Signal Processing: Image Communication 64 (2018) 21–32

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Coarse-to-fine salient object detection based on deep convolutional neural networks

Ying Li [a,b,*], Fan Cui [a], Xizhe Xue [a], Jonathan Cheung-Wai Chan [c]

[a] *School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China*
[b] *National Key Laboratory of Science and Technology on Space Microwave, Xi'an 710000, China*
[c] *Department of Electronics and Informatics, Vrije Universiteit Brussel, 1050 Brussels, Belgium*

ARTICLE INFO

Keywords:
Saliency detection
Convolutional neural network
Fully convolutional neural network

ABSTRACT

With explosive growth of image data, automatic image interpretation becomes more and more important. Saliency detection is one of the fundamental problems. To predict the saliency map, traditional saliency detection approaches use handcrafted features, which are not robust for complex scene. Recently, convolutional neural network (CNN) have shown good performance in computer vision problems. In this paper, we propose a coarse-to-fine approach combining pixel-wise FCN with superpixel-based CNN for detecting salient objects with precise boundaries. Firstly, the fully convolutional network (FCN) model is used to produce a coarse saliency map. Instead of patch-based CNN taking in overlapping patches as samples, the FCN model adopts the pixel-wise structure which can predict the location of the salient objects from the global aspect. Then, superpixel clustering is presented to decompose the image into homogeneous superpixels. For each superpixel, the local superpixel-based CNN model is created to integrate the coarse saliency map with the original image information for refining the detected salient objects with precise boundaries. Experimental results on large benchmark databases demonstrate the proposed method perform well when tested against the state-of-the-art methods.

1. Introduction

"Saliency" is considered to represent an object or a pixel that is more conspicuous than its neighbors. Salient object detection aims to capture the regions that stand out in an image. It is one of the most studied topics in computer vision and has been applied in object recognition [1], image segmentation [2], image compression [3] and image retargeting [4], quality assessment [5].

In terms of algorithm strategy, saliency detection approaches can be categorized into two subclasses, one is bottom-up data-driven methods [6–19], and the other is top-down task-driven methods [20–25]. For most bottom-up methods, low-level features are employed to calculate the saliency value, such as color [6], contrast [7,8], and distribution [9]. Achanta et al. [6] proposed a frequency-tuned method, which compute color difference of each pixel to generate saliency map. In [7], saliency was considered as the local contrast. Itti et al. [8] proposed a saliency model that integrates center-surrounded contrast over multiple scales to achieve identification of salient pixels. Later in [9], Perrazzi et al. presented a contrast-based saliency filter, which measures saliency by the combination of color uniqueness and spatial distribution. Cheng et al. [10] considered the global region contrast with the entire image to achieve a better performance. Graph-based visual saliency detection method proposed by Harel et al. [11] computes the equilibrium distribution of a Markov chain on a fully connected graph to measure saliency. To keep the structure of the objects, some region-based methods were proposed. Those methods segment images into coherent regions to obtain proper spatial structure. Goferman et al. [12] used a patch-based approach to get global properties. Cheng et al. [13] combined a soft abstraction to decompose an image into large perceptually homogeneous elements to achieve efficient saliency detection. Additionally, boundary cue is used to improve the saliency detection performance. Boundary prior treats the image boundary regions as labeled background. In [15], saliency maps are related to the distance to the image boundary. In [16,17], foreground and background information are used to rank the saliency value. In [18], contrast, sparsity and objectness are combined with regional similarities to evaluate the region saliency. These methods have achieved an effective and efficient performance in the earlier MASR dataset [25], which only contains one-target and simple background images. However, bottom-up methods cannot have a robust utilization in unconstrained scene.

On the other hand, top-down methods are task-driven which learn a supervised classifier for salient object detection. In DRFI [19], 93

[*] Corresponding author at: School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China.
E-mail address: lybyp@nwpu.edu.cn (Y. Li).

https://doi.org/10.1016/j.image.2018.01.012
Received 24 August 2017; Received in revised form 27 January 2018; Accepted 31 January 2018

hand crafted features are extracted to classify each region. Xi et al. [20] propose a SVM based methods with a color information as the input. In [21], multiple kernel boosting and adaptive fusion is used for evaluating the saliency value. Huang et al. [22] learned to combine multiple methods. However, hand-crafted features are too simple to mimic the human visual system. Recently, CNN has shown powerful capacity in computer vision with robust high-level features for classification and recognition. It can capture semantic salient objects such as animals, pedestrian, car, etc.

Several CNN based salient object detection methods have been proposed such as MDF [23] and LEGS [24]. Those methods extract deep features by CNN from a local or global aspect, has shown high performances. However, the drawback is that those methods treat each superpixel or region independently; it cannot get semantic features from a global view and overlapping caused redundancy. FCN [26] was proposed to achieve pixel-wise semantic segmentation, which can overcome the drawback of the traditional CNN method. It only needs to be run on the input image once to produce a complete saliency map with the same pixel resolution as the input image. Several FCN based saliency detection methods [27–31] have been proposed. However, the problem is that the FCN structure only obtains prediction at coarse resolution with fuzzy object boundary. It is because the multiple convolutional and pooling layers blur the object boundary.

