Contrast-Oriented Deep Neural Networks for Salient Object Detection

Guanbin Li and Yizhou Yu, *Senior Member, IEEE*

***Abstract*— Deep convolutional neural networks (CNNs) have become a key element in the recent breakthrough of salient object detection. However, existing CNN-based methods are based on either patchwise (regionwise) training and inference or fully convolutional networks. Methods in the former category are generally time-consuming due to severe storage and computational redundancies among overlapping patches. To overcome this deficiency, methods in the second category attempt to directly map a raw input image to a predicted dense saliency map in a single network forward pass. Though being very efficient, it is arduous for these methods to detect salient objects of different scales or salient regions with weak semantic information. In this paper, we develop hybrid contrast-oriented deep neural networks to overcome the aforementioned limitations. Each of our deep networks is composed of two complementary components, including a fully convolutional stream for dense prediction and a segment-level spatial pooling stream for sparse saliency inference. We further propose an attentional module that learns weight maps for fusing the two saliency predictions from these two streams. A tailored alternate scheme is designed to train these deep networks by fine-tuning pretrained baseline models. Finally, a customized fully connected conditional random field model incorporating a salient contour feature embedding can be optionally applied as a postprocessing step to improve spatial coherence and contour positioning in the fused result from these two streams. Extensive experiments on six benchmark data sets demonstrate that our proposed model can significantly outperform the state of the art in terms of all popular evaluation metrics.**

***Index Terms*— Conditional random fields (CRFs), deep contrast network, salient object detection.**

I. Introduction

VISUAL saliency detection aims to locate the most conspicuous regions in images according to the human visual system and has recently received increasing research interest. Image saliency detection is traditionally approached in the form of either eye-fixation prediction or salient object detection. The former focuses on the natural mechanism of visual attention and aims at accurately predicting human eye attended image locations. However, previous research has pointed out that salient object detection, which is more concerned with the integrity of the predicted object regions, is more conducive to a series of computer vision tasks, including semantic segmentation [2], object localization and detection [3], [4], content-aware image editing [5], visual tracking [6], and person reidentification [7]. Although numerous valuable models have been proposed, salient object detection remains challenging due to a variety of complex factors in real-world scenarios.

Perceptual studies [8], [9] have shown that visual contrast is the key factor that affects visual saliency. A series of conventional salient object detection algorithms based on local or global contrast modeling [10]–[12] has been successfully proposed. In previous research efforts, visual contrast modeling is generally focused on the differences among various handcrafted low-level features and coupled with heuristic saliency priors. Although handcrafted features tend to perform well in simple cases, they are not robust enough for more challenging scenarios. For example, it is hard for local contrast models to accurately segment out large homogeneous regions inside salient objects, while global contrast information may fail to handle images with cluttered background. Although there exist machine learning-based algorithms for salient object detection [13]–[16], they are basically focused on integrating various handcrafted features [14] or merging multiple saliency maps computed by different methods [16].

Recently, deep convolutional neural networks (CNNs) have been widely used in salient object detection [17]–[19] because of their powerful feature representations and have achieved substantially a better performance than the traditional methods. Methods based on deep CNNs can be roughly divided into two categories. Methods in the first category generally perform patchwise (or regionwise) training and inference. Specifically, an image is first divided into a set of regions or patches, and deep CNN-based regression or classification models is then trained to independently map each image patch or region to a saliency score or a binary class label (salient or non-salient). However, this results in serious storage and computational redundancies, making training and testing very time-consuming. For example, training a patch-oriented CNN model takes over two GPU days while requiring hundreds of megabytes of storage to save deep features extracted from one single image. Inspired by the latest trends of developing fully CNNs for pixel-level image understanding problems [20]–[22], methods in the second category train end-to-end models that directly map an input image of arbitrary size to a saliency map with the same size, performing dense feedforward computation

Manuscript received February 23, 2017; revised September 18, 2017, January 21, 2018, and March 8, 2018; accepted March 14, 2018. Date of publication April 12, 2018; date of current version November 16, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61702565 and in part by the CCF-Tencent Open Research Fund. This paper was presented at CVPR 2016 [1]. *(Corresponding author: Yizhou Yu.)*

G. Li is with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China (e-mail: liguanbin@mail.sysu.edu.cn).

Y. Yu is with the Department of Computer Science, The University of Hong Kong, Hong Kong (e-mail: yizhouy@acm.org).

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.2018.2817540

and backpropagation over the entire image. This type of methods has rapidly become the cornerstone of this field as they not only achieve very favorable performance but also are very efficient. However, it is still arduous for these methods to detect salient objects of different scales or salient regions with weak semantic information. Moreover, pixel-level correlation is typically not considered in such fully convolutional networks (FCNs), which usually give rise to incomplete salient regions with blurry contours.

In this paper, we develop hybrid contrast-oriented deep neural networks to overcome the aforementioned limitations of two types of contemporary CNN-based salient object detection methods. Our deep networks are composed of a fully convolutional stream for dense prediction and a segment-level spatial pooling (SP) stream for sparse saliency inference. We devise a multiscale FCN (MS-FCN) in the first stream, which receives an entire image as an input and directly learns to map it to a dense saliency prediction with pixel-level accuracy. Our MS-FCN can not only learn multiscale feature representations but also accurately judge the saliency of every pixel by mining visual contrast information hidden in multiscale receptive fields. The segment-level SP stream computes another sparse saliency map over superpixels by modeling the contrast between every superpixel and its spatially adjacent regions. It extracts multiscale regional features very efficiently by performing feature masking in the feature map of an intermediate layer of MS-FCN. At the end, we produce our final saliency map by merging the saliency maps from both streams with weight maps generated from a proposed attentional module in our deep network. Our MS-FCN can also be retrained to generate a contour map for salient objects. This contour map can be used to improve contour localization in the fused saliency map via a fully connected conditional random field (CRF).

In summary, this paper has the following contributions.

1) We propose end-to-end contrast-oriented deep neural networks for localizing salient objects using multiscale contextual information. They incorporate a fully convolutional stream for dense prediction and a segmentwise SP stream for sparse inference. A tailored alternate scheme is designed to train these deep networks by fine-tuning pretrained baseline models.
2) A multiscale VGG-16 or ResNet-101 network pretrained for image classification is repurposed as the fully convolutional stream to infer a dense saliency prediction directly from the raw input image in a single forward pass. This FCN can also be retrained to infer a salient object contour map, which can be represented as a feature embedding and incorporated in a fully connected CRF model to further improve contour localization in the final result.
3) We have also devised a segmentwise SP stream complementary to the fully convolutional stream in our deep network. This stream efficiently masks out segmentwise features from one designated feature map of MS-FCN and accurately models visual contrast among superpixels and well captures saliency discontinuities along region boundaries.

The rest of this paper is organized as follows. Section II reviews a related work on salient object detection. In Section III, we introduce our proposed contrast-oriented deep neural networks. The complete algorithm is presented in Section IV. Section V provides extensive performance evaluation as well as comparisons against state-of-the-art models. Finally, we conclude this paper in Section VI.

II. Related Work

Traditional salient object detection can be categorized into bottom–up approaches with handcrafted low-level features [10], [11], [14], [15], [23]–[28] and top–down approaches incorporating high-level knowledge [29]–[35]. Bottom–up methods are usually based on the center bias or background priors and infer saliency maps from global or local contrast represented as a combination of handcrafted low-level features (e.g., color, texture, and image gradient). Bottom–up computational models are primarily based on a center-surround scheme and compute saliency maps using a linear or nonlinear combination of low-level features, such as color, intensity, texture, and orientation of edges [10], [15], [24], [36]. Top–down methods are, in general, task-dependent and require a machine learning scheme to incorporate high-level knowledge into a process which was originally limited to specified objects or assumptions [33]–[35]. Graph-based methods have also been widely used to enhance spatial consistency and refine detected saliency maps [1], [11], [37]. Recently, deep learning-based methods have been widely used for salient object detection and have promoted its research into a new phase. Since the focus of this paper is deep learning-based salient object detection, we highlight the most relevant previous work in the following discussion.

In recent years, the successful application of deep CNNs has triggered a revolution in machine learning and artificial intelligence, and has yielded significant improvement in a variety of visual comprehension tasks, including image classification [38], object detection [39], and semantic segmentation [20], closing the gap to human-level performance. Motivated by this, several attempts have also been made to apply deep neural network models to salient object detection [1], [40]–[43]. Han *et al.* [44] first attempted to develop stacked denoising autoencoders to learn powerful representations for salient object detection in an unsupervised and bottom–up manner. In [45], a weighted sparse coding framework is proposed for image saliency detection. Recently, with the widespread application of CNNs in image analysis and comprehension tasks, it is not surprising to see a surging number of research papers where very good results have been achieved on salient object detection via the application of CNNs. Li and Yu [17], [40] trained a multilayer fully connected network for deriving the saliency value of every superpixel from its contextual CNN features. Wang *et al.* [19] proposed two deep neural networks, which take into account both low-level features and high-level objectness, for salient object detection at the patch level. A multicontext deep CNN framework incorporating both global and local contexts is presented in [18]. However, all these methods include fully

connected layers and infer saliency maps in an isolated patch-wise manner, and the crucial spatial information in the input image is ignored. However, since all the image patches are treated as independent samples during network training and inference, there is no shared computation among overlapping image segments, which results in significant redundancies and excessive computational cost during training and testing.

