

# DISC: Deep Image Saliency Computing via Progressive Representation Learning

Tianshui Chen, Liang Lin, Lingbo Liu, Xiaonan Luo, and Xuelong Li

**Abstract**—Salient object detection increasingly receives attention as an important component or step in several pattern recognition and image processing tasks. Although a variety of powerful saliency models have been intensively proposed, they usually involve heavy feature (or model) engineering based on priors (or assumptions) about the properties of objects and backgrounds. Inspired by the effectiveness of recently developed feature learning, we provide a novel deep image saliency computing (DISC) framework for fine-grained image saliency computing. In particular, we model the image saliency from both the coarse- and fine-level observations, and utilize the deep convolutional neural network (CNN) to learn the saliency representation in a progressive manner. In particular, our saliency model is built upon two stacked CNNs. The first CNN generates a coarse-level saliency map by taking the overall image as the input, roughly identifying saliency regions in the global context. Furthermore, we integrate superpixel-based local context information in the first CNN to refine the coarse-level saliency map. Guided by the coarse saliency map, the second CNN focuses on the local context to produce fine-grained and accurate saliency map while preserving object details. For a testing image, the two CNNs collaboratively conduct the saliency computing in one shot. Our DISC framework is capable of uniformly highlighting the objects of interest from complex background while preserving well object details. Extensive experiments on several standard benchmarks suggest that DISC outperforms other state-of-the-art methods and it also generalizes well across data sets without additional training. The executable version of DISC is available online: <http://vision.sysu.edu.cn/projects/DISC>.

**Index Terms**—Convolutional neural network (CNN), image labeling, representation learning, saliency detection.

## I. INTRODUCTION

AS psychophysical experiments suggested, humans appear to perceive surrounding environment almost effortlessly due to their attentional mechanisms guiding the

Manuscript received November 30, 2014; revised October 20, 2015 and December 5, 2015; accepted December 6, 2015. Date of publication January 5, 2016; date of current version May 16, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61320106008 and Grant 61232011, in part by the Guangdong Natural Science Foundation under Grant S2013050014548, and in part by the Guangdong Science and Technology Program under Grant 2013B010406005 and Grant 2015B010128009. (Corresponding author: Liang Lin.)

T. Chen, L. Lin, L. Liu, and X. Luo are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China (e-mail: tianshuichen@gmail.com; linliang@ieee.org; liulingbo918@qq.com; lnsln@mail.sysu.edu.cn).

X. Li is with the Center for OPTical Imagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China. (e-mail: xuelong\_li@ieee.org).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org> (File size: 2 MB).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNNLS.2015.2506664

gaze to salient and informative locations in the visual field. Mimicking such a visual saliency system is a long-standing research topic both in neuroscience [1] and in computer vision. Recently, instead of predicting sparse human eye fixation, many studies in computer vision focus on detecting the most informative and attention-grabbing regions (i.e., salient objects) in a scene [2]. These proposed salient object detection methods [3]–[6] also evolve to target on uniformly highlighting pixel-accurate saliency values, which is the aim of this paper. In addition, salient object detection has the great potential to benefit a wide range of applications ranging from image/video compressing [7] and editing [8] to object segmentation and recognition [9].

Due to the lack of a rigorous definition of image saliency, inferring the accurate saliency assignment for diversified natural images without a task orientation is a highly ill-posed problem. Therefore, many works of image saliency detection usually rely on various priors (or assumptions) for defining their saliency representations. Fig. 1 shows some examples. The contrast prior is arguably the most popular one, which can be further categorized as local contrast and global contrast according to the context where the contrast is computed. Local contrast-based methods [10], [11] exploit a pixel/patch difference in the vicinity to compute the image saliency. Without considering the global information, these methods, however, often miss the interior content while emphasizing the boundaries of salient objects, as shown in Fig. 1(d). In contrast, global contrast-based methods [12], [13] take the whole image as input to estimate the saliency of every pixel or image patch. Some results generated by these methods are shown in Fig. 1(e). Though the entire salient objects are generally highlighted, the object structure details may not be well preserved. The compactness prior [14] is also widely utilized in image saliency modeling, which suggests the elements of salient objects tend to be compactly grouped in the image domain. This prior is shown to better capture object details compared with the global contrast, but may fail to highlight the object uniformly, as the examples shown in Fig. 1(f). The background prior [4], similar with the compactness, tends to render high saliency values to the regions near the center of the image. This prior, however, may lead to unreliable results on detecting salient regions that have similar appearance to the background, as shown in Fig. 1(g). Some studies used a combination of different priors (e.g., the compactness and local contrast priors) to improve the performance, as shown in Fig. 1(h).

Furthermore, the priors for modeling image saliency can be derived from higher level knowledge. For example,

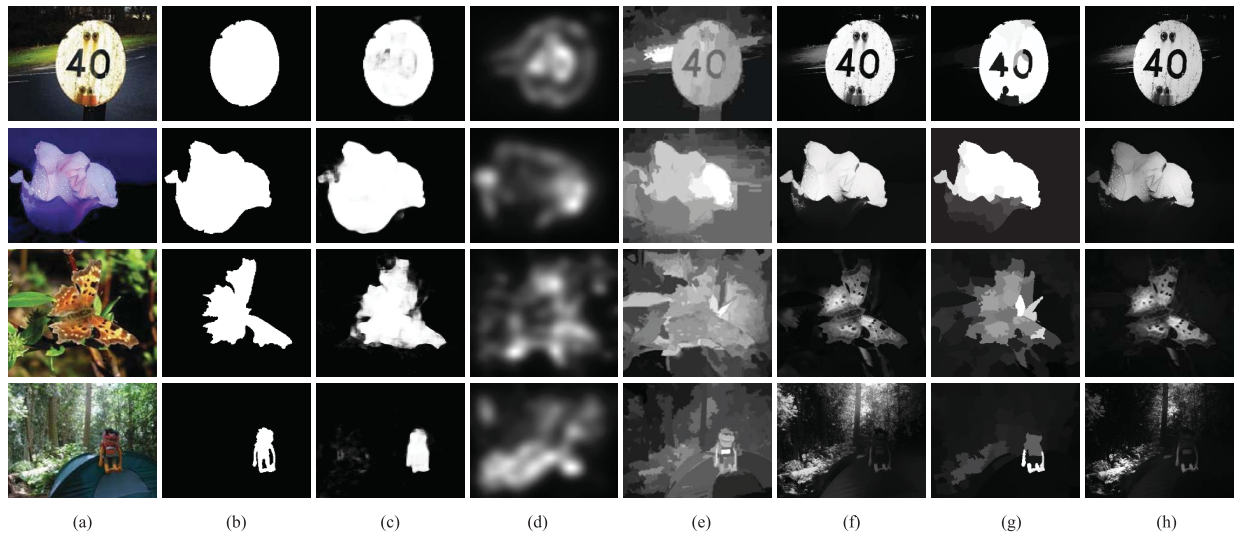


Fig. 1. Comparison of image saliency maps generated by several state-of-the-art approaches based on different assumptions, including local/global contrast, compactness, background priors, and the combination of them. In contrast, the proposed method is capable to produce better fine-grained saliency maps without any assumption or feature engineering. (a) Image. (b) Ground truth. (c) DISC. (d) Local contrast. (e) Global contrast. (f) Compactness. (g) Background. (h) Compactness and local contrast.

the human perception suggests that red colored objects are more pronounced as more than 50% of the cones in human eyes are red sensitive [15]. Some semantic information (e.g., object categories and attributes) has also been explored for completing the saliency models. The integration of these higher level priors, nevertheless, is usually *ad hoc*, and one of the common ways is to calculate the weighted average of the saliency maps generated from high-level cues and those from low-level features.

On the other side, several learning-based approaches have been discussed for the task-driven image saliency detection. For instance, Liu *et al.* [11] learned a conditional random field (CRF) with supervision to detect salient objects. Jiang *et al.* [16] proposed to learn the saliency maps using a random forest regressor. Though impressive results are achieved, those methods often depend on hand-crafted features, such as contrast histogram and color spatial distribution, and the design of these features basically follows the priors as we discussed above. Besides, most of the mentioned approaches conduct the saliency learning based on the oversegmentation of images (i.e., small regions or superpixels) and, thus, require an additional postrelaxation step (e.g., local filtering) to smooth the saliency values over pixels. The step of image segmentation may introduce errors and degenerate the saliency detection.

Different from those previous works, we present a novel end-to-end framework for fine-grained image saliency computing, and formulate it as a progressive representation learning problem. Our proposed model captures the image saliency in a coarse-to-fine manner, i.e., exploiting saliency cues with global and local contexts, respectively. In particular, the coarse-level image saliency roughly identifies the locations, scales, and rough regions for the salient objects, while the object details (e.g., boundaries and subtle structures) will be rendered by the fine-level saliency map. Instead of

defining two level representations with assumptions or hand-crafted image descriptors, we aim to learn the feature transformation directly from raw image pixels.

Inspired by its outstanding performance on the traditional classification and detection tasks [17], we propose a novel image saliency model named deep image saliency computing (DISC) using convolutional neural network (CNN) in this paper. In particular, our DISC framework is built upon two stacked CNNs to cope with the coarse-to-fine saliency representation learning, where the CNN is treated as a feature extractor and the saliency map is generated by an additional linear transformation. The CNN architecture, designed according to the AlexNet [17], comprises several convolutional layers and fully connected layers. We define the linear transformation in the form of support vector machine (SVM) classification, instead of using a softmax classifier like other CNN-based approaches. The combination of CNNs and SVMs has been discussed in [18], but they trained these two models separately. In contrast, we embed the SVM into the CNN, ensuring joint optimization for these two components.

