DHSNet: Deep Hierarchical Saliency Network for Salient Object Detection

Nian Liu Junwei Han*
School of Automation, Northwestern Polytechnical University
Xi'an, 710072, P. R. China
{liunian228, junweihan2010}@gmail.com

Abstract

Traditional salient object detection models often use hand-crafted features to formulate contrast and various prior knowledge, and then combine them artificially. In this work, we propose a novel end-to-end deep hierarchical saliency network (DHSNet) based on convolutional neural networks for detecting salient objects. DHSNet first makes a coarse global prediction by automatically learning various global structured saliency cues, including global contrast, objectness, compactness, and their optimal combination. Then a novel hierarchical recurrent convolutional neural network (HRCNN) is adopted to further hierarchically and progressively refine the details of saliency maps step by step via integrating local context information. The whole architecture works in a global to local and coarse to fine manner. DHSNet is directly trained using whole images and corresponding ground truth saliency masks. When testing, saliency maps can be generated by directly and efficiently feedforwarding testing images through the network, without relying on any other techniques. Evaluations on four benchmark datasets and comparisons with other 11 state-of-the-art algorithms demonstrate that DHSNet not only shows its significant superiority in terms of performance, but also achieves a real-time speed of 23 FPS on modern GPUs.

1. Introduction

Salient object detection aims at accurately and uniformly detecting objects that grab human attention in images. In recent years, researchers have developed many computational models for salient object detection and applied them to benefit many other applications, such as image summarization [1], segmentation [2], retrieval [3], and editing [4].

Traditional saliency detection methods rely on various saliency cues. The most widely explored one is contrast, which aims at evaluating the distinctiveness of each image region or image pixel with respect to local contexts or global ones. Local contrast based methods [5, 6] typically tend to highlight object boundaries while often miss object interiors. On the contrary, global contrast based methods [7, 8] are capable of highlighting object interiors uniformly. This kind of methods are better, but still unsatisfactory. On one hand, they usually fail to preserve object details. On the other hand, they are often difficult to detect salient objects with large sizes and complex textures, especially when image backgrounds are also cluttered or have similar appearances with foreground objects (see column (b) in Figure 1). Furthermore, conventional methods usually model contrast via hand-crafted features (e.g., intensity, color, and edge orientation [5]) and human designed mechanisms (e.g., the "Difference of Gaussians" (DoG) operator [5]) based on limited human knowledge on visual attention. Thus they may not generalize well in different scenarios.

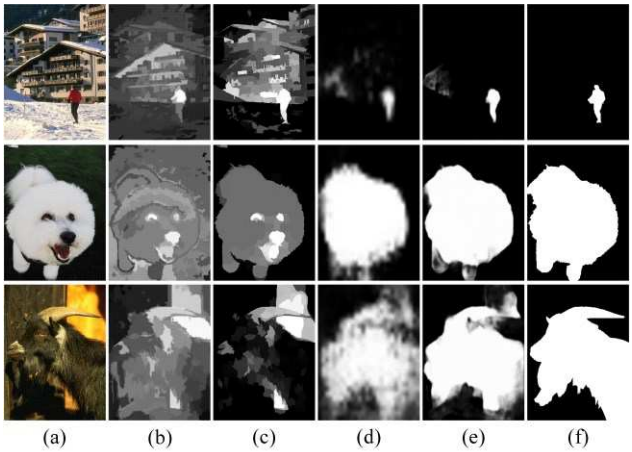

Figure 1: Comparison of results by different kinds of methods. For images in (a), we show the salient object detection results of a global contrast based method in (b), a background prior based method in (c), the results of the GV-CNN in (d), the final refined results of DHSNet in (e), and the ground truth in (f).

Some recent works also utilize various prior knowledge as informative saliency cues. Background prior [9-11] hypothesizes that regions near image boundaries are probably backgrounds. However, it often fails when salient objects touch image boundaries or have similar appearance

* Corresponding author.

with backgrounds (see column (c) in Figure 1). Compactness prior [12] advocates that salient object regions are compact and perceptually homogeneous elements. Objectness prior [13, 14] tends to highlight an image region which is likely to contain an object of a certain class. Although these priors can further provide informative information for salient object detection, they are usually explored empirically and modelled by hand-designed formulations.

Various saliency cues are also combined in some works to incorporate their complimentary interactions. Nevertheless, these works usually resort to simple combination schemes (e.g., simple arithmetic) or shallow learning models (e.g., CRF used in [15]), which are hard to mine complicated joint interactions between diverse saliency cues. Moreover, to preserve object details and subtle structures, many traditional methods adopt over-segmentations of images (e.g., superpixels used in [9-11, 16-19] and object proposals used in [14]) either as the basic computational units to predict saliency or as the post-processing methods to smooth saliency maps. Although these methods can further improve saliency detection results, they are usually very time-consuming, becoming the bottleneck of the computational efficiency of a salient object detection algorithm.

From the discussions above, we can see that, how to build real meaningful feature representations, how to simultaneously explore all potential saliency cues, how to find the optimal integration strategy, and how to efficiently preserve object details become the most intrinsic problems for further promoting salient object detection methods.

