SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2022, Vol. 65 182103:1–182103:14
https://doi.org/10.1007/s11432-021-3384-y

Densely nested top-down flows for salient object detection

Chaowei FANG[1†], Haibin TIAN[2†], Dingwen ZHANG[3*], Qiang ZHANG[2], Jungong HAN[4] & Junwei HAN[3*]

[1]*School of Artificial Intelligence, Xidian University, Xi'an* 710071*, China;*
[2]*School of Mechano-Electronic Engineering, Xidian University, Xi'an* 710071*, China;*
[3]*Brain and Artificial Intelligence Laboratory, School of Automation, Northwestern Polytechnical University, Xi'an* 710072*, China;*
[4]*Department of Computer Science, Aberystwyth University, Aberystwyth* SY23 3DB*, UK*

Received 14 February 2021/Revised 16 July 2021/Accepted 10 November 2021/Published online 18 July 2022

Abstract With the goal of identifying pixel-wise salient object regions from each input image, salient object detection (SOD) has been receiving great attention in recent years. One kind of mainstream SOD method is formed by a bottom-up feature encoding procedure and a top-down information decoding procedure. While numerous approaches have explored the bottom-up feature extraction for this task, the design of top-down flows remains under-studied. To this end, this paper revisits the role of top-down modeling in salient object detection and designs a novel densely nested top-down flows (DNTDF)-based framework. In every stage of DNTDF, features from higher levels are read in via the progressive compression shortcut paths (PCSPs). The notable characteristics of our proposed method are as follows. (1) The propagation of high-level features which usually have relatively strong semantic information is enhanced in the decoding procedure. (2) With the help of PCSP, the gradient vanishing issues caused by non-linear operations in top-down information flows can be alleviated. (3) Thanks to the full exploration of high-level features, the decoding process of our method is relatively memory-efficient compared to those of existing methods. Integrating DNTDF with EfficientNet, we construct a highly light-weighted SOD model, with very low computational complexity. To demonstrate the effectiveness of the proposed model, comprehensive experiments are conducted on six widely-used benchmark datasets. The comparisons to the most state-of-the-art methods as well as the carefully-designed baseline models verify our insights on the top-down flow modeling for SOD.

Keywords salient object detection, top-down flow, densely nested framework, convolutional neural networks

Citation Fang C W, Tian H B, Zhang D W, et al. Densely nested top-down flows for salient object detection. Sci China Inf Sci, 2022, 65(8): 182103, https://doi.org/10.1007/s11432-021-3384-y

1 Introduction

Salient object detection [1] aims at performing the pixel-level identification of the salient object region from an input image. Owing to its wide-ranging applications in the vision and multimedia community, such as scene understanding [2–6], onfocus detection [7], and image retrieval [8], numerous efforts have been made in recent years to develop effective and efficient deep salient object detection frameworks.

As shown in Figure 1, the existing deep salient object detection models can be divided into three typical frameworks. The first one is the bottom-up encoding flow-based salient object detection framework (see Figure 1(a)). A bottom-up encoder is used for feature extraction, and then a simple classification head is attached to the top of the encoder for predicting the pixel-wise saliency map. Such methods [9–13] occurred in relatively early ages in this research field by designing one or multiple forward network paths to predict the saliency maps. To take advantage of multi-stage feature representations, some recent studies [14–18] started to incorporate additional network blocks to further explore the side information

* Corresponding author (email: zhangdingwen2006yyy@gmail.com, junweihan2010@gmail.com)
† Chaowei FANG and Haibin TIAN have the same contribution to this work.

