Saliency Detection with Recurrent Fully Convolutional Networks

Linzhao Wang[1], Lijun Wang[1], Huchuan Lu[1](✉), Pingping Zhang[1], and Xiang Ruan[2]

[1] School of Information and Communication Engineering,
Dalian University of Technology, Dalian, China
{linzertling,wlj,jssxzhpp}@mail.dlut.edu.cn, lhchuan@dlut.edu.cn
[2] TIWAKI Corporation, Iwaki, Japan
ruanxiang@gmail.com

Abstract. Deep networks have been proved to encode high level semantic features and delivered superior performance in saliency detection. In this paper, we go one step further by developing a new saliency model using recurrent fully convolutional networks (RFCNs). Compared with existing deep network based methods, the proposed network is able to incorporate saliency prior knowledge for more accurate inference. In addition, the recurrent architecture enables our method to automatically learn to refine the saliency map by correcting its previous errors. To train such a network with numerous parameters, we propose a pre-training strategy using semantic segmentation data, which simultaneously leverages the strong supervision of segmentation tasks for better training and enables the network to capture generic representations of objects for saliency detection. Through extensive experimental evaluations, we demonstrate that the proposed method compares favorably against state-of-the-art approaches, and that the proposed recurrent deep model as well as the pre-training method can significantly improve performance.

Keywords: Saliency detection · Recurrent fully convolutional network

1 Introduction

Saliency detection can be generally divided into two subcategories: salient object segmentation [12,16,38] and eye-fixation detection [7,26]. This paper mainly focus on salient object segmentation, which aims to highlight the most conspicuous and eye-attracting object regions in images. It has been used as a pre-processing step to facilitate a wide range of vision applications and received increasingly more interest from the community. Although much progress has been made, it is still a very challenging task to develop effective algorithms capable of handling real world adverse scenarios.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46493-0_50) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part IV, LNCS 9908, pp. 825–841, 2016.
DOI: 10.1007/978-3-319-46493-0_50

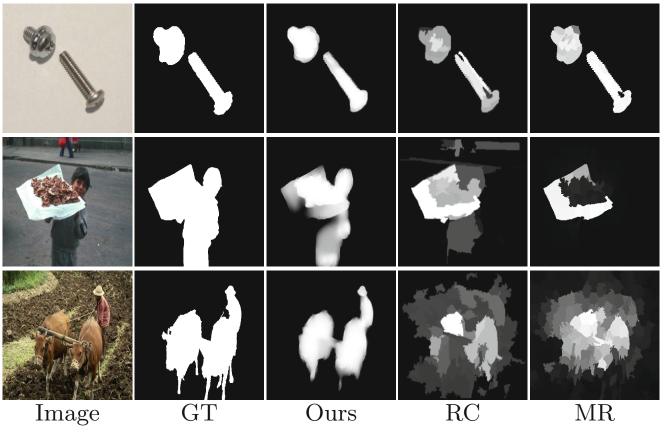


Fig. 1. Saliency detection results by different methods. From left to right: original image, groundtruth mask, our proposed RFCN, RC [2], MR [36]

Most existing methods address saliency detection with hand-crafted models and heuristic saliency priors. For instance, *contrast prior* formulates saliency detection as center-surrounding contrast analysis and captures salient regions either characterized by global rarity or locally standing out from their neighbors. In addition, *boundary prior* regards boundary regions as background and detects foreground objects by propagating background information to the rest image areas. Although these saliency priors have been proved to be effective in some cases (Fig. 1 first row), they are not robust enough to discover salient objects in complex scenes (Fig. 1 second row). Furthermore, saliency prior based methods mainly rely on low-level hand-crafted features which are incapable to capture the semantic concept of objects. As demonstrated in the third row of Fig. 1, high-level semantic information, in some cases, plays a central role in distinguishing foreground objects from background with similar appearance.

Recently, deep convolutional neural networks (CNNs) have delivered record breaking performance in many vision tasks, *e.g.* image classification [15,28], object detection [5,27], object tracking [32,33], semantic segmentation [21,22], *etc.* Existing methods suggest that deep CNNs can also benefit salinecy detection and are very effective to handle complex scenes by accurately identifying semantically salient objects (Fig. 1 third row). Though better performance has been achieved, there are still three major issues of prior CNN based saliency detection methods. Firstly, saliency priors, which are shown to be effective in previous work, are completely discarded by most CNN based methods. Secondly, CNNs predict the saliency label of a pixel only considering a limited size of local image patch. They mostly fail to enforce spatial consistency and may inevitably make incorrect predictions. However, with feed-forward architectures, CNNs can hardly refine the output predictions. Lastly, saliency detection are mainly formulated as binary classification problems, *i.e.*, either foreground or background. Compared with image classification tasks with thousands of categories, the supervision of binary labels is relatively weak to effectively train a deep CNN with a huge number of parameters.

To mitigate the above issues, we investigate recurrent fully convolutional networks (RFCNs) for saliency detection. In each time step, we feed forward both the input RGB image and a saliency prior map through the RFCN to obtain the predicted saliency map which in turn serves as the saliency prior map in the next time step. The prior map in the first time step is initialized by incorporating saliency priors indicative of potential salient regions. Our RFCN architecture has two advantages over existing CNN based methods: a) saliency priors are exploited to make training deep models more easier and yield more accurate prediction; b) in contrast to feed-forward networks, the output of our RFCN network is provided as the feedback signal, such that the RFCN is capable to refine the saliency prediction by correcting its previous mistakes until producing the final prediction in the last time step. To train the RFCN for saliency detection, a new pre-training strategy is developed, which leverage rich attribute information of semantic segmentation data for supervision. Figure 2 demonstrates the architecture overview of the proposed RFCN model.

