Signal Processing: Image Communication 78 (2019) 103–112

Contents lists available at ScienceDirect

Signal Processing: Image Communication

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

Accurate salient object detection via dense recurrent connections and residual-based hierarchical feature integration☆

Yanpeng Cao [a,b], Guizhong Fu [a,b], Jiangxin Yang [a,b,*], Yanlong Cao [a,b], Michael Ying Yang [c]

[a] State Key Laboratory of Fluid Power and Mechatronic Systems, School of Mechanical Engineering, Zhejiang University, Hangzhou, 310027, China
[b] Key Laboratory of Advanced Manufacturing Technology of Zhejiang Province, School of Mechanical Engineering, Zhejiang University, Hangzhou, 310027, China
[c] Scene Understanding Group, University of Twente, Hengelosestraat 99, 7514 AE Enschede, The Netherlands

ARTICLE INFO

Keywords:
Convolutional neural network
Recurrent convolutional layer
Salient object detection
Hierarchical feature fusion
Deep supervision

ABSTRACT

Recently, the convolutional neural network (CNN) has achieved great progress in many computer vision tasks including object detection, image restoration, and scene understanding. In this paper, we propose a novel CNN-based saliency detection method through dense recurrent connections and residual-based hierarchical feature integration. Inspired by the recent neurobiological finding that abundant recurrent connections exist in the human visual system, we firstly propose a novel dense recurrent CNN module (D-RCNN) to learn informative saliency cues by incorporating dense recurrent connections into sub-layers of convolutional stages. Then we present a residual-based architecture with short connections for deep supervision which hierarchically combines both coarse-level and fine-level feature representations. Our end-to-end method takes raw RGB images as input and directly outputs saliency maps without relying on any time-consuming pre/post-processing techniques. Extensive qualitative and quantitative evaluation results on four widely tested benchmark datasets demonstrate that our method can achieve more accurate saliency detection results solutions with significantly fewer model parameters.

1. Introduction

Saliency detection aims at finding the most distinctive objects in an image which are consistent with human visual perception. It is commonly utilized as a preliminary processing step to facilitate a wide range of applications such as object recognition [1], person re-identification [2], image retrieval [3], semantic segmentation [4], scene classification [5], visual tracking [6], video summarization [7] and so on.

Salient object detection has been attracting great attention, and various effective computational models have been developed [8–10]. Inspired by cognitive studies of human visual attention mechanisms [11,12], many existing approaches make use of different types of saliency cues (e.g., color [13], texture [14] and contrast [15,16]) and prior knowledge (e.g., background prior [17], center prior [18] and objectness prior [19]) to predict salient image regions. However, these methods rely on hand-crafted features and pre-defined priors, thus are not capable of generating accurate saliency detection results for images with complex object–scene contextual interactions and highly cluttered backgrounds.

Recently, a number of CNN-based models have been proposed and achieved impressive performances in large-scale saliency detection tasks [20–22]. They successfully utilized semantic information extracted on raw images to overcome the limitations of traditional hand-crafted ones. It is noted that most state-of-the-art methods are based on purely feed-forward CNN architectures. However, the latest studies of human visual system reveal that recurrent connectivities of synapses in the human brain are essential to perform high-level visual perception tasks (e.g., object recognition and saliency detection) [23,24]. Moreover, these models mainly consider the high-level features extracted in late convolutional stages to generate global saliency predictions, which are robust to cluttered backgrounds but unfavorably remove object boundaries and subtle structures. It is important to incorporate both global and local saliency cues to achieve accurate detection results [25,26].

To address the above-mentioned limits, we present a novel saliency detection method based on two major improvements including: (1) building more informative saliency cues through a novel dense recurrent CNN module (D-RCNN) and (2) integrating multi-level feature

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.image.2019.06.004.

* Corresponding author.
E-mail address: yangjx@zju.edu.cn (J. Yang).

https://doi.org/10.1016/j.image.2019.06.004
Received 18 November 2018; Received in revised form 28 March 2019; Accepted 12 June 2019
Available online 14 June 2019

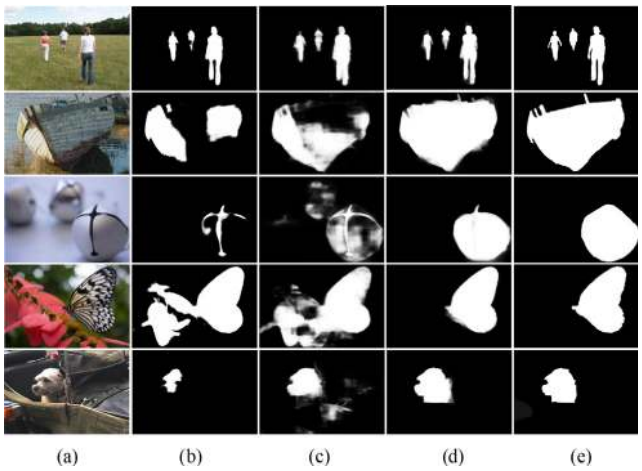

Fig. 1. Comparative results of different saliency detection methods. (a) Input images; (b) Results of DSS [25]; (c) Results of Amulet [27]; (d) Our results. (e) Ground truth.

maps via a residual-based architecture with short connections. The latest experiments reveal that ubiquitous recurrent connections in human brains are essential for high-level vision task performance. Inspired by this neurobiological finding, we first present the D-RCNN module to extract more representative salient cues by adding dense recurrent connections within each convolutional stages of the feed-forward CNN architecture. It is noted that the extracted high-level features provide global cues to robustly predict the location of salient regions while local features are useful to estimate accurate boundary between salient objects and background. Based on this observation, we further develop a residual-based architecture with short connections to hierarchically integrate both global and local feature maps for improving the performance of saliency detection in terms of both accuracy and robustness. Our end-to-end method takes whole images as input and directly outputs full-size saliency maps without relying on any computationally expensive post-processing methods. Some comparative results using different saliency detection methods are shown in Fig. 1. Both source code and trained model will be made publicly available in the future to facilitate research in the related domains.

The contributions of this paper can be summarized as follows:

- We propose a novel dense recurrent CNN module (D-RCNN), in which abundant recurrent convolutional layers (RCL) are added within each convolutional stages of the feed-forward CNN, to learn distinctive feature representations and informative saliency cues.
- We present an effective residual-based deep neural network architecture with short connections to optimize the integration of semantic features and detail features for improving the performance of saliency detection in terms of both robustness and accuracy.
- Experimental results on four public benchmark datasets and comparisons with other state-of-the-art approaches demonstrate the superiority of our proposed method, improving saliency prediction accuracy with significantly fewer model parameters.

2. Related works

We present a review of the most recent studies closely related to our work below.

2.1. Salient object detection

Many saliency detection models have been proposed since two decades ago [8–10,28–30]. Traditional saliency detection methods are typically based on various hand-crafted saliency cues, among which image contrast is the most widely used one. Ma et al. presented a local contrast-based saliency analysis model by evaluating the distinctiveness of each image region with respect to its local neighbors [15]. Achanta et al. proposed to calculate local image contrast at different scale levels to generate saliency maps [16]. A noticeable drawback of local contrast based methods is that they only highlight boundaries but cannot uniformly identify entire salient objects. To overcome the limit, Cheng et al. developed a regional contrast (RC) model to firstly divide an image into various local regions and then assign each region a global contrast value [31]. Although the global-contrast based models provide robust localization results of salient objects, they usually fail to preserve important object details and boundaries. Some researchers attempted to explore various prior knowledge, such as background prior [17], compactness prior [32], objectness prior [19], to extract informative cues for salient object detection. Zhang et al. presented a novel bottom-up salient object detection approach by exploiting the relationship between the saliency detection and null space learning [33]. Li et al. made use of fixation and boundary cues as foreground and background seeds to construct multiple graphs and then integrated the multiple graphs and seeds to generate smooth and accurate saliency maps [34]. However, these methods based on hand-crafted features or pre-defined priors are difficult to detect salient objects with complex textures and cluttered backgrounds.