To solve these problems, we propose a novel end-to-end deep hierarchical saliency detection framework, i.e., the DHSNet, via convolutional neural networks (CNN) [20]. DHSNet takes the whole images as the inputs and outputs saliency maps directly, hierarchically detecting salient objects from the global view to local contexts, from coarse scale to fine scales (see Figure 2). In details, we first adopt a CNN over the global view (GV-CNN) to generate a coarse global saliency map ($\mathbf{Sm}^G$) to roughly detect and localize salient objects. With the supervision of the global structured loss, the GV-CNN can automatically learn feature representations and various global structured saliency cues, such as global contrast, objectness, compactness, and their optimal combination. Consequently, the GV-CNN can obtain optimal global salient object detection results, being robust to complex foreground objects and cluttered backgrounds, even if they are very similar in appearance (see column (d) in Figure 1).

The generated $\mathbf{Sm}^G$ is much coarser than the input image since some detailed information, such as accurate object boundaries and subtle structures, are gradually discarded in the GV-CNN. To address this problem, we further propose to adopt a novel hierarchical recurrent convolutional neural network (HRCNN) to refine saliency maps in details by incorporating local contexts. The HRCNN is composed of several recurrent convolutional layers (RCL) [21] and upsampling layers (see Figure 2). RCLs incorporate recurrent connections into each convolutional layer, thus enhancing the capability of the model to integrate context information, which is very important for saliency detection models. In HRCNN, we refine the saliency map in several steps hierarchically and successively. In each step, we adopt a RCL to generate a finer saliency map by integrating the upsampled coarse saliency map predicted at the last step and the finer feature maps from the GV-CNN. The RCL in each step boosts the details for the former step, and provides a good initialization for the next step. As the scales of intermediate saliency maps become finer and finer and the receptive fields of the combined feature maps become smaller and smaller, the image details can be rendered step by step, without relying on image over-segmentations (see the final results in column (e) in Figure 1).

The contributions of this paper can be summarized as follows:

(1) We propose a novel end-to-end saliency detection model, i.e., the DHSNet, to detect salient objects. DHSNet can simultaneously learn powerful feature representations, informative saliency cues (for instance, global contrast, objectness, and compactness), and their optimal combination mechanisms from the global view, and subsequently learn to further refine saliency map details.

(2) We propose a novel hierarchical refinement model, i.e., the HRCNN, which can hierarchically and progressively refine saliency maps to recover image details by integrating local context information without using over-segmentation methods. The proposed HRCNN can significantly and efficiently improve saliency detection performance. Furthermore, it can also be used in other pixel-to-pixel tasks, such as scene labeling [22], semantic segmentation [23], depth estimation [24] and so on.

(3) Experimental results on four benchmark datasets and comparisons with other 11 state-of-the-art approaches demonstrate the great superiority of DHSNet on the salient object detection problem, especially on complex datasets. Furthermore, DHSNet is very fast on modern GPUs, achieving a real-time speed of 23 FPS.

2. Related Works

2.1. Convolutional Neural Networks

Recently, CNNs have achieved great successes in many computer vision tasks, including image classification [25, 26], object detection and localization [27, 28], face recognition [29] and so on. CNNs have also been

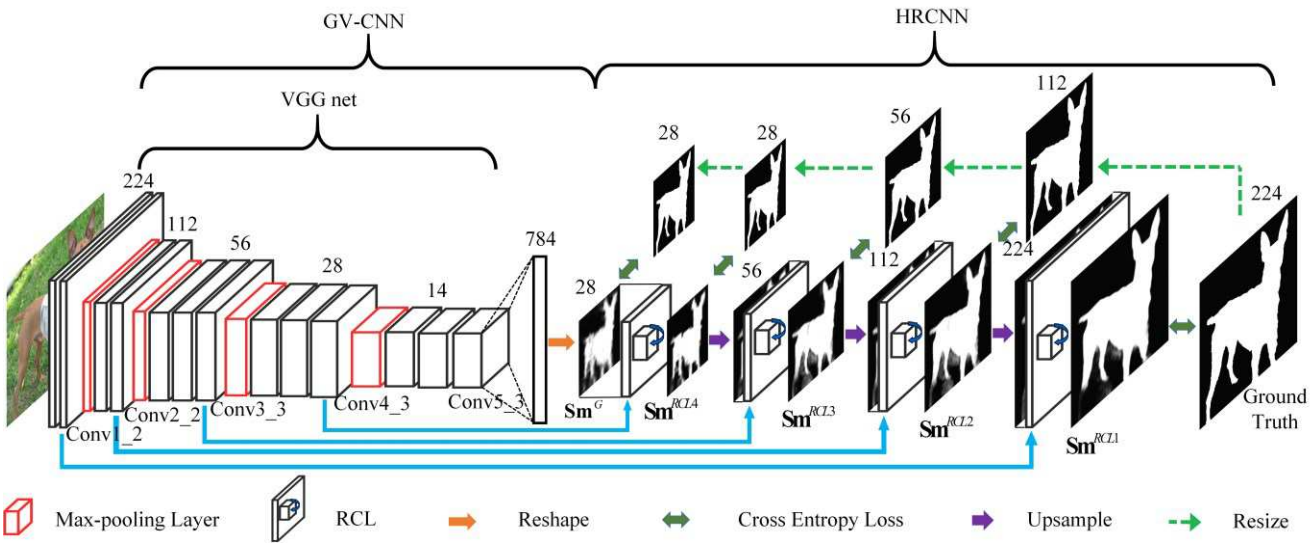

Figure 2: The architecture of the proposed DHSNet method. The spatial size of each image or feature map is given. In the VGG net, the names of the layers whose features are utilized in the HRCNN are shown. The name of each step-wise saliency map is also shown.

successfully applied in many pixel-wise prediction tasks [22-24]. Here we briefly review several works related to this paper.

