Deep Salient Object Detection with Contextual Information Guidance

Yi Liu, Jungong Han, Qiang Zhang, and Caifeng Shan, *Senior Member, IEEE*

Abstract—Integration of multi-level contextual information, such as feature maps and side outputs, is crucial for Convolutional Neural Networks (CNNs) based salient object detection. However, most existing methods either simply concatenate multi-level feature maps or calculate element-wise addition of multi-level side outputs, thus failing to take full advantages of them. In this work, we propose a new strategy for guiding multi-level contextual information integration, where feature maps and side outputs across layers are fully engaged. Specifically, shallower-level feature maps are guided by the deeper-level side outputs to learn more accurate properties of the salient object. In turn, the deeper-level side outputs can be propagated to high-resolution versions with spatial details complemented by means of shallower-level feature maps. Moreover, a group convolution module is proposed with the aim to achieve high-discriminative feature maps, in which the backbone feature maps are divided into a number of groups and then the convolution is applied to the channels of backbone feature maps within each group. Eventually, the group convolution module is incorporated in the guidance module to further promote the guidance role. Experiments on three public benchmark datasets verify the effectiveness and superiority of the proposed method over the state-of-the-art methods.

Index Terms—Salient object detection, convolutional neural networks (CNNs), group convolution, multi-level contextual information integration

I. INTRODUCTION

HUMAN beings possess the innate ability of identifying the most attractive regions or objects in an image. Salient object detection aims to imitate this ability by automatically identifying and segmenting the most attractive objects in an image. Due to its potential to improve computational efficiency, salient object detection has been studied for decades in various vision tasks, including segmentation [1], [2], image fusion [3], image retrieval [4], object recognition [5], etc.

Earlier methods [6]–[14] for salient object detection mostly employed primitive hand-crafted features; their performance is reasonable but far from satisfactory in complex scenes. Recently, deep convolutional neural networks (CNNs), thanks to their powerful feature representation abilities, have been successfully applied for salient object detection [15]–[33].

Yi Liu and Qiang Zhang are with Key Laboratory of Electronic Equipment Structure Design, Ministry of Education, Xidian University, Xi'an Shaanxi 710071, China, and also with Center for Complex Systems, School of Mechano-Electronic Engineering, Xidian University, Xi'an Shaanxi 710071, China. Email: yLiu_89@stu.xidian.edu.cn, qzhang@xidian.edu.cn. Qiang Zhang is the corresponding author.

Jungong Han is with WMG Data Science at University of Warwick, Coventry, CV4 AL7, U.K. Email: jungonghan77@gmail.com.

Caifeng Shan is with Philips Research, Eindhoven, The Netherlands. Email: caifeng.shan@philips.com.
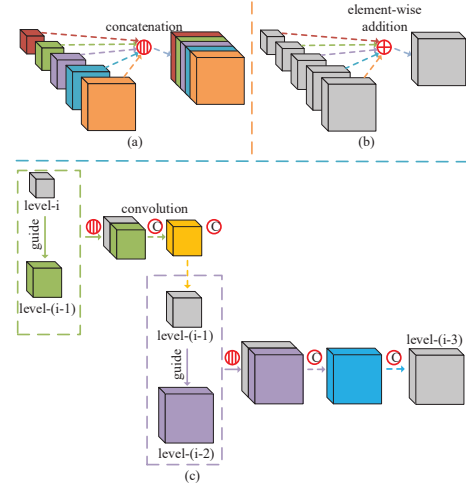
Fig. 1: Illustrations of different manners for integrating multi-level contextual information, where color cuboids represent feature maps and grey cuboids represent side outputs. Different-size cuboids represent different-level feature maps/side outputs. It is noted that some basic operations are omitted for clarity, such as deconvolution. (a) Multi-level feature maps are integrated through concatenation. (b) Multi-level side outputs are integrated through element-wise addition. (c) The proposed multi-level contextual information integration jointly employs feature maps and side outputs. Specifically, the side output of deeper-level (i), used as a guidance feature map, is concatenated with the feature maps of shallower-level ($i − 1$). Due to the fact that side outputs have coarsely predicted the salient object, deeper-level side outputs can provide a guidance for shallower-level feature maps to learn more accurate properties of the salient object. In turn, deeper-level side outputs can be propagated to their high-resolution versions with spatial details complemented by means of shallower-level feature maps.

CNNs are composed of a cascade of repeated convolutional layers, where deeper layers encode high-level semantic knowledge while shallower layers preserve fine details. On top of that, there are rich contextual information across multiple network layers. Lately, such multi-level contextual information is incorporated in the CNNs [15], [16], [19], [29], [34] to further improve the performance of salient object detection. Most of these methods either integrate multi-level feature maps [16], [29], [34] via concatenation (as shown in Fig. 1(a)) or integrate multi-level side outputs (i.e., saliency predictions) [19] through element-wise addition (as shown in Fig. 1(b)).

On one hand, multi-level feature maps can represent an image at different scales, which potentially provide multi-resolution saliency cues. For instance, shallower-level feature maps (high-resolution saliency cues) have small receptive fields and thus can help capture the local saliency. In the meanwhile, deeper-level feature maps (low-resolution saliency cues) with larger receptive fields enable to capture the complementarily global saliency. On the other hand, multi-level side outputs can provide saliency predictions at different scales, where i) deeper side outputs encode high-level semantic knowledge and thus can better locate salient objects, and ii) shallower side outputs are prone to capture rich spatial information such as object boundaries. In view of the above discussion, appropriate integration of multi-level side outputs can potentially improve the performance of saliency detection.

However, the current integration strategies, which fuse the feature maps or the side outputs, are still in the mire of two major limitations. Firstly, some feature maps may be too cluttered, which is likely to mislead the integration of multi-level feature maps. As illustrated in Fig. 2, the Non-Local Deep Feature (NLDF) model [16], which achieves the contrast features by subtracting the average features simply obtained via average pooling, cannot identify the salient parts similar to the background. Secondly, when multi-level side outputs miss some parts of the salient objects, it is no longer possible to make up them again by integrating these side outputs only, as illustrated in the fourth column of Fig. 2.
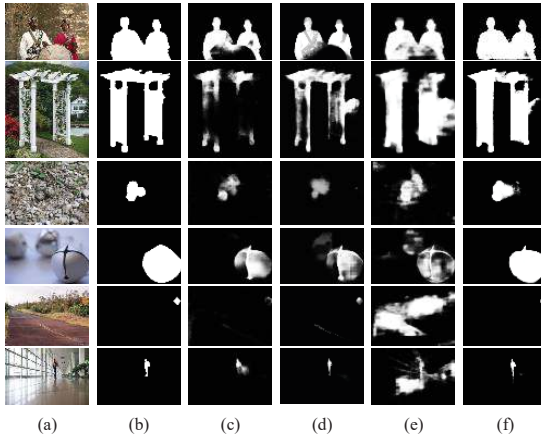

Fig. 2: Illustrations for existing salient object detection methods by employing multi-level contextual information integration. (a) Images; (b) GT; (c) NLDF [16]; (d) DCL [19]; (e) Amulet [15]; (f) Proposed method.

Differently, the aggregating multi-level convolutional feature framework (named Amulet) [15] considers both the feature maps and side outputs for multi-level contextual information integration in the Resolution-based Feature Combination (RFC) module and the Saliency Map Prediction (SMP) module. Concretely, multi-resolution feature maps are integrated into each resolution in RFC while shallower-level feature maps and deeper-level predictions are jointly considered by a weighted summation of them, which turns out to be better than the previous separate manner for information integration. Quite evidently, as displayed in the first two rows of Fig. 2,
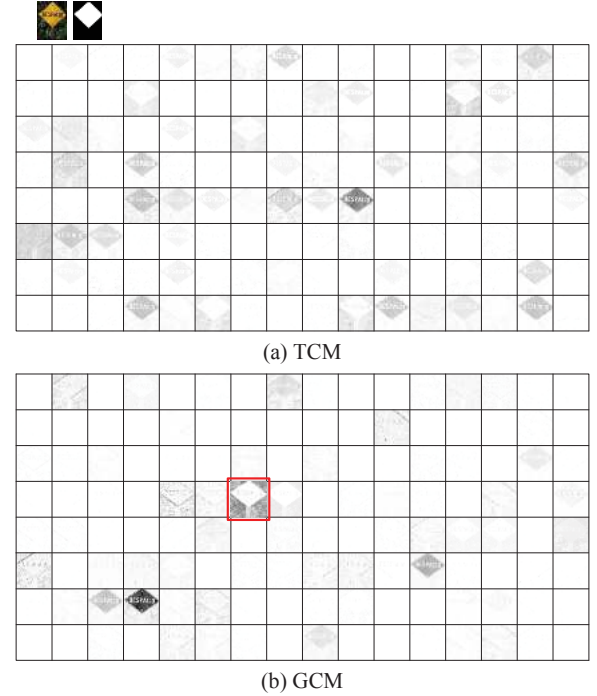

(a) TCM

(b) GCM

Fig. 3: TCM vs GCM. For a given image (shown in the top row), all the discriminative feature maps (128) of the shallowest layer obtained by TCM and GCM are shown here. It can be observed that the feature maps obtained by TCM are mostly trivial and thus not much discriminative to distinguish the salient object from the background. In contrast, the feature map obtained by GCM are more discriminative. Especially, the feature maps marked by a red box can easily predict the salient object based on the softmax function.

Amulet [15] achieves better detection results than NLDF [16] and the Deep Contrast Learning (DCL) model [19] that combines multi-level side outputs for saliency prediction. However, we argue that this simple weighted summation in SMP of Amulet [15] does not efficiently explore the complementarity of these two types of information. As a result, some undesired detection results will occasionally arise. For instance, in Fig. 2, a part of backgrounds are mistakenly labeled as the salient object by Amulet [15]. Furthermore, as shown in the last two rows of Fig. 2, it is hard to distinguish small salient objects from complex backgrounds by Amulet [15].