2.2. Convolution neural network

Recently, CNN-based models have been utilized to tackle the challenging saliency prediction tasks, significant bridging the gap between machine and human visual system [20–22,35]. Attempts are made to combine both hand-crafted low-level and learning-based high-level features to improve the performance of saliency detection [36]. Li et al. proposed a coarse-to-refine approach to detect salient objects with precise boundary by combining pixel-wise FCN with superpixel-based CNN [37]. Wang et al. presented a saliency detection algorithm by integrating both local estimation and global search, which are individually performed through two deep neural networks [38]. Li et al. proposed a multi-task deep salient object detection model by exploring the inherent correlations between saliency detection and semantic image segmentation [39]. Zhao et al. proposed a multi-context deep learning framework for robust salient object detection when the foreground objects share similar appearance with backgrounds [40]. A noticeable drawback of these deep network models is that the generated saliency maps typically contain blurred and inaccurate boundaries since it is difficult to discriminate pixels around the object boundaries.

Zhang et al. presented a generic aggregating multi-level convolutional feature framework for accurate salient object detection by incorporating both coarse semantics and fine details [27]. Hu et al. proposed a deep level set network to generate salient objects with accurate boundaries [41]. Hou et al. adopted the fully connected conditional random field (CRF) as a selective layer during the inference phase to improve spatial coherence and quality of their saliency maps [25]. However, such post-processing techniques are usually time-consuming thus significantly decrease the computational efficiency of saliency detection methods. Wang et al. proposed to neural networks with a novel pyramid pooling module and a multi-stage refinement mechanism [42]. Zhang et al. proposed a novel bi-directional message model to integrate multi-level features for salient object detection [43].

2.3. Recurrent convolution neural network

The latest neuroscience researches reveal that abundant recurrent connections of synapses exist in the human brain. They provide critical functionalities to support high-level visual perception tasks (e.g., object recognition and saliency detection) [23,24,44]. Therefore, it is important to include a recurrent mechanism within the feed-forward

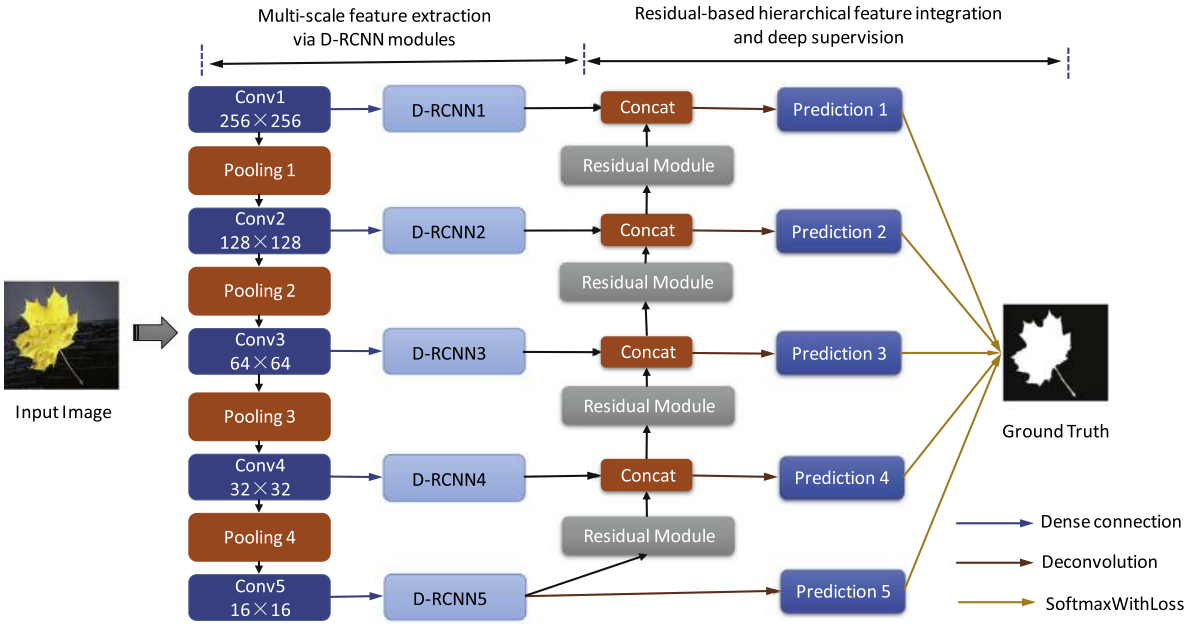

Fig. 2. The overall architecture of our proposed model. Firstly, multi-scale feature extraction at individual convolutional stages via a number of D-RCNN modules (D-RCNN1, D-RCNN2, …, D-RCNN5). Then the extracted features are hierarchically integrated through a residual-based architecture. The model is trained in an end-to-end manner to optimize saliency detection results generated at different scale levels.

CNN architectures for improving the accuracy and efficiency of saliency detection models. A more neurobiologically realistic recurrent neural network (RNN) model is initially proposed for sequential data processing tasks such as handwriting recognition [45,46] and speech recognition [47] and outperforms feed-forward models. Liang et al. presented a recurrent CNN (RCNN), which contains both feed-forward and recurrent connections, to utilize context information for more accurate object recognition [23]. Wang et al. proposed a saliency prediction model using recurrent fully convolutional networks (RFCNs) to refine saliency maps and to precisely describe the compact and boundary-preserving object regions [48]. Liu et al. proposed an end-to-end deep network, in which a hierarchical recurrent convolutional neural network is applied to refine the saliency maps by integrating local context information [26]. Deng et al. proposed a novel recurrent residual refinement network (R^3Net) equipped with residual refinement blocks (*RRBs*) to detect salient regions of an input image [49]. Wang et al. proposed a global Recurrent Localization Network (RLN) to localize accurate salient objects, exploiting contextual information by the weighted response map [50]. Zhang et al. proposed a progressive attention guided recurrent network with multi-path recurrent feedback to enhance multi-level contextual information integration [51].

Our proposed approach differs from the methods mentioned above in two significant aspects. Firstly, a novel dense recurrent CNN module (D-RCNN) is proposed to extract more informative saliency cues based on the latest neurobiological finding that abundant recurrent connections exist in human brains for high-level vision task performance. The proposed D-RCNN module extracts more representative image features by adding dense recurrent convolutional layers within each convolutional stages of a feed-forward CNN model. In contrast, existing RCNN-based models only add recurrent connections to the last layer of each stage [23,26]. Secondly, our method employs a residual-based architecture with short connections to integrate multi-level feature maps. This end-to-end architecture effectively utilizes global and local features for robust object prediction and accurate detail restoration, respectively. Therefore it successfully overcomes limits of other CNN-based saliency detection models such as generating inaccurate or blurry object boundaries [38–40] and relying on additional post-processing techniques [25,48].

3. Proposed method

As illustrated in Fig. 2, the proposed method consists of two main components including (1) multi-scale feature extraction via D-RCNN modules, (2) residual-based hierarchical feature integration and deep supervision. The two components are jointly trained in an end-to-end manner to optimize saliency detection results generated at different scale levels. When testing, the model feed-forwards a raw image through the network and directly outputs high-accuracy saliency maps without using any post-processing methods.