In summary, the contributions of this work are three folds. Firstly, we propose a saliency detection method using recurrent fully convolutional network which is able to refine the previous predictions. Secondly, saliency priors are incorporated into the network to facilitate training and inference. Thirdly, we design a RFCN pre-training method for saliency detection using semantic segmentation data to both leverage strong supervison from multiple object categories and capture the intrinsic representation of generic objects. The proposed saliency detection method yields more accurate saliency maps and outperforms state-of-the-art approaches with a considerable margin on four benchmark data sets.

2 Related Work

Existing saliency detection methods can be mainly classified into two categories, *i.e.*, either hand-crafted models or learning based approaches. Most hand-crafted methods ca be traced back to the feature-integration theory [30], where important visual features are selected and combined to model visual attention. Later on, Itti *et al.* [8] propose to measure saliency by center-surround contrast of color, intensity and orientation features. Xie *et al.* [34] formulate saliency detection in a Bayesian framework and estimate visual saliency by a likelihood probability. In [3], a soft image abstraction is developed by considering both appearance similarity and spatial distribution of image pixels for saliency measurement. Meanwhile, background prior is also commonly used by many hand-crafted models [6,10,36,38], where the fundamental hypothesis is that image boundary regions are more likely to be background. Salient regions can then be recognized by label propagation using boundary regions as background seeds.

Hand-crafted saliency methods are efficient and effective, however they are not robust in handling complex scenarios. Recently, learning based methods have

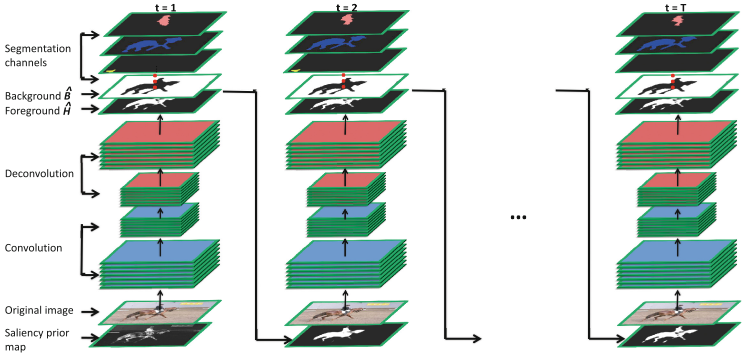


Fig. 2. Architecture overview of our RFCN model.

received more attention from the community. These methods can automatically learn to detect saliency by training detectors (*e.g.*, random forests [12,19], deep networks [17,31,37] *etc.*) on image data with annotations. Among others, deep networks based saliency models have shown very competitive performance. For instance, Wang *et al.* [31] propose to detect salient region by training a DNN-L and a DNN-G network for local estimation and global search, respectively. In [16], a fully connected network is trained to regress the saliency degree of each superpixel by taking multi-scale CNN features of the surrounding region. Both methods conduct patch-by-patch scanning in order to obtain the saliency map of the input image, which is very computational expensive. In addition, they directly train deep models on saliency detection data sets and ignore the problem of weak supervision from binary labels. To address the above issues, Li *et al.* [17] propose to detect saliency using a fully convolutional network (FCN) trained under a multi-task learning framework. Though bears a similar spirit, our method significantly differs from [17] in three aspects. Firstly, saliency priors are leveraged for network training and inference, which are ignored in [17]. Secondly, instead of using the feed-forward architecture in [17], we design a recurrent architecture capable of refining the generated predictions. Thirdly, our pre-training method for deep network allows to learn both class specific features and generic object representations using segmentation data. In contrast, [17] trains the network on segmentation data only for the task of distinguishing objects of different categories, which is essentially different from the task of salient object detection.

Recurrent neural networks (RNNs) have been applied to many vision tasks [20,25]. The recurrent architecture in our method mainly serves as a refinement mechanism to correct previous errors. Compared to existing RNNs that strongly rely on hidden units from last step, RFCN takes only the final output of last step as prior. Hence, it takes fewer steps to converge and is more easier to train.

3 Saliency Prediction by Recurrent Networks

A conventional CNN used for image classification consists of convolutional layers followed by fully connected layers, which takes an image of fixed spatial size as input and produces a label vector indicating the category of the input image. For tasks requiring spatial labels, like segmentation, depth prediction *etc.*, some methods apply CNNs for dense predictions in a patch-by-patch scanning manner. However, the overlap between patches leads to redundant computations and thus significantly increases computational overhead. Unlike existing methods, we consider the fully convolutional network (FCN) architecture [22] for our recurrent model, which generates predictions with the same size of the input image. In Sect. 3.1, we formally introduce FCN network for saliency detection. Section 3.2 presents our saliency methods based on RFCN network. Finally, we show how to train the RFCN network for saliency detection in Sect. 3.3.

3.1 Fully Convolutional Networks for Saliency Detection

Convolutional layers as building blocks of CNNs are defined on a translation invariance basis and have shared weights across different spatial locations. Both the input and the output of convolutional layers are 3D tensors called feature maps, where output feature map is obtained by convolving convolution kernels on the input feature map as

$$f_s(\boldsymbol{X}; \boldsymbol{W}, \boldsymbol{b}) = \boldsymbol{W} *_s \boldsymbol{X} + \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{X}$ is the input feature map; $\boldsymbol{W}$ and $\boldsymbol{b}$ denote kernel and bias, respectively; $*_s$ represents convolution operation with stride s. As a result, the resolution of the output feature map $f_s(\boldsymbol{X}; \boldsymbol{W}, \boldsymbol{b})$ is downsampled by a factor of s. Typically, convolutional layers are interleaved with max pooling layers and non-linear units (*e.g.*, ReLUs) to further improve translation invariance and representation capability. The output feature map of the last convolutional layer can then be fed into a stack of fully connected layers which discard the spatial coordinates of the input and generates a global label for the input image (See Fig. 3 (a)).