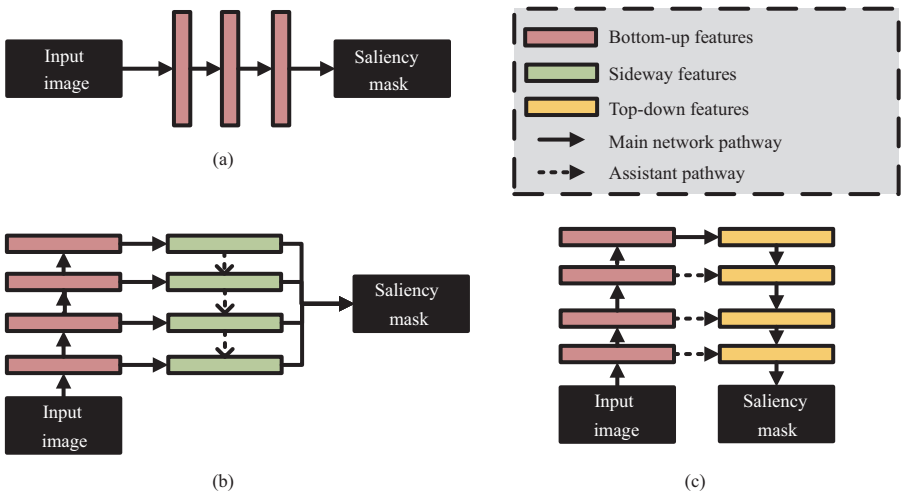

Figure 1 (Color online) A brief illustration of the three mainstream designs of the deep salient object detection frameworks. The bottom-up feature indicates the intermediate feature from the bottom-up pathway which extracts high-level information from low-level information. On the contrary, the top-down feature indicates the intermediate feature from the top-down pathway which decodes the saliency map from multi-scale features generated by the encoder. While the sideway feature indicates the feature produced by the sideway path which transits information from a bottom-up feature to the saliency prediction head. (a) Bottom-up encoding flow-based framework; (b) side information fusion-based framework; (c) top-down decoding flow-based framework.

residing in the features extracted by multiple stages of the forward pathway. The involvement of the learned side information plays a key role in predicting the desired salient object regions. These studies form the second type of learning framework, i.e., the side information fusion-based salient object detection framework (see Figure 1(b)). Although the side information fusion-based frameworks have achieved great performance gains when compared with the bottom-up encoding flow-based frameworks, one important cue for saliency detection, i.e., the top-down information, has not been adequately explored. To this end, the third type of salient object detection framework appeared, which is named as the top-down decoding flow-based framework (see Figure 1(c)). In this framework, the main network pathway is formed by an encoder-decoder architecture, where the decoder explores saliency patterns from the multi-scale semantic embeddings stage by stage and gradually enlarges the resolution of the coarse high-level feature map [19–24]. Notice that this framework may also use the side information to assist the decoding process, but the final saliency masks are obtained from the last decoder stage rather than the fusion stage of the side features.

From the aforementioned top-down decoding flow-based approaches, we observe that their core modeling components still focus on enhancing the side features and merging them into the decoding flow, whereas the top-down information flow remains primitive—propagating from the former decoding stage to the later one as is in the basic encoder-decoder architecture (see Figure 2(a)). Considering that high-level features possess a great wealth of semantic information, we propose a novel decoding framework, named densely nested top-down flows (see Figure 2(b)), to enhance the exploration of features extracted from relatively higher levels. In our method, feature maps obtained by each encoding stage are progressively compressed via shortcut paths and propagated to all subsequent decoding stages. The strengths of our method include the other two strong points. (1) The non-linear operations in the decoding stage are disadvantageous to the gradient back-propagation flow. Hence, the supervision signal propagated from the final prediction to the feature maps of top encoding levels might vanish. For example, if a neuron is not activated by the ReLU function, the gradient flow will be cut off, which means the supervision signal will not be propagated backward. The progressive compression shortcut paths have no non-linear operations, hence they can relieve the gradient vanishing problem. (2) The reuse of high-level features allows a light-weighted decoding network design while achieving high salient object detection performance. Features produced by the top layers of the encoder contain relatively strong semantic information which is beneficial to discriminating regions of salient objects from the background. Our method enhances the propagation of these features, resulting in a memory-efficient decoding framework.