To address these issues, inspired by the seminal work of developing end-to-end deep networks for semantic image segmentation [20], [21], a variant of fully CNNs have been introduced to solve the problem of salient object detection since the publication of our earlier conference version [1]. Li *et al.* [41] proposed to explore the correlations between saliency detection and semantic image segmentation using a multitask fully CNN. Liu and Han [46] proposed a hierarchical recurrent CNN to progressively refine the details of saliency maps from a coarse prediction result generated from the forward pass of a fully convolutional VGG-16 network. Kuen *et al.* [47] proposed a recurrent attentional convolutional-deconvolution network, which consists of a recurrent neural network (RNN) and a spatial transform module, to recurrently attend to selected image subregions for saliency refinement. Wang *et al.* [48] introduced a recurrent FCN (RFCN) to iteratively refine the saliency map with incorporated prior knowledge. These FCN-based models have greatly improved both accuracy and efficiency in saliency detection; there are still three aspects of the flaws. First of all, these models are mostly based on the topmost feature map of the network for saliency inference, and the over-reliance on the regional semantic feature may result in the pool detection performance on the salient region with weak semantic information. Second, all of these methods consider feature modeling at a single scale and may not accurately detect salient objects of very different sizes. Finally, as the value at each position of a saliency map generated from FCN-based models is derived from a context with a fixed size (receptive field), the contours of salient objects can hardly be well detected, and the generated saliency maps usually have inadequate spatial consistency. Our proposed method instead delves into the nature of saliency prediction, capturing the key aspect in this problem, which is contrast learning. The proposed method is not only able to infer a saliency probability map from the contrast information in a multiscale deep CNN but also from edge-preserving regionwise contrast information. In addition, it has been proven that fully connected CRFs can be formulated as RNNs. However, experimental results show that RNNs can hardly be trained to achieve comparable results as CRFs. Our proposed method therefore exploits the effectiveness of a contour-aware CRF. Our experimental results demonstrate the superiority of our proposed method in comparison to all existing FCN-based salient object detection techniques.

Note that the initial deep contrast network reported in CVPR 2016 [1] can be viewed as the first piece of work that aims at designing an end-to-end FCN for visual contrast modeling. To a certain extent, it inspired the subsequent development of FCN-based models in this field. Our updated contrast-oriented deep neural network for salient object detection has several improvements over its initial version. First, we adapt the state-of-the-art ResNet-101 network [49] for image classification to an FCN and use it to replace the VGG-16 network in the original fully convolutional stream, achieving a better performance. Second, the fully convolutional stream is run on multiple scaled versions of the original input image, while the segmentwise SP stream is trained using segments from multilevel image segmentation. These strategies make our deep model to more accurately detect salient objects at different scales. Third, we propose to add an attentional module that learns pixelwise soft weights for fusing the two saliency maps, respectively, generated from the two streams. Fourth, we discover that the proposed multiscale fully convolutional stream in our deep network can be retrained to detect salient region contours, which can be integrated into a fully connected CRF model to further improve contour localization in the final saliency map. Finally, we present a more comprehensive experimental comparison among multiple model variants and report the improved results on all benchmarks using all evaluation metrics.

III. Deep Contrast Network

As illustrated in Fig. 1, our proposed contrast-oriented deep neural network is composed of two complementary components, a fully convolutional stream for dense saliency prediction and a segmentwise SP stream for sparse saliency inference. Specifically, the first component is an MS-FCN, which receives an entire image as an input and is trained to map the input to a dense saliency map S_1 in an end-to-end mode by exploiting visual contrast across multiple levels of feature maps. The segmentwise SP stream is trained to infer the saliency map S_2 at the segment level by discovering the contrast among spatially adjacent regions on the basis of features masked out from one designated feature map of the first stream and a multilayer perceptron. At the end, these two intermediate saliency predictions from the above two network streams are merged according to weight maps prescribed by a trained attention module. The merged map becomes our final saliency map S.

A. Multiscale Fully Convolutional Network

Inspired by the groundbreaking application of FCNs in pixel-level image comprehension, we focus on constructing an end-to-end pixelwise regression network, which can directly map a raw input image to a dense saliency map. Considering the centrality of contrast modeling for saliency detection, we have the following considerations when designing the structure of this end-to-end network. First, the network should be deep enough to accommodate features from multiple levels, since visual saliency relies on modeling the contrast among both low-level appearance features and high-level semantic features. Second, the network needs to be able to explore the visual contrast across multiple feature maps and detect salient objects of various scales. Finally, due to the lack of training images with pixelwise labeling, it is much desired to fine-tune an existing pretrained network instead of training from scratch.

As VGG [50] and ResNet [49] are the two most representative and widely used deep classification networks with publicly

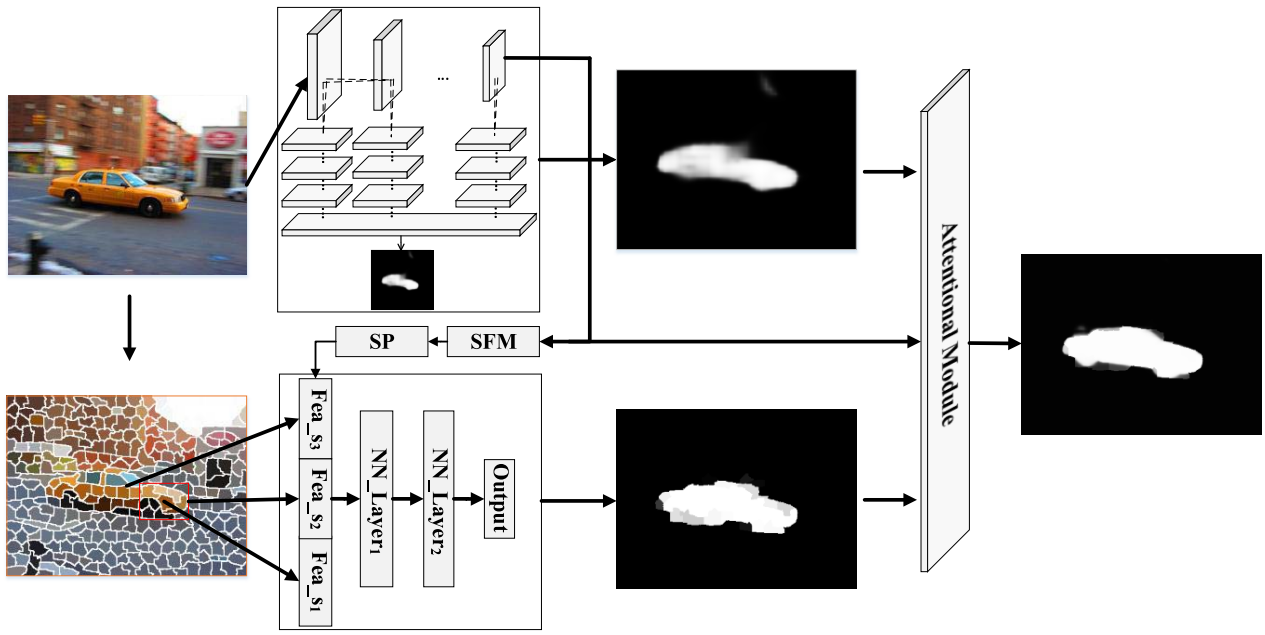


Fig. 1. Overall architecture of our proposed contrast-oriented deep neural network. It consists of a fully convolutional stream (top), a segmentwise SP stream (bottom), and an attentional module to fuse the intermediate saliency maps from the two streams. "SFM" refers to the segment feature masking layer, while "SP" refers to the SP operation.

available pretrained models, we choose them as our pretrained networks and adapt for our requirements. Here, we describe in detail the transformation of the VGG-16 network, and ResNet-101 can be similarly transformed to satisfy the requirements. To repurpose the VGG-16 network for dense saliency map generation, we first convert the two fully connected layers of VGG-16 into 1×1 convolutional ones as described in [20]. Moreover, as the original VGG-16 network consists of five max-pooling layers and each with stride 2, the resulting network can only yield low-resolution prediction maps with $1/32$ the input resolution. To make the resulting saliency map that has a higher resolution, we remove the downsampling operation in the last two max-pooling layers by simply setting their "stride" to 1, which results in downsampling by a factor of 8 instead of 32. At the same time, to maintain the same size of the receptive fields of the convolutional layers that follow, we refer to [21] and [51] and apply the dilation operation to the corresponding filter kernels. The dilation algorithm (also called à trous algorithm), which was originally proposed to improve the computational efficiency of undecimated wavelet transforms [52], has recently been incorporated into the Caffe framework [21], [51] as "dilated convolution" to efficiently control the resolution of feature maps within deep CNNs without the need to learn extra parameters. It works by inserting zeros between filter weights. Specifically, consider applying the dilated version of a convolutional filter w to an input feature map x and generating an output feature map y. The output value at position i is calculated as

$$y[i] = \sum_k x[i + r \cdot k] w[k] \tag{1}$$

where the dilation rate r corresponds to the stride with which we sample the input feature map. This is equivalent to applying convolution to the input feature map x with filters upsampled by inserting $r - 1$ zeros between any two originally adjacent filter elements along each dimension. This dilated convolution allows us to explicitly control the density of feature responses in our customized FCNs. In our implementation, after setting the stride of the last two pooling layers to 1, we replace all subsequent convolutional layers with dilated convolutional layers with dilation rate $r = 2$ or $r = 4$ ($r = 2$ for the three consecutive convolutional layers after the penultimate max-pooling layer and $r = 4$ for the last two newly converted 1×1 convolutional layers).