Alternatively, in this paper, we propose a novel guidance strategy to integrate multi-level contextual information by jointly employing feature maps and side outputs (as illustrated in Fig. 1(c)). The underlying idea behind is based on the observation that multi-level side outputs provide saliency predictions under multi-scale receptive fields, where deeper-level side outputs, corresponding to the large receptive fields, encode high-level semantics and thus can be used to coarsely localize the salient object. Therefore, it is reasonable that we use the deeper-level side outputs to guide the shallower-level feature maps through concatenation. By doing so it can bring benefits to both parties - shallower-level feature maps can learn the properties of salient objects more accurately given

the coarse saliency predictions from deeper-level side outputs; with the aid of a large number of saliency cues provided by shallower-level feature maps, deeper-level side outputs can be propagated into their high-resolution versions with fine details complemented.

Additionally, existing CNN-based salient object detection methods derive the discriminative feature maps by means of the Traditional Convolution Module (TCM), which performs convolutions on all the Backbone Feature Maps (BFMs)[1]. This usually ends up producing a lot of BFMs for each level. The downside is that there is a high chance that salient features are drowned amongst the trivial BFMs, thus making themselves difficult to be distinguished from the background. One example can be found from Fig. 3(a), in which feature maps obtained by TCM are not discriminative enough to highlight the salient object. To tackle this problem, in this paper, we introduce a Group Convolutional Module (GCM) that divides BFMs into groups such that the convolutions can be carried out on the BFMs within each group. By doing that, a number of discriminative feature maps are derived, which are finally concatenated together. The intention of using GCM is to generate fewer trivial feature maps within each group such that the salient features can be well identified, which addresses the drawback of TCM. As illustrated in Fig. 3(b), the GCM indeed produces the feature maps that can better distinguish the salient object.

Moreover, we embed the GCM into the proposed multi-level context guidance strategy to further promote the guidance role of deeper-level side outputs. Specifically, the shallower-level feature maps are first divided into a series of groups. Then, the deeper-level side output performs its guidance within each group, e.g., the deeper-level side output is used as a guidance feature map to be concatenated with several shallower-level feature maps within each group. Extensive experiments demonstrate the superiority of our proposal when compared to the state-of-the-art methods.

The main contributions of this paper are summarized as follows:

(1) A novel guidance strategy is proposed to integrate multi-level contextual information by jointly employing feature maps and side outputs, making full use of multi-level saliency cues and multi-level saliency predictions. As a departure from prior saliency detectors using contextual information, our strategy allows feature maps and side outputs to engage with each other during the integration of multi-level contextual information.

(2) A Group Convolutional Module (GCM) is proposed to produce more discriminative feature maps, which potentially increase the accuracy in identifying the salient object.

(3) Furthermore, the GCM is appropriately embedded into the guidance strategy to design a Group Guidance Module (GGM), which further enhances the guidance role.

The rest of this paper is organized as follows: Section II reviews related works; Section III illustrates the proposed deep salient object detection network in detail; Section IV conducts experiments to verify the effectiveness and superiority of the

proposed method over the state-of-the-art methods; Section V concludes this paper.

II. RELATED WORK

In this section, we review the related works from three aspects. We start with a comprehensive discussion on salient object detection. Afterwards, the related semantic segmentation works that involve multi-level feature integration will be introduced. Last, we discuss the guidance idea used in other computer vision applications.

A. Salient object detection

Earlier salient object detection methods mainly compute saliency based on hand-crafted features [7]–[14], [35]–[39]. Readers can refer to [6] for a comprehensive review on these methods. In recent years, CNNs have been successfully applied for saliency detection and have achieved substantial improvements due to their powerful representation ability [15]–[33], [40]–[42]. Many CNN-based works attempt to learn deep semantic properties of salient objects for further performance improvements. For example, Li *et al.* [22] learnt multi-scale deep features by CNNs for high-quality visual saliency. Li *et al.* [25] improved the perceptional saliency detection by designing a multi-task deep neural network to learn deep features for two correlated tasks, including saliency detection and semantic image segmentation. Hu *et al.* [26] proposed a deep neural network to learn a Level Set function for salient objects, which could produce more accurate boundaries and compact saliency. In addition, a superpixel-based guided layer was constructed to recover full-resolution saliency maps. Zhang *et al.* [27] proposed to learn deep uncertain convolutional features with a reformulated dropout to construct an uncertain ensemble of internal feature units in specific convolutional layers, thus improving the robustness and accuracy of saliency detection. Then, a unified deep neural network was designed for the uncertain feature extraction and saliency detection. While high-level features extracted by CNNs are good to evaluate objectness in an image, they are usually too weak to determine the precise localization. To remedy this problem, Lee *et al.* [23] jointly employed hand-crafted features and deep features via a unified framework to evaluate the saliency.

Apart from the deep semantics, an appropriate scope of context is another important property for salient objects. Specifically, (i) global context can extract the object saliency in a full image; and (ii) local context can better detect the local saliency in the meticulous areas. Therefore, integrating global context and local context will produce more accurate and comprehensive salient objects. Zhao *et al.* [21] applied deep CNNs for saliency detection, which was achieved by extracting global context in a full image and local context in meticulous areas to capture the object saliency. Wang *et al.* [24] designed two deep networks for firstly estimating the local saliency and subsequently searching the global saliency of a set of salient object regions, which were weighted summed to construct the final saliency map. From the view of global to local and coarse to fine, Liu *et al.* [20] proposed a deep hierarchical network to firstly achieve a coarse global saliency prediction and then

---

[1]In this paper, the feature maps of the backbone network are called as the Backbone Feature Maps (BFMs).

hierarchically and progressively refine the details of saliency maps by integrating local context information.

These methods tried to extract more perceptual-context saliency cues for salient object detection. However, they ignored the complementarities of the multi-level contextual information, which were provided by several stages of deep features produced by standard CNNs. Therefore, their performance is still far from satisfactory. Motivated by this, many CNNs based salient object detection methods [15], [16], [19] attempt to integrate hierarchical contextual information. Our work is most related to these methods, which will be elaborated below.

Many CNNs based works have found that i) deeper-level features extract high-level semantic knowledge and thus can help locate the salient objects; ii) shallower-level features capture low-level spatial details that can be used to detect the object boundaries. Based on these perceptual studies, a lot of works have attempted to integrate multi-level contextual information for salient object detection. Zhang *et al.* [15] proposed a RFC module for aggregating the multi-level feature maps into each resolution. Thus, high-level semantic knowledge and low-level spatial details are simultaneously combined at each resolution. Besides, a SMP module was further designed to consider the feature maps and prediction by a simple weighted summation. Luo *et al.* [16] adopted a step-wise unsampling procedure to upsample the deeper-level feature maps by a factor of 2, which were then concatenated with the shallower-level feature maps. Such an operation was performed layer-by-layer until the shallowest layer. In such way, deep-level feature maps will be gradually transformed into high resolution with the refinement of the shallow-level ones. Li *et al.* [19] proposed an end-to-end network consisting of a pixel-level fully convolutional stream, which combined multi-level saliency predictions to produce a pixel-level saliency map, and a segment-wise spatial pooling stream. Eventually, a superpixel-level saliency map could be generated by performing spatial pooling and saliency estimation over superpixels.

B. Semantic segmentation

Semantic segmentation is generally considered as a pixel-wise classification problem, in which each pixel is assigned with an object category label. Ronneberger *et al.* [43] proposed a U-Net architecture for the biomedical image segmentation, where the outputs from low resolution features were combined with high resolution ones for more accurate localization. Badrinarayanan *et al.* [44] designed a SegNet by constructing a hierarchy of decoders corresponding to each encoder for exploring different-scale information. Chen *et al.* [45] proposed DeepLabv1 for image segmentation by integrating the hole algorithm and fully connected Conditional Random Fields (CRFs) in the deep CNN. In [46], they further developed DeepLabv2 by integrating an Atrous Spatial Pyramid Pooling (ASPP) into DeepLabv1 for the sake of accurate object segmentation at multiple scales. In [47], DeepLabv3 was presented, which augmented ASPP with image-level features to capture the global context. DeepLabv3+ [48] was a further extension of DeepLabv3 in the sense that a decoder module

was added into the framework to refine the object boundaries. In general, most sematic segmentation methods directly use the BFMs for further prediction. Differently, we perform GCM on BFMs to produce high-discriminative features for more accurate predictions. Besides, these methods mostly perform information integration via combining multi-layer features. In contrast, we apply the deeper-layer prediction to guide the shallower layer features extraction for more accurate salient properties, where the accurate location of deep prediction and the rich spatial details from shallow features will be comprehensively integrated.

C. Guidance strategy usage in computer vision

Various strategies have been adopted to guide feature integration in different computer vision applications [49]–[55]. Wang *et al.* [49] proposed to capture the motion structure across time for the video inpainting by learning the temporal structure guidance, which could improve the temporal smoothness and the context consistency. A 2D Encoder-Decoder architecture was further adopted to recover the spatial details. Ren *et al.* [50] designed a cross-modal method by unifying both visual and auditory modalities to enhance the robustness against distractors. Wang *et al.* [51] made use of the motions within a video to distinguish different parts and thus extracted more accurate foreground appearance in a video. Sam *et al.* [52] constructed a top-down structure to use high-level feature maps as high-level scene context information to correct false density predictions of the crowd counting CNN. Pinheiro *et al.* [53] first produced multiple channels of coarse mask for the objects in an image, and then refined it with low-level spatial details that were reduced-dimensional feature maps. Shrivastava *et al.* [54] integrated higher and lower features by a top-down structure, which learned what semantic or context information to be preserved in the top-down feature transmission as well as the selection of relevant low-level features. Basically, the above works either apply different-modal information to construct cross-modal information integration [49]–[51] or adopt a top-down structure to combine high-level semantics with low-level spatial details in the form of feature maps [52]–[54]. Differently, we introduce a novel guidance strategy into the top-down structure for multi-level context information integration by jointly employing the prediction and feature maps. Specifically, the deeper-level predictions acts like a guidance feature map to guide the shallower-level features via concatenation. The deeper-level prediction promotes shallower layers to learn more accurate salient properties. GCM is further embedded to essentially promote the guidance role of the deeper level prediction.