Many works adopt diverse deep architectures to preserve details in pixel-wise tasks. For depth map prediction, Eigen *et al.* [24] first trained a CNN for making a coarse global prediction based on the entire image, then another CNN was used to refine this prediction locally. For semantic segmentation, [30] utilized deconvolutional layers and unpooling layers to gradually enlarge the resolutions of feature maps to predict a fine semantic segmentation results. Similarly, [31] utilized several "upconvolutional" layers which consisted of deconvolutional layers and unpooling layers to refine the optical flow predictions layer by layer. These two works share similar ideas of gradually refining feature maps or prediction results from coarse to fine with our model. However, the unpooling layers adopted in their models selectively transferred information from a coarser layer to a finer layer yet limitted the transferred information. Besides, their heavy decoder architectures introduced lots of parameters to learn and made their network hard to train. Last but not least, we embedded RCLs [21] in each refinement step, thus enhancing the capability of the model to integrate context information with limited parameters.

2.2. Convolutional Neural Networks for Saliency Detection

Some researchers have already applied deep neural networks to saliency detection, which includes two branches, i.e., eye fixation prediction [32, 33] and salient object detection [34-36]. Here we briefly review the works about the latter which are related to our work.

For salient object detection, Wang *et al.* [35] used a CNN to predict saliency score for each pixel in local context first, then they refined the saliency score for each object proposal over the global view. Li and Yu [34] predicted the saliency score for each superpixel by using multiscale CNN features. Similarly, Zhao *et al.* [36] predicted the saliency score for each superpixel by incorporating local context and global context simultaneously in a multi-context CNN. These three methods all achieved better results than traditional methods. However, none of them considered global context preferentially. Furthermore, they processed local regions (superpixels, and object proposals) separately, thus the correlation of the regions in different spatial locations was not utilized. These two weaknesses make their networks hard to learn enough global structures, thus their results are often distracted by local salient patterns in cluttered backgrounds and are not able to highlight salient objects uniformly. On the contrary, DHSNet adopts the whole image as the computational unit and propagates the global context information to local contexts hierarchically and progressively, being able to perceive global properties and avoid the distraction of local interferences from the beginning. Last but not least, all these three methods relied on image over-segmentations, making their algorithms very time-consuming. While DHSNet only needs to feedforward each testing image through the network, thus is much faster.

3. DHSNet for Salient Object Detection

As shown in Figure 2, DHSNet is composed of the GV-CNN and the HRCNN. The GV-CNN first coarsely detects salient objects in a global perspective, then the HRCNN hierarchically and progressively refines the details of the saliency map step by step. DHSNet is trained end-to-end. When testing we just feedforward the input

image through the network, without using any post-processing and image over-segmentation method, thus making DHSNet not only effective, but also efficient.

3.1. GV-CNN for Coarse Global Prediction

As shown in Figure 2, the GV-CNN consists of 13 convolutional layers of the VGG net [25], a subsequent fully connected layer, and a reshape layer. For an input image wrapped to size 224×224, the 13 convolutional layers of the VGG 16-layer network are first adopted to extract deep features. Afterwards, on top of the last convolutional layer (i.e., the third sublayer in the fifth group of convolutional layers, denoted as Conv5_3. The other convolutional layers in the VGG net can also be denoted by this analogy.) with size 14×14×512, a fully connected layer with sigmoid activation function and 784 nodes is deployed. Finally this layer is reshaped to size 28×28 as the coarse global saliency map $\mathbf{Sm}^G$. Supervised by the global structured loss, i.e., the averaged pixel-wise cross entropy loss between $\mathbf{Sm}^G$ and the ground truth saliency mask, the fully connected layer learns to detect and localize salient objects of the input image from the feature maps ahead by integrating various saliency cues. As [37] pointed out, convnet features can localize at a much finer scale than their receptive field sizes. Thus the GV-CNN can generate a relatively large saliency map (28×28) even though the size of layer Conv5_3 is small (14×14). The experiments in Section 4.5 show the effectiveness of the GV-CNN and its learned saliency cues.

Although the GV-CNN can coarsely detect and localize salient objects, the image details in $\mathbf{Sm}^G$, e.g., object boundaries and subtle structures, are still missing. The reasons are two-folds. First, the 4 max-pooling layers in the VGG net abandon some spatial information, making the layer Conv5_3 hard to reserve local details. Second, the amount of the parameters in the fully connected layer increases linearly with the enlargement of the size of $\mathbf{Sm}^G$, making the training difficult. Thus we have to choose a small size for $\mathbf{Sm}^G$. As a result, $\mathbf{Sm}^G$ is not satisfactory enough, both quantitatively and visually, and further refinements are needed.

3.2. HRCNN for Hierarchical Saliency Map Refinement

To further improve $\mathbf{Sm}^G$ in details, we propose a novel architecture, i.e., the HRCNN, to hierarchically and progressively render image details.