We briefly introduce the implementation of our approach as follows. The first CNN takes the whole image as input and measures the saliency score of each pixel in a global context, generating a coarse-level saliency map in a lower resolution. However, since the coarse CNN considers the whole image but pays less attention to local context information, it may mistakenly highlight some background regions or lose subtle salient structures. As a result, an incorrect coarse map will adversely affect the subsequent fine-level map generated by the second CNN. To address this issue, we utilize the superpixel-based local context information (SLCI) to further refine the coarse maps, which helps to keep the consistency of a spatial structure of the salient objects. The SLCI comprises two components, namely, intrasuperpixel smoothing (ISS) and intersuperpixel voting (ISV) that make use of local context

information in different scales. We formulate them as two special types of pooling layers and, then, embed them in the first CNN rather than treating them as a postprocessing step. The second CNN is guided by the coarse-level saliency map, measuring the accurate and fine-grained saliency in a local context. In particular, each pixel is fed to the second CNN together with its local observed patches from both the original image and the coarse-level saliency map. To avoid repeat computation, we load the entire image at a time and, thus, make the neighboring pixels share their observations during the learning and inference procedures. Moreover, we introduce a nonparametric map as an additional input channel to both CNNs, implicitly taking the spatial regularization (SR) into account to alleviate overfitting, and show that reasonable performance improvement can be achieved. Intuitively, we also refer the first and second CNNs as the coarse-level and the fine-level CNNs, respectively. The two CNNs are trained separately with supervision, yet they collaboratively conduct the inference for a testing image, i.e., producing the fine-grained saliency map in one shot. Some saliency maps generated by our approach are shown in Fig. 1(c).

This paper makes three main contributions to the community.

- 1) It presents a novel architecture capturing image saliency via progressive representation learning. This model is general to be extended to similar tasks, such as scene parsing.
- 2) SLCI is integrated in the proposed framework for salient object structure preserving. It is formulated as two operations and embedded in the first CNN as ISS and ISV layers.
- 3) Extensive experiments on the standard benchmarks of image saliency detection demonstrate that our proposed method significantly outperforms state-of-the-art approaches and generalizes well across data sets without additional training. We also evaluate carefully each component of our model, and discuss the key components that improve the performance.

The remainder of this paper is organized as follows. Section II presents a review of related work. We then present our DISC framework in Section III, followed by a description of the optimization method in Section IV and a description of domain adaption implementation in Section V. The experimental results, and analysis are exhibited in Section VI. Finally, the conclusion is drawn in Section VII.

## II. RELATED WORK

According to the objective and the technical components of this paper, we review the related work into three pipelines: 1) image saliency detection; 2) deep representation learning; and 3) deep saliency computing.

### A. Image Saliency Detection

Existing work of image saliency detection can be broadly divided into two categories: 1) bottom-up and 2) top-down approaches.

The bottom-up saliency models mainly focus on explaining visual attention according to different mathematical principles

or priors. Among them, various contrast-based methods have been intensively discussed. Itti *et al.* [10] presented the center-surround operators based on the multiscale image segmentation. Zhai and Shah [12] proposed to use image histograms to compute pixel-level saliency map. Achanta *et al.* [13] provided a frequency-tuned method that directly computed pixel saliency by subtracting the average image color. Cheng *et al.* [19] extended image histogram to 3-D color space, and proposed color histogram contrast (HC) and region contrast (RC). Background information is also widely explored for saliency modeling, including the boundary prior and the background-connectivity prior [4]. The compactness prior encourages the salient elements to be grouped tightly, and it was realized in [14] using two Gaussian filters. Moreover, several mathematical models have also been utilized to define the bottom-up saliency models, such as entropy [20] and Shannon's self-information [21]. Shen and Wu [15] proposed to utilize the low-rank representation for saliency detection, which is based on the assumption that the nonsalient background usually lies in a low-dimensional subspace, while the sparse noise indicates the salient regions.

The top-down approaches introduce visual knowledge commonly acquired through supervised learning to detect image saliency. Approaches in this category are highly effective on task-specified saliency detection. For example, Lin and Lin [22] proposed a computational visual saliency model based on feature-prior, position-prior, and feature-distribution, which are learned from support vector regressor, ground truth of training images, and features in the image using information theory, respectively. Mai *et al.* [23] trained a CRF model to aggregate various saliency map produced by different methods. Lu *et al.* [24] proposed a graph-based method to learn optimal seeds for object saliency. They learned the combination of different features that best discriminate between object and background saliency.

### B. Deep Representation Learning

Recently, representation learning via deep CNNs has obtained great success in various computer vision tasks, such as image classification [17], object detection [25], person re-identification [26], [27], and human centric analysis [28], [29]. Many works also apply a multiscale deep network to various computer vision tasks. Farabet *et al.* [30] used a multiscale CNN trained from raw pixels to extract dense feature for assigning a label to each pixel. However, it is time-consuming for multiple complex postprocessing required for accurate prediction. These works utilized multiscale input to realize multiscale representation learning. Instead, some works integrated a multiscale structure inside the CNN. Sermanet *et al.* [31] incorporated multiscale information inside the CNN architecture and utilized unsupervised multistage feature learning for pedestrian detection. It produced features that extract both global structures and local details, and improved the performance by a large margin. However, this novel architecture was difficult to adapt to high-resolution pixelwise labeling. Nowadays, more works tended to design different network architectures for different scale representation learnings. Wang *et al.* [32] designed a localization network

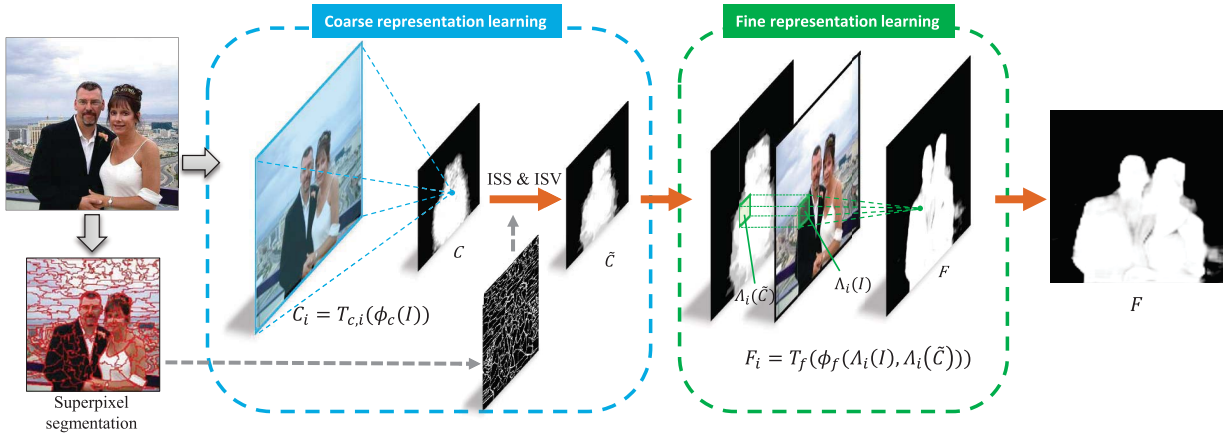


Fig. 2. Illustration of our proposed deep saliency computing model. The first CNN takes the whole image data as input and produces coarse map. Guided by the coarse map, the second CNN takes a local patch as input and generates the fine-grained saliency map.

and a segmentation network to rapidly locate and accurately segment the object from the images. It outperformed previous segmentation work on public benchmark both in accuracy and efficiency. However, it heavily relied on the accuracy of localization, which is not always satisfied, especially for the image with more than one object. Sun *et al.* [33] designed a cascaded regression framework with three level of CNNs for facial landmark detection. Eigen *et al.* [34] proposed a depth prediction model based on two deep CNNs to estimate the depth of each image pixel in a coarse-to-fine manner. The output of the first CNN was concatenated with the output of the first convolutional layer of the second CNN, and they are fed into the subsequent layers for accurate depth prediction.

### C. Deep Learning for Saliency Detection

Very recently, deep neural network models are also utilized in image saliency detection [35]–[37]. For example, He *et al.* [35] learned the hierarchical contrast features using CNNs and designed a multiscale architecture with sharing weights for robust salient region detection. Li and Yu [36] extracted multiscale deep features using CNNs pretrained on the ImageNet data set [38], and the multiscale representations were fused to produce the final saliency score by several fully connected layers. Wang *et al.* [37] presented a saliency detection algorithm by integrating both local estimation and global search, and utilized two deep neural networks to realize it. Some other attempts have been made for the application of human fixation prediction [39]–[42]. Vig *et al.* [40] combined CNNs and a linear SVM for predicting fixation, and Kümmerer *et al.* [41] learned a weight on linear combination of convolution channels without using fully connected layers. Using deep architectures, these above-mentioned works mainly integrated multiscale context information in a straightforward way (e.g., in parallel). In contrast, we develop our saliency model following a divide-and-conquer manner, i.e., progressively generating saliency maps, which finely accords with biological perception and existing coarse-to-fine object detection models. We also demonstrate the superior performance of our method over the other existing deep saliency model.

## III. DEEP IMAGE SALIENCY COMPUTING FRAMEWORK

In this section, we describe the proposed DISC framework in detail. We model the fine-level image saliency computing as a progressive representation learning problem with two stacked CNNs. The first CNN takes the original image as input and produces a coarse-level saliency map. Both the original image and the coarse-level saliency map are then fed into the second CNN, generating the final fine-level saliency map. To maintain the spatial structure of salient object for the coarse map, the SLCI is integrated in the first CNN as an intrasmoothing layer and an intervolting layer. The pipeline of DISC is shown in Fig. 2.

### A. Progressive Representation Learning

Before delving into the formulation, we present some notations that would be used later. Let  $I$ ,  $C$ , and  $F$  denote the original image data, coarse-level, and fine-level saliency maps, respectively;  $\Lambda_i(I)$  and  $\Lambda_i(C)$  denote the square patches of  $I$  and  $C$  centered at pixel  $i$ ;  $C_i$  and  $F_i$  denote the saliency score of pixel  $i$  in saliency maps  $C$  and  $F$ .

