} 139.1949 & 149.0961 & 141.4379 \end{bmatrix}$$

$$\alpha = \begin{bmatrix} 0.1460 & 0.8105 & 0.0435 \end{bmatrix}$$

K = 4 segments

For 4 segments, convergence was reached after 32 iterations, generating the following segmentation and reconstruction:

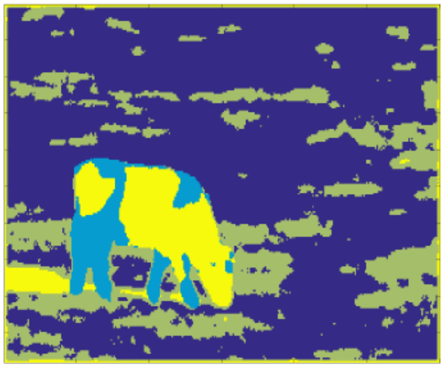

Figure 12: EM segmentation with K = 4

Figure 13: Reconstruction after segmentation

The values obtained for α, Σ and μ are as follows:

$$\Sigma_1 = \begin{bmatrix} 29.1829 & -0.0994 & 0.7465 \\ -0.0994 & 0.5558 & -0.1731 \\ 0.7465 & -0.1731 & 1.3944 \end{bmatrix}$$

$$\Sigma_2 = 10^3 \cdot \begin{bmatrix} 2.6499 & 0.0864 & 0.0348 \\ 0.0864 & 0.0140 & -0.0088 \\ 0.0348 & -0.0088 & 0.0244 \end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix} 55.9140 & 0.0433 & 6.2366 \\ 0.0433 & 2.6485 & -0.7286 \\ 6.2366 & -0.7286 & 3.2505 \end{bmatrix}$$

$$\Sigma_4 = \begin{bmatrix} 367.4413 & -78.4743 & 116.2718 \\ -78.4743 & 24.8426 & -30.8761 \\ 116.2718 & -30.8761 & 44.2182 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 91.8081 & 114.4414 & 149.0145 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} 125.8355 & 124.5895 & 140.7482 \end{bmatrix}$$

$$\mu_3 = \begin{bmatrix} 79.5576 & 114.7137 & 149.1387 \end{bmatrix}$$

$$\mu_4 = \begin{bmatrix} 31.7544 & 124.6505 & 134.3988 \end{bmatrix}$$

$$\alpha = \begin{bmatrix} 0.6375 & 0.0458 & 0.2246 & 0.0921 \end{bmatrix}$$