Recurrent Convolutional Layer. The core of the HRCNN is the RCL which was proposed by [21]. RCL incorporates recurrent connections into each convolutional layer. For a unit located at (i, j) on the kth feature map in an RCL, its

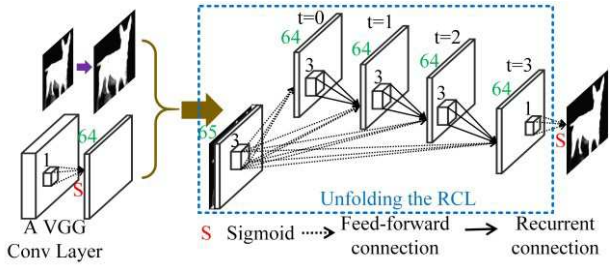

Figure 3: The detailed framework of a refinement step. The RCL is unfolded along with the time steps in the blue dotted box.

state at time step t is given by:

$$x_{ijk}(t) = g(f(z_{ijk}(t))), \tag{1}$$

where f is the ReLU [26] activation function and g is the local response normalization (LRN) function [26] to prevent the states from exploding:

$$g(f_{ijk}(t)) = \frac{f_{ijk}(t)}{\left(1+\frac{\alpha}{N}\sum_{k'=\max(0,k-N/2)}^{\min(K,k+N/2)}(f_{ijk'})^2\right)^{\beta}}, \tag{2}$$

where $f(z_{ijk}(t))$ is abbreviated as $f_{ijk}(t)$, K is the total number of feature maps, N is the size of the local neighbor feature maps which are involved in the normalization, α and β are constants to modulate the normalization.

In Eq. (1), $z_{ijk}(t)$ is the input of the unit, which incorporates a feedforward connection and a recurrent connection:

$$z_{ijk}(t) = (\mathbf{w}_k^f)^T \mathbf{u}^{(i,j)} + (\mathbf{w}_k^r)^T \mathbf{x}^{(i,j)}(t-1) + b_k. \tag{3}$$

where $\mathbf{u}^{(i,j)}$ and $\mathbf{x}^{(i,j)}(t-1)$ are the feedforward input from the previous layer and the recurrent input from the current layer at time step t-1, respectively. $\mathbf{w}_k^f$ and $\mathbf{w}_k^r$ are the feedforward weights and the recurrent weights, respectively. b_k is the bias.

A RCL with T time steps can be unfolded to a feed-forward subnetwork of depth $T+1$. We follow [21] to set $T=3$ and show the unfolded RCL in the blue dotted box in Figure 3. We can see that multiple recurrent connections make the subnetwork has multiple paths from the input layer to the output layer, which facilitates the learning. Besides, the effective receptive field of an RCL unit expands when the time step increases, making the units to be able to "see" larger and larger contexts without increasing the number of network parameters. Thus RCLs can help to incorporate local contexts efficiently in HRCNN to refine saliency maps. The experiment in Section 4.5 demonstrates the superiority of RCLs over traditional convolutional layers.

As shown in Figure 3, we use 64 feature maps in each RCL empirically to save computational costs and follow [21]

to use feed-forward and recurrent filters with size 3×3. The hyper-parameters of LRN in Eq. (2) are set as $\alpha = 0.001$, $\beta = 0.75$ and $N = 7$. Different from [21], we do not adopt dropout [21] in RCLs.

Hierarchical Saliency Map Refinement. As shown in Figure 2, we first combine $\mathbf{Sm}^{G}$ with layer Conv4_3 of the VGG net and adopt a RCL to generate a finer saliency map (as this saliency map is obtained by adopting a RCL over the local features in Conv4_3, we denote it as $\mathbf{Sm}^{RCL4}$ and the subsequent further refined saliency maps are denoted in the same way). As $\mathbf{Sm}^{RCL4}$ has a smaller size (28×28) compared with Conv3_3 (56×56), we first upsample $\mathbf{Sm}^{RCL4}$ to double its size, then we combine the upsampled $\mathbf{Sm}^{RCL4}$ with layer Conv3_3 to generate $\mathbf{Sm}^{RCL3}$. By doing the same thing, we combine the upsampled $\mathbf{Sm}^{RCL3}$ with layer Conv2_2 to generate $\mathbf{Sm}^{RCL2}$, and combine the upsampled $\mathbf{Sm}^{RCL2}$ with layer Conv1_2 to generate $\mathbf{Sm}^{RCL1}$, which is the final saliency map.

In Figure 3, we show the detailed framework of a refinement step, i.e., combining a coarse saliency map with a convolutional layer from the VGG net to generate a finer saliency map. We first use a convolutional layer with 64 1×1 convolutional kernels and sigmoid activation function to squash the features of the VGG layer. The reasons are two folds. First, we decrease the number of feature maps of the VGG layer to save computational costs. Second, by using sigmoid activation function, we squash the range of the activation values of the neurons to be [0,1], which is as the same as the combined saliency map. Without doing this, the combined saliency map will be overwhelmed since the activation values in each layer of the VGG net are usually very large with ReLU activation functions.

Next, the squashed VGG layer is concatenated with the upsampled coarse saliency map (except that $\mathbf{Sm}^{G}$ is directly concatenated with layer Conv4_3 without upsampling), resulting in 65 feature maps. Then we adopt a RCL to combine the coarse saliency map and the local features in the VGG layer. At last, the refined saliency map can be generated by adopting a convolutional layer with 1×1 kernels and sigmoid activation function.

Table 1 shows the size of each step-wise saliency map and the sizes of the receptive fields from which they are induced. From $\mathbf{Sm}^{G}$ to $\mathbf{Sm}^{RCL1}$, the sizes of step-wise saliency maps are gradually enlarged but the receptive fields gradually shrink (note that the receptive field of $\mathbf{Sm}^{G}$ is the whole image). Thus HRCNN refines the saliency maps in a coarse to fine and global to local manner. Consequently, image details can be rendered step by step by incorporating finer and finer features. The experiments in Section 4.5 also verify the effectiveness of the hierarchical refinement scheme.