In this paper, we propose a coarse-to-fine salient object detection method which combines the advantages of FCN and superpixel-based CNN. In the first step, FCN is used for pixel-to-pixel coarse prediction. The aim is: (1) extract global semantic information, (2) overcome drawbacks of redundancy and pre-procession issues of the traditional CNN-based saliency detection methods. In the second step, a superpixel-based CNN is used to refine the coarse result, which can extract the local details to maintain the object boundary.

2. Related work

2.1. Convolutional neural network

Convolutional neural network (CNN) is widely used and has shown promising results in the field of computer vision for image classification [32] and object detection [33], and it has significantly improved the efficiency of these fields. Ordinary CNN mainly consists of three parts: the convolution layer, pooling layer (subsampling layer), and fully connected layer. The output of the fully connected layer can be regarded as a learnt feature, which will be used in classification or detection when connected to a classifier at the end. An example of convolutional network is shown in Fig. 1.

(1) Convolution layer

During the forward propagation of the convolution layer, the input feature maps x are convolved with learnable kernels k with bias b, and through the activation function f it forms the output feature map. This step can be formulated as:

$$x_j^l = f\left(\sum_{i\in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right). \tag{1}$$

According to the back-propagation algorithm, in order to calculate the sensitivity of the convolution layer l, we need to multiply the next layer's sensitivity map with the activation derivative map at layer l element-wise. While each convolution layer l is normally followed by a pooling layer $l+1$, the activation derivative map and the subsampling layer's sensitivity map are not necessarily the same size. In order to calculate the sensitivity at layer l efficiently, we upsample the subsampling layer's sensitivity map to make it the same size as the convolutional layer's map. Then the sensitivity can be calculated by using the following formula.

$$\delta_j^l = \beta_j^{l+1}\left(f'\left(u_j^l\right)\circ up\left(\delta_j^{l+1}\right)\right) \tag{2}$$

where $\circ$ denotes element-wise multiplication, $up(\cdot)$ denotes an upsample operation, β denotes a parameter decided by the subsample operation, u_j^l denotes the total weighted sum of inputs to map j in layer l. Once the convolution layer sensitivities have been obtained, the gradients for the kernel weight and bias can be computed to update the parameters.

(2) Pooling layer

Early convolutional network methods adopted a subsampling layer to down-sample the input maps which are used to form output feature maps through the activation function. Such a subsampling layer is complex to use. In recent years, many researchers have adopted pooling as a replacement. The pooling layer's effect is similar to the subsampling layer's, but since the pooling layer has no activation function, it is more convenient to use. The two main kinds of pooling operation are max pooling and average pooling. The details of which are shown below.

As shown in Fig. 2, the data is separated into adjacent and non-overlapping blocks, each block is made up of four elements. In average pooling, the mean value of each block is computed for the corresponding output position. Similarly, the maximum value of each block is chosen in max pooling.

(3) Fully connected layer

This part is the same as a traditional network, such as Back Propagation network. The output maps of the last convolution layer or pooling layer are arranged into vectors which act as the input of a fully connected layer. The output of the fully connected layer can be regarded as the learnt feature. The classification operation can be completed by connecting a classifier, such as Softmax. Note that in order to complete the training of CNN, connecting a classifier is necessary. Depending on the choice of classifier, the sensitivity of the last layer will be different. However, the results of the convolution layer and pooling layer will remain constant.

2.2. Region-based salient object detection

Region-based methods first segment an image into regions with intensity edges and features are then extracted to describe those regions. When calculating the saliency map, region-based saliency prediction is mapped to each pixel. Those methods yield accurate object boundary. We describe those methods from two aspects below.

Region segmentation. Some methods adopt blocks or slide window [34] where the image is segmented into rectangles. Apparently, it is difficult to maintain the irregular boundary of objects. Graph-based image segmentation treats pixels as nodes in a graph, with the edge weights representing similarities. The segmentation is done by merging similar pixels. However, this approach is too slow for fast salient object detection. To balance the speed and good segmentation, superpixel approach is widely used to generate homogeneous regions [8,9,13]. Its goal is to obtain a coherent clustering of pixels, which is also named as 'over segmentation'. SLIC [35] is the most popular superpixel segmentation method. It randomly selects cluster centers on a regular grid, then assigns each pixel into the nearest cluster by $labxy$ color-image plane space. The $labxy$ space combines CIELAB color space $[l,a,b]$ with the position $[x,y]$. The distance of pixel i,j is defined as $labxy$ space Euclidean distance D_{labxy}:

$$\begin{aligned} D_{color} &= \sqrt{(l_i-l_j)^2+(a_i-a_j)^2+(b_i-b_j)^2} \\ D_{spatial} &= \sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \\ D_{labxy} &= \sqrt{D_{color}{}^2+\left(\frac{D_{spatial}}{\lambda}\right)^2}. \end{aligned} \tag{3}$$

When each pixel is assigned to the nearest cluster, the cluster center needs to be updated to be the mean $labxy$ space value. These steps are repeated until the cluster centers do not change.

Region feature representation. The second step of the region-based salient object detection methods is extracting features from regions to evaluate the saliency value. In [9], the global region contrast approach