III. PROPOSED SALIENT OBJECT DETECTION NETWORK

The framework of the proposed salient object detection network is shown in Fig. 4, where the backbone network (i.e., VGG16 [56]) firstly learns 5-level feature maps. GCM is then proposed to produce high-discriminative feature maps. Next, the guidance strategy is designed for multi-level contextual information integration. The final saliency map is computed by jointly employing those feature maps obtained by GGM,
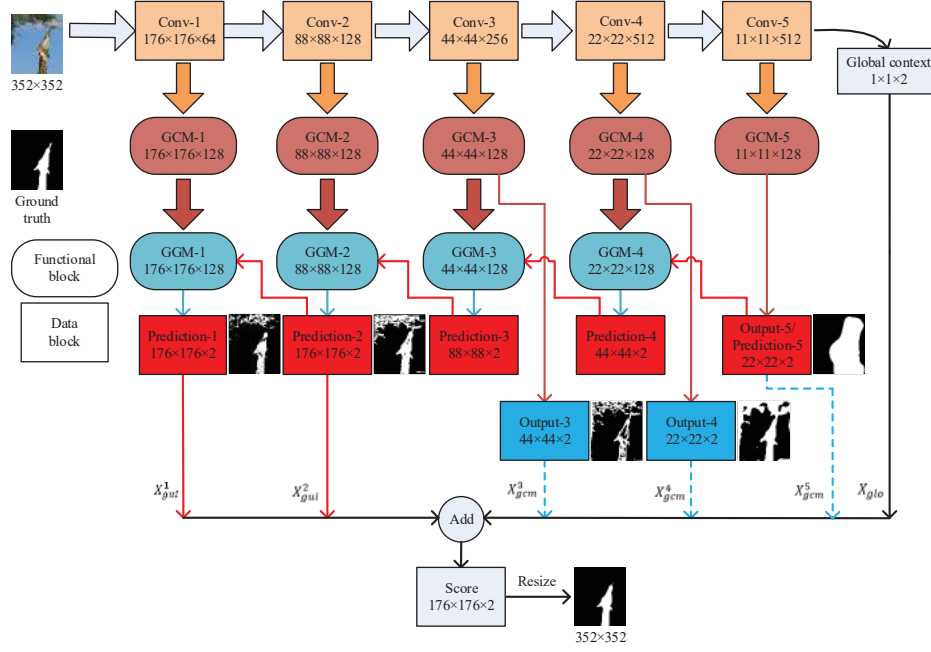
Fig. 4: The framework of the proposed salient object detection network. It is noted that some basic operations are omitted for clarity, such as convolution and deconvolution. The top row shows the backbone network (i.e., VGG16), which generates 5-level Backbone Feature Maps (BFMs). The second row is the GCM, which aims to achieve high-discriminative feature maps from BFMs. The third row illustrates GGM, which produces the Guided Feature Maps (GFMs). "Prediction-i" and "Output-i" represent the saliency predictions/side outputs obtained by GGM and GCM, respectively. GGM-4 is taken as an example for the description of the proposed guidance strategy. "GGM-4" takes the feature maps obtained by GCM-4 and "Prediction-5" as inputs. The deeper-level side output, i.e., "Prediction-5", is used as a guidance feature map to guide the shallower-level feature maps obtained by "GCM-4" through concatenation.

GCM, and the backbone network. The framework will be discussed in detail in the following.

A. Backbone network

Following the previous works [15], [16], [19], [23], [25], VGG16 network [56] is chosen as the backbone network in this paper. Considering VGG16 is originally proposed for image classification [56], we modify it to serve our purpose. Firstly, the last three fully connected networks of VGG16 [56] are removed. Secondly, the input image is cropped to 352×352 instead of 242×242 for keeping more image details. The output of the proposed network is 176×176, which is resized to 352×352 pixels with a bilinear interpolation. Different layers learn different levels of convolutional feature maps, i.e., Conv-1, Conv-2, Conv-3, Conv-4, and Conv-5 in our study, which are denoted as $\{\mathbf{F}^i\}$ $(i = 1, 2, 3, 4, 5)$. These feature maps are called Backbone Feature Maps (BFMs). Level-i has D^i-channel feature maps, which are denoted as $\{\mathbf{F}^i\} = \{\mathbf{F}^i_j\}$ $(j = 1, 2, \ldots, D^i)$. Table. I shows the details of the backbone network.

B. Group Convolution Module (GCM) for discriminative feature maps

Most existing CNNs-based methods obtain discriminative feature maps by the Traditional Convolution Module (TCM), which performs convolutions across all the channels of the

TABLE I: Details of the backbone network.

Block	Layer	Kernel	Stride	Zero padding	Output
Conv-1	2 conv	3*3	1	Yes	352*352*64
	max-pool	2*2	2	Yes	176*176*64
Conv-2	2 conv	3*3	1	Yes	176*176*128
	max-pool	2*2	2	Yes	88*88*128
Conv-3	3 conv	3*3	1	Yes	88*88*256
	max-pool	2*2	2	Yes	44*44*256
Conv-4	3 conv	3*3	1	Yes	44*44*512
	max-pool	2*2	2	Yes	22*22*512
Conv-5	3 conv	3*3	1	Yes	22*22*512
	max-pool	2*2	2	Yes	11*11*512

BFMs. However, the salient features may be drowned amongst the BFMs. This will lead to that the feature maps are not discriminative enough to distinguish the salient object from the complicated background (as illustrated in Fig. 3(a)). To address this, we propose a Group Convolution Module (GCM) as described below.

Step 1: Split BFMs into numerous groups. The BFMs of each level (except level-1), i.e., $\{\mathbf{F}^i\}$ $(i = 2, 3, 4, 5)$, are first empirically split into 128 non-overlapped groups $\{\mathbf{G}^i_j\}$ $(j = 1, 2, \ldots, 128)$. Each group consists of several BFMs, i.e., $\mathbf{G}^i_1 = \left(\mathbf{F}^i_1, \ldots, \mathbf{F}^i_{g^i}\right)$, $\mathbf{G}^i_2 = \left(\mathbf{F}^i_{g^i+1}, \ldots, \mathbf{F}^i_{2*g^i}\right), \ldots$, $\mathbf{G}^i_{128} = \left(\mathbf{F}^i_{127*g^i+1}, \ldots, \mathbf{F}^i_{128*g^i}\right)$, where $g^i = \frac{D^i}{128}$ is the number of BFMs within each group at

TABLE II: Details of the proposed GCM. Column 2: Input of the corresponding block; Column 3 to Column 8: Details of each group at each level; Column 9: Concatenation of outputs of all the groups.

Block	Input	Split	Layer	Kernel	Stride	Zero padding	Output	Concat
GCM-1	Conv-1 (176*176*64)	176*176*1	1 conv	3*3	1	Yes	176*176*2	Group-Conv-1 (176*176*128)
GCM-2	Conv-2 (88*88*128)	88*88*1	1 conv	3*3	1	Yes	88*88*1	Group-Conv-2 (88*88*128)
GCM-3	Conv-3 (44*44*256)	44*44*2	1 conv	3*3	1	Yes	44*44*1	Group-Conv-3 (44*44*128)
GCM-4	Conv-4 (22*22*512)	22*22*4	1 conv	3*3	1	Yes	22*22*1	Group-Conv-4 (22*22*128)
GCM-5	Conv-5 (11*11*512)	11*11*4	1 conv	3*3	1	Yes	11*11*1	Group-Conv-5 (11*11*128)

level-i. It is noted that $\{\mathbf{F}^1\}$ is divided into 64 groups, each of which consists of 1 BFM.

Step 2: Generate discriminative feature maps within each group. Convolutions are performed across all the channels of the BFMs within each group, i.e.,

$$\mathbf{A}_j^i = Conv(\mathbf{G}_j^i, d), \tag{1}$$

where $Conv$ denotes the convolution operation and d is the channel number of the output feature maps. Eq. (1) will obtain d-channel discriminative feature maps. In this paper, d is set to 1 for level-2, level-3, level-4, level-5, and 2 for level-1. In this way, there will be 128 channels of discriminative feature maps for each level.

Step 3: Concatenate all the discriminative feature maps. These feature maps are concatenated together at each level, i.e.,

$$\mathbf{A}^i = Concat(\{\mathbf{A}_j^i\})\{j = 1, 2, \ldots, 128\}, \tag{2}$$

where $Concat$ denotes the concatenation operation. Thus, we achieve 128 discriminative feature maps for level-i. Table. II illustrates the details of the proposed GCM.

To be more specific, Fig. 5 takes GCM-4 as an example to illustrate the proposed GCM. 512-channel BFMs of Conv-4 are first split into 128 groups, each of which consists of 4 channels of BFMs. Then, as in Eq. (1), a convolution with the kernel of 3×3 is performed across the 4-channel BFMs within each group to achieve 1 discriminative feature map. In this way, 128 discriminative feature maps will be obtained from 128 non-overlapped groups. Finally, these individual feature maps are concatenated by Eq. (2) to obtain 128-channel feature maps $\mathbf{A}^4$, i.e., Group-Conv-4. Similarly, Group-Conv5, Group-Conv3, and Group-Conv-2 will be obtained by the proposed GCM from Conv-5, Conv-3, and Conv-2, respectively. When computing Group-Conv-1, 2 convolutions with kernels of 3×3 are performed within each group to achieve 2 discriminative feature maps for each group. Following this, 64 groups will produce 128 discriminative feature maps, which are then concatenated to obtain 128-channel feature maps, i.e., Group-Conv-1.

The proposed GCM carries out the convolution operation across several channels of BFMs within each group, rather than all the channels of BFMs, as adopted by TCM. For TCM, the salient features must be protruded out from a large number of feature maps. This may be difficult because the salient features can be easily drown amongst feature maps. On the contrary, the salient features can be easily protruded out from several feature maps within each group in the proposed

GCM. As shown in Fig. 3, the feature maps computed by our proposed GCM are more discriminative than those obtained by TCM.
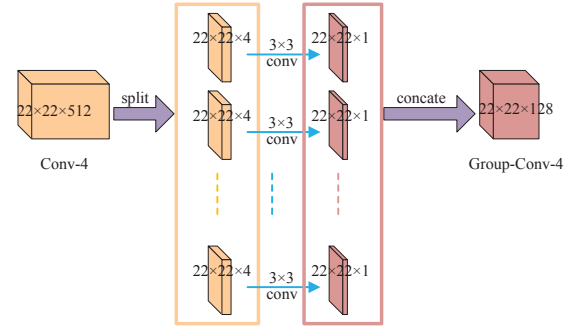

Fig. 5: Illustrations of the proposed GCM by taking GCM-4 as an example. The 512-channel BFMs of Conv-4 are first empirically divided into 128 non-overlapped groups, each of which consists of $512/128 = 4$ channels of BFMs. Then, a convolution with the kernel of 3×3 is performed across the 4-channel BFMs within each group to derive a discriminative feature map for each group. In such way, 128 non-overlapped groups will produce 128 discriminative feature maps, which are concatenated to form the 128-channel feature maps Group-Conv-4.