The coarse-level saliency map is generated first. We extract the feature of the whole image and, then, apply a linear transformation that assigns a corresponding saliency score to each pixel, which can be expressed as

$$\begin{aligned} C_i &= \mathcal{T}_{c,i}(\phi_c(I)) \\ &= \mathbf{w}_{c,i}^T \phi_c(I) + b_{c,i}, \quad i = 1, 2, \dots, N_c \end{aligned} \quad (1)$$

where  $\mathbf{w}_{c,i}$  and  $b_{c,i}$  are the parameters of the linear transform  $\mathcal{T}_{c,i}$  for pixel  $i$ .  $N_c$  is the pixel number of the coarse-level saliency map. The feature extractor  $\phi_c$  is implemented by a CNN, and the linear transformations  $\{\mathcal{T}_{c,i}\}_{i=1}^{N_c}$  are defined in the form of linear SVMs. We further refine  $C$  using SLCI to get a structure-preserved coarse map  $\tilde{C}$ . We describe SLCI in detail in Section III-B.

To compute the saliency score of each pixel  $i$  in the fine saliency map, we take both the local patch  $\Lambda_i(I)$  from the original image and its corresponding patch  $\Lambda_i(\tilde{C})$  from the coarse-level saliency map as input, and map them to a feature vector  $\phi_f(\Lambda_i(I), \Lambda_i(\tilde{C}))$ . A simple linear transformation  $\mathcal{T}_f$

is then utilized to map the feature vector to the corresponding saliency score

$$\begin{aligned} F_i &= \mathcal{T}_f(\phi_f(\Lambda_i(I), \Lambda_i(\tilde{C}))) \\ &= \mathbf{w}_f^T \phi_f(\Lambda_i(I), \Lambda_i(\tilde{C})) + b_f, \quad i = 1, 2, \dots, N_f \end{aligned} \quad (2)$$

where  $\mathbf{w}_f$  and  $b_f$  are the parameters of the linear transformation  $\mathcal{T}_f$ .  $N_f$  is the pixel number of the fine-level saliency map. Similarly, the feature extractor  $\phi_f$  is implemented by another CNN. For each CNN, we jointly train the feature extractor and the classifier for the saliency computing model.

### B. Superpixel-Based Local Context Information

As discussed above, the fine-level CNN generates the final saliency maps guided by the coarse maps. Recall that the saliency score of each pixel in the fine map is influenced merely by its small neighborhood from the original image and the corresponding coarse map. Hence, the accuracy of fine saliency map depends heavily on the quality of a coarse saliency map. Note that the first CNN takes the global information into consideration, but pays less attention to the nearby local context. We experimentally find that although the coarse CNN is able to highlight the overall salient regions, it suffers from two main problems: 1) the generated coarse maps may confuse some of small foreground or background regions if they have similar appearance and 2) it may fail to preserve the structure of the salient objects, especially when the background is very cluttered. Our aim is to capture the global information and simultaneously consider nearby context in order to produce high-quality coarse maps for the subsequent learning. Therefore, the SLCI, consisting of two types of superpixel-level refinements called intrasmoothing and extravoting, is utilized to make better use of local context information and preserve the spatial structure information. They are integrated in the first CNN as two pooling layers for coarse saliency map prediction. Note that the fine CNN predicts the saliency score of each pixel according to its neighborhood, and local context has been fully considered here, so we do not need to use SLCI for the fine CNN.

1) *Intrasuperpixel Smoothing*: ISS aims to assign close saliency scores to the pixels with similar appearance in local small regions. We first oversegment the input image using the entropy rate-based segmentation algorithm [43] to obtain  $N$  small superpixels per image. Let  $R_i$  denote the superpixel that contains pixel  $i$ . Given the prediction map  $C$ , the intrasmoothed saliency map  $\tilde{C}$  can be computed by

$$\tilde{C}_i = \frac{1}{|R_i|} \sum_{j \in R_i} C_j \quad (3)$$

where  $|\cdot|$  is the cardinality of the set. In this way, the saliency scores of pixels within a superpixel are replaced by the average score.

2) *Intersuperpixel Voting*: ISS focuses on saliency score smoothing in small regions, which can deal with a small-scale saliency structure. Furthermore, ISV takes larger regions into account for preserving a large-scale saliency structure, so that the salient object can be labeled more uniformly. We replace the saliency score of each region by the weighted average

of the saliency scores of the adjacent superpixels. Given a superpixel  $s$ , we first compute the LAB color histogram  $h^c(s)$  and the gradient histogram  $h^g(s)$  and, then, concatenate them to build the appearance feature histogram  $h(s)$ . Let  $F_s$  denote the saliency value of  $s$ . The intervoting can be expressed by

$$\tilde{C}_s = (1 - \lambda)C_s + \lambda \sum_{s' \in D(s)} w(s') \cdot C_{s'} \quad (4)$$

where the weight for  $s'$  is defined as

$$w(s') = \frac{\exp(-\|h(s) - h(s')\|)}{\sum_{s'' \in D(s)} \exp(-\|h(s) - h(s'')\|)} \quad (5)$$

$D(s)$  contains all superpixels that are adjacent to  $s$ , and  $\|h(s) - h(s')\|$  is the Euclidean distance between feature histogram  $h(s)$  and  $h(s')$ .  $\lambda$  is a scaling factor to balance the two terms. The voting weights of each region are defined according to the similarity in the color and structure space, which are two important cues for saliency computing [6]. We assign a larger voting weight to the region with similar color and structure feature to  $s$ .

It can be observed that ISS and ISV can be regarded as two special types of pooling methods. For ISS, the pooling map is calculated by averaging the predicted map in small regions with arbitrary shape. For ISV, we calculate the weighted average of the intrasmoothing response map in a larger region. We formulate these two operations as ISS-pooling and ISV-pooling layers and, then, integrate them as subcomponents in the first CNN. During the training stage, the gradients are computed according to (3) and (4).

### C. Spatial Regularization

In this section, we introduce the SR, which helps to reject false label assignments and leads to an improvement of the performance. As suggested in [11], objects near the center of the image are more likely to be salient. A mass of saliency detection algorithms, such as [6], [15], and [24], incorporate it in their saliency computing frameworks as a center-bias prior. In contrast, we simply average all of the aligned ground truth saliency map in the training set and take it as an additional input channel in both training and testing procedures for SR. We do not use a center-bias prior here for two reasons. First, previous works formulate the center-bias prior as a parametric spatial term or a postprocessing step, and inevitably have to tune the parameters carefully for better performance. Second, incorporating the center-bias prior may suppress some salient regions if they appear near the image boundaries. Conversely, the SR map is considered as an additional input channel, and fed to the CNNs with three-channels original image data. In this way, the network is capable to learn saliency representation from the original images, while implicitly taking the SR into account to alleviate overfitting.

### D. Network Architecture

As discussed above, two stacked CNNs are adopted for the coarse-to-fine saliency representation learning. This section introduces the architectures of DISC.

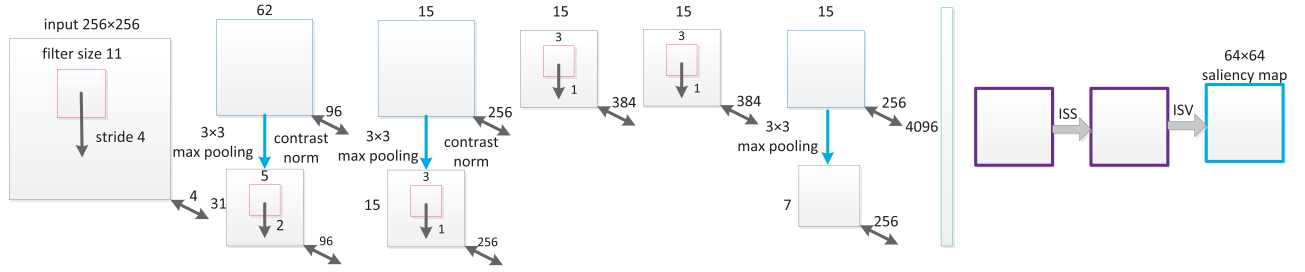


Fig. 3. Architecture of the first CNN for coarse-level saliency computing. A  $256 \times 256$  image is convolved with 96 different first layer filters, with a kernel size of  $11 \times 11$  and a stride of 4. The obtained feature maps are then passed through a rectified linear function, max-pooling operation, and local contrast normalizations, and similar operations are repeated in the next four convolutional layers. The feature vector  $\phi_c(I)$  of the whole image is then fed into the fully connected layer that produces the  $64 \times 64$  coarse-level saliency map. This map is further refined by ISS and ISV.

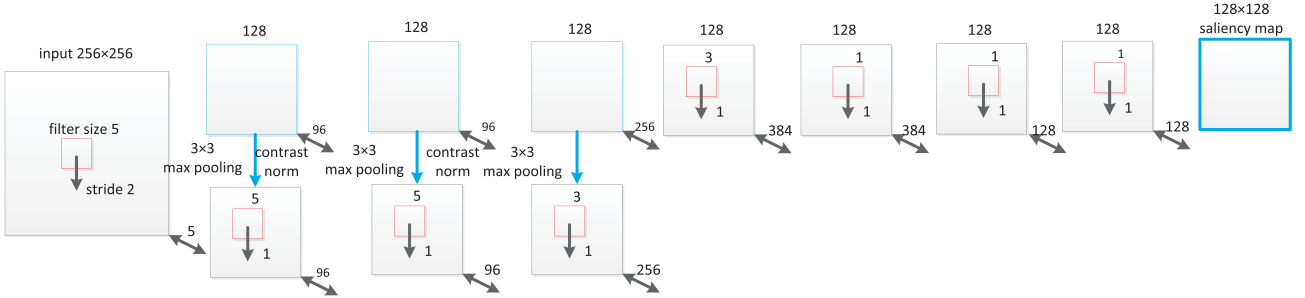


Fig. 4. Architecture of the second CNN for fine-level saliency computing. The network takes  $256 \times 256 \times 5$  image data as input and convolves the data with 96 different first layer filters, with a kernel size of  $5 \times 5$  and a stride of 2. The obtained feature maps are then passed through a rectified linear function, max-pooling operation, and contrast normalization. The next six convolutional layers take similar operations but with all stride of 1 and output the dense feature vectors  $\phi_f(\Lambda_i(I), \Lambda_i(C))(i = 1, 2, \dots, N_f)$ . The dense feature vectors are fed into the last convolutional layer, producing the  $128 \times 128$  fine-grained saliency map.