	$\mathbf{Sm}^{G}$	$\mathbf{Sm}^{RCL4}$	$\mathbf{Sm}^{RCL3}$	$\mathbf{Sm}^{RCL2}$	$\mathbf{Sm}^{RCL1}$
Size	28	28	56	112	224
RF	224	156	72	30	13

Table 1: The sizes of the step-wise saliency maps and the receptive fields (RF) from which they are induced.

To facilitate learning, we also adopt the deep supervision [38] scheme. To be specific, as shown in Figure 2, we resize the ground truth saliency mask to sizes ranging from 224 to 28 to supervise the corresponding learning of each step-wise saliency map.

4. Experiments

4.1. Datasets

We conducted evaluations on four widely used salient object benchmark datasets. **ECSSD** [18] includes 1,000 semantically meaningful but complex images. **MSRA10K** [7] contains 10,000 images with various objects. Most images contain only one salient object and the backgrounds are usually clear. **DUT-OMRON** [10] includes 5,168 images with one or more salient objects and relatively complex backgrounds. **PASCAL-S** [14] contains 850 real-world images selected from the PASCAL VOC segmentation dataset. Many images in this dataset have highly cluttered backgrounds and multiple complex foreground objects.

4.2. Evaluation metrics

We first adopted the precision-recall (PR) curve to evaluate DHSNet. Specifically, saliency maps were binarized using different thresholds varying from 0 to 1. Then compared with the ground truth, a series of precision-recall values can be obtained at different thresholds to plot the PR curve. We also adopted the F-measure [8] score to comprehensively consider precision and recall. By using an image adaptive threshold, we can obtain *Precision* and *Recall*, then F-measure is given by:

$$F_{\beta} = \frac{(1+\beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}. \quad (4)$$

where β^2 is set to 0.3 and the adaptive threshold is set to twice the mean saliency value of each saliency map as suggested in [8].

4.3. Implementation details

We randomly selected 6,000 images from MSRA10K dataset and 3,500 images from DUT-OMRON dataset as the training set, and another 800 images from MSRA10K and 468 images from DUT-OMRON as the validation set.

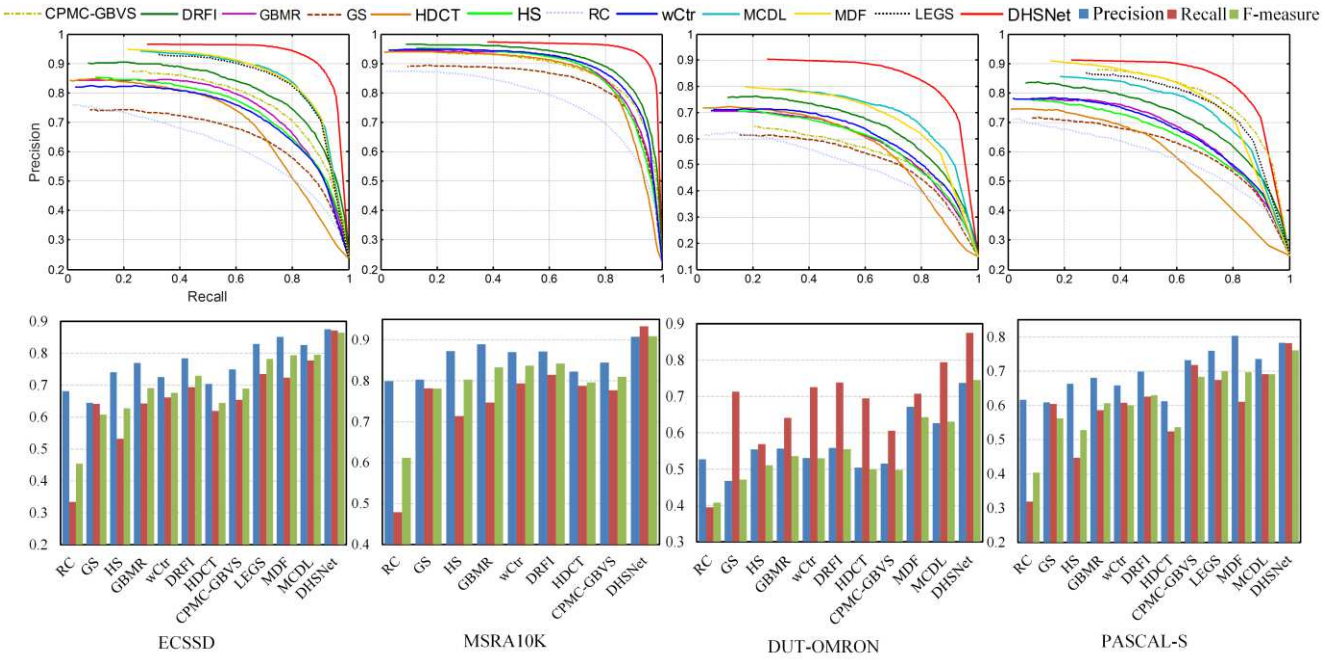

Figure 4: Quantitative model comparisons. We show PR curves (top) and F-measure scores (bottom) on 4 benchmark datasets.

Then we tested our model on the rest images and other datasets.

During training, we did image augmentation by horizontal-flipping and image cropping to relieve overfitting. In details, for each training image and the corresponding ground truth, we cropped out the most top, bottom, left, right, and middle 9/10 image as training samples. In addition to the original images and the horizontally-flipped ones, we increased the training set by 12 times. When fed into DHSNet, each image was first wrapped to size 224×224 and subtracted a mean pixel provided by VGG net at each position.