VGG-16 has five max-pooling layers performing downsampling operations. If we start from the pooling layer closest to the input image, these pooling layers have an increasingly larger receptive field containing contextual information. To design a deep convolutional network that is capable of mining visual contrast information crucial in saliency inference, we further develop a multiscale network from the above fully convolutional version of VGG-16. As shown in Fig. 2 (left), we connect three extra convolution layers to each of the first four max-pooling layers. The first extra layer uses 3×3 convolution kernels and has 128 channels, while the second one uses 1×1 convolution kernels and also has 128 channels. And the third extra layer has one 1×1 kernel and a single channel, which is used to produce the output saliency map. To make the output feature maps of the four sets of extra convolutional layers that have the same size ($8\times$ downsampling resolution), the stride of the first layer in these four sets is set to 4, 2, 1, and 1, respectively. Although the four resulted feature maps are of the same size, they are computed using receptive fields with different sizes and hence represent contextual features at four different scales. We further stack these four feature maps with the last output feature map of the

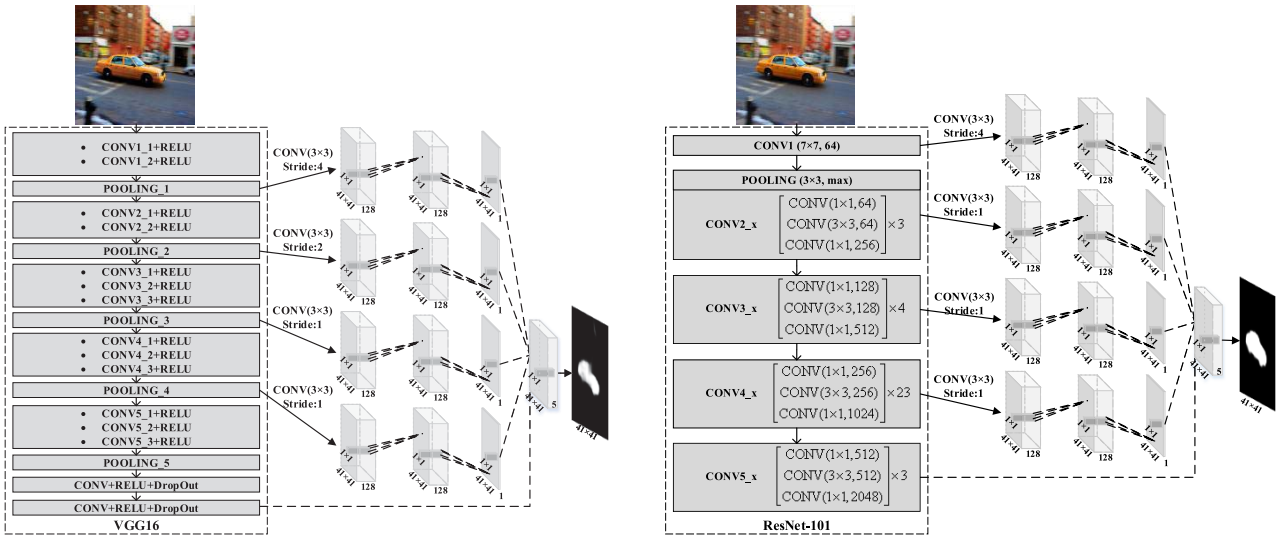


Fig. 2. Architecture of VGG-16-based MS-FCN (left) and ResNet-101-based MS-FCN (right). We connect three extra convolutional layers to each of the first four max-pooling layers of VGG-16 and convert it to a multiscale version. For ResNet-101, we divide the 101 layers into five groups and connect an extra subnetwork with three convolutional layers to each of the final layers in the first four groups to form the multiscale version.

above customized fully convolutional conversion. The stacked feature maps (five channels) are fed into a final convolution layer with a 1×1 kernel and a single output channel, which is modulated by the sigmoid activation function to produce the saliency probability map. Though the resulting saliency map of this network stream has a downsampling factor of 8 in comparison to the input image, it is smooth enough and allows us to use simple bilinear interpolation to restore the resolution of the original input at a negligible computational cost. We call this resized saliency map S_1.

Note that the ResNet-101 network has no hidden fully connected layers. To adapt ResNet-101 for dense saliency prediction, we simply replace its 1000-way linear classification layer with a linear convolutional layer with a 1×1 kernel and a single output channel. Similar to VGG-16, the resolution of the feature maps before the linear convolutional layer is only $1/32$ that of the original input image, because the original ResNet-101 consists of one pooling layer and four convolutional layers, each of which has stride 2. We call these five layers "downsampling layers." As described in [49], the 101 layers in ResNet-101 can be divided into five groups. Feature maps computed by different layers in each group share the same resolution. To increase the resolution of the final saliency map, we replace the last two downsampling layers with dilated convolution layers, skip subsampling by setting their stride to 1, and correspondingly increase the dilation rate of subsequent convolution kernels to enlarge their receptive fields. Therefore, all the features maps in the last three groups have the same resolution, $1/8$ original resolution, after network transformation. To develop a multiscale version of the above end-to-end extension of ResNet-101, as shown in Fig. 2 (right), we connect an extra subnetwork with three convolutional layers to each of the final layers in the first four groups. These additional layers have the same structure as those added to VGG-16. Similar to the multiscale extension of VGG-16, the four output feature maps from these four subnetworks are stacked together with the final output feature map of the transformed ResNet-101 and fed into a final convolutional layer with a 1×1 kernel and a single output channel for final saliency map inference.

B. Segment-Level Saliency Inference

Salient objects in images are usually presented in a variety of irregular shapes and the corresponding saliency map often exhibits discontinuities along the object boundaries. Our MS-FCN operates at a subsampled pixel level and equally treats each pixel in the input image without explicitly taking into account such saliency discontinuities. To better model visual contrast between regions and visual saliency along the region boundaries, we design a segmentwise SP stream in our network.

We first divide an input image into a set of superpixels and call each superpixel a segment. A mask is computed for every segment in the feature map generated from one selected convolutional layer of MS-FCN, which is named the feature masking layer. We choose the convolutional layer Conv5_3 as the feature masking layer in the MS-FCN based on VGG-16, and the last convolutional layer in the fourth layer group as the feature masking layer in the MS-FCN based on ResNet-101 as suggested in [49]. Since the activations at each location in the feature masking layer is controlled by a receptive field in the input image, we first project every location in the feature masking layer to the center of its receptive field as in [53]. For each segment in the input image, we first generate a binary mask within its bounding box. In this mask, pixels inside the segment are labeled "1," while others are labeled "0." Each pixel labeled as "1" in the binary mask is first assigned to the closest receptive field center and then backprojected onto the feature masking layer. Thus, each location in the feature

masking layer collects multiple "1" labels backprojected from its receptive field. The ratio between the number of collected "1" labels at the location and the number of pixels in the input image closest to its receptive field center is recorded. To yield a binary mask for the segment on the feature masking layer, the previously computed ratio at every location is thresholded at 0.5, and the set of locations with nonzero values after thresholding forms the segment mask. In the event that the ratio at all locations is below 0.5, the set of locations with nonzero ratios before thresholding forms the segment mask. The resulting segment mask is then applied to the output feature map of the feature masking layer by simply multiplying this binary mask with each channel of the feature map. We call the resulting features segment-masking features in our method. Note that the feature map generated from the feature masking layer has a downsampling factor of 8 instead of 32 in the original VGG-16 network or 16 in the original ResNet-101 network, since subsampling has been skipped in the last two downsampling layers as described in Section III-A. Therefore, the resolution of the feature map generated from the feature masking layer is sufficient for segment masking.

Since segments have irregular shapes and variable sizes when projected onto the feature masking layer, we further perform an SP operation to produce a feature vector of fixed length for each segment. It is a simplified version of spatial pyramid pooling described in [54]. Specifically, we divide the bounding box of a projected segment into $h \times w$ cells and perform max- or mean-pooling over valid positions (with mask label "1") in each grid cell. This results in $h \times w$ feature vectors of size C, which is the number of convolutional filters in the feature masking layer. Afterward, we concatenate the feature vectors extracted from all grid cells of the same segment to obtain the final feature vector with $h \times w \times C$ dimensions for that segment.