For efficient dense inference, [22] converts CNNs to fully convolutional networks (FCNs) (Fig. 3(b)) by casting fully connected layers into convolutional layers with kernels that cover their entire input regions. This allows the network to take input images of arbitrary sizes and generate spatial output by one forward pass. However, due to the stride of convolutional and pooling layers, the final output feature maps are still coarse and downsampled from the input image by a factor of the total stride of the network. To map the coarse feature map into a pixelwise prediction of the input image, FCN upsamples the coarse map via a stack of deconvolution layers (Fig. 3(c))

$$\hat{\boldsymbol{Y}} = U_S\left(F_S(\boldsymbol{I}; \boldsymbol{\theta}); \boldsymbol{\psi}\right), \tag{2}$$

where $\boldsymbol{I}$ is the input image; $F_S(\cdot; \boldsymbol{\theta})$ denotes the output feature map generated by the convolutional layers of FCN with total stride of S and parameterized

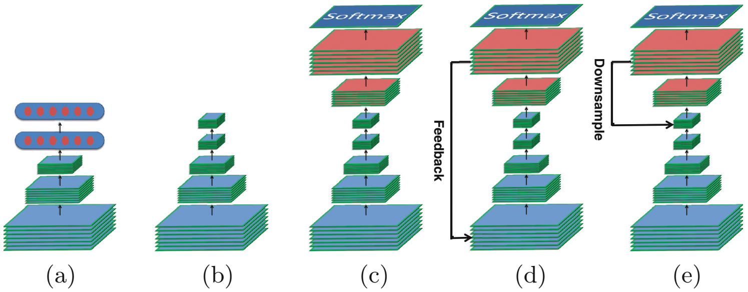


Fig. 3. Comparison of different deep models. (a) Convolution network. (b) Fully convolution network. (c) Fully convolution network with deconvolution layers. (d)(e) Recurrent fully convolution networks with different recurrent architectures.

by $\boldsymbol{\theta}$; $U_S(\cdot;\boldsymbol{\psi})$ denotes the deconvolution layers of FCN networks parameterized by $\boldsymbol{\psi}$ that upsamples the input by a factor of S to ensure the same spatial size of the output prediction $\hat{\boldsymbol{Y}}$ and the input image $\boldsymbol{I}$. Different from simple bilinear interpolation, the parameters $\boldsymbol{\psi}$ of deconvolution layers are jointly learned. To explore the fine-scaled local appearance of the input image, the skip architecture [22] can also be employed to combine output feature maps of both lower convolutional layers and the final convolutional layer for more accurate inference.

In the context of saliency detection, we are interested in measuring the saliency degree of each pixel in an image. To this end, the FCN takes the RGB image $\boldsymbol{I}$ of size $h \times w \times 3$ as input and generates the output feature map $\hat{\boldsymbol{Y}} = U_S\left(F_S(\boldsymbol{I};\boldsymbol{\theta});\boldsymbol{\psi}\right)$ of size $h \times w \times 2$. We denote the two output channels of $\hat{\boldsymbol{Y}}$ as background map $\hat{\boldsymbol{B}}$ and salient foreground map $\hat{\boldsymbol{H}}$, indicating the scores of all the pixels being background and foreground, respectively. By applying softmax function, these two scores are transformed into foreground probability as

$$p(l_{i,j} = fg|\boldsymbol{\theta},\boldsymbol{\psi}) = \frac{\exp(\hat{\boldsymbol{H}}_{i,j})}{\exp(\hat{\boldsymbol{H}}_{i,j}) + \exp(\hat{\boldsymbol{B}}_{i,j})}, \tag{3}$$

where $l_{i,j} \in \{fg, bg\}$ indicates the foreground/background label of the pixel indexed by (i,j). The background probability $p(l_{i,j} = bg|\boldsymbol{\theta},\boldsymbol{\psi})$ can be computed in a similar way. Given the training set $\{\boldsymbol{Z} = (\boldsymbol{I},\boldsymbol{C})\}_1^N$ containing both training image $\boldsymbol{I}$ and its pixelwise saliency annotation $\boldsymbol{C}$, the FCN network can be trained end-to-end for saliency detection by minimizing the following loss

$$\begin{aligned}\arg\min_{\boldsymbol{\theta},\boldsymbol{\psi}} -\sum_{\boldsymbol{Z}}\sum_{i,j} &\mathbf{1}(\boldsymbol{C}_{i,j} = fg)\ln p(l_{i,j} = fg|\boldsymbol{\theta},\boldsymbol{\psi}) \\ &+ \mathbf{1}(\boldsymbol{C}_{i,j} = bg)\ln p(l_{i,j} = bg|\boldsymbol{\theta},\boldsymbol{\psi}),\end{aligned} \tag{4}$$

where $\mathbf{1}(\cdot)$ is the indicator function. The network parameters $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ can then be iteratively updated using stochastic gradient descent (SGD) algorithm.

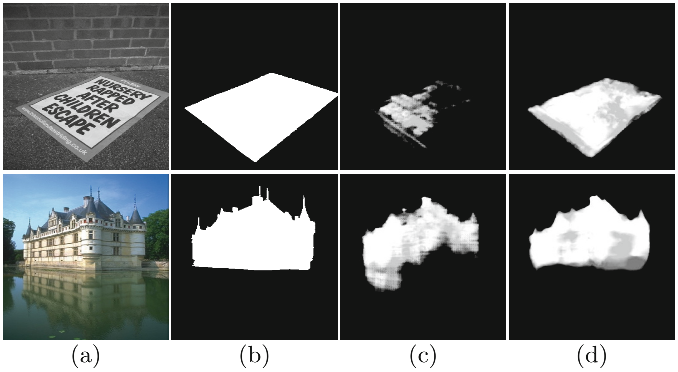


Fig. 4. Saliency maps generated by our model. (a) Original images. (b) Ground truth. (c)(d) Saliency maps without and with prior maps, respectively.