With the RCLs unfolded through time steps, the whole network was trained end-to-end by using back propagation algorithm [39]. The upsampling layers were simply implemented using the nearest-neighbor interpolation method. To facilitate training, we first trained the GV-CNN alone with the 13 convolutional layers initialized by the VGG net and the fully connected layer randomly initialized. We used a minibatch size of 12 and 40,000 iteration steps, and set the learning rate to 0.015 in the last layer and a 1/10 smaller one in the VGG layers. We also halved the learning rate every 4,000 iterations. Besides, we set momentum to 0.9 and weight decay factor to 0.0005. Then we trained the whole DHSNet with the GV-CNN part initialized by the pretrained model and the HRCNN part initialized randomly. Here we set the minibatch size to 5 and kept the 40,000 iteration steps, and set the learning rate to 0.03 in the HRCNN part and a 1/1000 smaller one in the GV-CNN layers. The learning rate decay policy, the momentum and the weight decay factor were kept as the same as those when training GV-CNN. We tested the cross entropy loss on the validation set every 2000 iteration steps and selected a model with the lowest validation loss as the best model to do testing.

We implemented DHSNet using caffe [40] toolbox. The testing codes were implemented using Matlab. A GTX Titan X GPU was used both in training and testing for acceleration.

4.4. Results

We compared DHSNet with other 11 state-of-the-art models, including RC [7], GS [11], HS [18], GBMR [10], DRFI [19], wCtr [41], HDCT [17], CPMC-GBVS [14], LEGS [35], MDF [34] and MCDL [36][2].

For quantitative evaluation, we show comparison results with PR curves and F-measure scores in Figure 4. We can see that, DHSNet outperforms all other methods by a large margin, especially on complex datasets, i.e., ECSSD, DUT-OMRON, and PASCAL-S. In terms of PR curves, as shown in the top row in Figure 4, DHSNet achieves much higher curves than all other methods from the beginning to the end on all the 4 datasets, indicating that DHSNet can achieve both the highest precision and the highest recall. From the bottom row in Figure 4, we can see that DHSNet almost achieves all of the highest precision, recall and F-measure scores under the adaptive threshold on all the 4

[2] LEGS didn't publish their results on DUT-OMRON. MDF and LEGS were trained on the MSRA-B dataset [15], which is covered by MSRA10K, and MCDL was trained on MSRA10K. Thus we didn't compare our model with these three models on this dataset.

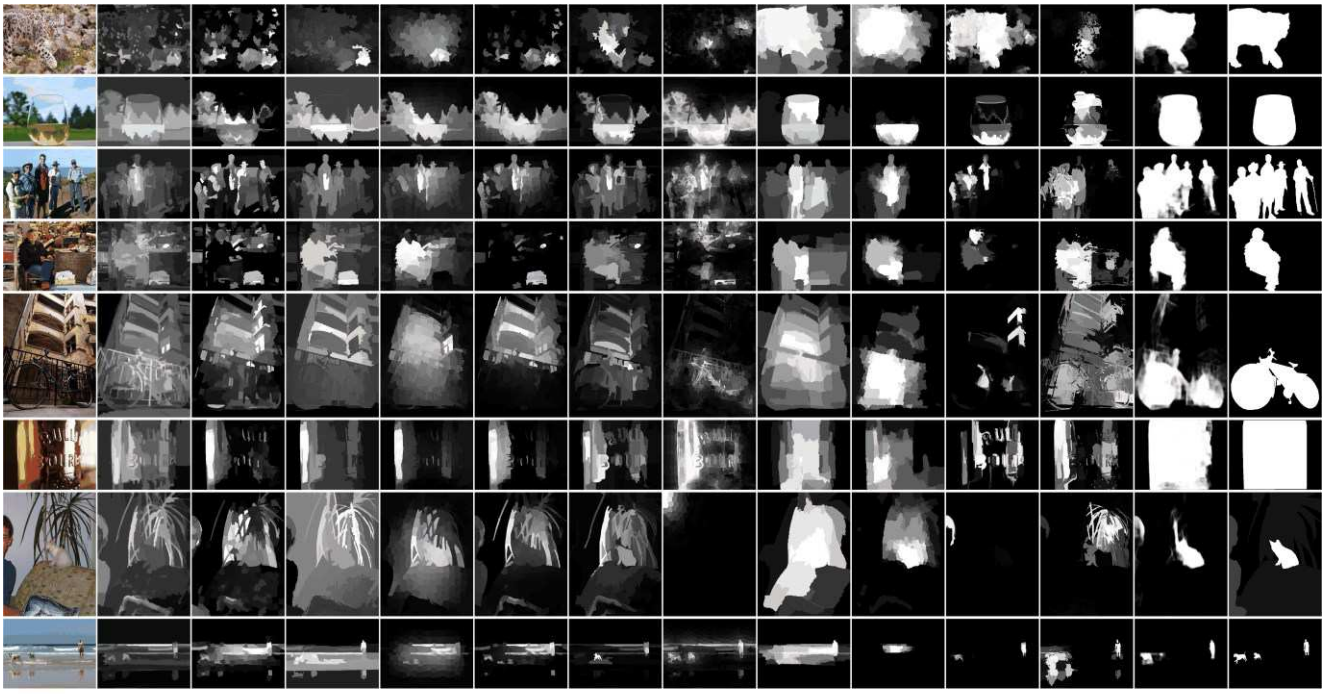

Figure 5: Qualitative model comparisons. The ground truth (GT) is shown in the last column.