The first CNN takes the whole image as input and produces the coarse-level saliency map. It contains five convolutional layers and one fully connected layer, followed by the ISS and ISV pooling layers, as shown in Fig. 3. It is similar to the general architecture proposed in [17], but we use four-channel data as input and replace the last two fully connected layers by ISS and ISV pooling layers. The input image data contain three RGB channels of the training image and one channel of the SR map. The five convolutional layers are served as the feature extractor, which takes the whole image data of size  $256 \times 256 \times 4$  as input and produces a feature vector of length  $7 \times 7 \times 256$ . The last fully connected layer computes the linear transformations of the feature vector and outputs 4096 saliency scores that are rearranged to a coarse-level map in a lower resolution. The ISS-pooling and ISV-pooling layers sequentially refine the coarse maps.

The second CNN, guided by the coarse-level map, computes the saliency score for each pixel based on local observation. It is designed as a fully convolutional network [44], as shown in Fig. 4. The input data contains five channels, i.e., three channels of original images with two channels of the SR and coarse maps. This network contains eight convolutional layers and first seven of which are regarded as a feature extractor. Except for the first convolutional layer, we set the stride as 1 based upon the following considerations. First, the feature extractor can produce dense feature vectors for accurate saliency computing. Second, the local patch for saliency computing is kept small, ensuring to better preserve the local details. The last convolutional layer takes the feature vector of each pixel as input and, then, computes its corresponding saliency value to form the fine-level saliency representation.

As the second CNN focuses on local context, it is intuitive to crop a local patch for each pixel and compute their saliency scores separately. However, it is time-consuming for network training and inference. For example, to get a  $128 \times 128$  saliency map, 16384 patches have to be computed. Note that most area of two adjacent patches is overlapping. To avoid these redundant computation, the CNN takes the whole image as input and produces dense outputs. In this way, the patches with overlap share much intermediate computation results, significantly reducing the training and inference time by hundreds of times.

#### IV. OPTIMIZATION

The two CNNs are trained using the stochastic gradient descent (SGD) algorithm with hinge loss sequentially.

##### A. Optimization Formulation

Suppose that there are  $N$  training images, the training sets for the two CNNs are  $\mathcal{X}_c = \{(I^k, Y^k)\}_{k=1}^N$  and  $\mathcal{X}_f = \{(I^k, C^k, Y^k)\}_{k=1}^N$ , respectively.  $I^k$  is the four-channel image data, including three channels of RGB value and a channel of center-bias map.  $Y^k$  is the corresponding saliency map of the  $k$ th image.  $C^k$  is the coarse map produced by the first CNN, which would be fed into the second CNN. The optimization objective is to minimize the total loss

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_f \quad (6)$$

where  $\mathcal{L}_c$  and  $\mathcal{L}_f$  are the objective functions for the first and second CNNs, respectively. We optimize the two terms separately.

For the first CNN, we formulate the coarse saliency computing for each pixel as a binary classification problem and utilize hinge loss to optimize the classifier. Thus,  $\mathcal{L}_c$  can be defined as

$$\mathcal{L}_c = \frac{1}{2} \sum_{i=1}^{N_c} \mathbf{w}_{c,i}^T \mathbf{w}_{c,i} + C \sum_{i=1}^{N_c} \sum_{k=1}^N \max(1 - Y_i^k (\mathbf{w}_{c,i}^T \phi_c(I^k) + b_{c,i}), 0)^2. \quad (7)$$

We adopt the hinge loss [45] in the square form for guaranteeing it is differential. According to (7), the gradients can be computed by

$$\frac{\partial \mathcal{L}_c}{\partial \phi_c(I^k)} = -2C \sum_{i=1}^{N_c} \sum_{k=1}^N Y_i^k \mathbf{w}_{c,i}^T \times \max(1 - Y_i^k (\mathbf{w}_{c,i}^T \phi_c(I^k) + b_{c,i}), 0). \quad (8)$$

Similarly, the fine-level saliency computing can also be regarded as a binary classification problem, and the objective function  $\mathcal{L}_f$  is defined as

$$\mathcal{L}_f = \frac{1}{2} \mathbf{w}_f^T \mathbf{w}_f + C \sum_{k=1}^N \sum_{i=1}^{N_f} \max(1 - Y_i^k (\mathbf{w}_f^T f_i + b_f), 0)^2 \quad (9)$$

where we denote  $\phi_f(\Lambda_i(I), \Lambda_i(C))$  as  $f_i$  for simplicity. The corresponding gradients can be calculated by

$$\frac{\partial \mathcal{L}_f}{\partial f_i} = -2C \sum_{k=1}^N \sum_{i=1}^{N_f} Y_i^k \mathbf{w}_f^T \max(1 - Y_i^k (\mathbf{w}_f^T f_i + b_f), 0). \quad (10)$$

The parameters of lower layers are learned by backpropagating the gradients from the top layer, which can be computed by differentiating the objective function with respect to the activations of the penultimate layer, that is,  $\phi_c(I^k)$ . According to (7) and (9), the gradients can be expressed as (8) and (10). From this point on, the backpropagation algorithm is exactly the same as the standard softmax-based deep networks.

We first remove the ISS and ISV layers from the first CNN and initial the parameters of the first CNN from a zero-mean Gaussian distribution. We then train it using SGD with a backpropagation method. Then, we integrate ISS and ISV layers in the first CNN and fine-tune it to get the final coarse model. The parameters of the second CNN are initialized and trained identically.

## V. TASK-ORIENTED ADAPTATION

Task-oriented salient object detection targets on highlighting specific classes of salient objects. It is significant in the situation that we are interested merely in some classes of objects. As stated above, the model trained in a generic salient object data set is inclined to highlight all the salient objects in the image, no matter what it is. Taking the first image in Fig. 5 as an example, the model trained on the MSRA

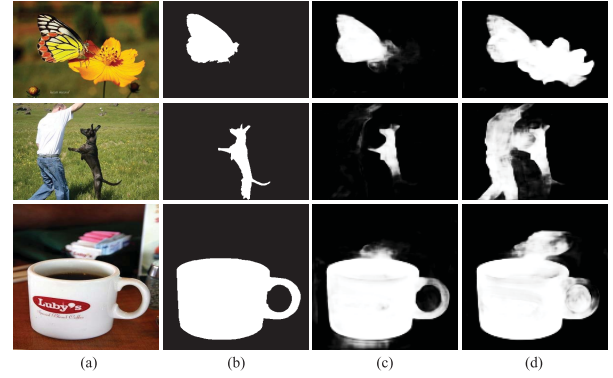


Fig. 5. Result of task-oriented salient object detection without and with retraining. (a) Image. (b) Ground truth. (c) With retraining. (d) Without retraining. All the salient objects are highlighted in (d), but only the specific object is highlighted after retraining in (c).

data set highlights both the butterfly and the flower, because both are salient. Hence, such a model cannot be directly used in a task-oriented task (e.g., only highlighting the butterfly).

Fortunately, it is easy to generalize the model to a task-oriented task. We first collect a data set of images that contain the specific classes of object that we are interested in. We then label the pixels that belong to these specific classes of object as 1 and others as  $-1$  for each image. Finally, we fine-tune the model on the collected data set. In this way, only the specific kinds of objects are highlighted, as shown in Fig. 5.

## VI. EXPERIMENTS

### A. Experiment Setting

1) *Date Set Description*: We evaluate our DISC framework on five public benchmark data sets: 1) MSRA-10k [19]; 2) SED1 [46]; 3) ECSSD [47]; 4) PASCAL1500 [48]; and 5) THUR15K [49]. The MSRA-10k contains 10000 images from the MSRA data set with pixel-level labeling for salient objects. Because most images contain only a single object located near the center of the image and background is generally clean, the accuracy of recent methods has been more than 90%, but our DISC model still improves the accuracy greatly. The SED1 data set is exploited recently, which contains 100 images of single objects. The ECSSD contains 1000 diversified patterns in both foreground and background, which includes many semantically meaningful but structurally complex images for evaluation. The PASCAL1500, created from PASCAL VOC 2012 segmentation challenge, is also a challenging data set, in which images contain multiple objects appearing at a variety of locations and scales with complicated background. THUR15K consists of 15000 images, with 6233 pixel-accurate ground truth annotations for five specific categories: 1) *Butterfly*; 2) *Coffee mug*; 3) *Dog jump*; 4) *Giraffe*; and 5) *Plane*.

2) *Evaluation Protocol*: We use precision–recall (PR) curves,  $F_{0.3}$  metric, and mean absolute error (MAE) to evaluate all the methods. Precision is the fraction of correct salient pixel assigned number in relation to all the detected salient pixel number, while recall is the fraction of correct salient pixel number in relation to the ground truth number.

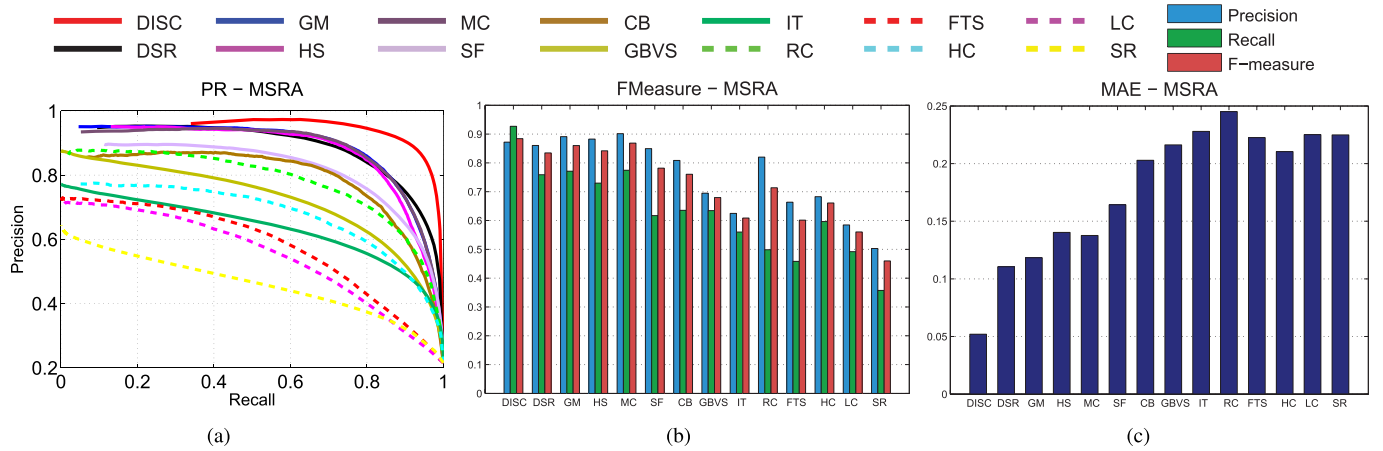


Fig. 6. Experimental results on the MSRA-10k data set. (a) PR curve. (b) PR bar with  $F$ -measure. (c) MAE for comparing our DISC model against previous works. Best viewed in color.