To discover segment-level visual contrast, we represent each segment with a concatenation of three feature vectors, respectively, for three nested and increasingly larger regions masked out from the designated feature map. These three regions include the bounding box of the considered segment, the bounding box of the immediate neighboring segments, as well as the entire feature map from the feature masking layer (with the considered segment excluded to indicate the position of the segment). The above-mentioned feature representation of each segment is further fed into two fully connected layers. The output of the second fully connected layer is fed into a "Sigmoid" layer which employs the sigmoid function to perform logistic regression and produces a distribution over binary saliency labels. We call the saliency map generated in this way S_2.

In fact, this segmentwise SP stream of our network is an accelerated version of our previous work proposed in [17]. Although they share the identical idea of inferring saliency from contrast among multiscale contextual regions, feature extraction and processing in the current method are much more efficient as hundreds of segmental features for the same image are instantaneously masked out from the feature map generated by the MS-FCN in a single forward pass. Moreover, our segmentwise SP stream also achieves better results as segment features are extracted from our MS-FCN, which has been fine-tuned for salient object detection, instead of from the original VGG-16 model for image classification.

C. Attentional Module for Saliency Map Fusion

To merge predicted saliency scores from the two different streams, there are three straightforward options: average pooling, max-pooling, and 1×1 convolution. However, all these strategies are image content independent. As our two network streams have complementary strengths in saliency map prediction, inspired by [55] and [56], we design a trainable attentional module to generate content-dependent weight maps for fusing the results from the two streams.

Let S_1 and S_2 be the probabilistic saliency maps from the two network streams. The final saliency map from our deep contrast network is calculated as a weighted sum of these two maps. The spatially varying weights are adaptively learned. Therefore, they are called weight maps. Let S be the fused saliency map, W_1 be the weight map for the saliency map generated from the MS-FCN stream, and W_2 be the weight map for the saliency map generated from the second stream. The merged saliency map is calculated by summing the elementwise product between each probability map (resized to $1/8$ the input image resolution) and its corresponding weight map

$$S = W_1 \odot S_1 + W_2 \odot S_2. \tag{2}$$

We refer to [56] and call W_1 and W_2 attention weights as they reflect how much attention should be paid to individual network streams as well as saliency scores at different spatial locations. These two attention weights can also be considered as feature maps that have the same size as the predicted saliency maps, and thus can be jointly trained in an FCN. In this paper, we employ a differentiable attention module to our deep network to infer these attention weights. As illustrated in Fig. 1, the proposed attention module receives as an input, the output feature map from the feature masking layer, and it contains two convolutional layers. The first layer has 512 filters with kernel size 3×3, while the second layer has two convolutional filters with kernel size 1×1. The output feature map has two channels, further fed into a SoftMax layer, which generates two score maps corresponding to the aforementioned two attention weights.

D. Deep Contrast Network Training

We propose an alternate training scheme to train our network. Specifically, in the initialization phase, we precompute the segments of all training images and train the segmentwise SP stream alone until convergence to obtain its initial network parameters. Segmentwise saliency labeling is performed by thresholding the average pixelwise labeling inside each segment, and the segment features are extracted using the VGG-16 or ResNet-101 image classification model pretrained on the ImageNet data set [57]. After initialization, we alternately update the weights in the two network streams. First, we fix the weights of the second stream and train the MS-FCN as well as the attention module for one epoch.

Note that the weights in the attention module for adaptively merging the predicted saliency maps from the two streams are trained simultaneously with the MS-FCN stream in an end-to-end mode. Next, we fix the weights in the MS-FCN as well as the attention module, and fine-tune the parameters in the second stream for one epoch using segment features extracted with an updated VGG-16 or ResNet-101 network embedded in the MS-FCN stream. We alternately train the two streams 8 times (16 epochs in total) until the whole training process converges. We define the following class-balanced cross entropy as the loss function for training the multiscale fully convolutional steam and the attention module of our network:

$$L = -\beta_i \sum_{i=1}^{|I|} G_i \log P(S_i = 1|I_i, W) - (1-\beta_i) \sum_{i=1}^{|I|} (1-G_i) \log P(S_i = 0|I_i, W), \quad (3)$$

where β_i represents the class-balancing weight, denoted as $\beta_i = (|I|\_/|I|)$ and $1-\beta_i = (|I|_+/|I|)$, where $|I|$, $|I|_+$, and $|I|\_$ indicate the total number of pixels, salient pixels, and nonsalient ones in image I, respectively. G represents the groundtruth annotation and W represents the collection of all network weights in the MS-FCN stream and the attention module. When fine-tuning the segmentwise SP stream, we use a batch of images as a unit and update parameters by minimizing the summed squared errors accumulated over all segments from the same batch of training images.

IV. Complete Algorithm

A. Superpixel Segmentation

The segmentwise SP stream of our network requires the input image to be decomposed into nonoverlapping segments. In order to better avoid artificial boundaries in the generated saliency map, each segment should be a perceptually homogeneous region, while at the same time, strong contours and edges should still be well preserved. In our earlier version [1], we use a geodesic distance-based [58] simple linear iterative clustering (SLIC) algorithm for superpixel generation. In this paper, we discover that graph-based image segmentation [59] produces segments with better edge preservation than the SLIC algorithm, and using segments generated from multiple levels of image segmentation can further improve the performance. Therefore, we refer to [59] and employ the graph-based image segmentation algorithm therein to generate three levels of segments with different parameter settings. We train a single segmentwise SP stream for all the segments across three levels of segmentation instead of learning different model parameters for segments from different levels of segmentation. When generating a saliency map from the segmentwise SP stream, we apply the same stream to infer a saliency map for each level of segmentation and then simply average the three resulting saliency maps.

B. Salient Contour Detection

While in most cases, our proposed deep contrast network works well, it sometimes produces saliency maps where salient region boundaries are not accurately localized, particularly for images containing small salient regions. Meanwhile, we find that our MS-FCN described in Section III-A, when retrained using annotated salient region contours, is also capable of detecting the contours of salient regions. The detected contours can be further encoded as feature vectors and embedded into a CRF framework to enhance spatial coherence and the preservation of salient region contours in saliency maps. To prepare training data for salient region contour detection, boundary pixels of salient regions in the groundtruth saliency maps are labeled "1," and all other pixels are labeled "0." Such salient region contour maps are taken as the groundtruth annotations when the MS-FCN is trained for salient region contour detection, and the class-balancing weight is updated according to the fraction of pixels on salient region contours.

Given a detected salient region contour map M, we apply the normalized cut [60] algorithm to generate per-pixel feature vectors, which are used in a fully connected CRF to improve boundary localization in our final saliency map. First, we construct a sparse graph where every pixel is connected to other pixels in its 11×11 neighborhood. The affinity matrix W of this graph is defined as follows:

$$W_{ij} = \exp\left(-\max_{p \in \overline{ij}} \left\{ \frac{M(p)^2}{\rho} \right\}\right) \quad (4)$$

where W_{ij} denotes the affinity between pixels i and j, p represents pixels along the line segment $(\overline{ij})$ connecting pixels i and j, $M(p)$ indicates the probability of pixel p being on a salient region contour, and ρ is a constant scaling factor, which is set to 0.1 in our experiments. The idea is that two pixels should have a similar saliency value if there is no salient region contour crossing the line segment connecting these two pixels. Given an affinity matrix W, we further define $D_{ii} = \Sigma_{i \neq j} W_{ij}$ and solve for generalized eigenvectors of the following system, (D-W) $\upsilon = \lambda D \upsilon$. We use these eigenvectors as additional features to improve spatial coherence. In our experiments, we use eigenvectors corresponding to the 16 smallest eigenvalues.

C. Spatial Coherence

Since both streams of our deep contrast network independently infer the saliency score of each individual pixel or segment without considering the impact of the correlation among pixels and segments on saliency prediction, the resulting saliency maps contain more or less incomplete or false positive salient objects. To mitigate this issue, we adopt a fully connected CRF [61] in a postprocessing step to enhance spatial coherence. The energy function of the CRF model is formulated as

$$E(L) = -\sum_i \log P(l_i) + \sum_{i,j} \theta_{ij}(l_i, l_j) \quad (5)$$

where L is the binary label prediction for all pixels (salient or not salient). $P(l_i)$ indicates the probability of pixel x_i being labeled l_i. As an initialization, $P(l_i = 1) = S_i$

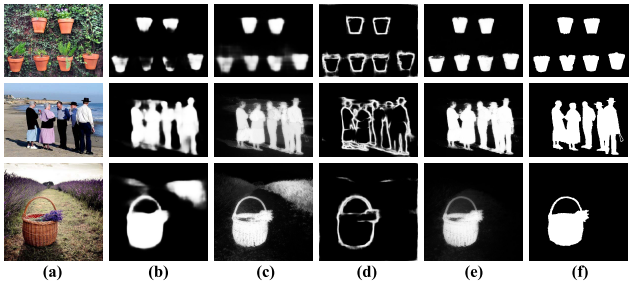


Fig. 3. Examples of saliency maps generated with and without a CRF (including CRFs with and without a contour feature embedding).

and $P(l_i = 0) = 1 - S_i$, where S_i refers to the predicted probabilistic saliency value at pixel x_i of the saliency map S generated from our deep contrast network. The pairwise potential $\theta_{ij}\left(l_i, l_j\right)$ is defined as

$$\theta_{ij} = \mu(l_i, l_j)\left[\omega_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2} - \frac{\|v_i - v_j\|^2}{2\sigma_\gamma^2}\right) + \omega_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\epsilon^2}\right)\right] \tag{6}$$

where $\mu(l_i, l_j) = 1$ if $l_i \neq l_j$, and zero, otherwise. It involves a summation of two Gaussian kernels. The first kernel is based on the observation that neighboring pixels should be assigned similar saliency scores if they have similar colors but do not have intervening salient region contours. It, therefore, depends on pixel positions (p), pixel intensities (I), and the contour feature embedding (v) discussed in Section IV-B. The importance of color similarity, spatial closeness, and salient region contours are controlled by three parameters (σ_α, σ_β, and σ_γ), respectively. The second kernel is only dependent on pixel positions with hyperparameter σ_ϵ controlling the scale of the Gaussian function. As pointed out in [62], it helps to enhance label smoothness and remove small isolated regions.