	RC	GS	HS	GBMR	wCtr	DRFI	HDCT	CPMC -GBVS	LEGS	MDF	MCDL	DHSNet
Time(s)	0.25	0.18	0.43	0.87	0.52	47.08	1.54	36.30	2.00*	8.00*	2.38*	0.04*

Table 2: Runtime of each method. *: GPU time.

datasets, except that on PASCAL-S, the precision of MDF is slightly better than DHSNet, however its recall and F-measure score are much lower than those of DHSNet. It is worth noting that DHSNet also outperforms the other three CNN-based methods (i.e., LEGS, MDF, and MCDL) a lot. The improvement DHSNet achieved with respect to these three methods is almost equivalent to their improvement with respect to traditional approaches, demonstrating the superiority of DHSNet.

We showed visual comparison in Figure 5. As we can see, DHSNet not only detects and localizes salient objects accurately, but also preserves object details subtly. It can handle various situations well, including foreground objects being very big (row 1 and 6 in Figure 5) or very small (row 8), images with multiple foreground objects (row 3 and 8), cluttered backgrounds and complex foregrounds (row 4, 5 and 7), and salient objects touching image boundaries (row 1, 2 and 6). Especially in row 1, 2 and 5, the foreground objects have similar appearance with backgrounds, which confuses most other methods, while DHSNet works well. We can also see that LEGS, MDF, and MCDL often are distracted by local salient patterns in cluttered backgrounds and are not able to highlight salient objects uniformly. In contrast, DHSNet can overcome these difficulties well.

We also evaluated the runtime of each method in Table 2. These evaluations were conducted on a machine with 2 2.8GHz 6-core CPUs and 32GB memory. LEGS, MDF, MCDL, and DHSNet were accelerated by a GTX Titan X GPU. We can see that DHSNet is the fastest, achieving a real-time speed of 23 FPS.

4.5. Model Component Analysis

The effectiveness of the hierarchical refinement scheme of HRCNN. We show the F-measure scores and the corresponding precision and recall of the step-wise saliency maps in Table 3. We can see that, from $\mathbf{Sm}^{G}$ to $\mathbf{Sm}^{RCL1}$, all the three metrics, i.e., precision, recall and F-measure are progressively enhanced, except that the recall saturates and fluctuates slightly after $\mathbf{Sm}^{RCL3}$. Qualitative results are shown in Figure 6. We can see that the hierarchical refinement scheme can progressively improve the details of saliency maps, not only eliminating false positive highlights,

Settings	Precision	Recall	F-measure
step-wise results of DHSNet			
$\mathbf{Sm}^{G}$	0.8063	0.8352	0.7941
$\mathbf{Sm}^{RCL4}$	0.8154	0.8451	0.8074
$\mathbf{Sm}^{RCL3}$	0.8350	0.8710	0.8277
$\mathbf{Sm}^{RCL2}$	0.8496	**0.8757**	0.8425
$\mathbf{Sm}^{RCL1}$	**0.8753**	0.8720	**0.8645**
DHSNet with the GV-CNN substituted by a FCN			
FCN $\mathbf{Sm}^{G}$	0.8067	0.8300	0.7923
FCN $\mathbf{Sm}^{RCL1}$	0.8615	0.8682	0.8516
DHSNet with RCLs substituted by traditional convolutional layers			
DHSNet(w/o RCLs)	0.8622	0.8685	0.8516
Comparison with other encoder-decoder networks			
DeconvNet [30]	0.8493	0.8661	0.8396

Table 3: The results for component analysis on ECSSD dataset. The best results are shown in bold face.

but also redetecting missing parts (e.g., the arm of the man in the top row).

The effectiveness of the GV-CNN. In figure 7, we show some saliency maps from $\mathbf{Sm}^{G}$ as intuitive examples of the saliency cues learned in GV-CNN. $\mathbf{Sm}^{G}$ in (a) just highlights the dog despite of the small-scale high-contrast patterns in the background (i.e., the flowers), which indicates that GV-CNN learned global contrast. In (b), although the squirrel has similar appearance with the background, GV-CNN can still detect it accurately due to the learned objectness. In (c), the successful detection of all the sailing ships indicates that GV-CNN is robust to scales and locations of foreground objects. Furthermore, we can see that the highlighted regions are basically compact and homogeneous, demonstrating that GV-CNN also follows the compactness prior well.

To further verify the effectiveness of GV-CNN quantitatively, we also adopted a fully connected network (FCN) [23] to substitute GV-CNN in DHSNet. The utilized FCN was implemented by adopting the "hole algorithm" [42] to the Conv5 layers and using a convolutional layer with 1 3×3 sized kernel and sigmoid activation function to generate the coarse saliency map with size 28×28 at last. The receptive field size of the units in the output layer is 228. However, only the pixels around the image center can perceive the whole image, and other pixels can only "watch" part of the image, which is the limitation of the FCN. As a result, although the performance of the coarse prediction the FCN made (denoted as FCN $\mathbf{Sm}^{G}$ in Table 3) approximates that of $\mathbf{Sm}^{G}$, the final refined results of FCN (FCN $\mathbf{Sm}^{RCL1}$ in Table 3) are still worse than those of

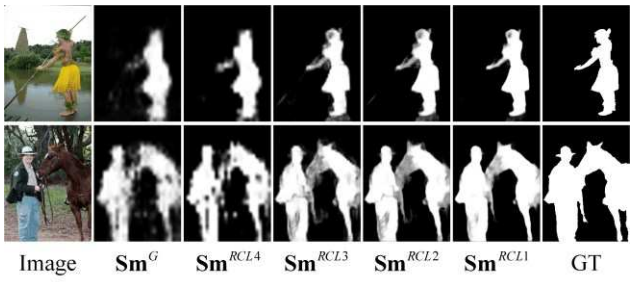

Figure 6: Visualization of the step-wise saliency maps.