3.2 Recurrent Network for Saliency Detection

The above FCN network is trained to approximate the direct nonlinear mapping from raw pixels to saliency values and ignores the saliency priors which are widely used in existing methods. Although, heuristic saliency priors have their limitations, they are easy to compute and shown to be very effective under a variety of cases. Thus, we believe that leveraging saliency prior information can facilitate faster training and more accurate inference. This has been verified by our experiments. We also note that the output prediction by FCN may be very noisy and lack of label consistency. However, the feed forward architecture of FCN fails to consider feedback information, which makes it impossible to correct prediction errors. Based on these observations, we make two improvements over the FCN network and design the RFCN by: (i) incorporating saliency prior into both training and inference; and (ii) recurrently refining the output prediction (Fig. 4).

Saliency Prior Maps. We encode prior knowledge into a saliency prior map which serves as the input to the network. We first oversegment the input image into M superpixels, $\{s_i\}_1^M$. The color contrast prior for s_i is calculated by

$$\mathcal{G}(s_i) = \frac{1}{\varGamma_i} \sum_{j=1}^{M} \|\mu_{s_i} - \mu_{s_j}\|^2 \exp(-\frac{\|p_{s_i} - p_{s_j}\|^2}{2\delta^2}), \tag{5}$$

where μ and p denote the mean RGB value and the center position of a superpixel, respectively; $\varGamma_i$ is the normalization factor; and δ is a scale parameter (fixed to 0.5). The intensity contrast $\mathcal{I}(s_i)$ and orientation feature contrast $\mathcal{O}(s_i)$ can be computed in a similar way by replacing the color values in (5) with corresponding feature values. The saliency prior map $\boldsymbol{P}$ is obtained by

$$\boldsymbol{P}(s_i) = \mathcal{U}(s_i) \times (\mathcal{G}(s_i) + \mathcal{I}(s_i) + \mathcal{O}(s_i)), \tag{6}$$

where $\boldsymbol{P}(s_i)$ denotes the saliency prior value of superpixel s_i; and the central prior [11] $\mathcal{U}(s_i)$ penalizes0 the distance from superpixel s_i to the image center.

Recurrent Architecture. To incorporate the saliency prior maps into our approach, we consider two recurrent architectures for RFCN network. As in Sect. 3.1, we divide the network into two parts, *i.e.*, convolution part $F(\cdot, \boldsymbol{\theta})$ and deconvolution part $U(\cdot, \boldsymbol{\psi})$. Our first recurrent architecture (Fig. 3 (d)) incorporates the saliency prior map $\boldsymbol{P}$ into the convolution part by modifying the first convolution layer as

$$f(\boldsymbol{I}) = \boldsymbol{W_I} * \boldsymbol{I} + \boldsymbol{W_P} * \boldsymbol{P} + \boldsymbol{b}, \tag{7}$$

where $\boldsymbol{I}$ and $\boldsymbol{P}$ denote input image and saliency prior,respectively; $\boldsymbol{W_I}$ and $\boldsymbol{W_P}$ represent corresponding convolution kernels; $\boldsymbol{b}$ is bias parameter. In the first time step, the RFCN network takes the input image and saliency prior map as input and produces the final feature map $\hat{\boldsymbol{Y}}^1 = U\left(F(\boldsymbol{I}, \boldsymbol{P}; \boldsymbol{\theta}); \boldsymbol{\psi}\right)$ comprising both foreground map $\hat{\boldsymbol{H}}^1$ and background map $\hat{\boldsymbol{B}}^1$. In the following each time step, the foreground map $\hat{\boldsymbol{H}}^{t-1}$ generated in the last time step is fed back as saliency prior map to the input. The RFCN then refine the saliency prediction by considering both the input image and the last prediction as

$$\hat{\mathbf{Y}}^t = U\left(F(\boldsymbol{I}, \hat{\boldsymbol{H}}^{t-1}; \boldsymbol{\theta}); \boldsymbol{\psi}\right). \tag{8}$$

For the above recurrent architecture, forward propagation of the whole network is conducted in every time step, which is very expensive in terms of both computation and memory. An alternative recurrent architecture is to incorporate the saliency prior maps into the deconvolution part ((Figure 3 (e))). Specifically, in the first time step, we feed the input image $\boldsymbol{I}$ into the convolution part to obtain the convolution feature map $F(\boldsymbol{I}; \boldsymbol{\theta})$. The deconvolution part then takes the convolution feature map as well as saliency prior map $\boldsymbol{P}$ as input to infer the saliency prediction $\hat{\boldsymbol{Y}}^1 = U\left(F(\boldsymbol{I}; \boldsymbol{\theta}), \boldsymbol{P}; \boldsymbol{\psi}\right)$. In the t-th time step, the predicted foreground map $\hat{\boldsymbol{H}}^{t-1}$ in the last time step serves as saliency prior map. The deconvolution part takes the convolution feature map $F(\boldsymbol{I}; \boldsymbol{\theta})$ as well as the foreground map $\hat{\boldsymbol{H}}^{t-1}$ to refine the saliency prediction $\hat{\boldsymbol{Y}}^t$:

$$\hat{\boldsymbol{Y}}^t = U\left(F(\boldsymbol{I}; \boldsymbol{\theta}), \hat{\boldsymbol{H}}^{t-1}; \boldsymbol{\psi}\right). \tag{9}$$

Note that, for each input image, forward propagation of deconvolution part is repeatedly conducted in each time step, whereas the convolution part is only required to be fed forward once in the first time step. Since the deconvolution part has approximately 10 times fewer parameters than the convolution part, this recurrent architecture can effectively reduce computational complexity and save memory. However, we find in our preliminary experiments that the second recurrent architecture can only achieve similar performance compared to the FCN based approach (*i.e.*, without recurrent). This may be attributed to the fact