From the perspective of elements involved in the convolution, our proposed GCM clearly differs from the partial convolution [57]. The partial convolution [57] introduces a binary mask into the standard convolution operation, where only those pixels masked by 1 are counted in the convolution. While our proposed GCM divides all the channels of feature maps into a few groups along the channel dimension, on each of which the convolution is implemented. Different from the partial convolution [57] that just leverages parts of the input features, GCM comprehensively explores all the input features to obtain more distinctive information.

C. Guidance strategy for integration of multi-level contextual information

As pointed out previously [15], [16], [19], deep layers are prone to extract semantic knowledge for localizing the salient objects, while shallow layers tend to preserve low-level spatial details that can better detect object boundaries. Here, we propose a novel guidance strategy to jointly employ feature maps and side outputs for integration of multi-level contextual information.

1) Direct Guidance Module (DGM): Given saliency inference, the side outputs provide coarse saliency predictions, which indicate the locations of the salient object and background at a coarse level. Suppose we use the side output of level-i to guide the feature maps of level-$(i-1)$ through concatenation. Here, the side outputs are denoted as $\mathbf{O}^i = \left\{\mathbf{O}^i_{fg}, \mathbf{O}^i_{bg}\right\}$ $(i = 1, 2, 3, 4, 5)$, where $\mathbf{O}^i_{fg}$ denotes the foreground probability and $\mathbf{O}^i_{bg}$ denotes the background probability. Usually, we employ the deeper-level foreground probability, e.g., $\mathbf{O}^i_{fg}$, to guide the generation of shallower-level feature maps. Specifically, if the foreground probability of $\mathbf{O}^i_{fg}$ at the ith level is represented as the prediction $\mathbf{P}^i$, the Guided Feature Maps (GFMs) of level-i (i.e., Guided-Conv-i, $i = 1, 2, 3, 4$) can be achieved by the following steps.

Step 1: Concatenate the deeper-level prediction with the shallower-level feature maps.

$$ConD^i = Concat(\mathbf{P}^{i+1}, \mathbf{A}^i)(i = 1, 2, 3, 4). \quad (3)$$

Step 2: Perform convolutions on the concatenated feature maps.

$$gui\mathbf{A}^i = Conv(ConD^i)(i = 1, 2, 3, 4). \quad (4)$$

$gui\mathbf{A}^i$ is the expected GFMs of level-i.

In this guidance strategy, the deeper-level side output is used to directly guide the shallower-level feature maps, which we call Direct Guidance Module (DGM).

Fig. 6 shows an example of the DGM by taking Guide-Conv-4 as an example. The deeper-level side output, i.e., Prediction-5, is first upsampled by a deconvolution with the stride of 2 into the same resolution with the shallower-level feature maps, i.e., Group-Conv-4, and then is concatenated with 128-channel feature maps of Group-Conv-4, resulting in 129-channel feature maps. Finally, a convolution with the kernel of 3×3 is performed on these 129-channel feature maps to achieve 128-channel GFMs, i.e., Guide-Conv-4, which are processed subsequently for saliency prediction, i.e, Prediction-4. Following this way, Group-Conv-3, Group-Conv-2, and Group-Conv-1 are successively guided by Prediction-4, Prediction-3, and Prediction-2, respectively.
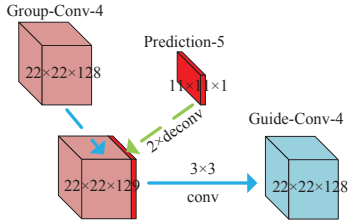

Fig. 6: Illustrations of the proposed DGM by taking Guide-Conv-4 as an example. The deeper-level side output, i.e., Prediction-5, is first upsampled by a deconvolution with the stride of 2 into the same resolution with the shallower-level feature maps, i.e., Group-Conv-4, and then is concatenated with 128-channel feature maps of Group-Conv-4, resulting in 129-channel feature maps. Finally, a convolution with the kernel of 3×3 is performed on these 129-channel feature maps to achieve 128-channel feature maps, i.e., Guide-Conv-4.

2) Group Guidance Module (GGM): Considering the effectiveness of the GCM, we propose to embed it into the proposed baseline of guidance strategy, i.e., DGM, to further promote the guidance role. We call this Group Guidance Module (GGM), which is illustrated in Algorithm 1. $gui\mathbf{A}^i$ in Algorithm 1 is the expected GFMs of level-i.

Algorithm 1 Group Guidance Module (GGM)

Step 1: Split the shallower-level feature maps into groups.

$$gui\mathbf{G}^i_1 = \left\{\mathbf{A}^i_1, \ldots, \mathbf{A}^i_{4*1}\right\}$$
$$gui\mathbf{G}^i_2 = \left\{\mathbf{A}^i_{4*1+1}, \ldots, \mathbf{A}^i_{4*2}\right\}$$
$$\cdots$$
$$gui\mathbf{G}^i_{32} = \left\{\mathbf{A}^i_{4*31+1}, \ldots, \mathbf{A}^i_{4*32}\right\}. \quad (5)$$

Step 2: Concatenate the deeper-level side output with the shallower-level feature maps of each group.

$$Con\mathbf{G}^i_k = Concat(\mathbf{P}^{i+1}, gui\mathbf{G}^i_k)$$
$$(i = 1, 2, 3, 4; k = 1, 2, \ldots, 32). \quad (6)$$

Step 3: Perform a convolution with the kernel of 3×3 on the concatenated feature maps to achieve a GFM within each group.

$$gui\mathbf{A}^i_k = Conv(Con\mathbf{G}^i_k)(i = 1, 2, 3, 4; k = 1, 2, \ldots, 32). \quad (7)$$

Step 4: Concatenate the GFMs of all the groups.

$$gui\mathbf{A}^i = Concat(gui\mathbf{A}^i_k)(k = 1, 2, \ldots, 32). \quad (8)$$
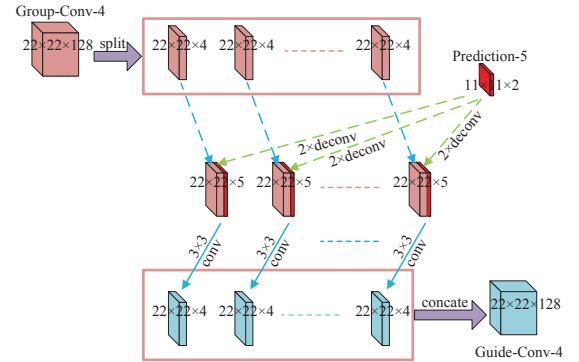

Fig. 7: Illustrations of the GGM by taking Guide-Conv-4 as an example. The 128-channel feature maps of Group-Conv-4 are first empirically split into 32 non-overlapped groups, each of which consists of 4-channel feature maps. Then, the deeper-level side output, i.e., Prediction-5, is used as a guidance feature map to be concatenated with the 4-channel feature maps within each group to achieve 5-channel feature maps for each group. A convolution with the kernel of 3×3 is performed on the 5-channel feature maps within each group to achieve 4-channel feature maps for each group. Following this way, we will achieve 32 4-channel feature maps. These feature maps are finally concatenated together to obtain Guide-Conv-4.

Fig. 7 shows an example for GGM by taking Guide-Conv-

TABLE III: Details of the proposed GGM. Column 2: Inputs of the corresponding blocks; Column 3 to Column 8: Details of each group at each level; Column 9: Concatenation of outputs of all the groups. To avoid over-fitting, a *dropout* layer is added after the "Concat" layer within the blocks of "Guide-Conv-1", "Guide-Conv-2", "Guide-Conv-3", and "Guide-Conv-4", which are not displayed in Table.

Block	Input	Split	Layer	Kernel	Stride	Zero padding	Output	Concat
GGM-1	Group-Conv-1 (176*176*128) $\mathbf{O}^2_{fg}$ (176*176*1)	176*176*4 -	Concat 1 conv 1 conv	1*1 1*1	1 1	Yes Yes	176*176*5 176*176*5 176*176*4	Guide-Conv-1 (176*176*128)
GGM-2	Group-Conv-2 (88*88*128) $\mathbf{O}^3_{fg}$ (88*88*1)	88*88*4 -	Concat 1 conv 1 Decov 1 conv	1*1 5*5 1*1	1 2 1	Yes Yes Yes	88*88*5 88*88*5 176*176*16 176*176*4	Guide-Conv-2 (176*176*128)
GGM-3	Group-Conv-3 (44*44*128) $\mathbf{O}^4_{fg}$ (44*44*1)	44*44*4 -	Concat 1 conv 1 Decov 1 conv	1*1 5*5 1*1	1 2 1	Yes Yes Yes	44*44*5 44*44*5 88*88*12 88*88*4	Guide-Conv-3 (88*88*128)
GGM-4	Group-Conv-4 (22*22*128) $\mathbf{O}^5_{fg}$ (22*22*1)	22*22*4 -	Concat 1 conv 1 Decov 1 conv	1*1 5*5 1*1	1 2 1	Yes Yes Yes	22*22*5 22*22*5 44*44*8 44*44*4	Guide-Conv-4 (44*44*128)

4 as an example. The 128-channel feature maps of Group-Conv-4 are first split into 32 groups, each of which consists of 4-channel feature maps. Then, the deeper-level side output, i.e., Prediction-5, is used as a guidance feature map to be concatenated with the 4-channel feature maps within each group to achieve 5-channel feature maps for each group. A convolution with the kernel of 3×3 is performed on the 5-channel feature maps within each group to achieve 4-channel GFMs for each group. In this way, we will derive 32 4-channel GFMs. These GFMs are finally concatenated together to obtain Guide-Conv-4, which is further processed for saliency prediction, i.e., Prediction-4. Similarly, Guide-Conv-3, Guide-Conv-2, and Guide-Conv1 are successively guided by Prediction-4, Prediction-3, and Prediction-2, respectively.