Varying the threshold of saliency object segmentation from 0 to 255, we can plot the PR curve.

$F_{0.3}$  metric applies an image adaptive threshold proposed in [13]. The adaptive threshold is defined as twice the mean saliency of the image

$$T_f = \frac{2}{W \times H} \sum_{x=1}^W \sum_{y=1}^H S(x, y) \quad (11)$$

where  $W$  and  $H$  denote the width and height of the image, respectively, and  $S(x, y)$  denotes saliency value of the pixel at position  $(x, y)$ . The average precise and the average recall are obtained with the adaptive threshold above, and the  $F$ -measure is defined as

$$F_{\beta^2} = \frac{(1 + \beta^2)\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}. \quad (12)$$

The same as [13], we set  $\beta^2 = 0.3$  to weigh precision more than recall.

As indicated in [14], PR curves and  $F_{0.3}$  metric provide a quantitative evaluation, while MAE provides a better estimate of the dissimilarity between the continuous saliency map and the binary ground truth, which is defined as follows:

$$\text{MAE} = \frac{1}{W \times H} \sum_{x=1}^W \sum_{y=1}^H |S(x, y) - GT(x, y)|. \quad (13)$$

3) *Implementation Details*: For the balance between computational efficiency and accuracy, we resize each input image to  $256 \times 256$  for both CNNs and the output size of the first CNN is  $64 \times 64$  while that of the second CNN is  $128 \times 128$ . We implement the two CNNs under the Caffe framework [50], and train them using SGD with the momentum of 0.9 and the weight decay of 0.0005. The learning rate for training the first CNN is initialized as  $10^{-6}$  with a batch size of 32 and that for training the second CNN is initialized as  $10^{-7}$  with a batch size of 2. We train the first CNN for about 90 epochs and the second CNN for roughly 55 epochs, and the training procedure costs nearly two days in all. During inference, we first assign  $-1$  and  $1$  for the pixels with saliency scores smaller than  $-1$  and larger than  $1$ ,

respectively, and, then, transform them to  $[0, 255]$  via a simple linear normalization. Our method can calculate a  $128 \times 128$  normalized saliency map within about 75 ms on a single NVIDIA GeForce GTX TITAN Black. It is relatively time-saving compared with the previous state-of-the-art approaches. For example, sparse reconstruction [3] costs about 3.536 s and hierarchical saliency (HS) [47] runs in about 397 ms on a desktop with an Intel i7 3.4-GHz CPU and 8-GB RAM.

#### B. Comparison With the State-of-the-Art Methods

In this section, we evaluate the proposed method on MSRA-10k [19] data set. In our experiment, we randomly divide MSRA-10k into two subsets, one subset of 9000 images for training and the other subset of 1000 images for verification. We repeat the experiment for three times, and report the average result. We compare DISC with 13 recent state-of-the-art approaches: 1) context-based (CB) saliency [51]; 2) dense and sparse representation (DSR) [3]; 3) graph-based manifold (GM) ranking [4]; 4) HS [47]; 5) Markov chain (MC) saliency [52]; 6) saliency filter (SF) [14]; 7) visual attention measure (IT) [10]; 8) graph-based visual saliency (GBVS) [53]; 9) frequency-tuned saliency (FTS) [13]; 10) spatial-temporal cues (LC) [12]; 11) HC [19]; 12) RC [19] and SR [54]. We adopt the implementations from the original authors to generate the results for CB, DSR, SM, HS, MC, and SF and use the codes provided by [19] to generate the results of LC, FTS, HC, RC, and SR. The results of IT and GBVS are produced using the codes provided in [53]. We normalize all the saliency scores to  $[0, 255]$ .

The results of PR curve,  $F_{0.3}$  metric, and MAE on MSRA-10k data set are shown in Fig. 6, respectively. Based on the PR curve, although some previous methods, such as DSR, GM, HS, and MC, have achieved more than 93% accuracy, DISC still has significant improvement over all of them, reaching 97.3%. On the other side, the minimal recall value of DISC is 34%, significantly higher than those of the other methods, because the saliency maps computed by DISC contain more salient pixels with the saliency scores of 255. Based on  $F_{0.3}$ , our method performs consistently better than the others, as the precision is comparable with the other



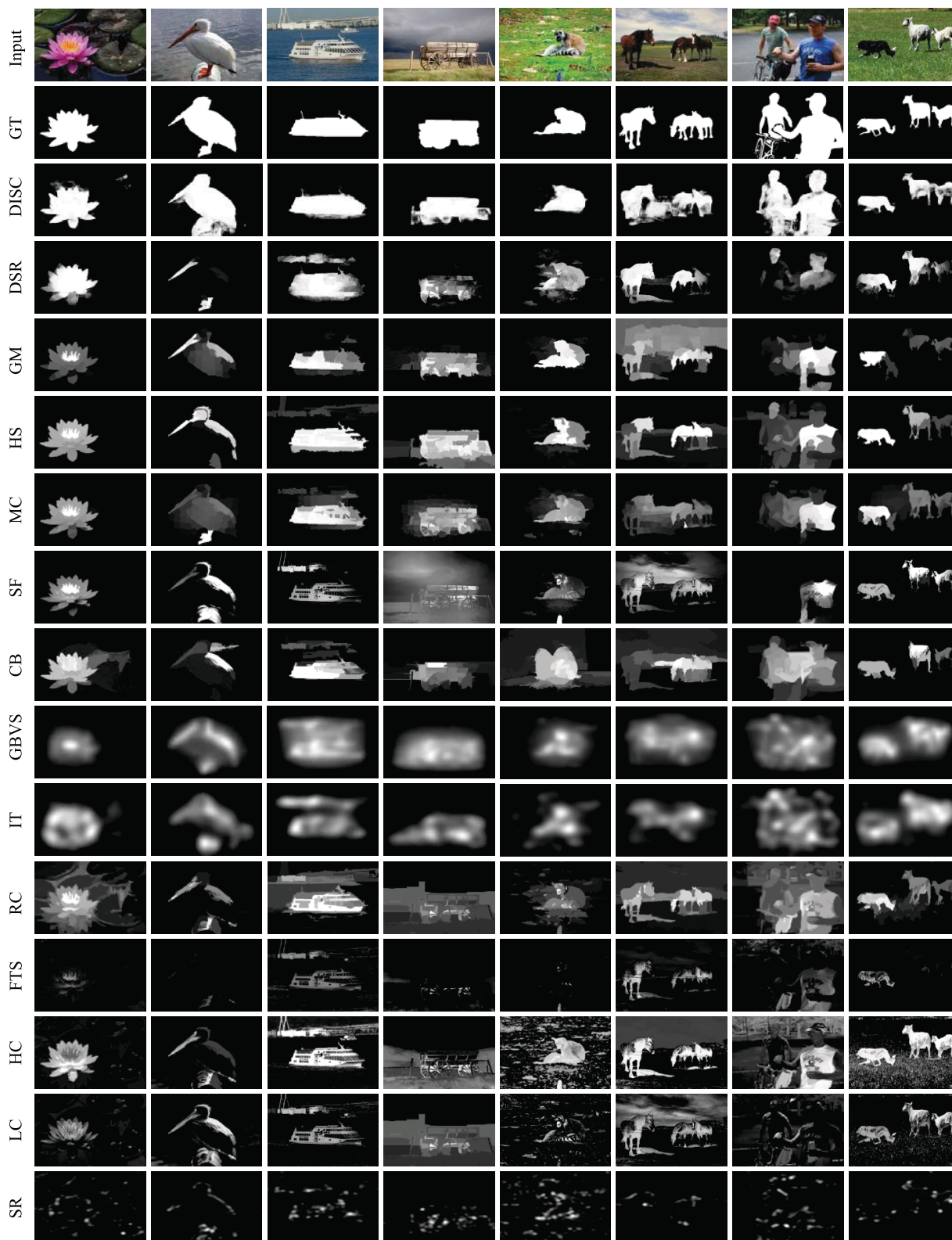


Fig. 7. Visual comparison with previous methods. The images are taken from MSRA-10k (first two columns), SED1 (third and fourth columns), ECSSD (fifth and sixth columns), and PASCAL1500 (last two columns). Our DISC model not only highlights the overall objects, but preserves boundary and structure details.

state-of-the-art result, but the recall is 15.3% higher than the best previous work. Besides, the MAE of DISC is significantly lower than the others, which suggests we preserve the details much better.

Some saliency maps produced by DISC and previous works are shown in Fig. 7 for visual comparison. It can be seen that our method not only highlights the overall salient objects, but also preserves the detail very well.

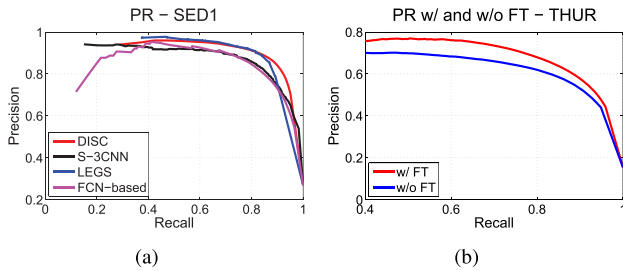


Fig. 8. (a) PR curve of DISC and other CNN-based methods on SED1 data set. (b) PR curve of DISC with and without fine-tuning on THUR data set. Best viewed in color.