As it has been proven in [61], this energy minimization process can be modeled as efficient approximate probabilistic inference by adopting a mean-field approximation to the original CRF. High-dimensional filtering can be employed to speed up the computation. We adapt the publicly available implementation of [61] to minimize the above energy function. The optimization process takes less than 0.5 s for an image with 300×400 pixels. After CRF model optimization, a saliency map S_{crf} can be generated from the pixelwise posterior probabilities of saliency labels. We visualize the effectiveness of our CRF in Fig. 3. As can be seen, the original saliency maps from the proposed method without CRF are rather coarse and the integrity (spatial coherence) of detected salient regions can hardly be maintained. Though saliency maps generated with a traditional CRF (without the contour feature embedding) can enhance the spatial coherence of detected salient regions to some extent, salient region contours still may not be well positioned and there may be false detections in the smooth background (e.g., the third row). The fourth column of Fig. 3 demonstrates the salient region contours detected by our proposed method. As can be seen, it is usually possible to accurately capture the boundaries of salient regions and its corresponding embedded features can further enhance the consistency of saliency prediction across salient region contours and correct prediction errors. A quantitative analysis of our CRF-based saliency refinement will be provided in Section V-C2.

V. Experimental Results

A. Experimental Setup

1) Data Sets: We evaluate our proposed method on six widely used saliency detection benchmarks, including MSRA-B [15], HKU-IS [17], PASCAL-S [35], DUT-OMRON [11], ECSSD [63], and SOD [64]. MSRA-B includes 5000 images, most of which holds a single salient object. HKU-IS is proposed in our previous work [17], which has 4447 images and most of the images include multiple separate salient objects. PASCAL-S is based on the validation set of PASCAL VOC2010 segmentation challenge [65] and contains 850 natural images. DUT-OMRON has 5168 challenging images, which have relatively complex and diversified contents. SOD has 300 images and was originally designed for image segmentation. It is very challenging as most of the images contain multiple objects and have low contrast or cluttered background. We train the proposed contrast-oriented deep neural networks based on the combination of both the training sets of the MSRA-B (2500 images) and the HKU-IS (2500 images). The two validation sets are also combined as our final validation, which contains a total of 1000 images. We test the model trained on this combined training set over all other data sets to verity the model's adaptability.

2) Evaluation Criteria: We employ precision–recall (PR) curves, F-measure, and mean absolute error (MAE) to quantitatively evaluate the performance of our method as well as other salient object detection methods. Given a saliency map with continuous values normalized to the range of 0–255, we compute the binary masks by using every possible fixed integer threshold. A pair of PR values can be computed by comparing each binary mask against the ground truth. The precision is defined as the ratio between detected groundtruth salient pixels and all predicted salient pixels in the binary mask, while the recall is defined as the ratio between detected groundtruth salient pixels and all groundtruth salient pixels. Once the PR pairs of all binary maps have been computed, the PR curve can be plotted by averaging all pairs of PR values over all saliency maps of a given data set. F-measure is defined as the harmonic mean of the average precision and the average recall, which can be calculated as

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{7}$$

where β^2 is set to 0.3 to place more emphasis on precision than recall, as suggested in [24]. During evaluation, we report the maximum F-measure (maxF) among all F-measure scores computed from PR pairs on the PR curve. We also use twice the mean value of every saliency map as the threshold to generate the corresponding binary map and report

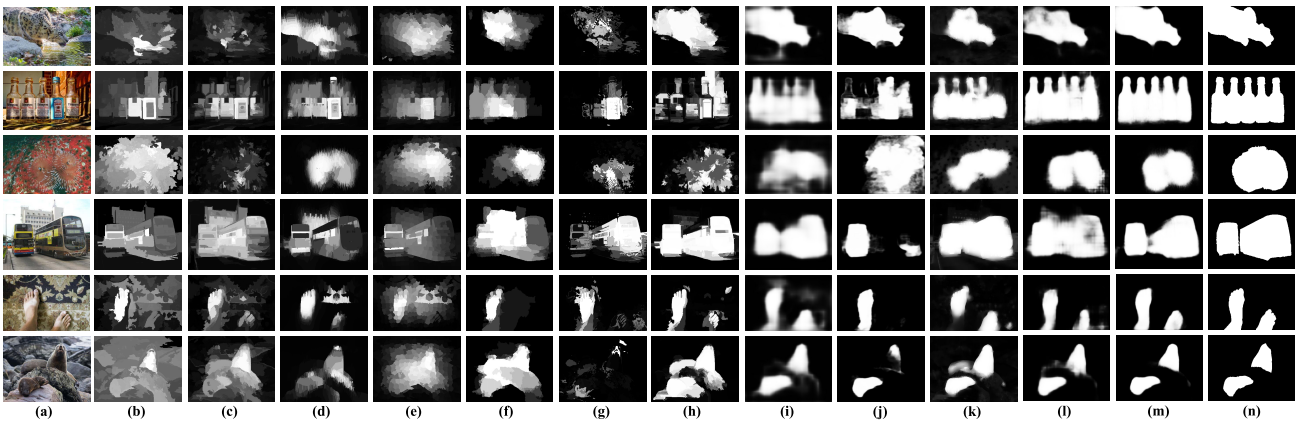


Fig. 4. Visual comparison between our methods (DCL and DCL$^+$) and other state-of-the-art methods. Source: input images. GT: groundtruth saliency maps. DCL$^+$: DCL with CRF refinement. DCL$^+$ consistently achieves the best results in a variety of complex scenarios.

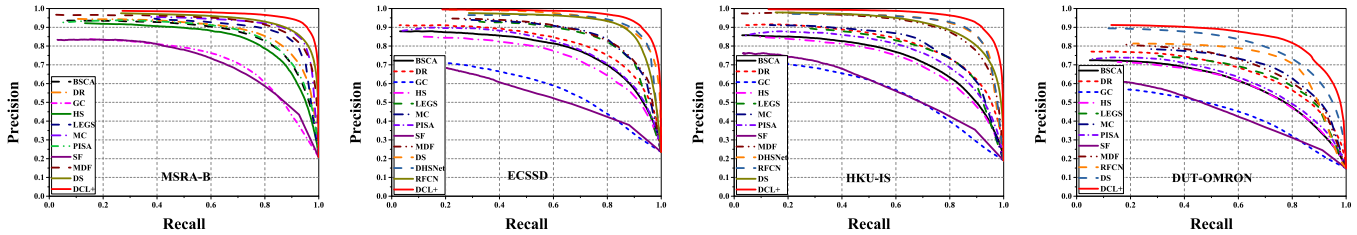


Fig. 5. PR curves of our method and 12 other state-of-the-art algorithms on 4 benchmark data sets. Our DCL$^+$ (DCL with CRF) consistently performs better than other methods across all the benchmarks.

the average precision, recall, and F-measure of all binary maps. As a complement, we also calculate the MAE [26] as follows to quantitatively measure the average absolute per-pixel difference between an estimated saliency map S and the corresponding groundtruth saliency map G

$$\text{MAE} = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x, y) - G(x, y)|. \tag{8}$$

3) Implementation: Our proposed model has been implemented on top of the open source code of DeepLab [21], which is based on the Caffe platform [66]. It was trained with a GTX Titan X GPU and Intel-i7 3.6GHz CPU.

