Computer Vision

Image Segmentation

Student: Yuchang Jiang

Date: 5 Nov. 2020

1. Image Processing

1.1 Implementation

Before segmentation, the first step is to smooth image: here Gaussian filter with 5x5 window size and sigma=5 is used. Then image is converted from RGB to L*a*b color space.

1.2 Result and Discussion

(a) original image (b) smoothed image (c) L*a*b image

Figure 1: result of processed image

It is better to convert RGB to L*a*b color space before segmentation because in L*a*b color space, lightness and colors are separate features, pixels with similar color but different lightness are more possible to be clustered together. Besides, L*a*b is more similar to visually perception so more suitable for image segmentation.

2. Mean-Shift Segmentation

2.1 Implementation

After processing image, mean-shift segmentation is implemented. Each pixel is treated as a data point with three features in L*a*b space. For each data point, $find\_peak$

function is called to find peak. With the defined radius, neighboring points inside the neighborhood of the current data point are selected and the mean (centroid) point is computed. Then current data point is shifted towards mean point and this step is repeated until the shift is smaller than defined threshold (set as 1 here). Those peaks found for pixels will be merged if they are close to previous peaks ('close' is defined as half of radius here). In the end, each pixel is associated with the index of peaks.

2.2 Result and Discussion

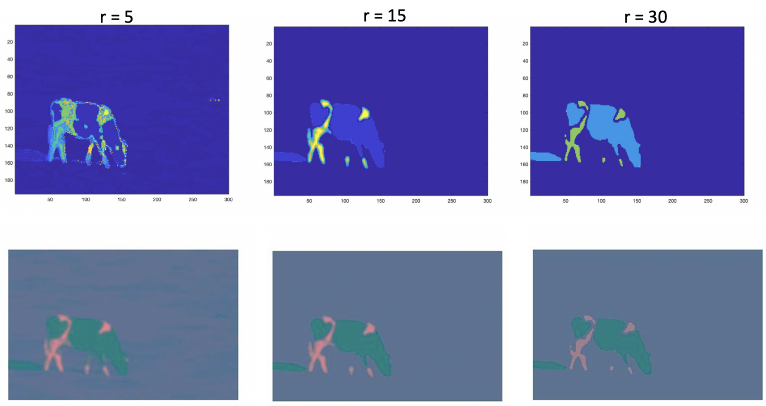

Figure 2: Result of mean-shift segmentation

Radius	5	15	30
Number of peaks	268	15	4

Table 1: Report of number of peaks

As shown in Figure 2, different radius (5, 15, 30) are tried for this method and both map results with peak index (top row) and peak color (bottom row) in L*a*b space are plotted. The number of resulted peaks is reported in Table 1. It is noticed a larger radius will lead to fewer peaks and more smooth result. Because large radius will make surrounding pixels have similar peaks, which can be merged, there are fewer peaks in the end.

3. EM Segmentation

3.1 Implementation

At first, all parameters are initialized. Alphas are initialized as uniform weight. Mean values are initialized based on range of three features in L*a*b space in order to spread them equally. Covariance matrix is initialized as diagonal matrix with computed range as diagonal elements. After initialization, expectation and maximization step are computed one after the other iteratively until the change of mean values is smaller than a defined threshold (here the threshold used is 0.8). In expectation step, the probability of data point given parameters in segment k is computed. Then in maximization, all parameters (alpha, mean, covariance) are re-computed by new result from expectation. In the end, each pixel has the probability of belonging to each segment. Then the segment with highest probability will be assigned to each pixel.