3.1. Dense recurrent convolutional neural network

We build our architecture based on the pre-trained VGG-16 model [52]. VGG-16 model has been successfully used as the backbone in many visual perception tasks and achieved state-of-the-art results. The VGG-16 model consists of five feed-forward convolutional stages (Conv1, Conv2, Conv3, Conv4, and Conv5) and each stage contains a number of sub-layers. In order to include a recurrent mechanism within the feed-forward CNN architectures, a number of recurrent convolutional layers are connected to the last sub-layer of convolutional stages in the VGG-16 network (Conv1-2, Conv2-2, Conv3-3, Conv4-3, and Conv5-3). The states of recurrent convolutional layers evolve over discrete time steps to integrate context information which is critical for high-level visual perception tasks. Inspired by the success of RCNN models [23,26,48], we propose a novel Dense Recurrent Convolutional Neural Network (D-RCNN) model to combine outputs of different sub-layers with a convolutional stage. Different from the RCNN model in which only the final output of a convolutional stage gets involved in recurrent interactions, D-RCNN adds recurrent connections within different sub-layers of a convolutional stage as illustrated in Fig. 3.

The net input $z_{ijk}(t)$ of a neural unit located at (i,j) on the kth feature map in the standard RCNN model [23,26] at time step t is calculated as

$$z_{ijk}(t) = (\mathbf{w}_k^f)^T \mathbf{u}^{(i,j)} + (\mathbf{w}_k^r)^T \mathbf{x}^{(i,j)}(t-1) + b_k, \quad (1)$$

where $\mathbf{u}^{(i,j)}$ is the feed-forward output of the last sub-layer of a convolutional stage, $\mathbf{x}^{(i,j)}(t-1)$ is the recurrent output at previous time $t-1$, $\mathbf{w}_k^f$ and $\mathbf{w}_k^r$ denote the vectorized feed-forward weights and recurrent

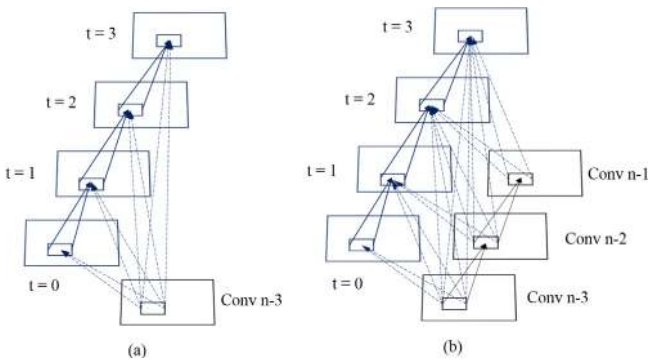

Fig. 3. The overall architecture of (a) a standard RCNN model [23,53] and (b) the proposed D-RCNN. Note the RCNN model only connects recurrent convolutional layers to the last sub-layer of a convolutional stage while the D-RCNN model adds dense recurrent connections to each sub-layers. Comparative evaluation results of these two models are provided in Section 4.3.1.

weights, respectively, and b_k is the bias. Instead of only integrating the output of the last convolution sub-layer in recurrent interactions, we add dense recurrent convolutional connections within different sub-layers of a convolutional stage. The input of a neural unit in D-RCNN becomes

$$z_{ijk}(t) = \sum_{m=1}^{min(t+1,L_{max})} \left((\mathbf{w}_{mk}^{f})^T \mathbf{u}_m^{(i,j)} \right) + (\mathbf{w}_k^r)\mathbf{x}^{(i,j)}(t-1) + b_k, \quad (2)$$

where $\mathbf{u}_m^{(i,j)}$ is the feed-forward output of the mth sub-layer in a convolutional stage from bottom to top, $\mathbf{w}_{mk}^{f}$ is the vectorized feed-forward weights of this sub-layer, and L_{max} denotes the maximum number of sub-layers in this convolutional stage. In the pre-trained VGG-16 model, L_{max} are set to 2, 2, 3, 3 and 3 for convolutional stage 1–5, respectively. The activity of this unit is calculated as

$$x_{ijk}(t) = f(z_{ijk}(t)), \quad (3)$$

where f is the Rectified Linear Unit (ReLU) activation function. It is noted that the state of a unit in D-RCNN depends on both the recurrent signal evolves over iterations and the feed-forward inputs from all sub-layers in a convolutional stage.

Although D-RCNN and RCNN models both consist of a number of RCLs, they are very different in two major aspects. First, the RCNN models typically add recurrent connections to the last sub-layer of a convolutional stage [23,26], thus only the final output of a convolutional stage is used to generate context information for saliency detection. In comparison, the D-RCNN model introduces dense recurrent connections to utilize outputs of each sub-layers for improving the ability of the model to integrate context information, thus constructs more representative features. Second, D-RCNN model provides more path options between the input layer to the output layer to improve gradient back-propagation during network training. For instance, the dense recurrent layers connected to the first sub-layer in a convolutional stage (e.g., Con1-1, Con2-1, Con3-1, Con4-1, and Con5-1) add shorter paths by bypassing all feed-forward convolutional layers and recurrent convolutional layers, while the RCNN model can only bypass the recurrent convolutional layers. More comparative evaluation results are provided in Section 4.3.1.

3.2. Residual-based hierarchical feature integration and deep supervision

Using the proposed D-RCNN modules, a number of scale-dependent feature maps $\mathbf{X}_n$ $(n = 1, 2.., 5)$ are extracted in convolutional stages Con-1, Con-2, Con-3, Con-4, and Con-5, respectively. It is well-known that the feature maps extracted in deeper layers encode high-level scene information to estimate the global location of salient regions, while the rich low-level features learned in shallower layers are useful to refine accurate boundaries of objects [25,26,54]. It is reasonable to utilize both high-level and low-level features extracted in layers with different depths for robust and accurate saliency detection. Instead of directly combining feature maps extracted in different layers through the simple concatenation technique [21,25,55], we introduce a residual-based module for effective step-by-step feature integration which leads to the performance gain of saliency detection by adding more gradient back-propagation path options to facilitate better network training. Residual module is initially proposed by He et al. for image recognition [56], and then has been successfully utilized for other computer vision tasks [57–59].

The detailed architecture of residual-based hierarchical feature integration (RHI) is shown in Fig. 4. In convolutional stage n, the output of D-RCNN module $\mathbf{X}_n$ is combined with the residual output $\mathbf{R}_n$ of stage $n+1$ to generate the fused feature map $\mathbf{F}_n$ as

$$\mathbf{F}_n = f^{cat}(\mathbf{X}_n, \mathbf{R}_n), \quad (4)$$

where f^{cat} represents the concatenation operation stacking $\mathbf{X}_n$ and $\mathbf{R}_n$ feature maps at the same spatial locations but across the feature channels. In the last convolutional stage ($n = 5$), we set $\mathbf{R}_5 = \mathbf{X}_5$ since it cannot be computed in a deeper convolutional stage. The fused feature map $\mathbf{F}_n$ then goes through a residual function, which consists of 2 convolutional layers, 2 ReLU layers and a 2×2 deconvolutional layer (double the size of feature map), to compute the residual feature map $\mathbf{R}_{n-1}$ for feature integration at stage $n-1$. The fused feature map $\mathbf{F}_n$ is utilized to generate saliency perdition in stage n through a deconvolutional layer. For example, when $n = 5$, the size of feature maps X_5 and R_5 is 16×16, and after the 16×16 Deconvolution layer, the size of output is $1 \times 256 \times 256$.