During training, we resize all the images and their corresponding groundtruth saliency maps to 321×321 and perform data augmentation by horizontal flipping. While training the MS-FCN stream, we set the learning rate for all newly added layers to 10^{-3} and the learning rate for the rest of the layers to 10^{-4}. We employ a "poly" learning rate updating policy [67] [the learning rate is scaled by $(1 - (\text{iter}/\text{max\_iter}))^{\text{power}}$ after each iteration, and power $= 0.9$]. We set the weight decay to 0.0005 and the momentum parameter to 0.9 during training. For the segmentwise SP stream, we refer to [59] and obtain 300 segments for each image from 3 levels of image segmentation achieved with different parameter settings. We set the grid size to 2×2 while performing spatial pooling over each segment, and the aggregated feature is of 6144 dimensions in the VGG-16-based MS-FCN and 12288 dimensions in the ResNet-101-based MS-FCN. This feature is further fed into a subnetwork consisting of two fully connected layers, each of which contains 300 neurons. As in [61], we determine the parameters of the fully connected CRF by performing cross validation on the validation set. Finally, the actual value of w_1, w_2, σ_α, σ_β, σ_γ, and σ_ϵ are set to 3.0, 5.0, 3.0, 50.0, 3.0, and 9.0 during evaluation, respectively.

We use DCL$^+$ and DCL to represent our best saliency detectors with and without CRF-based refinement, respectively. While it takes approximately 25 h to train our model, it only costs around 0.7 s for DCL to process an image of size 400×300 on a PC with NVIDIA Titan X GPU and Intel-i7 3.6GHz CPU. Note that this is far more efficient than regionwise deep saliency detectors which independently treat all image patches or superpixels during saliency estimation. However, CRF-based postprocessing is more expensive and requires additional 8 s, since we need to compute generalized eigenvectors used in the CRF model. Experimental results reported in Section V-B show that DCL alone without CRF refinement already performs better than most of the existing state-of-the-art methods. A specific comparison of the computational cost of different methods is summarized in Table II.

B. Comparison With the State of the Art

We compare our models (DCL and DCL$^+$) with nine other state-of-the-art algorithms, including saliency filters [26], global cues [10], hierarchical saliency [63], discriminative regional feature integration [14], pixelwise image saliency [68], background based single-layer cellular automata [69], local estimation and global search based deep network (LEGS) [19], multicontext deep

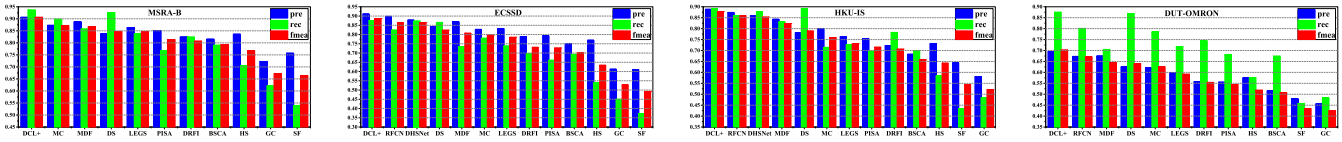


Fig. 6. Precision, recall, and F-measure achieved using an adaptive threshold for every image. Our proposed method consistently performs best among 13 different methods on 4 data sets.

TABLE I

QUANTITATIVE COMPARISON IN TERMS OF MAXIMUM F-MEASURE (LARGER IS BETTER) AND MAE (SMALLER IS BETTER). THE THREE BEST PERFORMING ALGORITHMS ARE MARKED IN RED, BLUE, AND GREEN, RESPECTIVELY. AS THE TESTING SET OF THE MSRA-B DATA SET IS USED AS PART OF THE TRAINING SET IN THE RELEASED MODEL OF DHSNET [46] AND RFCN [48], AND THE PART OF THE DUT-OMRON DATA SET IS ALSO USED IN TRAINING THE DHSNET MODEL, WE EXCLUDE THE CORRESPONDING RESULTS HERE

Data Set	Metric	SF	GC	HS	DRFI	PISA	BSCA	LEGS	MC	MDF	DS	RFCN	DHSNet	DCL	DCL+
MSRA-B	maxF	0.700	0.719	0.813	0.845	0.837	0.830	0.870	0.894	0.885	**0.898**	—	—	**0.929**	**0.931**
	MAE	0.166	0.159	0.161	0.112	0.102	0.130	0.081	**0.054**	0.066	0.067	—	—	**0.046**	**0.042**
ECSSD	maxF	0.548	0.597	0.727	0.782	0.764	0.758	0.827	0.837	0.832	0.900	0.899	**0.907**	**0.921**	**0.925**
	MAE	0.219	0.233	0.228	0.170	0.150	0.183	0.118	0.100	0.105	0.079	0.091	**0.059**	**0.061**	**0.058**
HKU-IS	maxF	0.590	0.588	0.710	0.776	0.753	0.723	0.770	0.798	0.861	0.866	**0.896**	0.892	**0.909**	**0.913**
	MAE	0.173	0.211	0.213	0.167	0.127	0.174	0.118	0.102	0.076	0.079	0.073	**0.052**	**0.050**	**0.041**
DUT-OMRON	maxF	0.495	0.495	0.616	0.664	0.630	0.617	0.669	0.703	0.694	**0.773**	0.747	—	**0.799**	**0.811**
	MAE	0.147	0.218	0.227	0.150	0.141	0.191	0.133	0.088	0.092	**0.084**	0.095	—	**0.070**	**0.064**
PASCAL-S	maxF	0.493	0.539	0.641	0.690	0.660	0.666	0.752	0.740	0.764	**0.834**	0.832	0.824	**0.851**	**0.857**
	MAE	0.240	0.266	0.264	0.210	0.196	0.224	0.157	0.145	0.145	0.108	0.118	**0.094**	**0.098**	**0.092**
SOD	maxF	0.516	0.526	0.646	0.699	0.660	0.654	0.732	0.727	0.785	**0.829**	0.805	0.823	**0.848**	**0.857**
	MAE	0.267	0.284	0.283	0.223	0.223	0.251	0.195	0.179	0.155	**0.127**	0.161	**0.127**	**0.122**	**0.120**

learning (MC) [18], multiscale deep feature (MDF) [17], deep saliency (DS) [41], RFCN [48], and deep hierarchical saliency network (DHSNet) [46]. The last three are fully CNN-based methods, which were published after the publication of our earlier conference version [1].

For qualitative evaluation, Fig. 4 provides a visual comparison of saliency detection results, and the results from our proposed method achieve much improvement over those from other state-of-the-art algorithms. Specifically, our method is capable of highlighting salient regions missed by other methods in various challenging cases, e.g., salient regions touching the image boundary (the first and fifth rows), low contrast between salient objects and the background (the third and sixth rows), and images with multiple separate salient objects (the last three rows).

Our method significantly outperforms all other methods, including those FCN-based deep models published after our earlier conference version [1], by a large margin on all public data sets in terms of the PR curve (Fig. 5) as well as average precision, recall, and F-measure (Fig. 6). Moreover, for the purpose of quantitative evaluation, we report a comparison of maximum F-measure and MAE in Table I. Our complete model (DCL$^+$) clearly outperforms the previous best performing method by 3.67%, 1.98%, 1.90%, 10.64%, 2.76%, and 3.38% in terms of maximum F-measure on MSRA-B (skipping RFCN and DHSNet on this data set), ECSSD, HKU-IS, DUT-OMRON (skipping DHSNet), PASCAL-S, and SOD, respectively. And at the same time, it, respectively, lowers the MAE by 22.22%, 1.69%, 21.15%, 23.81%, 2.13%, and 5.51%. It can also be observed that the proposed method (DCL) without CRF-based postprocessing already outperforms all evaluated methods on all considered data sets. We also compare run-time efficiency among the considered algorithms. As shown in Table II, our DCL model needs around 0.68 s to generate a saliency map in the testing phase, which is comparable to other fully convolutional methods (DS [41], RFCN [48], and DHSNet [46]), and is much more efficient than other region-based CNN models (LEGS [19], MC [18], and MDF [17]).

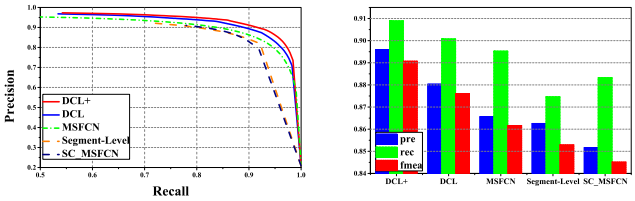


Fig. 7. Componentwise validation of the proposed model and the effectiveness of CRF-based refinement.

C. Ablation Studies

1) Componentwise Effectiveness of Deep Contrast Network: To validate the necessity and effectiveness of the two components contained in our deep contrast network, we take the VGG-16-based version as a representative and compare the saliency maps S_1 inferred from the first stream (MS-FCN), the saliency maps S_2 from the second stream, as well as the fused ones based on S_1 and S_2. As shown in Fig. 7, the fused saliency map consistently performs best under all evaluation metrics on the testing set of the MSRA-B data set, and the fully convolutional stream contributes to the merged prediction far more than the segmentwise spatial pooling stream. The two streams of our deep contrast network are complementary and are capable of discovering global and local contrast collaboratively through multiscale feature aggregation in both streams. To validate the effectiveness of MS-FCN, we have also generated saliency maps from the last scale of MS-FCN for comparison. As illustrated in Fig. 7, a single scale of MS-FCN (SC_MSFCN) may lead to significantly inferior performance when compared with the full version of

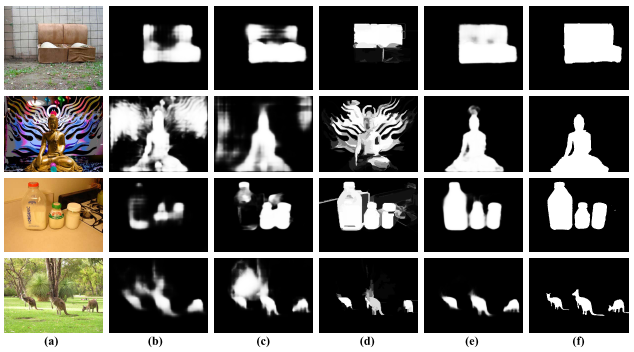


Fig. 8. Sample visualizations demonstrating the componentwise efficacy of our deep contrast network.