These comparisons suggest that DISC outperforms other state-of-the-art algorithms by large margins. The main reason for the superior performance is our coarse-to-fine deep architecture, which is capable of capturing different levels of image saliency information. More specifically, compared with the traditional saliency models using handcraft features, the deep models enable learning very powerful features by incorporating domain knowledge (i.e., how to define and model the image saliency) into neural networks and leveraging the large-scale data for effective learning. Actually, we believe that is very common reason for the success of deep learning in several vision tasks.

### C. Comparison With CNN-Based Methods

To further demonstrate the effectiveness of the proposed architecture, we provide the results of comparison between DISC and other CNN-based methods: 1) S-3CNN [36]; 2) LEGS [37]; and 3) an FCN-based method. S-3CNN and LEGS are two recent-public works, which utilize a multi-scale CNN structure for salient object detection. The fully convolutional network (FCN) [44] takes input of arbitrary size and produces a correspondingly sized dense label map, and it has shown convincing results for pixelwise labeling, such as semantic image segmentation. Thus, we also apply the FCN to salient object detection for comparison. We adopt the trained models and source codes provided by the original authors to generate the results for S-3CNN and LEGS. For FCN-based method, we utilize the best FCN-VGG16-8s architecture and train it using 9000 images of MSRA-10K data set. Since the training data sets are quite different, we test all the methods on SED1 data set, which has no overlap with all the data sets used for models training, for fair cross-data set comparison. The experiments are all carried out on a single NVIDIA GeForce GTX TITAN Black, and the results are shown in Fig. 8(a) and Table I. DISC performs consistently better than other CNN-based methods, and the running time is 75 ms per image, dramatically faster than S-3CNN and LEGS while slightly faster than the FCN-based method. It demonstrates the effectiveness and efficiency of our coarse-to-fine architecture.

### D. Performance of Generalization

In this section, we evaluate the generalization performance of DISC. It is a labor-intensive and time-consuming job to collect enough labeled data to learn particular model for

TABLE I  
PR WITH  $F$ -MEASURE, MAE, AND RUNNING TIME OF DISC AND OTHER THREE CNN-BASED METHODS ON SED1 DATA SET

	Precision	Recall	F-measure	MAE	Time (ms)
DISC	0.892	0.811	0.872	0.084	75
S-3CNN	0.852	0.746	0.825	0.132	17700
LEGS	0.912	0.740	0.865	0.107	2330
FCN-based	0.873	0.730	0.836	0.129	90

TABLE II  
 $F$ -MEASURE AND MAE WITH AND WITHOUT FINE-TUNING ON THE ECSSD AND PASCAL DATA SETS

Dataset	F-measure		MAE	
	w/ ft	w/o ft	w/ ft	w/o ft
ECSSD	0.779	0.784	0.117	0.114
PASCAL	0.738	0.712	0.116	0.114

each scenario. Therefore, transferring a learned model to current scenario without significant degradation is a more practical method. To assess how well our model generalized to other data sets, we evaluate DISC on three data sets, i.e., SED1, ECSSD, and PASCAL1500. As discussed in experiment setting, the images of three data sets are collected in three different scenarios. We directly test the performance on these three data sets with the model learned on MSRA-10K. The results are shown in Fig. 9. Although the model is trained on the other data set, it outperforms all other previous methods based on the three evaluation metrics. It is amazing because the three data sets are significantly different both in salient objects and background, which demonstrate the excellent generalization ability of DISC.

### E. Fine-Tuning on Other Data Sets

Although the result is quite outstanding on the ECSSD and PASCAL1500 data sets without fine-tuning, we retrain on these two data sets to get better result. We do not retrain our model on the SED1 data set, because it contains only 100 images and is too small for retraining. We randomly select 600 images from ECSSD and 1000 images from PASCAL1500 for fine-tuning, respectively, and the rest is for testing. As shown in Fig. 10 and Table II, the result after fine-tuning is slightly better than before. There are two factors that lead to such phenomenon. The first is the generalization ability of DISC is excellent, and it can deal with variable situations as discussed before. The second is that the samples for fine-tuning may be insufficient.

### F. Task-Oriented Adaptation

We select the 6233 images with pixel-accurate ground truth annotations to evaluate the performance of our model with task-oriented adaptation. We divide the 6223 images into two parts: 1) 5233 images for fine-tuning and 2) 1000 images for testing. We first produce the saliency maps for the 1000 testing images with the model learned on the MSRA-10K data set. Then, we fine-tune DISC with the 5233 training images. The PR curves with and without fine-tuning are shown

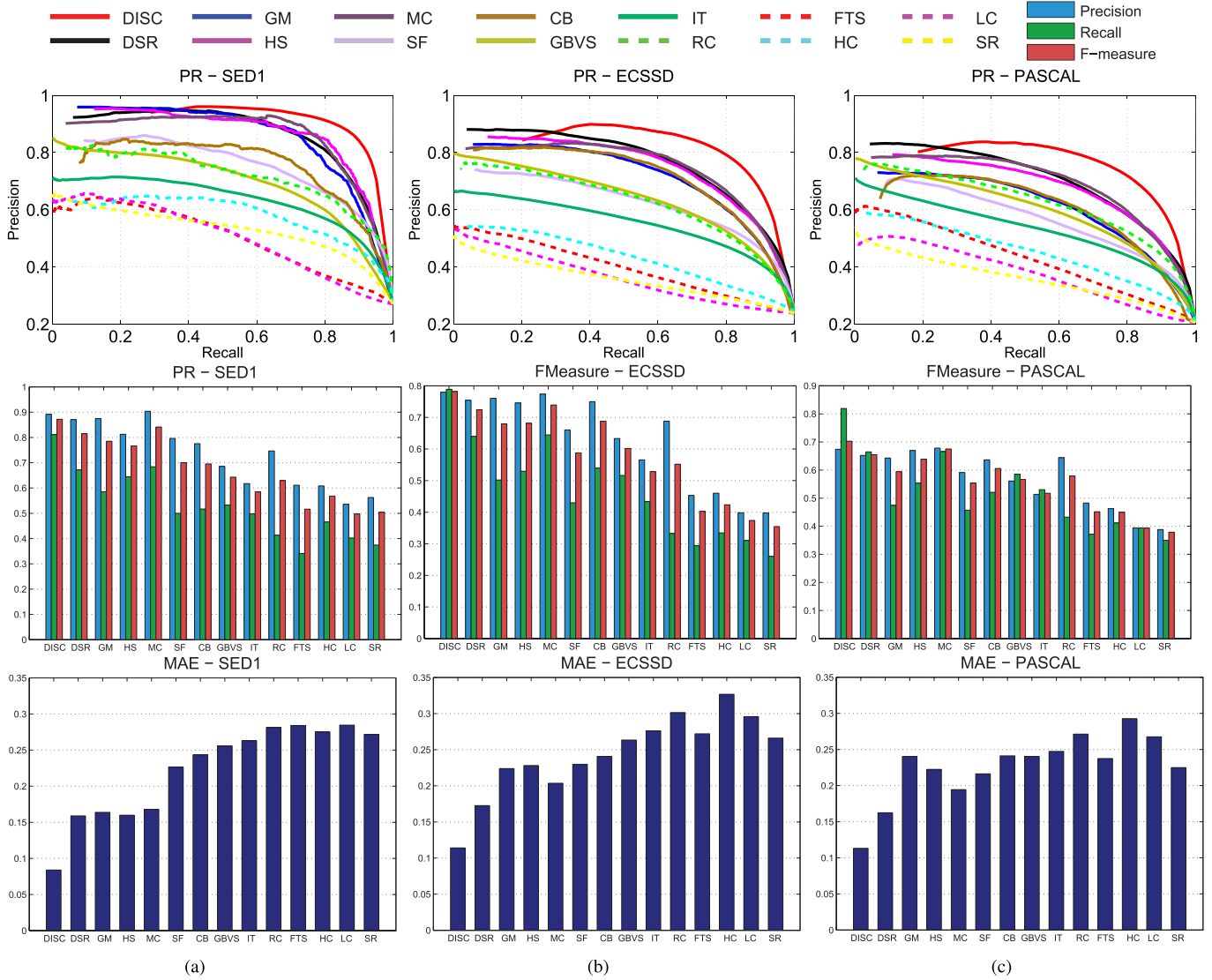


Fig. 9. Experimental results on (a) SED1, (b) ECSSD, and (c) PASCAL1500 data sets compared with previous works. PR curves (the first row), PR bar with  $F$ -measure (the second row), and MAE (the third row) show superior generalization ability of a DISC framework. Note that our method still achieves the state-of-the-art performance when it is learned on a small-scale set (i.e., MSRA-10k) without fine-tuning on the target data sets. Best viewed in color.

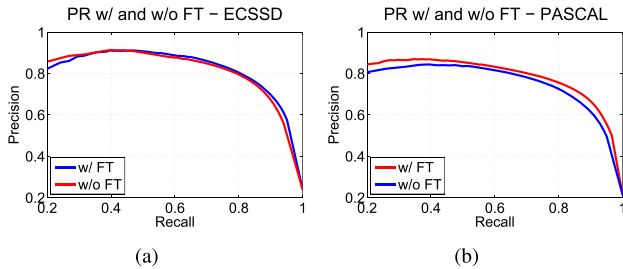


Fig. 10. Experimental results of fine-tuning on the ECSSD and PASCAL1500 data sets. PR curve of our DISC framework with and without fine-tuning (FT) on (a) ECSSD and (b) PASCAL1500. Best viewed in color.

in Fig. 8(b), while the corresponding PR with  $F$ -measure and MAE is listed in Table III. Since the ground truth only labels the specific objects, but the model without retaining highlights all the salient objects, the precision of result without fine-tuning is very low. After fine-tuning, the precision improves

TABLE III  
PR WITH  $F$ -MEASURE AND MAE WITH AND WITHOUT FINE-TUNING ON THE THUR DATA SET

	Precision	Recall	$F$ -measure	MAE
w/ FT	0.664	0.810	0.693	0.084
w/o FT	0.586	0.817	0.626	0.118

significantly, while the recall keeps nearly the same. It suggests that the pixel number we mislabel as saliency decreases. In the other word, we do only highlight the specific classes of objects, ignoring others.