Let $\{(I_m, Y_m), m = 1, 2, \ldots, M\}$ denote the training dataset, where $I_m = \left\{ i_j^m, j = i, \ldots, |I_m| \right\}$ is the input image with $|I_m|$ pixels and $Y_m = \left\{ y_j^m, j = i, \ldots, |Y_m| \right\}$, $y_j^m \in [0, 1]$ denotes the corresponding ground truth saliency map of image I_m. $y_j^m = 1$ is foreground pixel and $y_j^m = 0$ is a background pixel. In addition, $\mathbf{W}^f$, $\mathbf{W}^r$ and $\mathbf{W}^{res}$ denote the parameters of the feed-forward VGG-16 model, the recurrent D-RCNN modules, and the residual-based feature integration modules, respectively. The sigmoid cross entropy loss function L_f^n in convolutional stage n is defined as

$$\begin{aligned} L_f^n = -\sum_j \{ & y_j \log Pr(o_j | I; \mathbf{W}^f, \mathbf{W}^r, \mathbf{W}^{res}) \\ & + (1 - y_j) \log Pr(o_j | I; \mathbf{W}^f, \mathbf{W}^r, \mathbf{W}^{res}) \}, \end{aligned} \quad (5)$$

where $Pr(o_j | I; \mathbf{W}^f, \mathbf{W}^r, \mathbf{W}^{res})$ represents the confidence score of a pixel belongs to the foreground. The confidence score is calculated using the sigmoid function as

$$Pr(o_j | I; \mathbf{W}^f, \mathbf{W}^r, \mathbf{W}^{res}) = \frac{1}{1 + e^s}, \quad (6)$$

where s is the pixel value in the last convolution layer of our architecture. The final multi-loss function L_f is defined as

$$L_f(\mathbf{W}^f, \mathbf{W}^r, \mathbf{W}^{res}) = \sum_{l=1}^{N} \alpha_l L_f^l(\mathbf{W}^f, \mathbf{W}^r, \mathbf{W}^{res}), \quad (7)$$

where $N = 5$, and $\alpha_l = 1 (l = 1, 2..., 5)$. The Adaptive Moment Estimation (Adam) method [60] is used to compute the optimal parameters as

$$(\mathbf{W}^{f*}, \mathbf{W}^{r*}, \mathbf{W}^{Res*}) = \arg min L_f(\mathbf{W}^f, \mathbf{W}^r, \mathbf{W}^{res}), \quad (8)$$

As mentioned above, it is critical to develop an effective feature fusion scheme to combine both global and local features for accurate saliency detection. Instead of directly combining scale-dependent feature representations through a simple concatenation/sum operation [21,25,27], we make use of a residual module to perform hierarchical feature integration from coarse to fine. The residual module, which consists of a number of feed-forward layers and shortcut connections,

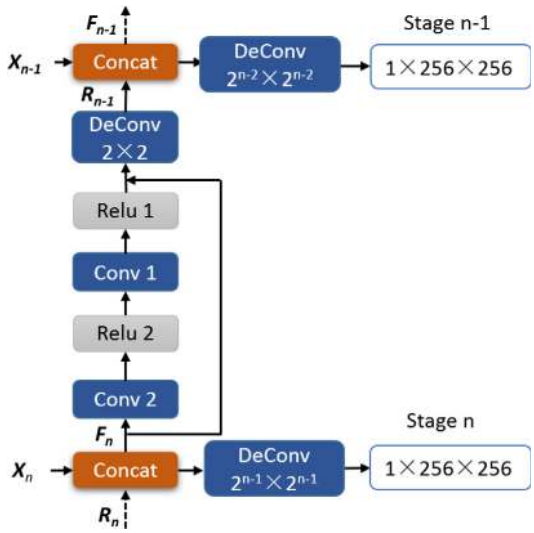

Fig. 4. The detailed architecture of the residual-based step by step feature integration between two adjacent stages.

provides multiple path options to facilitate better network training. More specifically, the longer paths (going through feed-forward layers in the residual modules) improve the capability of a network to construct more complex/distinctive features, while the shorter paths (bypassing feed-forward layers through shortcut connections) strengthen the gradient back-propagation process and make models easier to optimize [56]. The comparative evaluation of a number of alternative fusion architectures is provided in Section 4.3.2.

It is worth mentioning that the architecture of our proposed method is very different from the ones used in Amulet [27] and DHS [26] in two aspects. In the feature extraction stage, Amulet and DHS utilize the output of the last feature layer in each scale. In comparison, we apply a dense recurrent convolutional neural network to enhance the extracted features in different scales. In the feature integration stage, Amulet utilizes a simple concatenation function to combine feature maps extracted at different convolutional states and DHS made use of a recurrent neural network (RNN) to combine coarse and fine features adaptively. In comparison, we propose a residual-based architecture for hierarchical feature integration.

4. Experiments

In this section, we describe the training/testing datasets and evaluation criteria and provide implementation details of our proposed model. The proposed approach is systematically evaluated and compared with the state-of-the-art alternatives.

4.1. Datasets and evaluation metrics

We make use of 10,000 images from MSRA10k dataset [31] as the training dataset. Most of the images in this dataset contain only one salient object. Some standard image augmentation techniques (e.g., image rotation, cropping, and flipping) are applied to increase the varieties of training data. In total, we obtain 80,000 images with high-quality pixel-wise annotations. For the performance evaluation, we consider four public saliency detection benchmark datasets including ECSSD [61], PASCAL-S [62], HKU-IS [63], and DUT-OMRON [8]. ECSSD contains 1000 semantically meaningful but structurally complex natural images with objects of different sizes. PASCAL-S contains 850 challenging natural images which are selected from the validation set of the PASCAL VOC 2010 segmentation dataset. HKU-IS is a recently released dataset containing 4447 images with high-quality pixel-wise annotations. Images of this dataset are chosen to include multiple disconnected salient objects or objects touching the image boundary. DUT-OMRON is another challenging dataset which has 5168 images and each image contains one or more salient objects and complex backgrounds. All these datasets are manually annotated with pixel-wise ground-truth.

We adopt the most commonly used evaluation metrics, F-measure and Mean absolute error (MAE), to evaluate the performances of different saliency detection methods. F-measure is a harmonic mean of average precision and average recall, which is calculated as

$$F_\beta = \frac{(1+\beta^2)\times Precision \times Recall}{\beta^2 \times Precision + Recall}, \tag{9}$$

where $Precision$ and $Recall$ are computed by thresholding the predicted saliency map and comparing the binary map with the ground truth. Here we set the balance parameter $\beta^2 = 0.3$ to emphasize the importance of precision [64]. We also consider another evaluation index MAE which is calculated as

$$MAE = \frac{1}{W \times H}\sum_{x=1}^{W}\sum_{y=1}^{H}|S(x,y) - G(x,y)|, \tag{10}$$

where W and H are the width and height of the input image, $S(x,y)$ is the computed saliency map, and $G(x,y)$ is the ground truth. This metric provides a comprehensive evaluation of the overall detection performance in both salient and non-salient regions.

4.2. Implementation details

Our network is implemented using the publicly available Caffe library [65]. All input images are resized to 256 × 256 for training and testing. The parameters of multi-scale feature extraction layers are initialized according to the pre-trained VGG-16 model. Parameters of newly added convolutional layers are initialized randomly using the "Xavier" method [66]. The parameters of recurrent convolutional layers are provided in Table 1. Note the channel number of each convolutional layer is set to 64 to reduce the size of our model, and larger kernels are employed in deeper layers to facilitate larger receptive fields. Our network is trained using Adaptive Moment Estimation (Adam) method [60]. The batch size is set to 4 for all experiments. We use the "step" learning policy, and set base learning rate to 10^{-4}, gamma to 0.1, step size to 20,000, weight decay to 10^{-4}, momentum to 0.9, and *iter_size* to 8. To avoid exploding gradient problems in the training process [67], we clip gradients when the L2 norm of the gradients exceeds 35. It takes about 35 h to train our model on a single NVIDIA TITAN X GPU (12G memory) and a 2.6 GHz Intel Xeon processor. The source code will be made publicly available in the future.

4.3. Experimental results

We first evaluate the effectiveness of two main components of our method including feature extraction based on D-RCNN modules and residual-based hierarchical feature integration. Then we provide qualitative and quantitative evaluation results of our saliency detection method and a number of state-of-the-art ones. For a fair comparison, we only consider a number of approaches based on the pre-trained VGG-16 model [52].