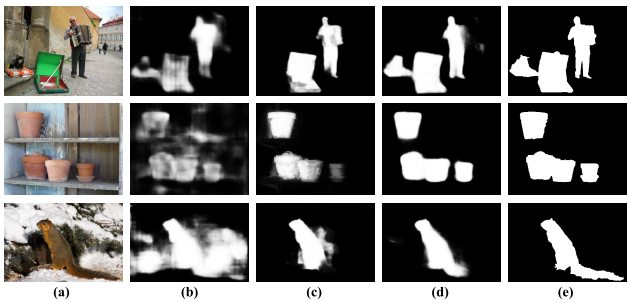


Fig. 9. Effectiveness of ResNet-101 in our DCL model.

MS-FCN in terms of the PR curve as well as average precision, recall, and F-measure. Fig. 8 shows sample visualizations to demonstrate the complementary nature of the two streams inside the DCL network. As shown in Fig. 8, although the fully convolutional stream and the segmentwise spatial pooling stream can produce promising saliency maps, they are far from perfect. The MS-FCN tends to generate very smooth saliency maps but cannot well maintain the integrity of salient regions, while the segmentwise stream predicts saliency maps in the unit of superpixels; it can hardly capture the global contrast and cannot well handle images with a complex background. However, the fused DCL model exploits the advantages of both and produces more accurate saliency predictions, which confirms the complementarity of these two subnetworks. In particular, there are examples (e.g., the second image in Fig. 8) where the two streams have different mistakenly predicted regions, but our proposed network still preferentially integrates, respectively, predicted salient pixels and produces more accurate results. This further demonstrates the robustness of our network and the strong complementarity of the two network streams.

2) Effectiveness of Contour Guided CRF: As described in Section IV-C, we incorporate a fully connected CRF with embedded contour features to further improve spatial coherence and contour positioning in the saliency maps generated from our deep contrast network. We compare the performance of the generated saliency maps with and without CRF as postprocessing. As shown in Fig. 7, CRF significantly increases the accuracy of the saliency maps generated for the testing images of the MSRA-B data set. We also show a visual comparison in Fig. 3 to illustrate the effectiveness of conventional CRF postprocessing and CRF incorporating salient region contours. As shown in Fig. 3, conventional CRF improves the spatial consistency of predicted results to a certain extent, while incorporating salient region contours enhances the confidence of saliency predictions, especially for pixels near detected salient region boundaries.

D. Improvements After Conference Version

After the conference version of this paper, we have made the following five major modifications to our method: 1) adding an attention module to infer spatially varying weights for saliency map fusion; 2) employing the ResNet-101 network in the fully convolutional stream; 3) running the fully convolutional stream on multiple scaled versions of the original input image and fusing the results using max-pooling; 4) training and testing the segmentwise spatial pooling stream using segments from multilevel image segmentation; and 5) performing salient region contour detection and incorporating detected contours in the fully connected CRF during postprocessing. In Table III, we evaluate how each of these factors affects the maximum F-measure and MAE on the DUT-OMRON data set. As shown in Table III, these five factors together contribute a 7.13% improvement in the maximum F-measure and a 20.0% decline in MAE in comparison to the best reported results in the earlier conference version of this paper.

1) Effectiveness of Attention Module: As described in Section III-C, instead of simply adding a 1×1 convolutional layer on top of the saliency maps from the two network streams, we design an attention module to infer spatially varying weight maps. To validate its effectiveness, we conduct a performance comparison between a deep contrast network with a trained attention module and another deep contrast network with a simple 1×1 convolutional layer. As shown in Table III, adopting the attention module for saliency map fusion improves the maximum F-measure on the DUT-OMRON data set by 1.77% while lowering the MAE by 2.38%. Because of the effectiveness of this mechanism, we always integrate this module in our network in subsequent experiments.

2) Effectiveness of ResNet-101 in MS-FCN: As described in Section III-A, we have attempted to replace the VGG-16 network with a transformed ResNet-101 network in the fully convolutional stream of our deep network. To demonstrate its effectiveness, we have trained a new deep contrast network model for comparison. This new model is trained using the same setting as Section V-D1 except that the transformed VGG-16 network is replaced with a pretrained and transformed ResNet-101. As shown in Table III, adopting ResNet-101 instead of VGG-16 significantly improves the maximum F-measure on the DUT-OMRON data set by 3.62% while lowering the MAE by 7.32%. We have also reached the same conclusion as the VGG based DCL network that ResNet-101 in the single-scale setting generates oversmoothed saliency maps with prediction errors and performs much worse than the multiscale version with side branches. As shown in the second and third columns of Fig. 9, our proposed DCL network

TABLE II
COMPARISON OF RUNNING TIME

	SF	GC	HS	DRFI	PISA	BSCA	LEGS	MC	MDF	DS	RFCN	DHSNet	DCL
Time(s)	0.115	0.25	0.43	47.08	0.65	2.03	2.00*	2.38*	8.00*	0.25*	4.60*	0.24*	0.68*

*: GPU time.

TABLE III
PERFORMANCE EVALUATION OF DIFFERENT MODEL FACTORS ON THE DUT-OMRON DATA SET

MSFCN				Segment-Level		CRF		Metric	
VGG16	ResNet-101	Attentional Module	Multi-Scale Input	SLIC Superpixel	Multi-Scale Segmentation	w/o contour	w/ contour	maxF	MAE
√				√				0.733	0.084
√				√		√		0.757	0.080
√		√		√				0.746	0.082
	√	√		√				0.773	0.076
	√	√	√	√				0.792	0.071
	√	√	√		√			0.799	0.070
	√	√	√		√	√		0.804	0.068
	√	√	√		√		√	0.811	0.064

with multiscale ResNet-101 generates much more confident and cleaner results than DCL with the original single-scale ResNet-101.

3) Effectiveness of Multiple Scaled Inputs: Inspired by [56], we adopt a multiscale input strategy when generating a saliency map from the fully convolutional stream. Specifically, we obtain three scaled versions of the original input image with the scaling factor, respectively, set to 1, 0.75, and 0.5, and independently feed these scaled images to the fully convolutional stream. The three resulting saliency maps are fused by taking the maximum response across scales for each position (i.e., max pooling). As shown in Table III, a multiscale input brings an extra 2.46% improvement in the maximum F-measure while lowering the MAE by 6.58%. Sample visualizations are shown in the fourth column of Fig. 9, where fusing saliency predictions from multiscale inputs gives rise to more accurate saliency maps especially when there exists multiple salient objects of different scales in the testing image.

4) Effectiveness of Multilevel Image Segmentation: As described in Section IV-A, the final saliency map from the revised segmentwise spatial pooling stream is the average of three saliency maps, each of which is computed using all superpixels from one of three levels of image segmentation. As shown in Table III, multilevel image segmentation further improves the maximum F-measure by 0.88% and lowers the MAE by 1.40%.

5) Effectiveness of Salient Region Contours: As described in Section IV, we revise the CRF-based postprocessing step in this version by integrating an additional feature vector computed from detected salient region contours. Salient region contours are detected using a separately trained contour detection model, which has the same network structure as the MS-FCN stream. We compare saliency maps computed without CRF, with CRF but without contour saliency features, and with contour guided CRF. As shown in Table III, postprocessing our saliency maps with a dense CRF always yields performance improvement. For the VGG-16-based deep contrast network, running CRF as a postprocessing step boosts the maximum F-measure by 3.27% and lowers the MAE by 4.76%. For the ResNet-101-based deep contrast network, which already achieves a much better performance itself, adding a dense CRF still brings a 0.63% improvement in the maximum F-measure and a 2.86% decrease in MAE. It is worth noting that contour guided CRF results in more accurate saliency maps with a 1.50% improvement in the maximum F-measure and an 8.57% decrease in MAE.

VI. CONCLUSION

In this paper, we have proposed end-to-end contrast-oriented deep neural networks for salient object detection. Our deep networks contain two complementary subnetworks and are capable of extracting a wide variety of visual contrast information. The first subnetwork is based on an MS-FCN and is intended to infer pixelwise saliency by looking into contexts (receptive field) of multiple scales around each pixel. The second subnetwork is designed to capture the contrast information among adjacent regions, which can not only maintain the consistency of saliency prediction within homogeneous regions but also better detect discontinuities along salient region boundaries. An attentional module with learnable weights is introduced to adaptively fuse the two saliency maps from the two subnetworks. Finally, to produce more accurate saliency predictions, we incorporate a CRF with a contour feature embedding to further enhance the spatial coherence and contour localization of the produced saliency map. Experimental results show that the proposed model achieves the state-of-the-art performance on six public benchmark data sets under various evaluation metrics.