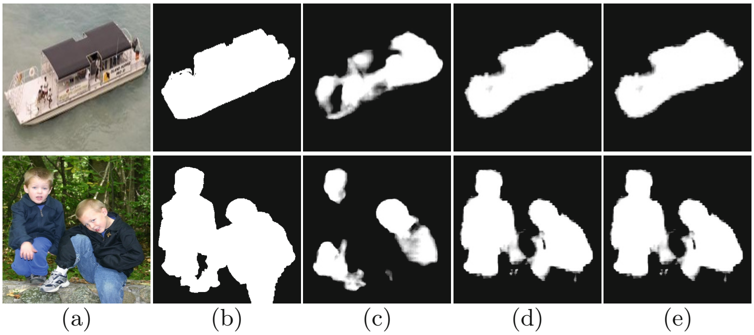


Fig. 5. Saliency maps predicted by the proposed RFCN in different time steps. (a) Original images. (b) Ground truth. (c)–(e) Saliency maps predicted by RFCN in the 1st–3rd time step, respectively.

that the prior saliency map is severely downsampled to the same spatial size of the last convolution feature map $F(\boldsymbol{I};\boldsymbol{\theta})$ (downsampled by a factor of 1/32 from the input). With less prior information, the downsampled prior saliency map can hardly facilitate network inference. Therefore, we adopt the first recurrent architecture in this work. In our experiments, we observe that the accuracy of the saliency maps almost converges after the second time step (Compare Fig. 5(a) and (e)). Therefore, we set the total time step of the RFCN to $T = 2$.

3.3 Training RFCN for Saliency Detection

Our RFCN training approach consists of two stages: pre-training and fine-tuning. Pre-training is conducted on the PASCAL VOC 2010 semantic segmentation data set. Saliency detection and semantic segmentation are highly correlated but essentially different in that saliency detection aims at separating generic salient objects from background, whereas semantic segmentation focuses on distinguishing objects of different categories. Our pre-training approach enjoys strong supervision from segmentation data and also enables the network to learn general representation of foreground objects. Specifically, for each training pair $\boldsymbol{Z} = (\boldsymbol{I}, \boldsymbol{S})$ containing image $\boldsymbol{I}$ and pixelwise semantic annotation $\boldsymbol{S}$, we generate an object map $\boldsymbol{G}$ to label each pixel as either foreground (fg) or background (bg) as follow

$$\boldsymbol{G}_{i,j} = \begin{cases} bg & \text{if } \boldsymbol{S}_{i,j} = 0 \\ fg & \text{otherwise} \end{cases}, \tag{10}$$

where $\boldsymbol{S}_{i,j} \in \{0, 1, \ldots, C\}$ denotes the semantic class label of pixel (i,j), and $\boldsymbol{S}_{i,j} = 0$ indicates the pixel belonging to background. In the pre-training stage, the final feature map $\hat{\boldsymbol{Y}}^t$ (Sect. 3.1) generated by the RFCN consists of $C + 3$ channels, where the first $C + 1$ channels correspond to the class scores for semantic segmentation and the last 2 channels, *i.e.*, $\hat{\boldsymbol{H}}^t$ and $\hat{\boldsymbol{B}}^t$ (Sect. 3.1), denotes

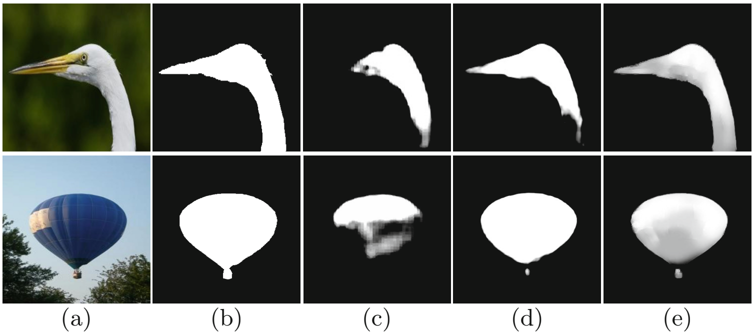


Fig. 6. Saliency detection results on different stages. (a) Original images. (b) ground truth. (c) results of pre-trained RFCN. (d) results of fine-tuned RFCN. (e) result after post-processing.

the foreground/background scores. By applying softmax function, we obtain the conditional probability $p(c_{i,j}|\boldsymbol{I}, \hat{\boldsymbol{H}}^{t-1}, \boldsymbol{\theta}, \boldsymbol{\psi})$ and $p(l_{i,j}|\boldsymbol{I}, \hat{\boldsymbol{H}}^{t-1}, \boldsymbol{\theta}, \boldsymbol{\psi})$ predicted by the RFCN for segmentation and foreground detection, respectively. The loss function for pre-training across all time steps is defined as

$$L(\boldsymbol{\theta}, \boldsymbol{\psi}) = -\sum_{t=1}^{T} \sum_{\boldsymbol{Z}} \sum_{i,j} \ln p(c_{i,j} = \boldsymbol{S}_{i,j}|\boldsymbol{I}, \hat{\boldsymbol{H}}^{t-1}, \boldsymbol{\theta}, \boldsymbol{\psi}) + \ln p(l_{i,j} = \boldsymbol{G}_{i,j}|\boldsymbol{I}, \hat{\boldsymbol{H}}^{t-1}, \boldsymbol{\theta}, \boldsymbol{\psi}), \quad (11)$$

where T is the total time step and $\hat{\boldsymbol{H}}^{0}$ is initialized by the saliency prior map $\boldsymbol{P}$ (Sect. 3.2). Pre-training is conducted via back propagation through time.

After pre-training, we modify the RFCN network architecture by removing the first $C+1$ channels of the last feature map and only maintaining the last two channels, *i.e.*, the predicted foreground and background maps. Finally, we fine-tune the RFCN network on the saliency detection data set as described in Sect. 3.2. As demonstrated in Fig. 6(c), the pre-trained model, supervised by semantic labels of multiple object categories, captures generic object features and can already discriminate foreground objects (of unseen categories in pre-training) from background. Fine-tuning on the saliency data set can further improve the performance of the RFCN network (Fig. 6(d)).