For DGM, the guidance role of the deeper-level side output could be weak because it can be easily submerged in a flood of shallower-level feature maps. Differently, GGM has a stronger guidance role for the deeper-level prediction due to the embedding of GCM, which is helpful for promoting the guidance role of the deeper-level prediction within each group. In view of the above discussion, GGM is chosen as the guidance strategy in this paper. Table. III presents the details of the proposed GGM. To avoid over-fitting, a dropout layer is added after the "Concat" layer within the blocks of "Guide-Conv-1", "Guide-Conv-2", "Guide-Conv-3", and "Guide-Conv-4", which is not displayed in Table. III for clarity.

Our proposed GGM differs from the SMP module [15] in two aspects. First, SMP in Amulet [15] achieves the level-i prediction by performing a deconvolution and a convolution on the level-i feature maps and level-$(i + 1)$ prediction, respectively, and then adding them up. In contrast, our proposed GGM exploits the level-$(i+1)$ prediction as a guidance feature map to guide the level-i feature maps via the concatenation operation. Since the deeper-level prediction can coarsely locate the salient object, our guidance strategy can help the shallower-level feature maps learn more accurate properties of the salient object. As well, with the aid of fine details provided by the

shallower-level feature maps, the proposed guidance strategy enables the deeper-level prediction to be well propagated into their high-resolution versions. Furthermore, instead of simply concatenating the feature maps and prediction, we embed the proposed GCM in the guidance strategy, which promotes the guidance role of the deeper-level prediction. Secondly, in view of the fact that the desired saliency map is essentially the foreground prediction, it is better to use only the foreground prediction to guide the feature maps, which is applied in our proposed GGM module. However, SMP in Amulet [15] combines both the foreground prediction and background prediction with the feature maps, which may lead to some noises due to the involvement of the background prediction. To sum up, compared with SMP in Amulet [15], our proposed GGM can better explore the complementarity of feature maps and prediction.

3) Analysis of the contextual information guidance strategy: We analyze the propagation formulations behind the contextual information guidance strategy based on DGM.

Eq. (4) is rewritten as

$$\mathbf{x}^{gui}_i = Conv\left(Concat\left(\mathbf{P}^{i+1}, \mathbf{x}_i\right)\right), \qquad (9)$$

where $\mathbf{x}^{gui}_i$ and $\mathbf{x}_i$ represent the input of DGM at level-i and the output of GCM at level-i, respectively.

Denoting the loss function as ε, from the chain rule of backpropagation [58], we have

$$\frac{\partial \varepsilon}{\partial \mathbf{x}_i} = \frac{\partial \varepsilon}{\partial \mathbf{x}^{gui}_i} \frac{\partial \mathbf{x}^{gui}_i}{\partial \mathbf{x}_i}$$
$$= \frac{\partial \varepsilon}{\partial \mathbf{x}^{gui}_i} \left(\frac{\partial Conv\left(\mathbf{x}_i\right)}{\partial \mathbf{x}_i} + \frac{\partial \mathbf{P}^{i+1}}{\partial \mathbf{x}_i}\right). \qquad (10)$$

Eq. (10) indicates that the gradient $\frac{\partial \varepsilon}{\partial \mathbf{x}_i}$ exhibits some favorable properties. **(i)** $\frac{\partial \varepsilon}{\partial \mathbf{x}^{gui}_i} \frac{\partial Conv(\mathbf{x}_i)}{\partial \mathbf{x}_i}$ propagates information directly at the current level-i without concerning any other stages of deep features. **(ii)** $\frac{\partial \varepsilon}{\partial \mathbf{x}^{gui}_i} \frac{\partial \mathbf{P}^{i+1}}{\partial \mathbf{x}_i}$ propagates information through the deeper prediction of level-$(i + 1)$, which ensures the guidance of deeper-level prediction for the

shallower-level feature maps. Additionally, the second term avoids the loss of semantic knowledge to some extent as well.

D. Saliency inference

In this section, we introduce how to produce the saliency map based on the proposed GCM and GGM.

Supposing we have three deeper-level side outputs generated by GCM and two shallower-level predictions produced by GGM, where the former three are denoted as $\mathbf{X}_{gcm}^3$, $\mathbf{X}_{gcm}^4$, $\mathbf{X}_{gcm}^5$, and the latter two are denoted as $\mathbf{X}_{gui}^1$, $\mathbf{X}_{gui}^2$, these 5 side outputs are added together to produce the final saliency map. To better recover the spatial details, a series of deconvolutional layers with kernels of 5×5 and strides of 2 are performed on the low-resolution side outputs to achieve high-resolution versions.

In addition, "Conv-5" can well capture the global context. Therefore, we also add the side output of "Conv-5" to the final saliency map, denoted as $\mathbf{X}_{glo}$. Similar to [16], three convolutional layers are added after "Conv-5" to achieve the global context. One more convolution is further added to obtain the global context output.

The final saliency map is computed as a linear combination of $\mathbf{X}_{gui}^1$, $\mathbf{X}_{gui}^2$, $\mathbf{X}_{gcm}^3$, $\mathbf{X}_{gcm}^4$, $\mathbf{X}_{gcm}^5$, and $\mathbf{X}_{glo}$ using 6 linear operators $\left(\mathbf{W}_{gui^1}, \mathbf{b}_{gui^1}\right)$, $\left(\mathbf{W}_{gui^2}, \mathbf{b}_{gui^2}\right)$, $\left(\mathbf{W}_{gcm^3}, \mathbf{b}_{gcm^3}\right)$, $\left(\mathbf{W}_{gcm^4}, \mathbf{b}_{gcm^4}\right)$, $\left(\mathbf{W}_{gcm^5}, \mathbf{b}_{gcm^5}\right)$, and $\left(\mathbf{W}_{glo}, \mathbf{b}_{glo}\right)$, where $\mathbf{W}_*$ and $\mathbf{b}_*$ represent the parameters of weights and biases. The softmax function is used to compute the probability for each pixel of being salient or not, i.e.,

$$\hat{y}(\mathbf{v}_i) = p\left(y(\mathbf{v}_i) = c\right) = \frac{e^{\Phi(\mathbf{v}_i)}}{\sum_{c' \in \{0,1\}} e^{Z(\mathbf{v}_i)}}, \qquad (11)$$

where

$$\begin{aligned}\Phi\left(\mathbf{v}_i\right) = \\ \mathbf{W}_{gui^1}^c \mathbf{X}_{gui}^1(\mathbf{v}_i) + \mathbf{b}_{gui^1}^c + \mathbf{W}_{gui^2}^c \mathbf{X}_{gui}^2(\mathbf{v}_i) + \mathbf{b}_{gui^2}^c + \\ \mathbf{W}_{gcm^3}^c \mathbf{X}_{gcm}^3(\mathbf{v}_i) + \mathbf{b}_{gcm^3}^c + \mathbf{W}_{gcm^4}^c \mathbf{X}_{gcm}^4(\mathbf{v}_i) + \mathbf{b}_{gcm^4}^c + \\ \mathbf{W}_{gcm^5}^c \mathbf{X}_{gcm}^5(\mathbf{v}_i) + \mathbf{b}_{gcm^5}^c + \mathbf{W}_{glo}^c \mathbf{X}_{glo}(\mathbf{v}_i) + \mathbf{b}_{glo}^c,\end{aligned}$$
$$(12)$$

and

$$\begin{aligned}Z\left(\mathbf{v}_i\right) = \\ \mathbf{W}_{gui^1}^{c'} \mathbf{X}_{gui}^1(\mathbf{v}_i) + \mathbf{b}_{gui^1}^{c'} + \mathbf{W}_{gui^2}^{c'} \mathbf{X}_{gui}^2(\mathbf{v}_i) + \mathbf{b}_{gui^2}^{c'} + \\ \mathbf{W}_{gcm^3}^{c'} \mathbf{X}_{gcm}^3(\mathbf{v}_i) + \mathbf{b}_{gcm^3}^{c'} + \mathbf{W}_{gcm^4}^{c'} \mathbf{X}_{gcm}^4(\mathbf{v}_i) + \mathbf{b}_{gcm^4}^{c'} + \\ \mathbf{W}_{gcm^5}^{c'} \mathbf{X}_{gcm}^5(\mathbf{v}_i) + \mathbf{b}_{gcm^5}^{c'} + \mathbf{W}_{glo}^{c'} \mathbf{X}_{glo}(\mathbf{v}_i) + \mathbf{b}_{glo}^{c'}.\end{aligned}$$
$$(13)$$

In Eq. (11), $p\left(\cdot\right)$ computes the probability, where $c \in \{0, 1\}$ and $c' \in \{0, 1\}$. In Eq. (12) and Eq. (13), $\mathbf{v}_i$ represents the location of pixel i. $y(\mathbf{v}_i)$ and $\hat{y}(\mathbf{v}_i)$ represent the ground truth and the predicted saliency value of the pixel i, respectively.

Similar to [16], we adopt the joint loss function by employing the cross-entropy loss and IoU Boundary loss. The cross-entropy loss function is defined as

$$CE\left(\mathbf{v}\right) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c \in \{0,1\}} \left(y(\mathbf{v}_i) = c\right) \left(\log\left(\hat{y}(\mathbf{v}_i) = c\right)\right).$$
$$(14)$$

The IoU boundary loss is defined as

$$IoU\ Loss = 1 - \frac{2|C_j \cap \hat{C}_j|}{|C_j| + |\hat{C}_j|}, \qquad (15)$$

where $\hat{C}_j$ and C_j are the gradient magnitudes of saliency map and the ground truth corresponding to region j, respectively. The gradient magnitude is computed by using a Sobel operator followed by a tanh activation on the saliency map. $|\cdot|$ represents the number of non-zero entries in a mask.

The joint loss function is

$$Joint\ Loss = CE + IoU\ Loss. \qquad (16)$$

Following [19], the CRF method in [59] is adopted for further smoothness.

IV. EXPERIMENTS

In this section, we evaluate our proposed salient object detection network, and compare it with a number of state-of-the-art (SOTA) methods on three public benchmark datasets. Besides, some ablation experiments are performed to illustrate the effectiveness of the proposed GCM and guidance strategy in our method.

A. Experimental setup

1) Datasets: We evaluate the proposed method on three benchmark datasets, including ECSSD [60], DUT-OMRON [12], and HKU-IS [22]. ECSSD [60] contains 1000 images with multiple salient objects and structurally complex scenes. DUT-OMRON [12] is composed of 5168 images, each of which contains one or more salient objects with cluttered backgrounds. HKU-IS [22] includes 4447 images with multiple low-contrast salient objects. This dataset has been split into 2500 training images, 500 validation images, and 1447 test images, and we fairly evaluate our method and the SOTA methods on the test set (i.e., HKU-IS-TE) of this dataset.