4.3.1. D-RCNN modules

Inspired by the latest neurobiological finding, we integrate recurrent convolutional layers into the feed-forward CNN architecture (VGG-16 model [52]) to improve its ability to extract distinctive and scale-dependent feature maps in individual convolutional stages. Different from the existing RCNN-based models [23,26], our D-RCNN module adds dense recurrent connections within each convolutional stages of the feed-forward CNN, as illustrated in Fig. 3. We perform

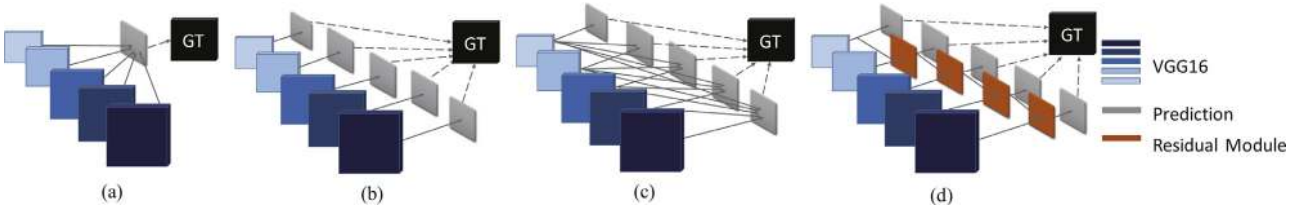

Fig. 5. Illustration of different architectures for multi-scale feature integration and deep supervision including (a) DCL [21]; (b) HED [55]; (c) DSS [25]; and (d) Ours. GT denotes the ground truth saliency map.

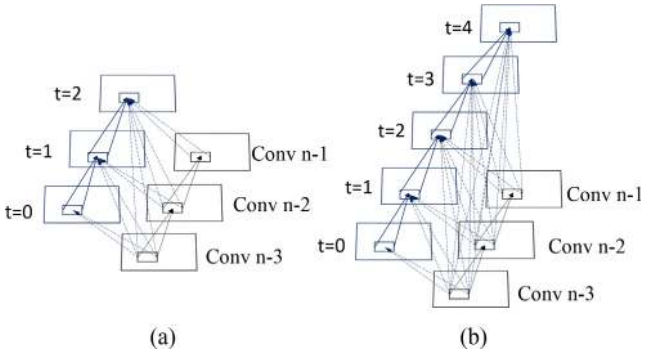

Fig. 6. The architecture of D-RCNN modules with (a) two and (b) four recurrent steps.

Table 1
The detailed configurations of recurrent layers in D-RCNN models in different convolutional stages.

Layer name	Kernel size	Pad	Channel
Conv1-a	1 × 1	1	64
Conv1-b	3 × 3	1	64
Conv1-c	3 × 3	1	64
Conv2-a	3 × 3	1	64
Conv2-b	3 × 3	1	64
Conv2-c	3 × 3	1	64
Conv3-a	3 × 3	1	64
Conv3-b	3 × 3	1	64
Conv3-c	3 × 3	1	64
Conv3-d	3 × 3	1	64
Conv4-a	3 × 3	1	64
Conv4-b	3 × 3	1	64
Conv4-c	3 × 3	1	64
Conv4-d	5 × 5	2	64
Conv5-a	5 × 5	2	64
Conv5-b	5 × 5	2	64
Conv5-c	5 × 5	2	64
Conv5-d	7 × 7	3	64

saliency detection using **D-RCNN** and two other baseline models on ECSSD [61] and PASCAL-S [62] benchmark datasets. The first baseline model (**Plain**) is a conventional feed-forward CNN without any recurrent connections (VGG-16 model [52]). The second baseline model (**RCNN**) is constructed by adding recurrent convolutional layers to the last sub-layer of each convolutional stages. Moreover, we set up experiments to evaluate the performance of D-RCNN and RCNN models with different recurrent steps. More specifically, we set the recurrent time step t to 2, 3 and 4 for both RCNN and D-RCNN modules. The architectures of D-RCNN modules with 2 and 4 recurrent steps are provided in Fig. 6. For a fair comparison, we directly utilize the output of RCNN/D-RCNN in the last stage of VGG16 to generate saliency predictions without using any feature integration or deep supervision techniques. The performances (F_β and MAE) of **Plain** (no recurrent layers), **RCNN**, and **D-RCNN** models are quantitatively compared in Table 2.

Table 2
The comparative evaluation of different modules (**Plain**, **RCNN**, and **D-RCNN**) on ECSSD and PASCAL-S benchmark datasets. Here we directly utilize the output of RCNN/D-RCNN in the last stage of VGG16 to generate saliency predictions without using any feature integration or deep supervision techniques. Note higher F_β and lower MAE indicate better saliency detection performance.

Model		Dataset			
		ECSSD		PASCAL-S	
		F_β	MAE	F_β	MAE
RCNN	t = 2	0.821	0.085	0.751	0.139
	t = 3	0.826	0.085	0.756	0.139
	t = 4	0.818	0.082	0.740	0.135
D-RCNN	t = 2	0.838	0.080	0.765	0.126
	t = 3	**0.849**	**0.075**	**0.769**	**0.126**
	t = 4	0.843	0.078	0.762	0.128
Plain	–	0.785	0.105	0.728	0.158

We observe that better detection performance can generally be achieved by incorporating recurrent connections into the feed-forward CNN model. This improvement enhances the ability of the model to integrate the context information for more accurate salient region detection. With three recurrent steps ($t = 3$), F_β index is increased from 0.785 (**Plain**) to 0.826 (**RCNN**) and MAE is decreased from 0.105 (**Plain**) to 0.085 (**RCNN**) in ECSSD dataset. In PASCAL-S dataset, these two evaluation metrics are also improved by 3.8% and 12.0%, respectively. Moreover, our proposed D-RCNN model can further boost the accuracy of detection results by adding dense recurrent convolutional layers between sub-layers within a single convolutional stage. With three recurrent steps ($t = 3$), F_β is significantly increased from 0.826 (**RCNN**) to 0.849 (**D-RCNN**) and MAE is further decreased from 0.085 (**RCNN**) to 0.075 (**D-RCNN**) in ECSSD dataset. It is worth mentioning that D-RCNN modules with different recurrent steps (2, 3 and 4) all achieve more accurate saliency detection results. Experimental results demonstrate that D-RCNN provides a more effective way to extract representative feature maps in individual convolutional stages by utilizing information from more sub-layers and adding more path options between the input and output layers. We empirically found that the D-RCNN module with three recurrent steps ($t = 3$) achieves the highest F_β and the lowest MAE values, therefore we adopt D-RCNN ($t = 3$) for feature extraction in following experiments. In the future, we plan to set up experiments to investigate other alternatives to optimize the architecture of D-RCNN module (e.g., using how many recurrent steps in each convolutional stages).

4.3.2. *Residual-based hierarchical feature integration and deep supervision*

Then we compare the residual-based hierarchical feature integration (RHI) with a number of alternatives in an attempt to identify the optimal multi-stage feature integration architecture for salient region detection. Besides our proposed RHI, we consider three other alternatives including DCL [21], HED [55], and DSS [25] as illustrated in Fig. 5. In DCL architecture, feature maps extracted in multiple stages are directly combined to generate a final prediction for network training. In HED architecture, features maps extracted in different scale are individually utilized to generate multiple prediction results for