The overall framework is shown in Figure 3. As can be seen, we use the U-Net-like architecture as the main network stream, upon which we further design a novel densely nested top-down flow path to introduce the rich top-down information to the decoding stages. To reduce the computational complexity

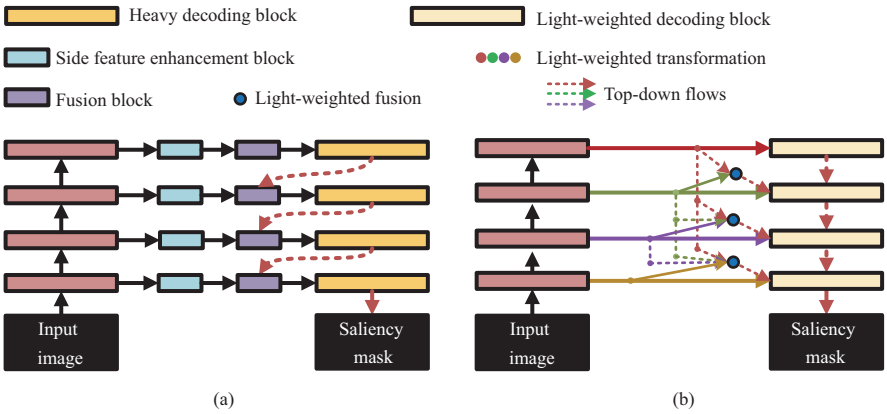

Figure 2 (Color online) In the conventional design for the top-down decoding flow-based salient object detection framework (a), every decoding stage only leverages the feature map of the corresponding encoding stage. In our design, all features extracted by topper encoding stages are propagated into each decoding stage. Our method explores richer semantic information from relatively top stages of the backbone during each decoding stage and complies with the light-weighted principle. (b) Our designs for the top-down decoding flow-based framework.

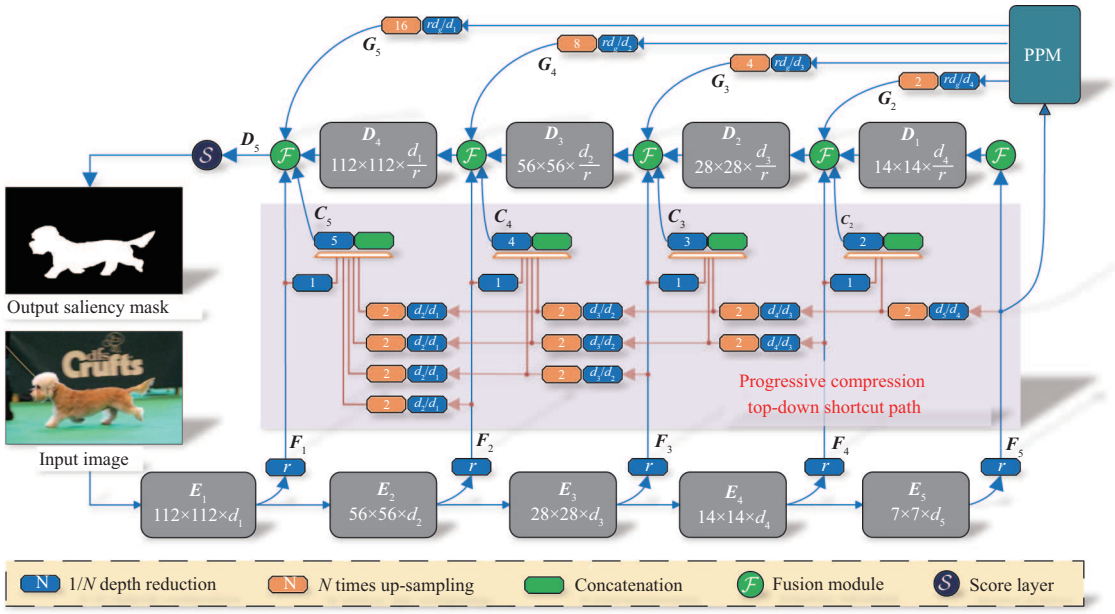

Figure 3 (Color online) A brief illustration of the proposed salient object detection framework, which is built on a basic U-Net-like architecture with the proposed DNTDF to complement the rich top-down information into the decoding pathway. Inspired by [23], the PPM is used to involve the global context features. The whole network is built by light-weighted network designs. More details of the network architecture can be referred to in Section 3.