### G. Evaluation and Analysis

In this section, we would like to analyze and discuss each component of the proposed model so as to evaluate the actual contribution of corresponding component.

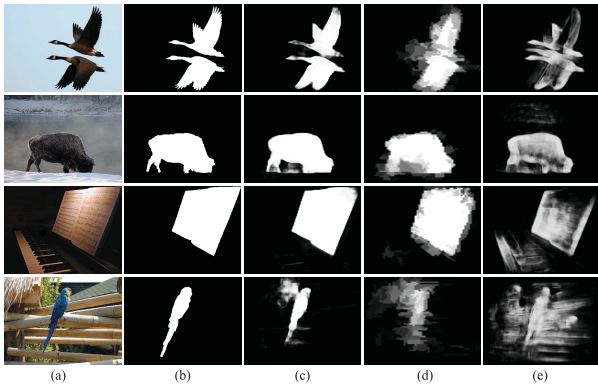


Fig. 11. Visual comparison of coarse maps and fine maps (with and without guidance). The samples are taken from the MSRA and ECSSD data sets. (a) Image. (b) Ground truth. (c) Fine map with guidance. (d) Coarse map. (e) Fine map without guidance.

TABLE IV

PR WITH  $F$ -MEASURE AND MAE COARSE MAPS AND FINE MAPS (WITH AND WITHOUT GUIDANCE) ON THE MSRA DATA SET

	Precision	Recall	F-measure	MAE
Fine map w/ guidance	0.872	0.927	0.884	0.052
Coarse map	0.819	0.900	0.837	0.076
Fine map w/o guidance	0.744	0.785	0.753	0.143

### 1) Contribution of Progressive Representation Learning:

The progressive representation learning framework is designed to learn the saliency representation in a coarse-fine manner, with the coarse representation capturing the object location and global structural information, while the fine representation further refining the object details. We evaluate the contribution of our progressive representation learning framework by comparing its performance with those when using only coarse representation, and using the fine maps without guidance. We show some examples of these three methods in Fig. 11. It is clear that the coarse maps are capable of highlighting the locations and shapes of the salient objects roughly, but the details, especially object boundaries and subtle structures, are easily lost. Overall, these coarse maps are well suited to help learn the fine representation. In contrast, generating the fine maps without guidance usually has two main drawbacks. First, it may miss the interior contents while emphasizing the boundaries of salient objects if the background is relatively clean (e.g., first two examples in Fig. 11). Second, it is likely to mistakenly highlight some of the background regions particularly when the background is cluttered (e.g., the last two examples in Fig. 11). The proposed DISC framework combines the advantage of coarse and fine maps, producing more accurate and structure-preserved results. We also present the quantitative comparisons in Fig. 12(a) and Table IV. They show that the performance of our results dramatically suppresses the results of coarse maps and fine maps without guidance. This comparison well demonstrates the effectiveness of progressive representation learning.

2) *Contribution of Superpixel-Based Local Context Information:* In this section, we analyze the contribution of SLCI.

TABLE V

PR WITH  $F$ -MEASURE AND MAE WITH AND WITHOUT SLCI ON THE MSRA DATA SET

	Precision	Recall	F-measure	MAE
w/ SLCI	0.872	0.927	0.884	0.052
w/o SLCI	0.873	0.908	0.881	0.056

TABLE VI

PR WITH  $F$ -MEASURE AND MAE OF OUR METHOD TRAINED WITH AND WITHOUT SR ON THE MSRA DATA SET

	Precision	Recall	F-measure	MAE
w/ SR	0.872	0.927	0.884	0.052
w/o SR	0.860	0.921	0.873	0.053

Here, we simply remove the ISS and ISV pooling layers from the first CNN and, then, retrain two networks. Some saliency maps generated with and without SLCI are shown in Fig. 13. We first compare the results of the coarse maps. It can be seen that the removal of SLCI leads to poor structure-preserved performance. This is not surprising because the first CNN has to learn the coarse saliency representation based solely on the visual cues while neglecting local structural constraints. As a result, it inevitably confuses some small regions from salient objects or background that appear similar, and also produces counterintuitive structures or shapes for the target objects [see Fig. 13(f)]. Aided by the SLCI, the local structural information can be better preserved. As discussed in Section VI-G1, the fine map depends heavily on the quality of the coarse map. Thus, the corresponding fine maps also suffer from similar problems above. We propose to embed the SLCI into the first CNN, which improves structure-preserved performance, as shown in Fig. 13(c) and (e). Meanwhile, quantitative comparisons on three metrics are provided in Fig. 12(b) and Table V, which show the performance is better than that without SLCI.

3) *Contribution of Spatial Regularization:* Figs. 6 and 9 have shown that the recall of DISC is significantly higher than the previous best approaches, while the precision keeps comparable with them. This indicates more salient regions are recalled by DISC, and one possible reason is that the boundary exclusion problem is alleviated. Furthermore, we also evaluate the contribution of SR by comparing the performance of DISC with and without SR. We exclude the SR input channel in both CNNs while remaining the network architectures unchanged and, then, retrain both of the CNNs. To analyze the effectiveness of SR, we average the feature maps of the first convolutional layer, as shown in Fig. 14. We find that the averaged maps without SR seem to focus only on the object contours, and SR helps to highlight the object more uniformly. Meanwhile, the performance consistently outperforms that without SR according to three evaluation metrics, as shown in Fig. 12(c) and Table VI.

4) *Hinge Loss Versus Cross Entropy Loss:* We finally present the experimental results that evaluate the benefit of hinge loss. For each CNN, we first connect a sigmoid layer with the top layer, and replace the hinge loss with

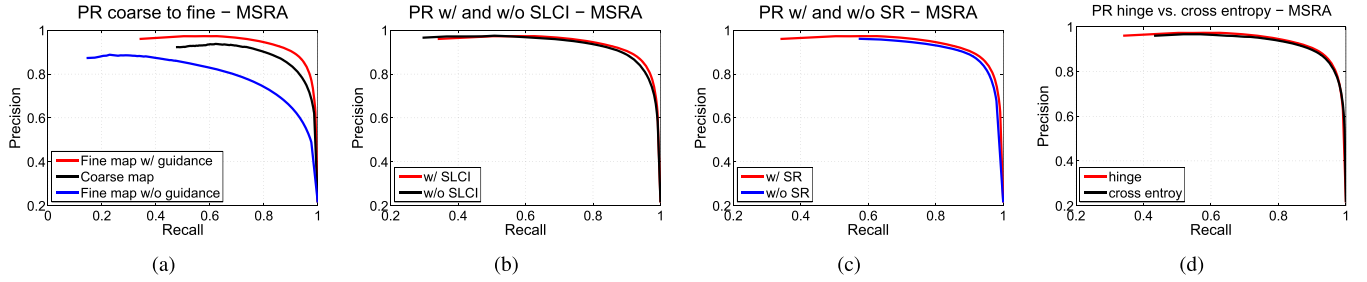


Fig. 12. (a) PR curve of coarse maps and fine maps (with and without guidance) on the MSRA-10k data set. (b) PR curve of our DISC with and without SLCI on the MSRA-10k data set. (c) PR curve with and without SR on the MSRA-10k data set. (d) PR curve of our method trained with hinge loss versus cross entropy loss on the MSRA-10k data set. Best viewed in color.

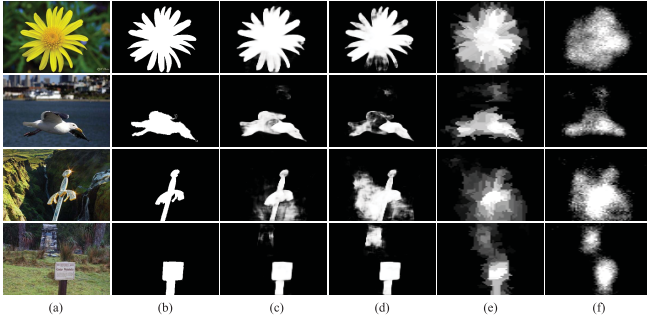


Fig. 13. Visual comparison of coarse/fine maps with and without SLCI. The samples are taken from the MSRA-10k and ECSSD data sets. (a) Image. (b) Ground truth. (c) Fine map with SLCI. (d) Fine map without SLCI. (e) Coarse map with SLCI. (f) Coarse map without SLCI.

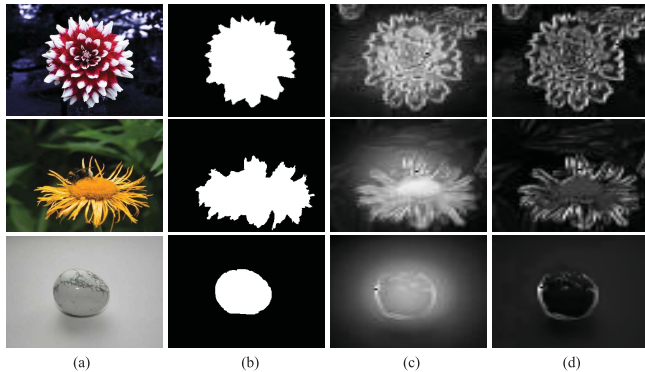


Fig. 14. Visual comparison of feature map of the first convolutional layer with and without SR. The samples are taken from the MSRA-10k data sets. (a) Image. (b) Ground truth. (c) Feature map with SR. (d) Feature map without SR.

the cross entropy loss, with the other layers left unchanged. Accordingly, the label of pixels belonging to salient object is set as 1, while others are set as 0. We then retrain two CNNs for comparison. The results on the MSRA data set are shown in Fig. 12(d) and Table VII. The performance of our method trained with hinge loss is slightly better than that trained with cross entropy loss. In addition, we experimentally find a relative decrease by 8% of MAE with hinge loss compared with cross entropy loss, which improves the visual quality of the generated saliency maps.