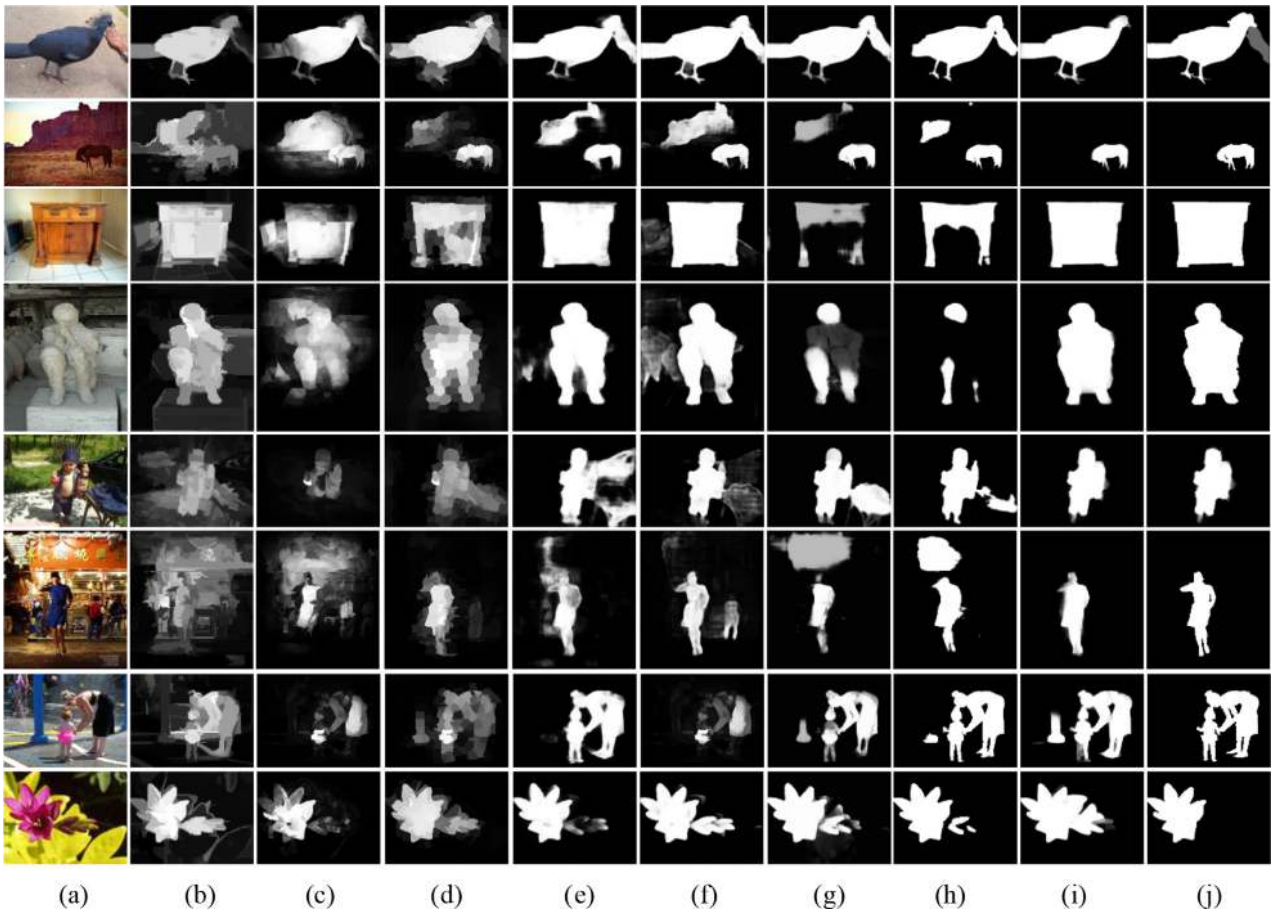

Fig. 7. Visual comparison with state-of-the-art detection methods. (a) Input image; (b) DRFI [68]; (c) DSR [69]; (d) DS [39]; (e) Amulet [27]; (f) UCF [35]; (g) DCL [21]; (h) DSS [25]; (i) Ours; (j) Ground Truth. Saliency maps or source codes of different methods are provided by their authors.

deep supervision. In DSS architecture, a series of short connections are introduced to combining global features extracted in deeper layers and local features extracted in shallower layers. In comparison, RHI architecture employs a number of residual modules to combine multi-scale features from coarse to fine hierarchically. Their comparative results are shown in Table 3. In our experiments, we use the "step" learning policy, and set base learning rate to 10^{-4}, gamma to 0.1, step size to 20,000, weight decay to 10^{-4}, momentum to 0.9, and *iter_size* to 8. It is noted that models incorporating different feature integration schemes (HED [55], DSS [25] and our proposed RHI) can all converge properly except the one based on the DCL feature integration architecture [21]. Therefore the performance of DCL feature integration scheme is significantly worse than the results of others. Our experimental results suggest that directly combining feature maps extracted in multiple stages to generate a final prediction for network training is not very stable. It is also observed that RHI outperforms other architectures since residual modules can both deploy longer paths for constructing distinctive feature maps and shorter paths for fast gradient back-propagation.

Our model is constructed based on the VGG-16 model which contains 5 convolutional stages. During the training process, 5 different predictions are computed in each stage to supervise the learning process. In Table 4, we evaluate the detection accuracy of different predicted saliency maps. It is noted that the last prediction achieves the highest F_β and lowest MAE. Therefore, only the last prediction is calculated as the final saliency map to decrease the deployment time of the model.

4.3.3. Comparison with state-of-the-art

In this section, we compare our proposed method with the state-of-the-art saliency detection methods including DRFI [68], DSR [69], NS [33], DS [39], Amulet [27], DHS [26], UCF [35], DCL [21] and DSS [25] in terms of both prediction accuracy and execution speed. The first two methods are based on traditional hand-crafted features and the remaining ones are deep learning based. For a fair comparison, we use either the implementations with recommended parameters or the saliency maps provided by the authors.

Fig. 7 provides several visual comparison results where our method outperforms the approaches mentioned above and some failure examples. It is observed that our method successfully highlight salient regions while suppressing background distraction, thus it can generate

Table 3
Comparative evaluation of different feature integration techniques on ECSSD and PASCAL-S dataset. Note higher F_β and lower MAE indicate better saliency detection performance.

Models	Dataset			
	ECSSD [61]		PASCAL-S [62]	
	F_β	MAE	F_β	MAE
DCL [21]	0.280	0.235	0.330	0.241
HED [55]	0.860	0.103	0.745	0.145
DSS [25]	0.869	0.071	0.790	0.105
RHI	**0.887**	**0.053**	**0.802**	**0.093**

Table 4
The performance of five different predictions (P1, P2, …, P5) in ECSSD dataset. Note higher F_β and lower MAE indicate better saliency detection performance.

	P1	P2	P3	P4	P5
F_β	0.853	0.874	0.881	0.885	**0.887**
MAE	0.066	0.065	0.056	0.054	**0.053**

Table 5
Comparison of quantitative results (average F_β and MAE) of different saliency detection methods. The top three results are highlighted in red, green, and blue, respectively. DL denotes the method is deep learning based. SP denotes the method utilizes superpixels over-segmentation. CRF denotes the method adopts a conditional random field (CRF) model during the inference phase. Note higher F_β and lower MAE indicate better saliency detection performance.

Model \ Dataset	ECSSD [62]		PASCAL-S [63]		HKU-IS [65]		DUT-OMRON [64]		Methods
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	
Ours	0.887	0.052	0.802	0.093	0.868	0.044	0.725	0.066	DL
Amulet [9]	0.868	0.059	0.768	0.098	0.843	0.050	0.647	0.098	DL
UCF [36]	0.844	0.069	0.768	0.115	0.823	0.061	0.621	0.120	DL
DHS [28]	0.867	0.060	0.778	0.094	0.854	0.053	-	-	DL
DS [40]	0.826	0.122	0.691	0.176	0.787	0.078	0.603	0.120	DL+SP
ELD [37]	0.810	0.08	0.718	0.123	0.776	0.072	0.611	0.092	DL+SP
DCL [24]	0.864	0.075	0.798	0.112	0.868	0.055	0.694	0.086	DL+CRF
DSS [8]	0.904	0.052	0.817	0.093	0.902	0.040	0.740	0.063	DL+CRF
DSR [71]	0.690	0.171	0.614	0.204	0.675	0.143	0.525	0.139	Hand-crafted
DRFI [70]	0.733	0.164	0.614	0.226	0.720	0.138	0.548	0.155	Hand-crafted
NS [34]	0.717	0.158	0.637	0.192	-	-	0.621	0.147	Hand-crafted

Table 6
The running time and model size of several deep-learning based methods. All methods are conducted in a PC with a single NVIDIA TITAN X GPU (12G memory) and a 2.6 GHz Intel Xeon processor. Note DSS method is not purely deep-leaning based and requires another 0.4 s for its CRF model.