of the decoding stages, we add a 1×1 channel compression layer to each side information pathway. In fact, all the feature pathways in the proposed decoding framework only need to pass through a number of 1×1 convolutional layers together with very small amounts of 3×3 convolutional layers, making the entire network highly lightweight. Meanwhile, the full exploration of top-down information in the proposed densely nested top-down flows (DNTDF) contributes to even better performance than the state-of-the-art salient object detection (SOD) methods.

In summary, this study has the following three-fold main contributions. (1) We revisit an important yet under-studied issue, i.e., the top-down flow modeling, of the recent SOD frameworks. (2) We design a highly light-weighted learning framework, via introducing a novel densely nested top-down flow architecture. (3) Comprehensive experiments are conducted, demonstrating the promising detection capacity and the lower computational complexity of the proposed framework. Meanwhile, the insights on top-down modeling are well verified: the proposed densely connected top-down flows are capable of enhancing the usage of high-level encoding features in the decoding process; a lightweight design for the decoder can

contribute to high SOD performance.

2 Related work

Traditional salient object detection methods are designed based on the hand-crafted features [25–30]. Recently, convolutional neural networks (CNNs) have been extensively applied in salient object detection. Thanks to the powerful feature extraction capability of CNN, a great breakthrough has been made in devising effective SOD algorithms. CNN-based SOD frameworks can be categorized into three kinds, including bottom-up encoding flow-based, side information fusion-based, and top-down decoding flow-based framework as shown in Figure 2.

In bottom-up encoding flow-based framework, one or multiple forward network paths are designed to predict the saliency maps. For example, Liu et al. [9] proposed a multi-resolution convolutional neural network, which has three bottom-up encoding pathways to deal with patches captured from three different resolutions of the input image. A similar idea is also proposed by Li and Yu [10], where a three-pathway framework is devised to extract multi-scale features for every super-pixel and two fully connected layers are adopted to fuse the features and predict the saliency value. Zhao et al. [11] proposed a multi-context deep learning framework, which fuses a global-context forward pathway and a local-context forward pathway to obtain the final prediction. Li et al. [12] built a multi-task deep learning model for salient object detection, where a shared bottom-up encoding pathway is used to extract useful deep features and two parallel prediction heads are followed to accomplish the semantic segmentation and saliency estimation, respectively. To learn to refine the saliency prediction results, Wang et al. [13] proposed a recurrent fully convolutional network architecture. Specifically, they combine multiple encoding pathways, where saliency prediction results from the former encoding pathway are used to form the input of the latter one.

The side information fusion-based salient object detection framework aims to further explore the side information from the features extracted in each stage of the forward pathway. Specifically, based on the network architecture of the holistically-nested edge detector [31], Hou et al. [14] introduced the skip-layer structures to provide rich multi-scale feature enhancement for exploring the side information. Zhao and Wu [15] proposed a simple but effective network architecture. They enhanced the low-level and high-level side features in two separate network streams, where the former is passed through a spatial attention-based stream while the latter is passed through a channel attention-based stream. Wu et al. [16] also built a two-stream side information flow. However, different from [15], the two-stream side information flow is designed to fuse the multi-stage features for salient region identification and salient edge detection, respectively. Su et al. [17] used a boundary localization stream and an interior perception stream to explore different side features for obtaining the high-selectivity features and high-invariance features, respectively. Recently, Gao et al. [18] proposed gOctConv, a flexible convolutional module to efficiently transform and fuse both the intra-stage and cross-stage features for predicting the saliency maps.

