CV2015Spring—Assignment #1

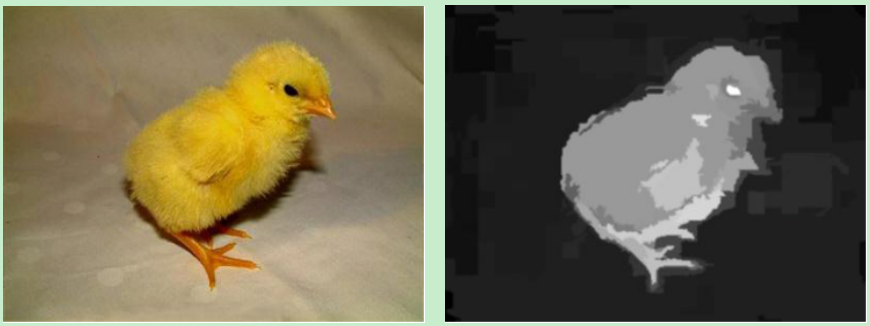
Due: Thursday, Apr 9 10:00 AM

Zhu Yafei

March 26, 2015

1. Assignment requirement

For this assignment, you will implement a version of the salient object detection technique. See Figure 1 for an example.



(a) (b)

Figure 1: Salient object detection.

Your method must be region-based, and at least one feature and one prior should be used. I will give you some tips for the implementation in the following sections.

2. Tips

The whole framework of the implementation for salient object detection is shown in Figure 2.

Next, I will introduce the implementation and requirement of each part of the framework for you.

2.1 Input image

A simple dataset can be downloaded from the website[1]. Pick one or several images from the dataset randomly for test.

[1] http://research.microsoft.com/en-us/um/people/jiansun/SalientObject/salient_object.htm

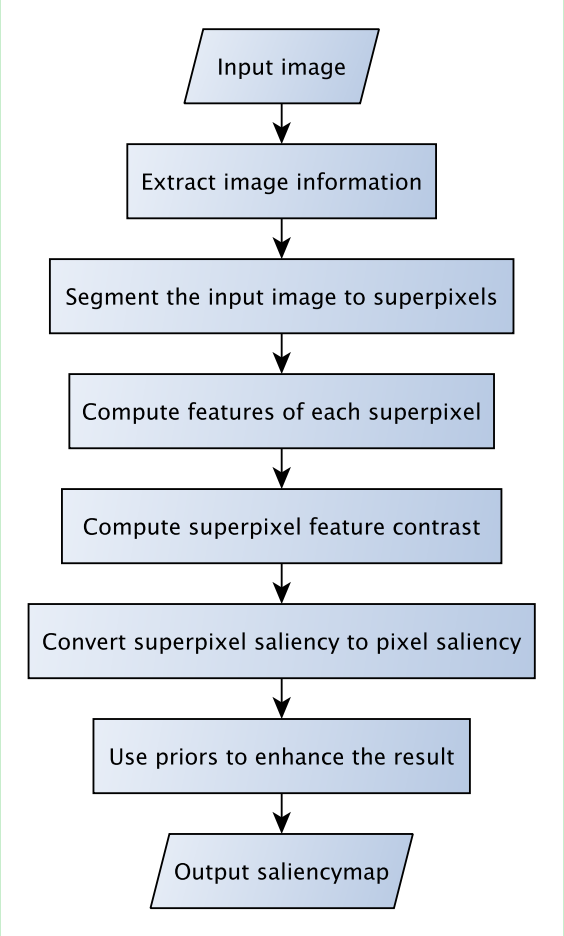


Figure 2: Framework of the implementation for salient object detection.

2.2 Step 1: Extract image information (15 points)

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

Output A $m \times n$ matrix.

Implementation This step is based on the regional feature you choose in step 3, you have two choices:

- Color information
  - You should quantify each color channel to have less number of values, then compute a value instead of a vector to represent the color of each pixel.
- Texture information
  - You should compute the LBP value of each pixel here.

Hint The extraction of color/texture information can refer to the matlab code from the website[2], which also contains the computation of color histogram and texture histogram of superpixels in step 3 and the conversion of superpixel saliency to pixel saliency in step 5.

2.3 Step 2: Segment the input image to superpixels (20 points)

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

Output Superpixel segmentation matrix (a $m \times n$ 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. For example, if you segment an image to n superpixels, the value of pixels in the result matrix should range from 0 to $n - 1$.

Hint There are a lot of superpixel segmentation algorithms, I recommend you to use SLIC[3], which is fast and easy to implement.

2.4 Step 3: Compute features of each superpixel (15 points)

Input Image information matrix, superpixel segmentation matrix.

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

Implementation you have two choices:

- Compute color histogram (RGB histogram/Lab histogram) feature of each superpixel
- Compute texture histogram feature of each superpixel

Hint Useful function in matlab for histogram computation: *hist*.

2.5 Step 4: Compute superpixel feature contrast (20 points)

Input h histograms.

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

[2] http://jianghz.com/projects/saliency_drfi/index.html
[3] http://www.vlfeat.org

Instructions Saliency can be defined as uniqueness in terms of local or global regional contrast. For simplicity, I recommend you to compute global regional contrast, which means that the saliency of a superpixel should be computed as its contrast of all the other superpixels in the image.

Theory The formulation for histogram distance is as follows:

$$\chi^2(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=1}^{b} \frac{2(h_{1i} - h_{2i})^2}{h_{1i} + h_{2i}} \tag{1}$$

where $\mathbf{h}_1$ and $\mathbf{h}_2$ are color histograms of two distinct regions, h_{1i} and h_{2i} are the ith component of $\mathbf{h}_1$ and $\mathbf{h}_2$ respectively, b is the number of histogram bins. Moreover, both histograms are normalized, i.e. their entries sum up to one.

2.6 Step 5: Convert superpixel saliency to pixel saliency (15 points)

Input h histograms, superpixel segmentation matrix.

Output An initial saliency map (a grayscale map).

Implementation Assign all the pixels of the same superpixel the same saliency value.

2.7 Step 6: Use priors to enhance the result (15 points)

Input Initial saliency map,.

Output Final saliency map.

Implementation Here you have two choices:

- Center prior
- Color prior

Multiply the initial saliency map with the prior you choose, you can obtain the final saliency map.

Hint You can download the source code from this website[4] for reference.

[4] http://sse.tongji.edu.cn/linzhang/va/SDSP/SDSP.htm

3. Submission instructions

3.1 What to hand in?

- Your matlab code (output your medial result in your code)
- A report containing the following:
  - Your name at the top
  - A brief explanation of your implementation strategy (in English)

3.2 Where to hand in?

Submit to Piazza in form of a followup below my assignment note.