	Running time (s)	Model size (MB)
Ours	**0.038**	**67**
Amulet [27]	0.043	118
UCF [35]	0.063	132
DHS [26]	0.043	358
DSS [25]	0.08 + 0.4	237

connected salient regions and impose sharp boundaries between foreground and background as illustrated in the 1–3 rows. Images in the 4–6 rows contain extremely clustered background and the foreground objects have similar appearances with backgrounds. In these challenging cases, most of the compared methods fail to identify the salient objects while our method successfully detects them with high precision. We also show some failed saliency detection examples which are the cases that the background image has high contrast (e.g., Row 7). It is noted that our detection results are not as good as the ones involving an additional inference phase (e.g., using the CRF model [25]) to improve the spatial coherence of saliency detection results when foreground objects have similar appearances with backgrounds (e.g., Row 8). However, such post-processing techniques are time-consuming thus significantly decrease the computational efficiency of saliency detection methods.

Table 5 shows the comparative results of our method and state-of-the-art ones. It is observed that deep learning based methods significantly outperform the hand-crafted ones which reaffirm the superiority of deep neural network models for saliency detection. Compared with three other purely deep-learning based methods (Amulet [27], UCF [35], DHS [26]), our method achieves higher F_β values in all four datasets (increasing F_β by 1.0% in ECSSD, 4.8% in PASCAL-S, 1.7% in HKU-IS, and 4.5% in DUT-OMRON. Also, it produces lower MAE results in the DUT-OMRON and HKU-IS datasets. DSS is the only method outperforms ours, but it adopts a time-consuming conditional random field (CRF) model to improve spatial coherence and quality of saliency maps which significantly decrease its computational efficiency.

Another advantage of our method is that it runs faster and requires smaller storage space as illustrated in Table 6. All models were tested with 256 × 256 × 3 RGB images. The running times of these methods are evaluated using a PC with a single NVIDIA TITAN X GPU (12G memory) and a 2.6 GHz Intel Xeon processor. Our method run 13.2% faster than the second faster one (Amulet [27]) and its model size is decreased almost by half. Our method can process 26 FPS which is almost a real-time speed. It is worth mentioning that although DSS method performs better than ours, it involves a conditional random field (CRF) model to refine saliency maps which is very time-consuming (requires extra 0.4 s for CRF) and unsuitable for real-time implementation.

5. Conclusion

Recently, a number of CNN-based models are utilized to improve performances of the challenging saliency prediction task. However, most state-of-the-art methods are based on purely feed-forward CNN architectures which do not contain important recurrent connections for performing high-level visual perception tasks. Moreover, it still remains an open question what the optimal strategy to combine global and local saliency cues for accurate salient object detection is. In this paper, we present two significant improvements in an attempt to address the above problems. Firstly, we present a novel D-RCNN module to extract more representative salient cues by adding dense recurrent connections within each convolutional stages of the feed-forward CNN. Secondly, we develop a residual-based architecture to hierarchically integrate both global and local feature maps for improving both the accuracy and robustness of saliency detection. Both qualitative and quantitative evaluation results on multiple benchmark datasets demonstrate that our method achieves more accurate saliency detection results than most state-of-the-art solutions with significantly fewer model parameters. Our approach runs almost in real-time (26 fps), therefore it can be utilized as a pre-processing technique to improve the performance of other high-level computer vision applications such as target recognition, person re-identification, abnormalities detection, scene understanding and so on. In the future, we plan to optimize the architecture of D-RCNN module (e.g., what is the optimal way to set up connections between recurrent layers and the convolutional layer of the feed-forward CNN). Also, it would be interesting to apply D-RCNN and RHI to other pre-trained CNN models such as ResNet [56] and GoogLeNet [70] to verify the effectiveness of the proposed methods.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (No. 51575486, No. 51605428 and U1664264).

References

[1] Zhixiang Ren, Shenghua Gao, Liang Tien Chia, Wai Hung Tsang, Region-based saliency detection and its application in object recognition, IEEE Trans. Circuits Syst. Video Technol. 24 (5) (2014) 769–779.
[2] R. Zhao, W. Oyang, X. Wang, Person re-identification by saliency learning, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2) (2017) 356–370.
[3] Yanzhang Wu, Hongzhe Liu, Jiazheng Yuan, Qikun Zhang, Is visual saliency useful for content-based image retrieval?, Multimedia Tools Appl. 77 (11) (2018) 13983–14006.
[4] Y. Wei, X. Liang, Y. Chen, X. Shen, M.M. Cheng, J. Feng, Y. Zhao, S. Yan, Stc: A simple to complex framework for weakly-supervised semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (11) (2017) 2314–2320.

