CV2015Spring—Assignment #1

Wu Bin

April 12, 2015

1. The process of this assignment

At the beginning, I decided to use Python+OpenCV to finish the assignment, but I gave up at last because I couldn't figure out some OpenCV lib problems. So, I adjust my strategy to use C++ and OpenCV to complete this assignment.

1.1 Input image

I choose an image from the database and rename it to test.png for test.

Figure 1: Input image named test.png.

1.2 Step 1-1: Extract image information

Input The input color image ($m \times n \times 3$ matrix), m is the width of the input image, n is the height of the input image.

Output $m \times n$ matrix.

Implementation In this step, I choose Color information for this assignment. First of all, I quantize each color channel(RGB) from 0-255 to 0-32, then I convert to a number($32 \times 32 \times 32$=32768). The results are as follow, and the numbers are saved in a $m \times n$ matrix.

Figure 2: Resize

Figure 3: Quantize

1.3 Step 1-2: Segment the input image to superpixels

Input The input color image ($m \times n \times 3$ matrix), m is the width of the input image, n is the length of the input image.

Output Superpixel segmentation matrix ($m \times n$ matrix), the value of the pixel in this matrix is just a label to indicate the superpixel it belongs to, and pixels of the same superpixel are labeled the same value.

Implementation I use an extensive library of OpenCV to do Superpixel segmentation and get result. The library is OpenCV_Contrib. As for my test picture, the number of superpixel areas is 304.

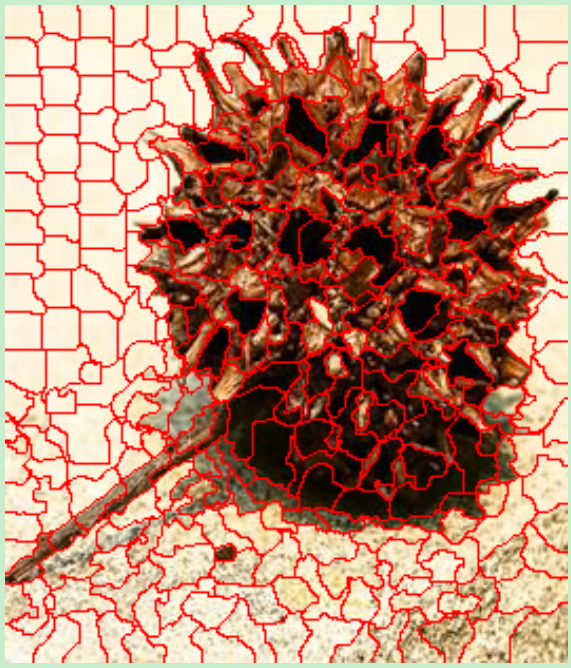

Figure 4: Superpixel Segmentation.

1.4 Step 2: Compute features of each superpixel

Input Image information matrix, superpixel segmentation matrix.

Output h histograms, h is the number of superpixels after segmentation.

Implementation There are 304 histograms.

1.5 Step 3: Compute superpixel feature contrast

Input h histograms.

Output h values, each value indicates the global feature contrast of a superpixel.

Implementation I compute global regional contrast, and the average contrast is the saliency value of every superpixel area.

1.6 Step 4: Convert superpixel saliency to pixel saliency

Input h values, superpixel segmentation matrix.

Output An initial saliency map ($m \times n$ matrix).

Implementation All the pixels of the seam superpixel are assigned the same saliency value. The initial saliency map is as follow.

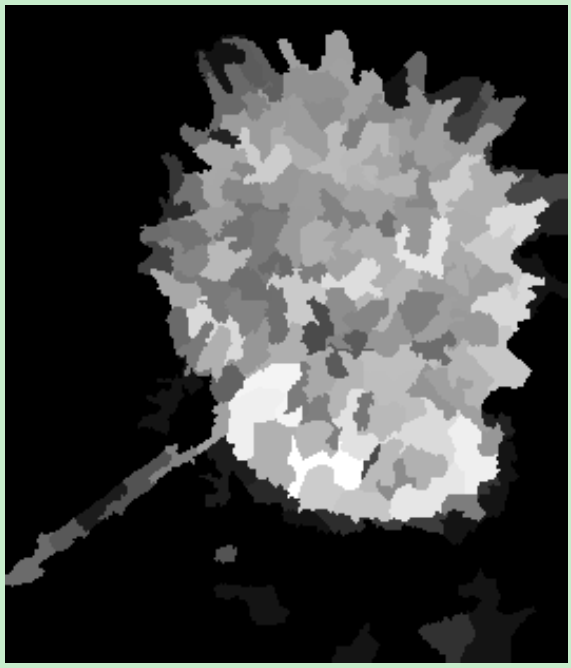

Figure 5: Initial saliency map.

1.7 Step 5: Use priors to enhance the result

Input Initial saliency map,.

Output Final saliency map.

Implementation I use Center Prior to enhance my result. And the Center prior and final saliency map are as follow.

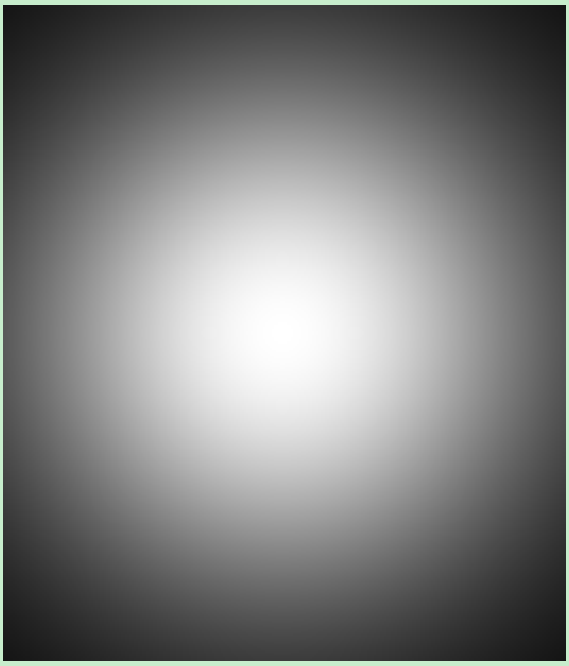

Figure 6: Center prior.

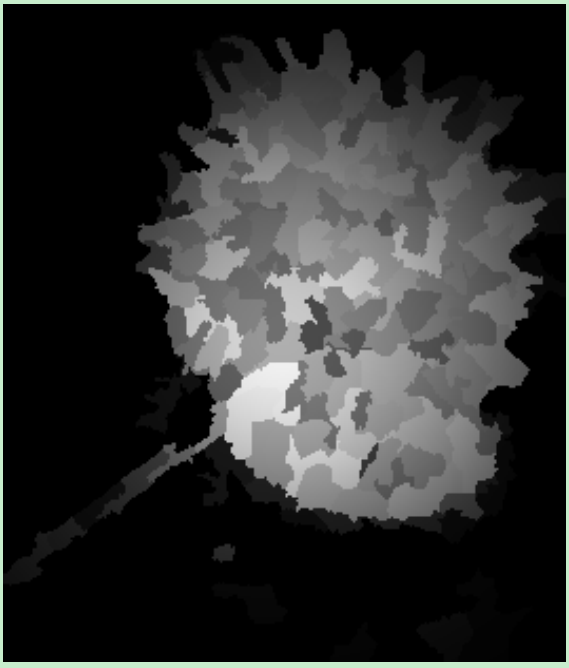

Figure 7: Final saliency map.

2. Result

From the assignment, I learned how to process images with OpenCV and how to use python doing some computer vision works.