2) Evaluation metrics: We apply multiple widely used evaluation metrics to evaluate the proposed method, including precision-recall curve [61], F-measure curve [61], and Mean Absolute Error (MAE) [62]. Given a continuous saliency map S, a binary mask B is achieved by thresholding. Precision is defined as $\mathrm{Pr}ecision = \frac{|B \cap G|}{|B|}$, and recall is defined as $\mathrm{Re}call = \frac{|B \cap G|}{|G|}$, where G is the corresponding ground truth. The PR curve is plotted by numerous pairs of precision and recall under different thresholds.

The F-measure metric is defined as

$$F_\beta = \frac{\left(1 + \beta^2\right) \mathrm{Pr}ecision \times \mathrm{Re}call}{\beta^2 \mathrm{Pr}ecision + \mathrm{Re}call}, \qquad (17)$$

where $\beta^2 = 0.3$, as suggested in [61]. The F-measure curve is plotted by 255 F-measure values, which are computed by 255 pairs of precision and recall values under 255 thresholds.

MAE is defined as

$$MAE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |S\left(i, j\right) - G\left(i, j\right)|, \qquad (18)$$

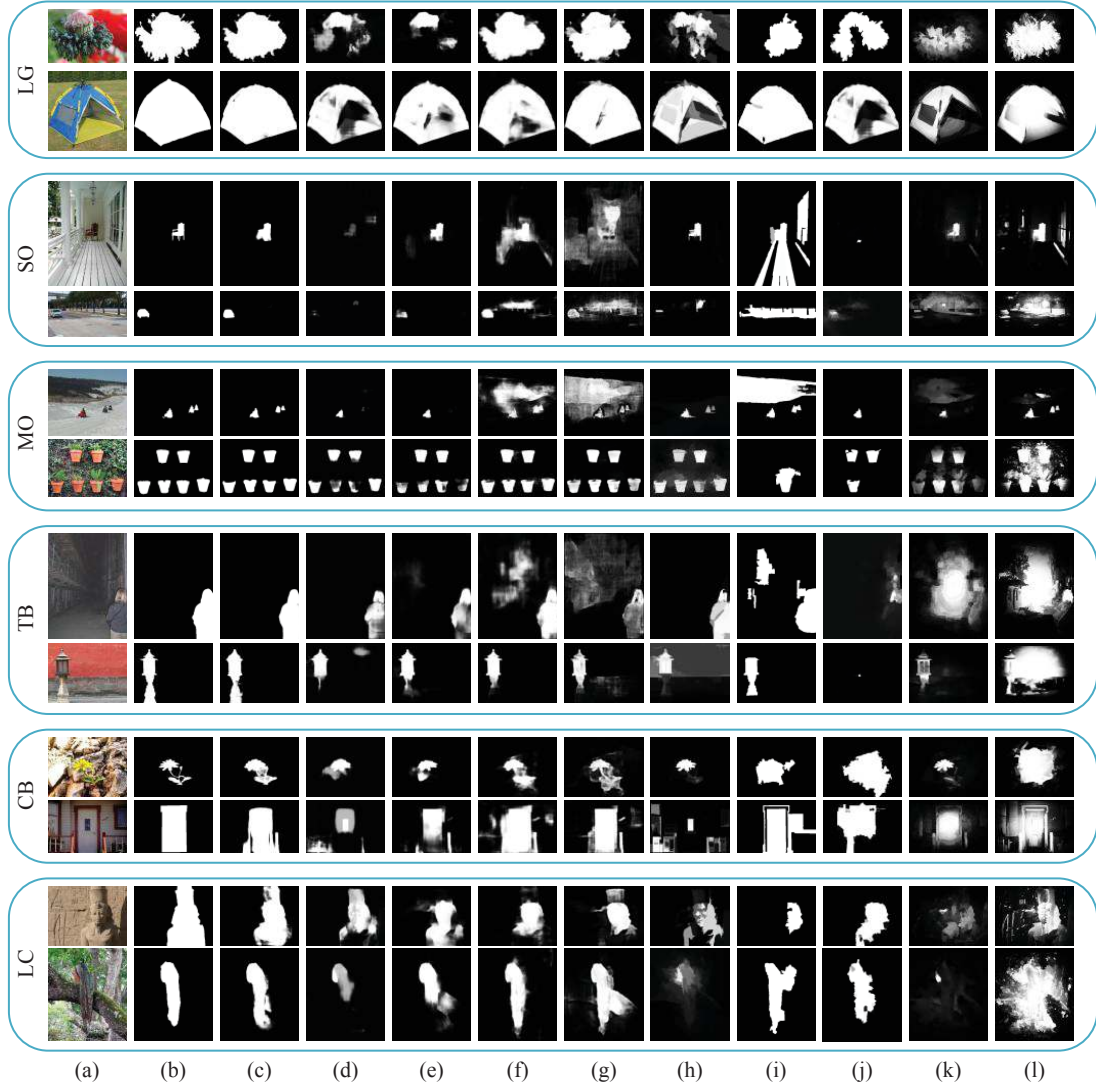where W and H represent the width and height of the input image, respectively.

Fig. 8: Visual comparisons of different methods. (a) Original images; (b) Ground truth; (c) OUR; (d) DCL [19]; (e) NLDF [16]; (f) Amulet [15]; (g) UCF [27]; (h) MDF [22]; (i) ELE [63]; (j) DLS [26]; (k) DSR [64]; (l) MST [65]. (d)-(f): R-SOTA methods. (g)-(l): G-SOTA methods.

3) Implementation details: The proposed model is implemented in Tensorflow [66]. The weights in the backbone, i.e., the VGG16 architecture, are initialized with the pretrained weights of VGG16 [56]. The other weights are initialized randomly with a truncated normal ($\sigma = 0.01$), and the biases are initialized to 0. The Adam optimizer [67] is used to train our model with an initial learning rate of 10^6, $\beta_1 = 0.9$, and $\beta_2 = 0.999$.

The MSRA10K [68] dataset, which contains 10000 images with high contrast, is used to train the proposed salient object detection network. Horizontal flipping is used for data augmentation. The inputs are resized to 352×352. With a NVIDIA Titan X GPU, it takes about 2 hours to finish the whole training procedure for 1 epoch with a single image batch size, which is due to the large training data. We take the trained model of epoch 17 as the test model. The test time for an image is about 0.08 seconds. For the training of GGM for each input image, we first compute the Prediction-5 by the

deepest-level feature maps of GCM-5. Prediction-5 is used as a guidance feature map to guide the feature maps of GCM-4. Those feature maps of GCM-5 are not guided anymore because they are at the deepest layer.

B. Comparisons with SOTA methods

In this section, we compare our method with 6 General-SOTA (G-SOTA) methods, including 4 CNN-based methods (UCF [27], MDF [22], ELE [63] and DLS [26]) and 2 traditional methods (DSR [64] and MST [65]). Besides, the proposed method is also compared with 3 Relative-SOTA (R-SOTA) CNN-based methods (NLDF [16], Amulet [15], and DCL [19]) that are based on the integration of multi-level contextual information.

1) Visual comparisons: Fig. 8 shows the visual comparisons of the proposed method with the SOTA methods on multiple difficult cases, including large object (LO), small object (SO), multiple objects (MO), object touching the im-

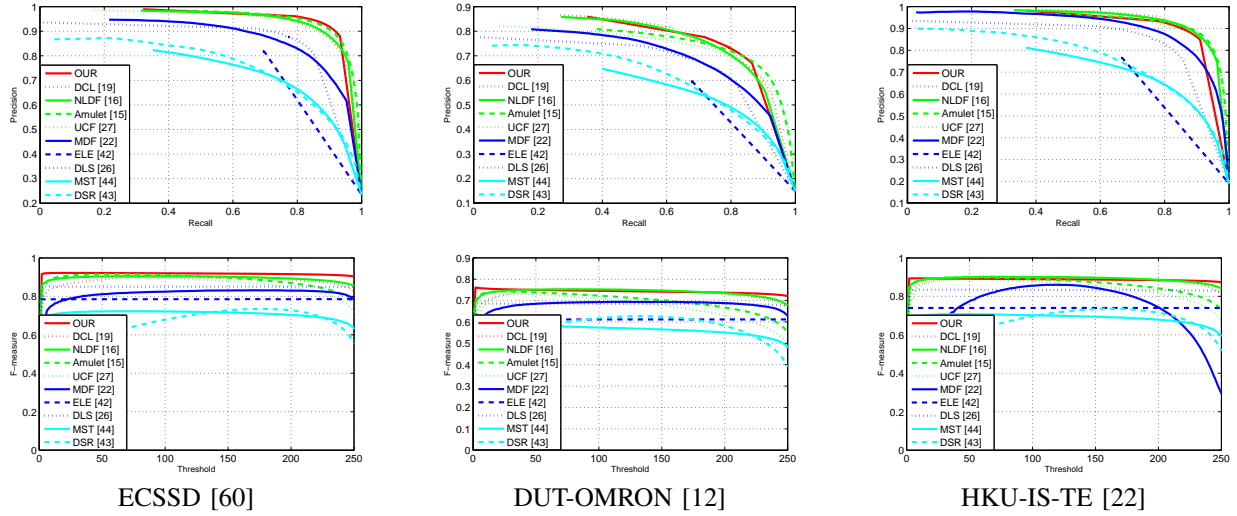ECSSD [60]　　　　　　DUT-OMRON [12]　　　　　　HKU-IS-TE [22]

Fig. 9: PR and F-measure curves of different methods.

TABLE IV: MAE of different methods for (a) ECSSD [60], (b) DUT-OMRON [12], and (c) HKU-IS-TE [22].

