

# CV2015Spring—Assignment #1

Due: Apr 12, 2015 (10:00AM)

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.

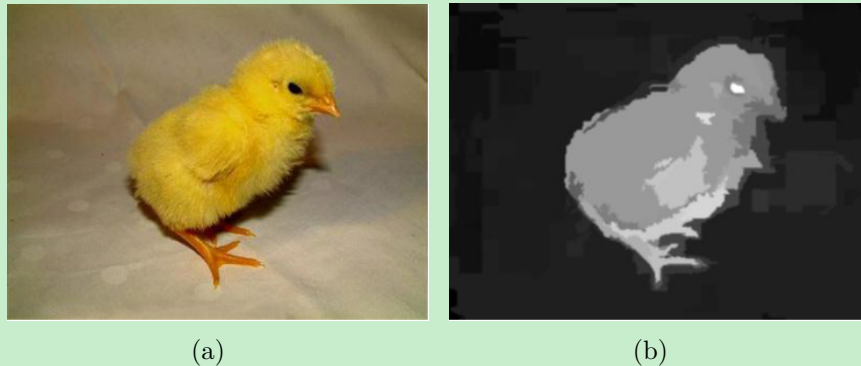


Figure 1: Salient object detection.

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, it may serve as a reference for your assignment. Next, I will introduce the implementation and requirement of each part of the framework for you.

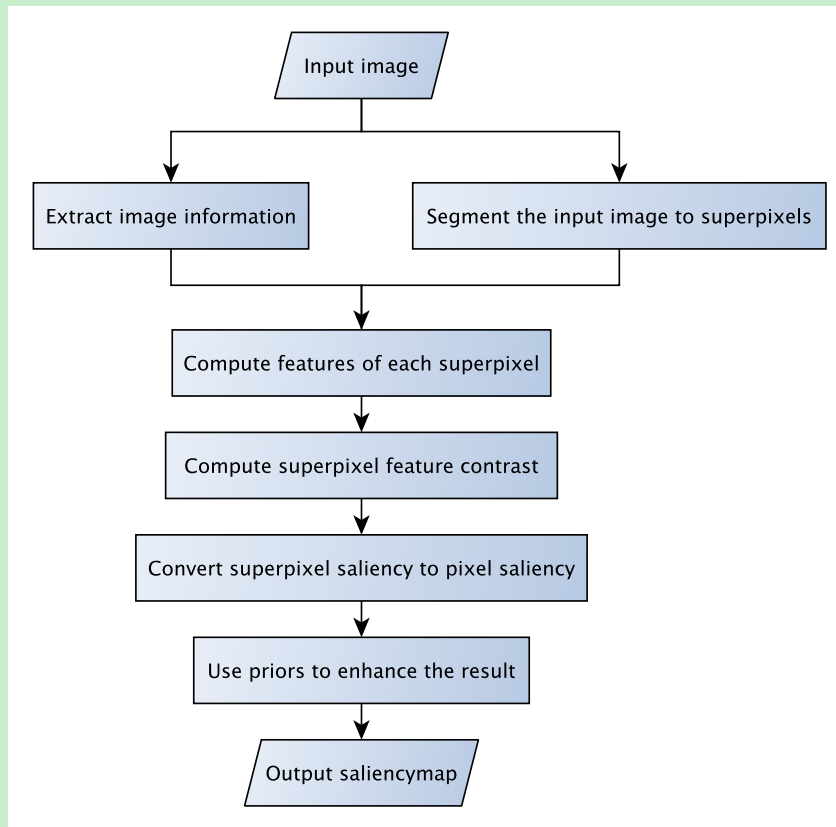


Figure 2: Framework of the implementation for salient object detection.

## 2.1 Input image

A simple dataset can be downloaded from the website<sup>1</sup>. Pick one or several images from the dataset randomly for test.

## 2.2 Step 1-1: Extract image information (15 points)

**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** This step determines the regional feature you want to use in step 2, and you have two choices (choose at least one feature for this assignment):

- Color information

<sup>1</sup>[http://research.microsoft.com/en-us/um/people/jiansun/SalientObject/salient\\_object.htm](http://research.microsoft.com/en-us/um/people/jiansun/SalientObject/salient_object.htm)

- You should quantize each color channel (RGB) to reduce the number of colors (such as from 256 to 16), then the number of color is reduced to  $16 \times 16 \times 16 = 4096$ . In order to create a bin for each color, you can use a number ( $1 \sim 4096$ ) instead of (R, G, B) values to represent a color uniquely.
- 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<sup>2</sup>, which also contains the computation of color histogram and texture histogram of superpixels in step 2 and the conversion of superpixel saliency to pixel saliency in step 4.

## 2.3 Step 1-2: Segment the input image to superpixels (20 points)

**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.

**Hint** There are a lot of superpixel segmentation algorithms, I recommend you to use SLIC<sup>3</sup>, which is fast and easy to implement.

## 2.4 Step 2: 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** The feature you choose here is based on the information you extracted in step 1-1:

- Compute color histogram feature of each superpixel, which means counting number of pixels for each color and store it in histogram's bins.
- Compute texture histogram feature of each superpixel

---

<sup>2</sup>[http://jianghz.com/projects/saliency\\_drfi/index.html](http://jianghz.com/projects/saliency_drfi/index.html)

<sup>3</sup><http://www.vlfeat.org>

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

## 2.5 Step 3: Compute superpixel feature contrast (20 points)

**Input**  $h$  histograms.

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

**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 feature contrast of all the other superpixels in the image.

**Theory** The formulation for histogram distance is as follows:

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

where  $\mathbf{h}_1$  and  $\mathbf{h}_2$  are color histograms of two distinct regions,  $h_{1i}$  and  $h_{2i}$  are the  $i$ th 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 4: Convert superpixel saliency to pixel saliency (15 points)

**Input**  $h$  values, superpixel segmentation matrix.

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

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

## 2.7 Step 5: Use priors to enhance the result (15 points)

**Input** Initial saliency map,.

**Output** Final saliency map.

**Implementation** Here you have two choices (choose at least one prior for this assignment):

- 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<sup>4</sup> for reference.

## 3. Submission instructions

### 3.1 What to hand in?

- Your matlab code (Show 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.

---

<sup>4</sup><http://sse.tongji.edu.cn/linzhang/va/SDSP/SDSP.htm>