TABLE VII  
PR WITH  $F$ -MEASURE AND MAE OF OUR DISC MODEL TRAINED WITH HINGE LOSS AND CROSS ENTROPY LOSS

Loss	Precision	Recall	F-measure	MAE
hinge	0.872	0.927	0.884	0.052
cross entropy	0.853	0.932	0.870	0.056

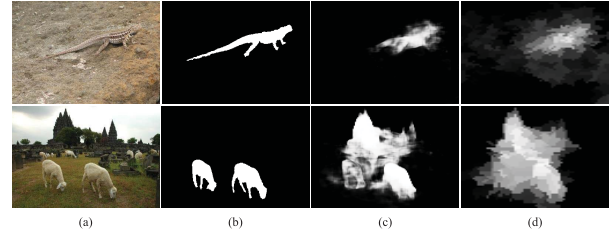


Fig. 15. Some samples that challenge our proposed DISC framework. (a) Image. (b) Ground truth. (c) Fine map. (d) Coarse map.

#### H. Limitation

In Fig. 15, we present some unsatisfying results generated by DISC. As discussed in Section III-B, the accuracy of fine saliency maps is influenced deeply by the quality of coarse saliency maps. In the experiment, we find that two situations will result in a poor quality coarse map. First, the coarse map fails to distinguish some foreground regions from the background when they are similar in appearance (see the first example). Second, it cannot effectively extract the foreground from cluttered background (see the second example). The SLCI will help preserve the structure of the target object, but it still cannot work well for complex background. The main limitation of our method seems to be the dependence of coarse maps. This problem could be tackled by incorporating high-level knowledge, such as object semantic shapes, to further refine the coarse saliency map.

## VII. CONCLUSION

In this paper, we have presented an effective learning framework for accurate image saliency computing. Compared with existing image saliency models, our framework achieves superior performance without relying on any feature engineering or heuristic assumption about image saliency. Two deep CNNs are utilized in a progressive manner to directly map the image data to detail-preserved image saliency maps. The proposed

deep architecture is very general and can be borrowed into other similar vision tasks. Extensive experimental evaluation on five public benchmarks has validated the advantages of our approach.

There are several possible directions in which we intend to extend this paper. The first is to study our model in the context of generic objectness detection, which aims to fast generate a batch of hypotheses of object localizations. Second, our approach can be revised to adapt to video data, and combined with the current research of video tracking algorithms.

## REFERENCES

- [1] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Rev. Neurosci.*, vol. 2, no. 3, pp. 194–203, 2001.
- [2] A. Borji, D. N. Sihite, and L. Itti, "Salient object detection: A benchmark," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 414–429.
- [3] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *Proc. 14th IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 2976–2983.
- [4] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3166–3173.
- [5] Q. Wang, Y. Yuan, P. Yan, and X. Li, "Saliency detection by multiple-instance learning," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 660–672, Apr. 2013.
- [6] K. Wang, L. Lin, J. Lu, C. Li, and K. Shi, "PISA: Pixelwise image saliency by aggregating complementary appearance contrast measures with edge-preserving coherence," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3019–3033, Oct. 2015.
- [7] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 1103–1127, Nov. 2000.
- [8] J. Han, K. N. Ngan, M. Li, and H.-J. Zhang, "Unsupervised extraction of visual attention objects in color images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 141–145, Jan. 2006.
- [9] L. Lin, X. Wang, W. Yang, and J.-H. Lai, "Discriminatively trained and-or graph models for object shape detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 959–972, May 2015.
- [10] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.
- [11] T. Liu *et al.*, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, Feb. 2011.
- [12] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *Proc. 14th ACM Int. Conf. Multimedia*, Santa Barbara, CA, USA, Oct. 2006, pp. 815–824.
- [13] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 1597–1604.
- [14] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. 25th IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 733–740.
- [15] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 853–860.
- [16] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 2083–2090.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, CA, USA, Dec. 2012, pp. 1097–1105.
- [18] F. J. Huang and Y. LeCun, "Large-scale learning with SVM and convolutional for generic object categorization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2006, pp. 284–291.
- [19] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 409–416.
- [20] T. Kadir and M. Brady, "Saliency, scale and image description," *Int. J. Comput. Vis.*, vol. 45, no. 2, pp. 83–105, 2001.
- [21] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2005, pp. 155–162.
- [22] R.-J. Lin and W.-S. Lin, "A computational visual saliency model based on statistics and machine learning," *J. Vis.*, vol. 14, no. 9, p. 1, 2014.
- [23] L. Mai, Y. Niu, and F. Liu, "Saliency aggregation: A data-driven approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 1131–1138.
- [24] S. Lu, V. Mahadevan, and N. Vasconcelos, "Learning optimal seeds for diffusion-based salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 2790–2797.
- [25] X. Liang, S. Liu, Y. Wei, L. Liu, L. Lin, and S. Yan, "Towards computational baby learning: A weakly-supervised approach for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015.
- [26] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recognit.*, vol. 48, no. 10, pp. 2993–3003, Oct. 2015.
- [27] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4766–4779, Dec. 2015.
- [28] X. Liang *et al.*, "Human parsing with contextualized convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015.
- [29] L. Lin, K. Wang, W. Zuo, M. Wang, J. Luo, and L. Zhang, "A deep structured model with radius-margin bound for 3D human activity recognition," *Int. J. Comput. Vis.*, to be published.
- [30] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.
- [31] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3626–3633.
- [32] X. Wang, L. Zhang, L. Lin, Z. Liang, and W. Zuo, "Deep joint task learning for generic object extraction," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 523–531.
- [33] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3476–3483.
- [34] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 2366–2374.
- [35] S. He, R. W. H. Lau, W. Liu, Z. Huang, and Q. Yang, "SuperCNN: A superpixelwise convolutional neural network for salient object detection," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 330–344, 2015.
- [36] G. Li and Y. Yu. (2015). "Visual saliency based on multiscale deep features." [Online]. Available: <http://arxiv.org/abs/1503.08663>
- [37] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3183–3192.
- [38] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [39] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu, "Predicting eye fixations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 362–370.
- [40] E. Vig, M. Dorr, and D. Cox, "Large-scale optimization of hierarchical features for saliency prediction in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 2798–2805.
- [41] M. Kümmeler, L. Theis, and M. Bethge. (2014). "Deep gaze I: Boosting saliency prediction with feature maps trained on ImageNet." [Online]. Available: <http://arxiv.org/abs/1411.1045>
- [42] C. Shen and Q. Zhao, "Learning to predict eye fixations for semantic contents using multi-layer sparse network," *Neurocomputing*, vol. 138, pp. 61–68, Aug. 2014.
- [43] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 2097–2104.
- [44] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.

- [45] X. Xu, I. W. Tsang, and D. Xu, "Soft margin multiple kernel learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 749–761, May 2013.
- [46] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8.
- [47] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 1155–1162.
- [48] W. Zou, K. Kpalma, Z. Liu, and J. Ronsin, "Segmentation driven low-rank matrix recovery for saliency detection," in *Proc. 24th Brit. Mach. Vis. Conf.*, Bristol, U.K., Sep. 2013, pp. 1–13.
- [49] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu, "SalientShape: Group saliency in image collections," *Vis. Comput.*, vol. 30, no. 4, pp. 443–453, 2014.
- [50] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, Nov. 2014, pp. 675–678.
- [51] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li, "Automatic salient object segmentation based on context and shape prior," in *Proc. Brit. Mach. Vis. Conf.*, Dundee, U.K., Aug./Sep. 2011, pp. 1–12.
- [52] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang, "Saliency detection via absorbing Markov chain," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 1665–1672.
- [53] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2006, pp. 545–552.
- [54] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8.



**Tianshui Chen** received the B.E. degree from the School of Information and Science Technology, Sun Yat-sen University, Guangzhou, China, in 2013, where he is currently pursuing the Ph.D. degree in computer science with the School of Data and Computer Science.

His current research interests include computer vision and machine learning.



**Liang Lin** received the B.S. and Ph.D. degrees from the Beijing Institute of Technology (BIT), Beijing, China, in 1999 and 2008, respectively.

He was a Joint Ph.D. Student with the Department of Statistics, University of California at Los Angeles (UCLA), Los Angeles, CA, USA, from 2006 to 2007. He was a Post-Doctoral Research Fellow with the Center for Vision, Cognition, Learning, and Art, UCLA. He is currently a Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. He has

authored or co-authored over 80 papers in top academic journals and conferences. His current research interests include new models, algorithms and systems for intelligent processing, and understanding of visual data.

Prof. Lin received the Best Paper Runner-Up Award in the ACM Symposium on Nonphotorealistic Animation and Rendering in 2010, the Google Faculty Award in 2012, and the Best Student Paper Award in the IEEE International Conference on Multimedia & Expo in 2014. His Ph.D. dissertation was nominated for the China National Excellent Ph.D. Thesis Award in 2010. He was supported by several promotive programs or funds for his works, such as Guangdong NSFs for Distinguished Young Scholars in 2013. He has served as an Associate Editor of the *Neurocomputing* journal and *The Visual Computer*, and a Guest Editor of *Pattern Recognition*.



**Lingbo Liu** received the B.E. degree from the School of Software, Sun Yat-sen University, Guangzhou, China, in 2015, where he is currently pursuing the M.Sc. degree in computer science with the School of Data and Computer Science.

His current research interests include computer vision, machine learning, and parallel computation.



**Xiaonan Luo** is currently a Professor with the School of Information and Science Technology, Sun Yat-sen University, Guangzhou, China. He also serves as the Director of the National Engineering Research Center of Digital Life. His current research interests include computer vision, image processing, and computer graphics and CAD.

**Xuelong Li** is currently a Full Professor with the State Key Laboratory of Transient Optics and Photonics, Center for OPTical IMagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.