3.4 Post-processing

The trained RFCN network is able to accurately identify salient objects. To more precisely delineate the compact and boundary-preserving object regions, we adopt an efficient post-processing approach. Given the final saliency score map $\hat{\boldsymbol{H}}^{T}$ predicted by the RFCN, we first segment the image into foreground and background regions by thresholding $\hat{\boldsymbol{H}}^{T}$ with its mean saliency score. A spatial confidence $\boldsymbol{SC}_{i,j}$ and a color confidence $\boldsymbol{CC}_{i,j}$ are computed for each pixel (i, j). The spatial confidence is defined considering the spatial distance of the pixel to the center of the foreground region

$$\boldsymbol{SC}_{i,j} = \exp(-\frac{\|loc_{i,j} - loc_s\|_2}{\sigma}), \tag{12}$$

where $loc_{i,j}$ and loc_s denote the coordinates the pixel (i, j) and the center of foreground, respectively; σ is a scale parameter. The color confidence is defined to measure the similarity of the pixel to foreground region in RGB color space

$$\boldsymbol{CC}_{i,j} = \frac{N_{i,j}}{N_s}, \tag{13}$$

where $N_{i,j}$ is the number of foreground pixels that have the same color feature with pixel (i, j) and N_s is the total number of foreground pixels.

We then weight the predicted saliency scores by spatial and color confidences to dilate the foreground region

$$\tilde{\boldsymbol{H}}_{i,j} = \boldsymbol{SC}_{i,j} \times \boldsymbol{CC}_{i,j} \times \hat{\boldsymbol{H}}^T. \tag{14}$$

After an edge-aware erosion procedure [4] on the dilated saliency score map $\tilde{\boldsymbol{H}}$, we obtain the final saliency map. As demonstrated in Fig. 6 (e), the post-processing step can improve the detection precision to a certain degree.

4 Experiments

4.1 Experimental Setup

Detailed architecture of the proposed RFCN can be found in the supplementary materials[1]. We pre-train the RFCN on the PASCAL VOC 2010 semantic segmentation data set with 10103 training images belonging to 20 object classes. The pre-training is converged after 200k iterations of SGD. We then fine-tune the pre-trained model for saliency detection on the THUS10K [2] data set for 100k iterations. In the test stage, we apply the trained RFCN in three different scales and fuse all the results into the final saliency maps [12]. Our method is implemented in MATLAB with the Caffe [9] wrapper and runs at 4.6 s per image on a PC with a 3.4 GHz CPU and a TITANX GPU. The source code will be released (see footnote 1).

We evaluate the proposed algorithm (RFCN) on five benchmark data sets: SOD [24], ECSSD [35], PASCAL-S [19], SED1 [1], and SED2 [1]. The evaluation result on SED2 and additional analysis on the impact of recurrent time step are included in the supplementary materials. Three metrics are utilized to measure the performance, including precision-recall (PR) curves, F-measure and area under ROC curve (AUC). The precision and recall are computed by thresholding

[1] http://ice.dlut.edu.cn/lu/index.html.

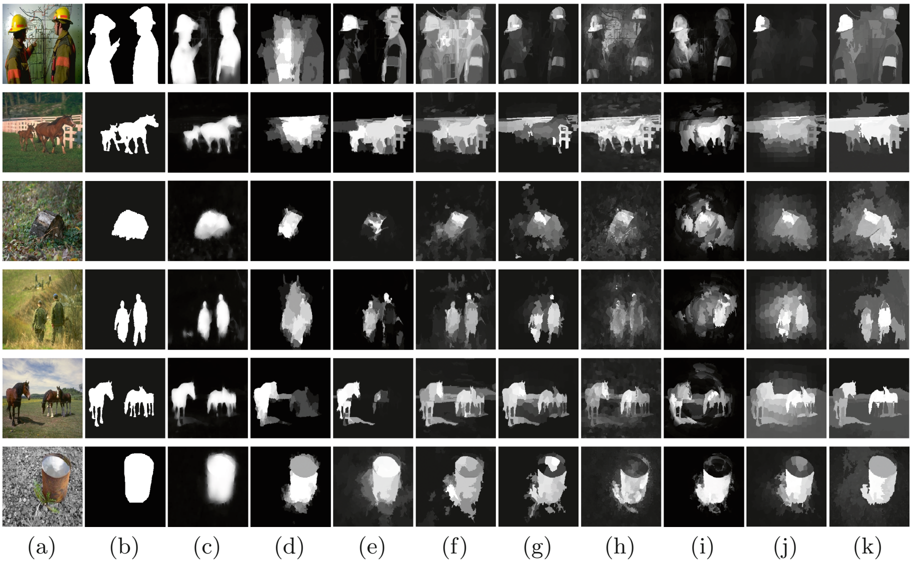
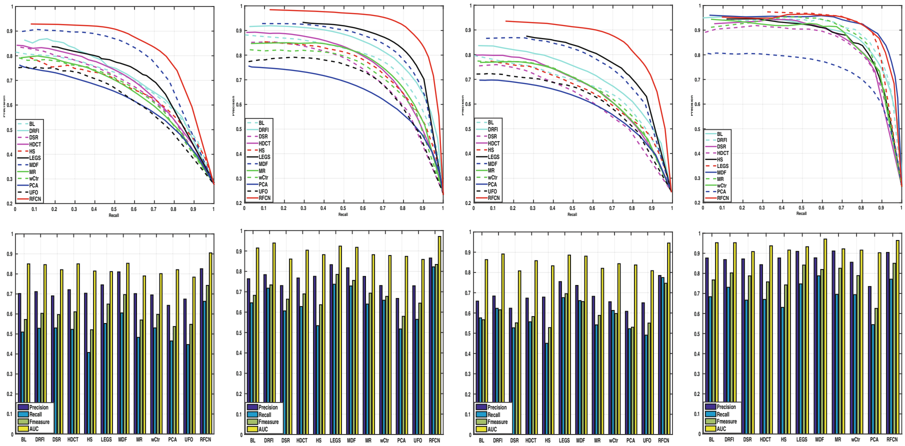


Fig. 7. Comparisons of saliency maps. Top, middle and bottom two rows are images from the SOD, ECSSD, PASCAL-S and SED1 data sets, respectively.(a) Original images, (b) ground truth, (c) our RFCN method, (d) LEGS, (e) MDF, (f) DRFI, (g) wCtr, (h) HDCT, (i) DSR, (j) MR, (k) HS.