3.2 Result and Discussion

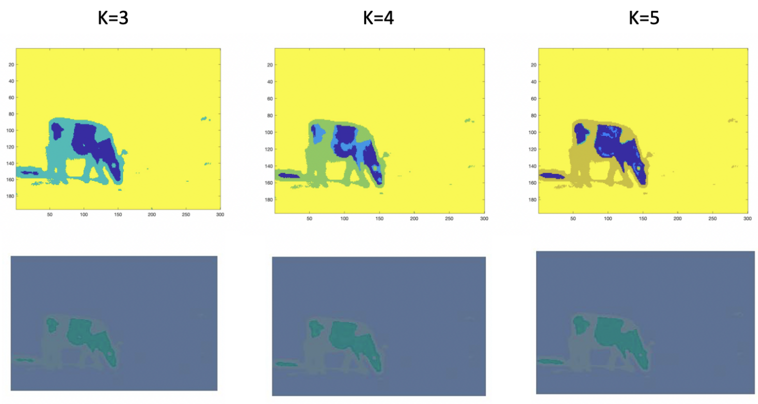

Figure 3: Result of EM segmentation

As shown in Figure 3, EM segmentation with different number of components (3, 4, 5) are performed.(The threshold used to terminate iterations is 0.8). These results look similar to each other because all of them show a dominant segment: background (or grass). In the result of K=5, more details are displayed: some textures on the cow are segmented while those details are omitted in the result of K=3.

More detailed results (θ) are shown in Figure 4 to Figure 6. By observing the alphas, the last alpha is much larger than others in all three EM results. This component may correspond to the large background area in original image (grass area).

For θ resulted in K=3:

```
alpha =

    0.0409    0.0900    0.8690


mu =

   14.7438  128.6651  128.4602
   89.5485  123.1248  140.5977
   88.9410  114.4285  149.1072


cov(:,:,1) =

   12.1792    2.7256   -1.2078
    2.7256    2.2312   -0.7732
   -1.2078   -0.7732    2.6021


cov(:,:,2) =

   1.0e+03 *

    2.9760    0.0606    0.0965
    0.0606    0.0209   -0.0189
    0.0965   -0.0189    0.0375


cov(:,:,3) =

   58.8537    0.2824    0.5181
    0.2824    0.8904   -0.1672
    0.5181   -0.1672    1.5731
```

Figure 4: Result of EM segmentation(K=3)

For θ resulted in K=4:

```
alpha =

    0.0263    0.0201    0.0848    0.8687


mu =

   13.0594  127.6638  128.8689
   19.5152  129.9292  128.6117
   93.7854  122.7508  141.2626
   88.9457  114.4275  149.1071



cov(:,:,1) =

    3.4563   -1.0984    0.7204
   -1.0984    1.4889   -1.2100
    0.7204   -1.2100    1.9151


cov(:,:,2) =

   27.3707    1.7044    2.5421
    1.7044    1.7011   -0.2541
    2.5421   -0.2541    5.6220


cov(:,:,3) =

   1.0e+03 *

    2.8592    0.0885    0.0580
    0.0885    0.0198   -0.0165
    0.0580   -0.0165    0.0333


cov(:,:,4) =

   58.7783    0.2913    0.5155
    0.2913    0.8885   -0.1677
    0.5155   -0.1677    1.5716
```

Figure 5: Result of EM segmentation(K=4)

For θ resulted in K=5:

```
alpha =

    0.0448    0.0038    0.0022    0.0810    0.8681


mu =

   16.4220  128.4273  129.0299
   11.0000  128.0623  128.5940
   34.3204  129.8385  132.1137
   97.2830  122.5032  141.7612
   88.9531  114.4262  149.1069
```

```
cov(:,:,1) =

   21.8979    1.8496    2.3901
    1.8496    3.8403   -2.0507
    2.3901   -2.0507    4.8132


cov(:,:,2) =

    0.0000    0.0000    0.0000
    0.0000    0.3532   -0.1164
    0.0000   -0.1164    0.3958


cov(:,:,3) =

    7.2225   -1.0396    0.1966
   -1.0396    0.7044   -0.8537
    0.1966   -0.8537    5.8646


cov(:,:,4) =

   1.0e+03 *

    2.7377    0.1070    0.0286
    0.1070    0.0198   -0.0159
    0.0286   -0.0159    0.0314


cov(:,:,5) =

   58.6541    0.2991    0.5120
    0.2991    0.8853   -0.1684
    0.5120   -0.1684    1.5691
```

Figure 6: Result of EM segmentation(K=5)