To take advantage of top-down information, the third type of salient object detection framework emerges, i.e., the top-down decoding flow-based framework. In this framework, the main network pathway is formed by an encoder-decoder architecture, where the decoder recognizes out saliency patterns after fusing multi-scale features progressively. Notice that this framework may also use the side information to assist the decoding process, but the final saliency masks are obtained from the last decoder stage instead of the fusion stage of the side features. This type of framework has become the mainstream solution for SOD under both fully-supervised [24,32] and few-shot [6,33] settings. One representative study is proposed by Zhang et al. [24], where a U-Net [34]-like architecture is used as the basic network and a bi-directional message passing model is introduced into the network to extract rich side features to help each decoding stage. Following this study, Liu et al. [23] designed a pooling-based U-shape architecture, where they introduced a global guidance module and a feature aggregation module to guide the top-down pathway. Feng et al. [22] proposed an attentive feedback module (AFM) and used it to better explore the structure of objects in the side information pathway. Liu et al. [21] proposed the local attended decoding and global attended decoding schemes for exploring the pixel-wise context-aware attention for each decoding stage. More recently, in order to better explore the multi-level and multi-scale features, Pang et al. [20] designed aggregation interaction modules and self-interaction modules and inserted them into the side pathway flow and decoding flow, respectively. Zhao et al. [19] proposed a gated decoding flow, where multi-level gate units are introduced in the side pathway to transmit informative context patterns to each decoding stage. Ref. [35] devised two branches, including a salient region prediction branch and a contour

prediction branch, which are interleaved with each other through feature exchange. In [32], a cascaded top-down decoding framework with feedback connections is proposed to refine multi-scale features. Both Refs. [32, 35] employed multiple parallel or cascaded stages to construct the decoding architectures. Though they can derive powerful representations for the decoding stage, heavier computation burdens are caused as well. In this paper, we propose a brand-new top-down flow mechanism, namely densely connected top-down flows. Each decoding stage aggregates all higher-level encoding features with the help of progressive compression shortcut paths. This helps to alleviate the gradient vanishing problem in the decoding stage. The full exploration of encoding features contributes to a light-weighted decoding design that can achieve appealing performance in SOD.

A densely nested top-down flows-based decoding framework is proposed to encourage the reuse of high-level features in every stage of the decoding process. Compared with existing top-down decoding flow-based methods, the superiorities of our method are as follows. The gradient vanishing problem caused by the nonlinear operations in the decoding procedure can be mitigated, and a memory efficient decoding network is employed to facilitate the fusion of multi-stage features while maintaining high detection performance. Our method is closely related to [36] in which each layer aggregates outputs of all preceding layers in the same dense block. Their main differences are: (1) our proposed dense connections aim at exploring multiple top-level encoding features in every decoding stage; (2) instead of directly accumulating preceding features, shortcut paths are devised to compress high-level features in our method, complying with the lightweight design principle.

3 Proposed method

The purpose of this paper is to settle the saliency object detection problem. Given an RGB image $\boldsymbol{X}$ with the size of $h \times w$, we propose a novel efficient deep convolutional architecture to predict a saliency map $\boldsymbol{P} \in [0,1]^{h \times w}$. Every element in $\boldsymbol{P}$ indicates the saliency probability value of the corresponding pixel. A novel densely nested top-down flow architecture is built up to make full use of high-level feature maps. The semantic information of top layers is propagated to bottom layers through progressive compression shortcut paths. Furthermore, interesting insights are provided to design light-weighted deep convolutional neural networks for salient object detection. Technical details are illustrated in subsequent sections.