(a) ECSSD [60]

Methods	Ours	DCL [19]	NLDF [16]	Amulet [15]	UCF [27]	MDF [22]	ELE [63]	DLS [26]	MST [65]	DSR [64]
MAE	**0.0494**	0.0747	0.0626	0.0589	0.0691	0.1050	0.1201	0.0860	0.1568	0.1715

(b) DUT-OMRON [12]

Methods	Our	DCL [19]	NLDF [16]	Amulet [15]	UCF [27]	MDF [22]	ELE [63]	DLS [26]	DSR [64]	MST [65]
MAE	**0.0746**	0.0863	0.0796	0.0976	0.1203	0.0916	0.1215	0.0895	0.1609	0.1388

(c) HKU-IS-TE [22]

Methods	Ours	DCL [19]	NLDF [16]	Amulet [15]	UCF [27]	MDF [22]	ELE [63]	DLS [26]	DSR [64]	MST [65]
MAE	**0.0466**	0.0552	0.0480	0.0501	0.0612	0.1292	0.1118	0.0696	0.1382	0.1409

age boundary (TB), complicated background (CB), and low contrast between foreground and background (LC).

Taking into account all the mentioned cases in Fig. 8, it can be easily seen that our proposed method can high-light the whole salient object(s) with satisfactory uniformity. Specifically, the proposed method can detect salient objects with different sizes wholly. In contrast, the G-SOTA methods just detect parts of the large salient object or even fail at identifying the small one (as shown in Fig. 8(d), (e), and (k) for the group of SO). For those images with multiple objects, some of the G-SOTA methods miss some salient objects, which can be well solved by the proposed method. This is because the G-SOTA methods ignore the multi-size/multi-level contextual information[2], which easily misses different-size objects and even mis-detects some salient objects in the case of multiple objects. Those salient objects touching the boundary can also be accurately detected by the proposed method, but the previous G-SOTA methods perform poorly in this case. This is attributed to the boundary prior (DSR [64] and MST [65]) that assumes the image boundary as background, and the lack of comprehensively global information (UCF [27], MDF [22], ELE [63], and DLS [26]). Especially, those

salient objects in images with complicated background and low contrast are challenging for the previous G-SOTA methods based on the fact that the G-SOTA methods just detect parts of salient objects or fail at identifying them. This is owing to the fact that the G-SOTA methods cannot extract features discriminative enough to distinguish the unobtrusive salient objects. Fortunately, our method can still highlight the salient objects with good uniformity.

Moreover, compared with the R-SOTA methods, the proposed method achieves much better wholeness, foreground uniformity, and background suppression for the salient object, and better robustness to salient objects with different sizes. As illustrated in Fig. 2 and Fig. 8, the proposed method can overcome the limitations of the existing practices for integration of multi-level feature maps and side outputs. To be specific, compared with the traditional multi-level information integration (NLDF [16], DCL [19], and Amulet [15]), the proposed method can evade the misleading information and thus predict complete salient object with better background suppression, and make up those parts of salient objects missed by DCL [19], which is owing to correction by the rich feature maps of the shallow layers.

2) Quantitative comparisons: Fig. 9 illustrates the PR and F-measure curves of different methods. Table. IV displays MAE values of the proposed method and the compared ones.

---

[2]MDF [22] achieves the multi-size information through cropping the input image, which will crop out the salient object and thus can not detect it.

It can be easily seen that the proposed method achieves best performance in terms of all the evaluation metrics for ECSSD [60] and DUT-OMRON [12]. For HKU-IS-TE [22], the proposed method performs best with respect to the F-measure curves and MAE, and slightly worse than DCL [19] with respect to the PR curves. Actually, the images of ECSSD [60] and DUT-OMRON [12] are more difficult than HKU-IS-TE [22] for salient object detection. Therefore, it is obvious that the proposed method performs more robustly than the previous methods in complex cases, which meets the requirements of the real scene. This is attributed to the guidance strategy, i.e., GGM, that effectively integrates the high-level knowledge and the low-level cues, and the high-contrast features extracted by the proposed GCM.

C. Ablation analysis

In this section, we conduct a serious of ablation analyses to better understand the proposed method.

1) With/without GCM: Fig. 10 and Fig. 11 show the improvements of the proposed GCM from the quantitative and visual perspectives, respectively. It can be easily observed from Fig. 10 that the performance of the backbone is greatly promoted by the proposed GCM. Similar conclusions can be drawn from Fig. 11 as well. As discussed in the previous sections, the proposed GCM can extract the salient features amongst the trivial ones. Therefore, GCM helps detect low-contrast foreground (as shown in the first two rows of Fig. 11) and non-noticeable small-size salient objects (as shown in the last two rows of Fig. 11).
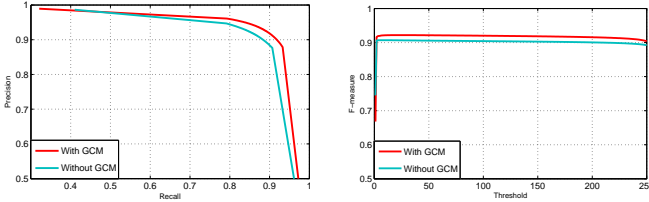

Fig. 10: Illustrations for the performance of GCM on ECSSD [60]. Left: PR curves; Right: F-measure curves.
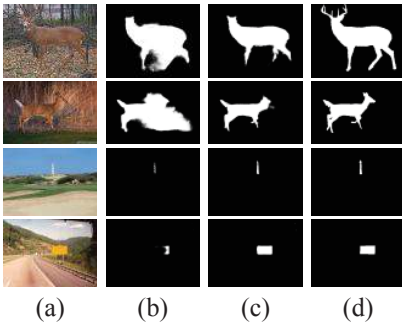

Fig. 11: Illustrations for the effectiveness of GCM. (a) Original images; (b) Without GCM; (c) With GCM; (d) Ground truth.

2) With/without GGM: Fig. 12 and Fig. 13 illustrate the performance of GGM quantitatively and visually, respectively. Seen from Fig. 12 and Fig. 13, it is obvious that GGM improves the performance. GGM provides a more comprehensive integration of multi-level contextual information by using the deeper-level side output to guide the shallower-level feature maps. This cross practice between side outputs and feature maps efficiently combines their advantages. Therefore, GGM provides complete detections for the salient objects (as shown in the first two rows of Fig. 13) and clean background suppression (as shown in the last two rows of Fig. 13).
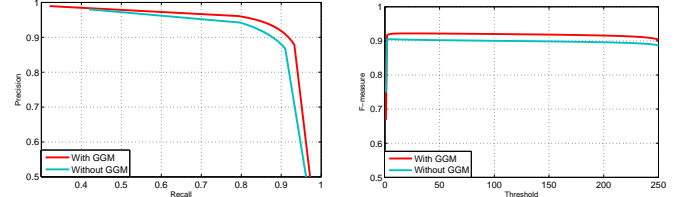

Fig. 12: Illustrations for the performance of GGM on ECSSD [60]. Left: PR curves; Right: F-measure curves.
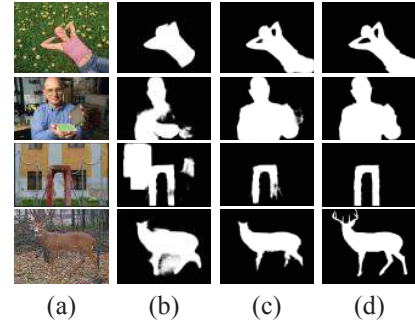

Fig. 13: Illustrations for the effectiveness of GGM. (a) Original images; (b) Without GGM; (c) With GGM; (d) Ground truth.
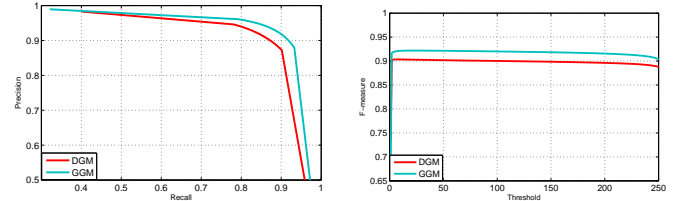

Fig. 14: Quantitative comparisons on ECSSD [60]: DGM vs GGM. Left: PR curves; Right: F-measure curves.

3) Guidance strategy: DGM vs GGM: As discussed in Section III-C, there are a baseline guidance strategy, i.e., DGM, and a comprehensive guidance strategy, i.e., GGM, for the proposed guidance strategy. Fig. 14 and Fig. 15 illustrate the comparisons between DGM and GGM. It can be easily found from Fig. 14 that GGM gets better quantitative performance than DGM. Moreover, it can be also noticed from Fig. 15 that GGM achieves much better foreground wholeness (as shown in the first three rows of Fig. 15) and background suppression (as shown in the last row of Fig. 15)
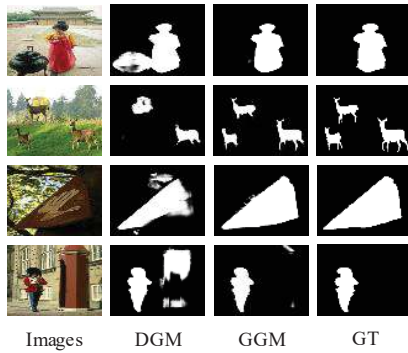
Fig. 15: Visual comparisons: DGM vs GGM.

than DGM. The superiority of GGM over DGM is owing to the embedding of the proposed GCM in GGM, which aggregates the advantage of GCM to further improve the proposed multi-level contextual information integration.

D. Failure cases

Fig. 16 shows some failure cases for our proposed method. The scenes in those images contain complex semantic knowledge. Scene understandings are needed to detect the salient objects within these images, which is challenging for our proposed method. To address this problem, we will take into account the knowledge of scene understanding [69], [70] and scene parsing [71], [72] to improve the performance of our method in the future.
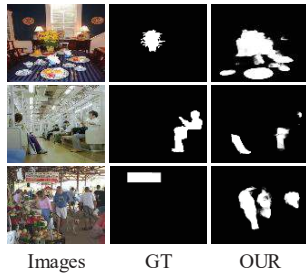

Fig. 16: Some failure cases for our proposed method.

V. CONCLUSIONS

In this paper, we have presented a deep salient object detection network, in which a novel guidance strategy is proposed for effective integration of multi-level contextual information, and a group convolution module is proposed to improve the feature discriminability. Moreover, incorporating the proposed GCM in the contextual information guidance strategy further promotes the guidance role of deeper-level side output for the shallower-level feature maps. In the future, we will integrate scene understanding and scene parsing in our work to improve the performance. Besides, we will apply our salient object detector to facilitate the representation ability of existing deep networks [73], [74] and real-world applications, including image retrieval [75], [76] and image classification [77].

REFERENCES

[1] J. Han, K. N. Ngan, M. Li, and H.-J. Zhang, "Unsupervised extraction of visual attention objects in color images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 141–145, 2006.