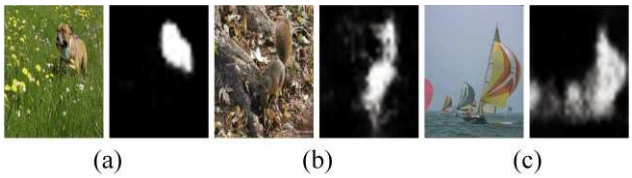

Figure 7: Intuitive examples of the saliency cues learned in GV-CNN.

DHSNet with GV-CNN, which is probably because the incorrect detections in FCN $\mathbf{Sm}^{G}$ are magnified by HRCNN.

The effectiveness of the RCLs. To demonstrate the effectiveness of the deployed RCLs, we followed [21] to substitute RCLs with traditional convolutional layers with more feature maps (128 feature maps were used to keep the number of parameters in each refinement step approximately unchanged). The results in Table 3 show that when RCLs are substituted, all the precision, recall and F-measure will drop, which demonstrates the superiority of adopting RCLs in the salient object detection problem.

Comparison with other encoder-decoder networks. We also compared the proposed DHSNet with the DeconvNet [30]. The results in the last row of Table 3 demonstrate the superiority of DHSNet over traditional encoder-decoder networks on the salient object detection task. Further analysis will be done in our future work.

5. Conclusions

In this paper, we proposed DHSNet as a novel end-to-end salient object detection model. It first detected salient objects coarsely from a global view, then hierarchically and progressively improve image details by integrating local contexts. DHSNet not only obtained state-of-the-art results, but also achieves real-time speed.

Acknowledgements: This work was supported in part by the National Science Foundation of China under Grants 61522207 and 61473231.

References

[1] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. in *CVPR*, 2008.

[2] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof. Saliency driven total variation segmentation. in *ICCV*, 2009.

[3] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu. Visual-textual joint relevance learning for tag-based social image search. *TIP,* 22(1): 363-376, 2013.

[4] C. Rother, L. Bordeaux, Y. Hamadi, and A. Blake. Autocollage. *ACM Trans. Graphics (TOG),* 25(3): 847-852, 2006.

[5] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI,* 20(11): 1254-1259, 1998.

[6] D. A. Klein and S. Frintrop. Center-surround divergence of feature statistics for salient object detection. in *ICCV*, 2011.

[7] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu. Global contrast based salient region detection. in *CVPR*, 2011.

[8] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. in *CVPR*, 2009.

[9] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu. Background Prior-Based Salient Object Detection via Deep Reconstruction Residual. *IEEE Trans. CSVT,* 25(8): 1309-1321, 2015.

[10] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. in *CVPR*, 2013.

[11] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors. in *ECCV*, 2012.

[12] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. in *CVPR*, 2012.

[13] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai. Fusing generic objectness and visual saliency for salient object detection. in *ICCV*, 2011.

[14] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. in *CVPR*, 2014.

[15] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *PAMI,* 33(2): 353-367, 2011.

[16] C. Gong, D. Tao, W. Liu, S. J. Maybank, M. Fang, K. Fu, and J. Yang. Saliency Propagation from Simple to Difficult. in *CVPR*, 2015.

[17] J. Kim, D. Han, Y.-W. Tai, and J. Kim. Salient region detection via high-dimensional color transform. in *CVPR*, 2014.

[18] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. in *CVPR*, 2013.

[19] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. in *CVPR*, 2013.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE,* 86(11): 2278-2324, 1998.

[21] M. Liang and X. Hu. Recurrent Convolutional Neural Network for Object Recognition. in *CVPR*, 2015.

[22] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *PAMI,* 35(8): 1915-1929, 2013.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. in *CVPR*, 2015.

[24] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. in *NIPS*, 2014.

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. in *ICLR*, 2015.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. in *NIPS*, 2012.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. in *ECCV*, 2014.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. in *CVPR*, 2014.

[29] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. in *CVPR*, 2014.

[30] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. in *ICCV*, 2015.

[31] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. in *ICCV*, 2015.

[32] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu. Predicting eye fixations using convolutional neural networks. in *CVPR*, 2015.

[33] M. Kümmerer, L. Theis, and M. Bethge. Deep Gaze I: Boosting saliency prediction with feature maps trained on ImageNet. in *ICLR Workshop*, 2015.

[34] G. Li and Y. Yu. Visual Saliency Based on Multiscale Deep Features. in *CVPR*, 2015.

[35] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep Networks for Saliency Detection via Local Estimation and Global Search. in *CVPR*, 2015.

[36] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. in *CVPR*, 2015.

[37] J. L. Long, N. Zhang, and T. Darrell. Do Convnets Learn Correspondence? in *NIPS*, 2014.

[38] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. in *AISTATS*, 2015.

[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature,* 323(6088): 533-538, 1986.

[40] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. in *ACM Multimedia*, 2014.

[41] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. in *CVPR*, 2014.

[42] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. in *ICLR*, 2015.