REFERENCES

[1] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proc. IEEE Conf. CVPR*, Jun. 2016, pp. 478–487.
[2] Y. Wei *et al.*, "STC: A simple to complex framework for weakly-supervised semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2314–2320, Nov. 2017.

[3] V. Navalpakkam and L. Itti, "An integrated model of top-down and bottom-up attention for optimizing detection speed," in *Proc. IEEE CVPR*, vol. 2. Jun. 2006, pp. 2049–2056.

[4] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *Proc. IEEE Conf. ICCV*, Nov. 2017, pp. 464–472.

[5] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 10-1–10-8, Jul. 2007.

[6] H. Wu, G. Li, and X. Luo, "Weighted attentional blocks for probabilistic object tracking," *Vis. Comput.*, vol. 30, no. 2, pp. 229–243, 2014.

[7] S. Bi, G. Li, and Y. Yu, "Person re-identification using multiple experts with random subspaces," *J. Image Graph.*, vol. 2, no. 2, pp. 151–157, 2014.

[8] W. Einhäuser and P. König, "Does luminance-contrast contribute to a saliency map for overt visual attention?" *Eur. J. Neurosci.*, vol. 17, no. 5, pp. 1089–1097, 2003.

[9] D. Parkhurst, K. Law, and E. Niebur, "Modeling the role of salience in the allocation of overt visual attention," *Vis. Res.*, vol. 42, no. 1, pp. 107–123, 2002.

[10] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.

[11] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 3166–3173.

[12] Q. Wang, Y. Yuan, and P. Yan, "Visual saliency by selective contrast," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1150–1155, Jul. 2013.

[13] S. Lu, V. Mahadevan, and N. Vasconcelos, "Learning optimal seeds for diffusion-based salient object detection," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 2790–2797.

[14] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by UFO: Uniqueness, focusness and objectness," in *Proc. IEEE Conf. ICCV*, Dec. 2013, pp. 1976–1983.

[15] 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.

[16] L. Mai, Y. Niu, and F. Liu, "Saliency aggregation: A data-driven approach," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 1131–1138.

[17] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 1131–1138.

[18] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. CVPR*, Jun. 2015, pp. 1265–1274.

[19] 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. CVPR*, Jun. 2015, pp. 3183–3192.

[20] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. CVPR*, Jun. 2015, pp. 3431–3440.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. (2014). "Semantic image segmentation with deep convolutional nets and fully connected CRFs." [Online]. Available: https://arxiv.org/abs/1412.7062

[22] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Conf. ICCV*, Dec. 2015, pp. 1395–1403.

[23] D. Gao and N. Vasconcelos, "Bottom-up saliency is a discriminant process," in *Proc. IEEE Conf. ICCV*, Oct. 2007, pp. 1–6.

[24] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 1597–1604.

[25] D. A. Klein and S. Frintrop, "Center-surround divergence of feature statistics for salient object detection," in *Proc. IEEE Conf. ICCV*, Nov. 2011, pp. 2214–2219.

[26] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 733–740.

[27] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 2814–2821.

[28] 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.

[29] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. IEEE Conf. ICCV*, Sep./Oct. 2009, pp. 2106–2113.

[30] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *Proc. IEEE Conf. ICCV*, Nov. 2011, pp. 914–921.

[31] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1915–1926, Oct. 2012.

[32] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 853–860.

[33] R. Liu, J. Cao, Z. Lin, and S. Shan, "Adaptive partial differential equation learning for visual saliency detection," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 3866–3873.

[34] Y. Jia and M. Han, "Category-independent object-level saliency detection," in *Proc. IEEE Conf. ICCV*, Dec. 2013, pp. 1761–1768.

[35] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 280–287.

[36] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[37] J. Lei *et al.*, "A universal framework for salient object detection," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1783–1795, Sep. 2016.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[39] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 580–587.

[40] G. Li and Y. Yu, "Visual saliency detection based on multiscale deep CNN features," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5012–5024, Nov. 2016.

[41] X. Li *et al.*. (2015). "DeepSaliency: Multi-task deep neural network model for salient object detection." [Online]. Available: https://arxiv.org/abs/1510.05484

[42] G. Li, Y. Xie, L. Lin, and Y. Yu, "Instance-level salient object segmentation," in *Proc. IEEE Conf. CVPR*, Jul. 2017, pp. 247–256.

[43] G. Li, Y. Xie, and L. Lin. (Mar. 2018). "Weakly supervised salient object detection using image labels." [Online]. Available: https://arxiv.org/abs/1803.06503

[44] J. Han, D. Zhang, S. Wen, L. Guo, T. Liu, and X. Li, "Two-stage learning to predict human eye fixations via SDAEs," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 487–498, Feb. 2016.

[45] N. Li, B. Sun, and J. Yu, "A weighted sparse coding framework for saliency detection," in *Proc. IEEE Conf. CVPR*, Jun. 2015, pp. 5216–5223.

[46] N. Liu and J. Han, "DHSNet: Deep hierarchical saliency network for salient object detection," in *Proc. IEEE Conf. CVPR*, Jun. 2016, pp. 678–686.

[47] J. Kuen, Z. Wang, and G. Wang. (2016). "Recurrent attentional networks for saliency detection." [Online]. Available: https://arxiv.org/abs/1604.03227

[48] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *Proc. Conf. ECCV*, 2016, pp. 825–841.

[49] K. He, X. Zhang, S. Ren, and J. Sun. (2015). "Deep residual learning for image recognition." [Online]. Available: https://arxiv.org/abs/1512.03385

[50] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[51] H. Li, R. Zhao, and X. Wang. (2014). "Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification." [Online]. Available: https://arxiv.org/abs/1412.4526

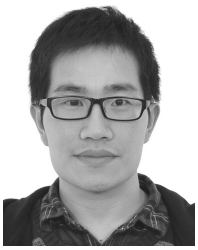
[52] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA, USA: Academic, 1999.

[53] R. Girshick, "Fast R-CNN," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1440–1448.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. Conf. ECCV*, 2014, pp. 346–361.

[55] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015, pp. 1–15.

[56] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. (2015). "Attention to scale: Scale-aware semantic image segmentation." [Online]. Available: https://arxiv.org/abs/1511.03339
[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 248–255.
[58] A. Criminisi, T. Sharp, C. Rother, and P. Pérez, "Geodesic image and video editing," *ACM Trans. Graph.*, vol. 29, no. 5, pp. 1–24, 2010.
[59] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
[60] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[61] P. Krähenbühl and V. Koltun. (2012). "Efficient inference in fully connected CRFs with Gaussian edge potentials." [Online]. Available: https://arxiv.org/abs/1210.5644
[62] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 2–23, Dec. 2007.
[63] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 1155–1162.
[64] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2. Jul. 2001, pp. 416–423.
[65] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.
[66] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[67] W. Liu, A. Rabinovich, and A. C. Berg. (2015). "ParseNet: Looking wider to see better." [Online]. Available: https://arxiv.org/abs/1506.04579
[68] 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.
[69] Y. Qin, H. Lu, Y. Xu, and H. Wang, "Saliency detection via cellular automata," in *Proc. IEEE Conf. CVPR*, Jun. 2015, pp. 110–119.



Guanbin Li received the Ph.D. degree from The University of Hong Kong, Hong Kong, in 2016.

He is currently a Research Associate Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. He has authored and co-authored over 20 papers in top-tier academic journals and conferences. His current research interests include computer vision, image processing, and deep learning.

Dr. Li was a recipient of the Hong Kong Postgraduate Fellowship. He serves as an Area Chair for the conference of the International Conference on Computer Vision Theory and Applications. He has been serving as a reviewer for numerous academic journals and conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CYBERNETICS, IEEE Conference on Computer Vision and Pattern Recognition 2018, and International Joint Conference on Artificial Intelligence 2018.

Yizhou Yu (SM'10) received the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 2000.

From 2000 to 2012, he was a Faculty Member with the University of Illinois Urbana–Champaign, Champaign, IL, USA. He is currently a Professor with The University of Hong Kong, Hong Kong. His current research interests include deep learning methods for computer vision, computational visual media, geometric computing, video analytics, and biomedical data analysis.

Dr. Yu was a recipient of the 2002 U.S. National Science Foundation CAREER Award and the 2007 NNSF China Overseas Distinguished Young Investigator Award. He has served on the Editorial Board of the *IET Computer Vision*, the IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, *The Visual Computer*, and the *International Journal of Software and Informatics*. He has also served on the program committee of many leading international conferences, including ACM Special Interest Group on Computer Graphics (SIGGRAPH), SIGGRAPH Asia, and the International Conference on Computer Vision.