(a) SOD (b) ECSSD (c) PASCAL-S (d) SED1

Fig. 8. Performance of the proposed algorithm compared with other state-of-the-art methods on the SOD, ECSSD, PASCAL-S and SED1 databases, respectively.

Table 1. F-measure and AUC (Area Under ROC Curve) on the SOD, ECSSD, PASCAL-S and SED1 data sets. The best two results are shown in italic and bold fonts respectively. The proposed methods rank first and second on the four data sets.

*	SOD		ECSSD		PASCAL-S		SED1	
	F-measure	AUC	F-measure	AUC	F-measure	AUC	F-measure	AUC
RFCN	0.7426	0.9053	0.8340	0.9714	0.7468	0.9453	0.8502	0.9640
MTDS	0.6978	0.9233	0.7589	0.9009	0.7310	0.9287	-	-
LEGS	0.6492	0.8117	0.7887	0.9230	0.6951	0.8857	0.8414	0.9328
MDF	0.6966	0.8532	0.7557	0.9180	0.6562	0.8806	0.8194	0.9710
BL	0.5723	0.8503	0.6825	0.9147	0.5668	0.8633	0.7675	0.9528
DRFI	0.6031	0.8464	0.7337	0.9391	0.6159	0.8913	0.8024	0.9528
wCtr	0.5978	0.8014	0.6774	0.8779	0.5972	0.8433	0.7889	0.9159
DSR	0.5968	0.8210	0.6636	0.8604	0.5513	0.8079	0.7877	0.9086
MR	0.5697	0.7899	0.6932	0.8820	0.5881	0.8205	0.8255	0.9223
HS	0.5210	0.8145	0.6363	0.8821	0.5278	0.8330	0.7426	0.9161
PCA	0.5370	0.8212	0.5796	0.8737	0.5298	0.8371	0.6256	0.9030
UFO	0.5480	0.7840	0.6442	0.8587	0.5502	0.8088	-	-

the saliency map, and comparing the binary map with the ground truth. The PR curves demonstrate the mean precision and recall of saliency maps at different thresholds. The F-measure can be calculated by $F_\beta = \frac{(1+\beta^2)Precision\times Recall}{\beta^2\times Precision+Recall}$, where *Precision* and *Recall* are obtained using twice the mean saliency value of saliency maps as the threshold, and set $\beta^2 = 0.3$.

4.2 Performance Comparison with State-of-the-art

We compare the proposed algorithm (RFCN) with twelve state-of-the-art methods, including MTDS [17], LEGS [31], MDF [16], BL [29], DRFI [12], UFO [13], PCA [23], HS [35], wCtr [38], MR [36], DSR [18] and HDCT [14]. We use either the implementations or the saliency maps provided by the authors for fair comparison. Note that MTDS, LEGS and MDF are deep learning based methods. Among others, MTDS exploits fully convolution network for saliency detection and leverages segmentation data for multi-task training. As demonstrated in Fig. 8 and Table 1, the proposed RFCN method can consistently outperform existing methods across almost all the data sets with a considerable margin in terms of PR curves, F-measure as well as AUC scores. Compared with other deep learning based methods, the three contributions of our method (*i.e.*, integration of saliency priors, recurrent architecture and pre-training approach) ensures more accurate saliency detection. Figure 7 shows that our saliency maps can reliably highlight the salient objects in various challenging scenarios.

Table 2. Different settings of variants of the proposed RFCN method. “Pre-train” denotes that the network is pre-trained on segmentation data only for semantic segmentation task. “Pre-train$_+$” indicates the proposed pre-training method using segmentation data for both semantic segmentation and foreground background separation.

Settings	FCN	FCN$_p$	RFCN-A	RFCN-B	RFCN
Prior map		√	√	√	√
Feed-forward	√	√			
Recurrent			√	√	√
Pre-train	√	√	√		
Pre-train$_+$				√	√
Post-processing					√

Table 3. Comparison of the proposed approach. The best results are shown in bold fonts.

*	SOD		ECSSD		PASCAL-S		SED1	
	F-measure	AUC	F-measure	AUC	F-measure	AUC	F-measure	AUC
FCN	0.6985	0.7810	0.8116	0.8864	0.7179	0.8387	0.7924	0.8452
FCN$_p$	0.7248	0.8210	0.8252	0.9145	0.7315	0.8733	0.8335	0.8823
RFCN-A	0.7276	0.8213	0.8312	0.9119	0.7372	0.8784	0.8500	0.8967
RFCN-B	**0.7487**	0.8331	**0.8534**	0.9310	**0.7734**	0.9028	**0.8527**	0.9006
RFCN	0.7426	**0.9053**	0.8340	**0.9714**	0.7468	**0.9453**	0.8502	**0.9640**

4.3 Ablation Studies

To analyze the relative contributions of different components of our methods, we evaluate four variants of the proposed RFCN method with different settings as demonstrated in Table 2. The performance in terms of F-measure and AUC are reported in Table 3. The comparison between FCN and FCN$_p$ suggests that saliency priors ignored by existing deep learning based methods can indeed benefit network training and inference. The comparison between FCN$_p$ and RFCN-A indicates that the proposed recurrent architecture is capable of correcting previous errors and refining the output saliency maps. In addition, the RFCN-B method with the proposed pre-training strategy can significantly outperform the RFCN-A method simply pre-trained for segmentation, which verifies that our pre-training method can effectively leverage the strong supervision of segmentation and simultaneously enable the network to caputre generic feature representation of foreground objects. After the proposed post-processing step, our RFCN method achieves considerable improvements over RFCN-B in terms of AUC scores with a slight performance degrade in terms of F-measure.