[2] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele, "Exploiting saliency for object segmentation from image level labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5038–5047.

[3] J. Han, E. J. Pauwels, and P. De Zeeuw, "Fast saliency-aware multi-modality image fusion," *Neurocomputing*, vol. 111, pp. 70–80, 2013.

[4] P. Wang, J. Wang, G. Zeng, J. Feng, H. Zha, and S. Li, "Salient object detection for searched web images via global saliency," in *Proceedings of IEEE Confernece on Computer Vision and Pattern Recognition*, 2012, pp. 3194–3201.

[5] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 989–1005, 2009.

[6] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706–5722, 2015.

[7] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[8] D. A. Klein and S. Frintrop, "Center-surround divergence of feature statistics for salient object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2214–2219.

[9] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[10] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 33, no. 2, pp. 353–367, 2011.

[11] H. Lu, X. Li, L. Zhang, X. Ruan, and M.-H. Yang, "Dense and sparse reconstruction error based saliency descriptor," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1592–1603, 2016.

[12] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3166–3173.

[13] H. Peng, B. Li, H. Ling, W. Hu, W. Xiong, and S. J. Maybank, "Salient object detection via structured matrix decomposition," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 818–832, 2017.

[14] J. Kim, D. Han, Y.-W. Tai, and J. Kim, "Salient region detection via high-dimensional color transform and local spatial support," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 9–23, 2016.

[15] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision*, 2017, pp. 202–211.

[16] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin, "Non-local deep features for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6609–6617.

[17] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu, "A stagewise refinement model for detecting salient objects in images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4019–4028.

[18] D. Zhang, J. Han, and Y. Zhang, "Supervision by fusion: Towards unsupervised learning of deep salient object detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4048–4056.

[19] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 478–487.

[20] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 678–686.

[21] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1265–1274.

[22] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5455–5463.

[23] G. Lee, Y.-W. Tai, and J. Kim, "Deep saliency with encoded low level distance map and high level features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 660–668.

[24] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3183–3192.

[25] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang, "Deepsaliency: Multi-task deep neural network model for salient object detection," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3919–3930, 2016.

[26] P. Hu, B. Shuai, J. Liu, and G. Wang, "Deep level sets for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 540–549.

[27] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 212–221.

[28] J. Kim and V. Pavlovic, "A shape-based approach for salient object detection using deep learning," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 455–470.

[29] Y. Tang and X. Wu, "Saliency detection via combining region-level and pixel-level predictions with cnns," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 809–825.

[30] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 825–841.

[31] S. S. Kruthiventi, V. Gudisa, J. H. Dholakiya, and R. Venkatesh Babu, "Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5781–5790.

[32] S. He, J. Jiao, X. Zhang, G. Han, and R. W. Lau, "Delving into salient object subitizing and detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1059–1067.

[33] X. Chen, A. Zheng, J. Li, and F. Lu, "Look, perceive and segment: Finding the salient objects in images via two-stream fixation-semantic cnns," *Learning*, vol. 39, no. 4321, p. 1050, 2017.

[34] G. Li, Y. Xie, L. Lin, and Y. Yu, "Instance-level salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 247–256.

[35] R. Cong, J. Lei, H. Fu, Q. Huang, X. Cao, and C. Hou, "Co-saliency detection for rgbd images based on multi-constraint feature matching and cross label propagation," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 568–579, 2018.

[36] H. Fu, X. Cao, and Z. Tu, "Cluster-based co-saliency detection," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3766–3778, 2013.

[37] X. Cao, Z. Tao, B. Zhang, H. Fu, and W. Feng, "Self-adaptively weighted co-saliency detection via rank constraint," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 4175–4186, 2014.

[38] Y. Liu, J. Han, Q. Zhang, and L. Wang, "Salient object detection via two-stage graphs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 1023–1037, 2018.

[39] L. Yi, Z. Qiang, H. Jungong, and W. Long, "Salient object detection employing robust sparse representation and local consistency," *Image and Vision Computing*, vol. 69, pp. 155–167, 2018.

[40] N. Liu and J. Han, "A deep spatial contextual long-term recurrent convolutional network for saliency detection," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3264–3274, 2018.

[41] X. Cao, C. Zhang, H. Fu, X. Guo, and Q. Tian, "Saliency-aware nonparametric foreground annotation based on weakly labeled data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1253–1265, 2016.

[42] N. Liu, J. Han, and M.-H. Yang, "Picanet: Learning pixel-wise contextual attention for saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3089–3098.

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[44] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[45] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[46] ——, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[47] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[48] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.

[49] C. Wang, H. Huang, X. Han, and J. Wang, "Video inpainting by jointly learning temporal structure and spatial details," *arXiv preprint arXiv:1806.08482*, 2018.

[50] J. Ren, Y. Hu, Y.-W. Tai, C. Wang, L. Xu, W. Sun, and Q. Yan, "Look, listen and learnła multimodal lstm for speaker identification," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[51] C. Wang, Y. Guo, J. Zhu, L. Wang, and W. Wang, "Video object co-segmentation via subspace clustering and quadratic pseudo-boolean optimization in an mrf framework," *IEEE Transactions on Multimedia*, vol. 16, no. 4, pp. 903–916, 2014.

[52] D. B. Sam and R. V. Babu, "Top-down feedback for crowd counting convolutional neural network," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[53] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conference on Computer Vision*. Springer, 2016, pp. 75–91.

[54] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta, "Beyond skip connections: Top-down modulation for object detection," *arXiv preprint arXiv:1612.06851*, 2016.

[55] Y. Wang, H. Huang, C. Wang, T. He, J. Wang, and M. Hoai, "Gif2video: Color dequantization and temporal interpolation of gif images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1419–1428.

[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representation*, 2015.

[57] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of European Conference on Computer Vision*, 2018, pp. 85–100.

[58] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[59] V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Proceedings of Advances in Neural Information Processing Systems*, 2011, pp. 109–117.

[60] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1155–1162.

[61] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1597–1604.

[62] F. Perazzi, P. Kr?henbhl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 733–740.

[63] C. Xia, J. Li, X. Chen, A. Zheng, and Y. Zhang, "What is and what is not a salient object? learning salient object detector by ensembling linear exemplar regressors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4399–4407.

[64] X. Li, H. Lu, L. Zhang, R. Xiang, and M. H. Yang, "Saliency detection via dense and sparse reconstruction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2976–2983.

[65] W. C. Tu, S. He, Q. Yang, and S. Y. Chien, "Real-time salient object detection with a minimum spanning tree," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2334–2342.

[66] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *Operating System Design and Implementation*, vol. 16, 2016, pp. 265–283.

[67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[68] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. M. Hu, "Global contrast based salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 409–416.

[69] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, "Blitznet: A real-time deep network for scene understanding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4154–4162.

[70] R. Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi, "Newtonian image understanding: Unfolding the dynamics of objects in static images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3521–3529.

[71] W. C. Hung, Y. H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M. H. Yang, "Scene parsing with global context embedding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2650–2658.

[72] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6230–6239.

[73] Y. Pang, M. Sun, X. Jiang, and X. Li, "Convolution in convolution for network in network," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1587–1597, 2017.

[74] Y. Pang, J. Cao, and X. Li, "Cascade learning by optimally partitioning," *IEEE transactions on cybernetics*, vol. 47, no. 12, pp. 4148–4161, 2016.

[75] G. Wu, J. Han, Y. Guo, L. Liu, G. Ding, Q. Ni, and L. Shao, "Unsupervised deep video hashing via balanced code for large-scale video retrieval," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1993–2007, 2018.

[76] G. Wu, J. Han, Z. Lin, G. Ding, B. Zhang, and Q. Ni, "Joint image-text hashing for fast large-scale cross-media retrieval using self-supervised deep learning," *IEEE Transactions on Industrial Electronics*, 2018.

[77] G. Ding, Y. Guo, K. Chen, C. Chu, J. Han, and Q. Dai, "Decode: deep confidence network for robust image classification," *IEEE Transactions on Image Processing*, 2019.

Qiang Zhang received the B.S. degree in automatic control, the M.S. degree in pattern recognition and intelligent systems, and the Ph.D. degree in circuit and system from Xidian University, China, in 2001,2004, and 2008, respectively. He was a Visiting Scholar with the Center for Intelligent Machines, McGill University, Canada. He is currently a professor with the Automatic Control Department, Xidian University, China. His current research interests include image processing, pattern recognition.

CaifengS han received the PhD degree in computer vision from Queen Mary, University of London. His research interests include computer vision, pattern recognition, image and video analysis, machine learning, bio-medical imaging, and related applications. He has authored more than 90 scientific papers and 50 patent applications. He has been Associate Editor and Guest Editor of many scientific journals including IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT), IEEE Transactions on Multimedia (T-MM), IEEE Journal of Biomedical and Health Informatics (J-BHI), Journal of Visual Communication and Image, Signal Processing (Elsevier), Machine Vision and Applications, Journal of Real-Time Image Processing, and IET Computer Vision. He has organized several international conferences and workshops, and served as Program Committee Member and Reviewer for numerous international conferences and journals. He is a Senior Member of IEEE.

Yi Liu received the B. S. degree from Nanjing Institute of Technology, Nanjing, China, in 2012, and the M. S. degree from the Dalian University, Dalian, China, in 2015. He is currently working towards the Ph. D. degree in Control Theory and Control Engineering at Xidian University, China. He is a visiting student at Lancaster University and an internship student at University of Warwick at the present. His current research interests include computer vision and salient object detection.

Jungong Han is currently a tenured Associate Professor with WMG Data Science at University of Warwick, U.K. He is also an adjunct professor at Xidian University, China. Previously, he was an International (Senior) Lecturer (tenured) at Lancaster University, a senior lecturer at Northumbria University (2015-2017), a senior scientist with Philips CI (2012-2015), a research staff (2010-2012) with the Centre for Mathematics and Computer Science, and a researcher (2005-2010) with the Technical University of Eindhoven in Netherlands. Dr. Han's research interests include multimodality data fusion, computer vision, and artificial intelligence. He has written and co-authored over 100 papers, in which one first-authored paper has been cited, up to date, for more than 560 times. He is an associate editor of Elsevier Neurocomputing and Springer Multimedia Tools and Applications.