[5] Fan Zhang, Bo Du, Liangpei Zhang, Saliency-guided unsupervised feature learning for scene classification, IEEE Trans. Geosci. Remote Sens. 53 (4) (2015) 2175–2184.
[6] A Aghamohammadi, M.C. Ang, A Sundararajan E, N.K. Weng, M Mogharrebi, S.Y. Banihashem, A parallel spatiotemporal saliency and discriminative online learning method for visual target tracking in aerial videos, Plos One 13 (2) (2018) e0192246.
[7] Naveed Ejaz, Irfan Mehmood, Muhammad Sajjad, Sung Wook Baik, Video summarization by employing visual saliency in a sufficient content change method, Int. J. Comput. Theory Eng. (2014) 26–29.
[8] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, Ming Hsuan Yang, Saliency detection via graph-based manifold ranking, in: CVPR, 2013, pp. 3166–3173.
[9] Xiaodi Hou, Liqing Zhang, Saliency detection: a spectral residual approach, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2007, pp. 1–8.
[10] Neil Bruce, John Tsotsos, Saliency based on information maximization, in: Advances in Neural Information Processing Systems, 2006, pp. 155–162.
[11] L. Itti, C. Koch, Computational modelling of visual attention, Nat. Rev. Neurosci. 2 (3) (2001) 194–203.
[12] Derrick Parkhurst, Klinton Law, Ernst Niebur, Modeling the role of salience in the allocation of overt visual attention, Vis. Res. 42 (1) (2002) 107–123.
[13] Jiwhan Kim, Dongyoon Han, Yu-Wing Tai, Junmo Kim, Salient region detection via high-dimensional color transform, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 883–890.
[14] Christian Scharfenberger, Alexander Wong, Khalil Fergani, John S Zelek, David A Clausi, Statistical textural distinctiveness for salient region detection in natural images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 979–986.
[15] Yu Fei Ma, Hong Jiang Zhang, Contrast-based image attention analysis by using fuzzy growing, in: Eleventh ACM International Conference on Multimedia, 2003, pp. 374–381.
[16] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, Sabine Süsstrunk, Salient Region Detection and Segmentation, in: springer lecture notes in computer science, vol. 5008, 2008, pp. 66–75,
[17] Yichen Wei, Fang Wen, Wangjiang Zhu, Jian Sun, Geodesic saliency using background priors, in: European Conference on Computer Vision, Springer, 2012, pp. 29–42.
[18] Chuan Yang, Lihe Zhang, Huchuan Lu, Graph-regularized saliency detection with convex-hull-based center prior, IEEE Signal Process. Lett. 20 (7) (2013) 637–640.
[19] Kai Yueh Chang, Tyng Luh Liu, Hwann Tzong Chen, Shang Hong Lai, Fusing generic objectness and visual saliency for salient object detection, in: International Conference on Computer Vision, 2011, pp. 914–921.
[20] Lijuan Duan, Chunpeng Wu, Jun Miao, Laiyun Qing, Yu Fu, Visual saliency detection by spatially weighted dissimilarity, in: Computer Vision and Pattern Recognition, 2011, pp. 473–480.
[21] G. Li, Y. Yu, Deep contrast learning for salient object detection, in: CVPR, 2016, pp. 478–487.
[22] Hongyang Li, Jiang Chen, Huchuan Lu, Zhizhen Chi, Cnn for saliency detection with low-level feature integration, Neurocomputing 226 (2017) 212–220.
[23] Ming Liang, Xiaolin Hu, Recurrent convolutional neural network for object recognition, in: Computer Vision and Pattern Recognition, 2015, pp. 3367–3375.
[24] G. Deco, T.S. Lee, The role of early visual cortex in visual integration: a neural model of recurrent interaction., Eur. J. Neurosci. 20 (4) (2015) 1089–1100.
[25] Qibin Hou, Ming Ming Cheng, Xiao Wei Hu, Ali Borji, Zhuowen Tu, Philip Torr, Deeply supervised salient object detection with short connections, in: CVPR, 2017.
[26] Nian Liu, Junwei Han, Dhsnet: Deep hierarchical saliency network for salient object detection, in: CVPR, 2016, pp. 678–686.
[27] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, Ruan Xiang, Amulet: Aggregating multi-level convolutional features for salient object detection, in: ICCV, 2017.
[28] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.
[29] Ali Borji, Laurent Itti, Exploiting local and global patch rarities for saliency detection, in: 2012 conference on computer vision and pattern recognition, IEEE, 2012, pp. 478–485.
[30] Shijian Lu, Cheston Tan, Joo-Hwee Lim, Robust and efficient saliency modeling from image co-occurrence histograms, IEEE Trans. Pattern Anal. Mach. Intell. 36 (1) (2014) 195–201.
[31] M.M. Cheng, N.J. Mitra, X. Huang, P.H.S. Torr, S.M. Hu, Global contrast based salient region detection, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 569–582.
[32] A. Hornung, Y. Pritch, P. Krahenbuhl, F. Perazzi, Saliency filters: Contrast based filtering for salient region detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 733–740.
[33] Ying Ying Zhang, Shuo Zhang, Ping Zhang, Zhang Xin Gang, Signal Processing: Image Communication 70 (2019) 271–281.
[34] Shiqi Li, Cheng Zeng, Yan Fu, Shiping Liu, Optimizing multi-graph learning based salient object detection, Signal Process. Image Commun. 55 (2017) 93–105.
[35] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, Baocai Yin, Learning uncertain convolutional features for accurate saliency detection, in: ICCV, 2017.
[36] G. Lee, Y.W. Tai, J. Kim, Deep saliency with encoded low level distance map and high level features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 660–668.
[37] Ying Li, Fan Cui, Xizhe Xue, Jonathan Cheung-Waichan, Coarse-to-fine salient object detection based on deep convolutional neural networks, Signal Process. Image Commun. 64 (2018) 21–32.
[38] Lijun Wang, Huchuan Lu, Ruan Xiang, Ming Hsuan Yang, Deep networks for saliency detection via local estimation and global search, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3183–3192.
[39] X. Li, L. Zhao, L. Wei, M.H. Yang, F. Wu, Y. Zhuang, H. Ling, J. Wang, Deepsaliency: Multi-task deep neural network model for salient object detection., TIP 25 (8) (2016) 3919–3930.
[40] Rui Zhao, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Saliency detection by multi-context deep learning, in: Computer Vision and Pattern Recognition, 2015, pp. 1265–1274.
[41] Ping Hu, Bing Shuai, Jun Liu, Gang Wang, Deep level sets for salient object detection, in: CVPR, 2017, pp. 540–549.
[42] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, Huchuan Lu, A stage-wise refinement model for detecting salient objects in images, in: The IEEE International Conference on Computer Vision, 2017.
[43] Lu Zhang, Ju Dai, Huchuan Lu, You He, Gang Wang, A bi-directional message passing model for salient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[44] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, Trevor Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2625–2634.
[45] Alex Graves, Marcus Liwicki, Horst Bunke, Horst Bunke, A novel connectionist system for unconstrained handwriting recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (5) (2009) 855–868.
[46] Alex Graves, Offline handwriting recognition with multidimensional recurrent neural networks, in: International Conference on Neural Information Processing Systems, 2008, pp. 545–552.
[47] Alex Graves, Abdel Rahman Mohamed, Geoffrey Hinton, Speech recognition with deep recurrent neural networks, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6645–6649.
[48] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, Xiang Ruan, Saliency detection with recurrent fully convolutional networks, in: European Conference on Computer Vision, Springer, 2016, pp. 825–841.
[49] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, Pheng-Ann Heng, R^3NEt: Recurrent residual refinement network for saliency detection, in: IJCAI, 2018.
[50] Tiantian Wang, Lihe Zhang, Shuo Wang, Huchuan Lu, Gang Yang, Xiang Ruan, Ali Borji, Detect globally, refine locally: A novel approach to saliency detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3127–3135.
[51] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, Gang Wang, Progressive attention guided recurrent network for salient object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 714–722.
[52] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Comput. Sci. (2014).
[53] C.J. Spoerer, P. Mcclure, N. Kriegeskorte, Recurrent convolutional neural networks: A better model of biological object recognition., Front. Psychol. 8 (2017) 1551.
[54] Eleonora Vig, Michael Dorr, David Cox, Large-scale optimization of hierarchical features for saliency prediction in natural images, in: Computer Vision and Pattern Recognition, 2014, pp. 2798–2805.
[55] Saining Xie, Zhuowen Tu, Holistically-nested edge detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1395–1403.
[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[57] Wei Ke, Jie Chen, Jianbin Jiao, Guoying Zhao, Qixiang Ye, Srn: side-output residual network for object symmetry detection in the wild, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 302–310.
[58] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, Enhanced deep residual networks for single image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 136–144.
[59] Eduardo Romera, José M Alvarez, Luis M Bergasa, Roberto Arroyo, Erfnet: Efficient residual factorized convnet for real-time semantic segmentation, IEEE Trans. Intell. Transp. Syst. 19 (1) (2018) 263–272.
[60] D.P. Kingma, L.J. Ba, Adam: A method for stochastic optimization, Comput. Sci. (2014).
[61] Qiong Yan, Li Xu, Jianping Shi, Jiaya Jia, Hierarchical saliency detection, in: CVPR, 2013, pp. 1155–1162.
[62] Yin Li, Xiaodi Hou, Christof Koch, James M. Rehg, Alan L. Yuille, The secrets of salient object segmentation, in: CVPR, 2014, pp. 280–287.

[63] G. Li, Y. Yu, Visual saliency based on multiscale deep features, in: CVPR, 2015, pp. 5455–5463.
[64] Ali Borji, Dicky N. Sihite, Laurent Itti, Salient object detection: a benchmark, IEEE Trans. Image Process. 24 (12) (2015) 5706–5722.
[65] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell, CaFfe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM International Conference on Multimedia, ACM, 2014, pp. 675–678.
[66] Xavier Glorot, Yoshua Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.
[67] Razvan Pascanu, Tomas Mikolov, Yoshua Bengio, On the difficulty of training recurrent neural networks, in: International Conference on Machine Learning, 2013, pp. 1310–1318.
[68] Jingdong Wang, Huaizu Jiang, Zejian Yuan, Ming Ming Cheng, Xiaowei Hu, Nanning Zheng, Salient object detection: A discriminative regional feature integration approach, Int. J. Comput. Vis. 123 (2) (2017) 251–268.
[69] Xiaohui Li, Huchuan Lu, Lihe Zhang, Ruan Xiang, Ming Hsuan Yang, Saliency detection via dense and sparse reconstruction, in: ICCV, 2013, pp. 2976–2983.
[70] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, Going deeper with convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.