5 Conclusions

In this paper, we propose a recurrent fully convolutional network based saliency detection methods. Heuristic saliency priors are incorporated into the network to facilitate training and inference. The recurrent architecture enables our method to refine saliency maps based on previous output and yield more accurate predictions. A pre-training strategy is also developed to exploit the strong supervision of segmentation data sets and explicitly enforce the network to learn generic feature representation for saliency detection. Extensive evaluations verify that the above three contributions can significantly improve performance of saliency detection. State-of-the-art performance has been achieved by the proposed method in five widely adopted data sets.

Acknowledgement. The work is supported by the National Natural Science Foundation of China under Grant 61528101 and Grant 61472060.

References

1. Alpert, S., Galun, M., Brandt, A., Basri, R.: Image segmentation by probabilistic bottom-up aggregation and cue integration. PAMI **34**(2), 315–327 (2012)
2. Cheng, M., Mitra, N.J., Huang, X., Torr, P.H., Hu, S.: Global contrast based salient region detection. PAMI **37**(3), 569–582 (2015)
3. Cheng, M.M., Warrell, J., Lin, W.Y., Zheng, S., Vineet, V., Crook, N.: Efficient salient region detection with soft image abstraction. In: ICCV, pp. 1529–1536 (2013)
4. Gastal, E.S., Oliveira, M.M.: Domain transform for edge-aware image and video processing. ACM Trans. Graph. (TOG) **30**, 69 (2011)
5. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR, pp. 580–587 (2014)
6. Han, J., Zhang, D., Hu, X., Guo, L., Ren, J., Wu, F.: Background prior-based salient object detection via deep reconstruction residual. CSVT **25**(8), 1309–1321 (2015)
7. Huang, X., Shen, C., Boix, X., Zhao, Q.: Salicon: reducing the semantic gap in saliency prediction by adapting deep neural networks. In: ICCV (2015)
8. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. PAMI **11**, 1254–1259 (1998)
9. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, pp. 675–678 (2014)
10. Jiang, B., Zhang, L., Lu, H., Yang, C., Yang, M.H.: Saliency detection via absorbing markov chain. In: CVPR, pp. 1665–1672 (2013)
11. Jiang, H., Wang, J., Yuan, Z., Liu, T., Zheng, N., Li, S.: Automatic salient object segmentation based on context and shape prior. In: BMVC, vol. 6, p. 9 (2011)
12. Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., Li, S.: Salient object detection: a discriminative regional feature integration approach. In: CVPR, pp. 2083–2090 (2013)

13. Jiang, P., Ling, H., Yu, J., Peng, J.: Salient region detection by ufo: uniqueness, focusness and objectness. In: ICCV, pp. 1976–1983 (2013)
14. Kim, J., Han, D., Tai, Y.W., Kim, J.: Salient region detection via high-dimensional color transform. In: CVPR, pp. 883–890 (2014)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
16. Li, G., Yu, Y.: Visual saliency based on multiscale deep features. In: CVPR, pp. 5455–5463 (2015)
17. Li, X., Zhao, L., Wei, L., Yang, M., Wu, F., Zhuang, Y., Ling, H., Wang, J.: Deepsaliency: multi-task deep neural network model for salient object detection. arXiv preprint arXiv:1510.05484 (2015)
18. Li, X., Lu, H., Zhang, L., Ruan, X., Yang, M.H.: Saliency detection via dense and sparse reconstruction. In: ICCV, pp. 2976–2983 (2013)
19. Li, Y., Hou, X., Koch, C., Rehg, J., Yuille, A.: The secrets of salient object segmentation. In: CVPR, pp. 280–287 (2014)
20. Liang, M., Hu, X.: Recurrent convolutional neural network for object recognition. In: Computer Vision and Pattern Recognition, pp. 3367–3375 (2015)
21. Liang-Chieh, C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR (2015)
22. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR, pp. 3431–3440 (2015)
23. Margolin, R., Tal, A., Zelnik-Manor, L.: What makes a patch distinct? In: CVPR, pp. 1139–1146 (2013)
24. Movahedi, V., Elder, J.H.: Design and perceptual validation of performance measures for salient object segmentation. In: CVPR, pp. 49–56 (2010)
25. Pinheiro, P.H., Collobert, R.: Recurrent convolutional neural networks for scene labeling. In: ICML, pp. 82–90 (2014)
26. Ramanathan, S., Katti, H., Sebe, N., Kankanhalli, M., Chua, T.-S.: An eye fixation database for saliency detection in images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 30–43. Springer, Heidelberg (2010)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS, pp. 91–99 (2015)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
29. Tong, N., Lu, H., Ruan, X., Yang, M.H.: Salient object detection via bootstrap learning. In: CVPR, pp. 1884–1892 (2015)
30. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. Cogn. Psychol. **12**(1), 97–136 (1980)
31. Wang, L., Lu, H., Ruan, X., Yang, M.H.: Deep networks for saliency detection via local estimation and global search. In: CVPR, pp. 3183–3192 (2015)
32. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: ICCV, pp. 3119–3127 (2015)
33. Wang, L., Ouyang, W., Wang, X., Lu, H.: Stct: sequentially training convolutional networks for visual tracking. In: CVPR (2016)
34. Xie, Y., Lu, H.: Visual saliency detection based on bayesian model. In: ICIP, pp. 645–648 (2011)
35. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: CVPR, pp. 1155–1162 (2013)

36. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: CVPR, pp. 3166–3173 (2013)
37. Zhao, R., Ouyang, W., Li, H., Wang, X.: Saliency detection by multi-context deep learning. In: CVPR, pp. 1265–1274 (2015)
38. Zhu, W., Liang, S., Wei, Y., Sun, J.: Saliency optimization from robust background detection. In: CVPR, pp. 2814–2821 (2014)