3.1 Overview of network architecture

The overall network is built upon an encoder-decoder architecture, as shown in Figure 3. After removing the fully connected layers, the backbone of an existing classification model, such as ResNet [37] and EfficientNet [38], is regarded as the encoder. Given an input image $\boldsymbol{X}$, the encoder is composed of five blocks of convolution layers, which yield 5 feature maps, $\{\boldsymbol{E}_i\}_{i=1}^5$. Every block reduces the horizontal and vertical resolutions into half. Let w_i, h_i and d_i denote the height, width and depth of $\boldsymbol{E}_i$, respectively. We have $h_{i+1} = h_i/2$ and $w_{i+1} = w_i/2$.

The target of the decoder is to infer the pixel-wise saliency map from these feature maps. First of all, a compression unit is employed to reduce the depth of each scale of the feature map,

$$\boldsymbol{F}_i = \mathcal{C}_r(\boldsymbol{E}_i, \boldsymbol{W}_i^c), \tag{1}$$

where $\mathcal{C}_r(\cdot,\cdot)$ indicates the calculation procedure of the depth compression unit, consisting of a ReLU layer [39] followed by a 1×1 convolution layer with the kernel of $\boldsymbol{W}_i^c$. r represents the compression ratio, which means the depth of $\boldsymbol{F}_i$ is d_i/r. Inspired from [23], the pyramid pooling module (PPM) [40, 41] is used to extract a global context feature map $\boldsymbol{G}$ (with size of $h_5 \times w_5 \times d_g$) from the last scale of feature map $\boldsymbol{F}_5$ produced by the encoder. Afterwards, a number of convolution layers are set up to fuse these compressed feature maps $\{\boldsymbol{F}_i\}_{i=1}^5$ and the global feature map $\boldsymbol{G}$, and output a soft saliency map, based on the U-shape architecture. The distinguishing characteristics of our encoder are reflected in the following aspects. (1) In every stage of the decoder, the features of the top stages of the encoder are accumulated through progressive compression shortcut paths, forming into the feature representation for SOD together with the additional information learned in the current stage. (2) Our decoder is comprised of 1×1 convolutions and a few 3×3 convolutions, which only take up a small number of parameters and consume a small amount of computational cost. The above decoder designs constitute our so-called densely-nested top-down flows.

3.2 Densely nested top-down flow

In deep convolution neural networks, features extracted by top layers have strong high-level semantic information. These features are advantageous at capturing the discriminative regions of salient objects. Especially, when the network is pretrained on large-scale image recognition datasets, such as ImageNet [42], the top feature maps are intrinsically capable of identifying out salient foreground objects according to [43]. However, their spatial resolutions are usually very coarse which means that it is difficult to locate fine object boundaries from them. On the other hand, bottom layers produce responses to local textural patterns which is beneficial to locate the boundaries of the salient object. Multi-resolution CNN models [9, 10] use multi-scale copies of the input image to explore both low-level and high-level context information. However, such kinds of methods are usually cumbersome and cost a heavy computation burden. Inspired by the holistic-nested network architecture [31], fusing multi-scale feature maps produced by different convolution blocks of the encoder is the other popular choice in SOD [14, 44]. U-Net [34], which currently prevails in deep SOD methods [19, 20, 23, 24], accumulates multi-scale feature maps in a more elegant manner. As shown in Figure 1(c), the decoder usually shares the same number of stages with the encoder, and every stage in the decoder merges the feature map of the corresponding stage of the encoder forming a U-shape architecture. However, in a standard U-Net, the features produced by the encoder are fused into the decoder via a simple linear layer. There exists room for improvement in more fully utilizing these features, especially these relatively high-level features. The difficulty for propagating gradients back into the topper layers of the encoder increases as the gradient back propagation process needs to pass through more decoding stages.

For purpose of settling the above issues, we propose a novel top-down flow-based framework, named densely nested top-down flows. First of all, the ultimate output of the encoder $\boldsymbol{F}_5$ is fed into the first stage of the decoder via a transition operation,

$$\boldsymbol{D}_1 = \mathcal{U}p_{\times 2}(\mathcal{F}(\boldsymbol{F}_5, \boldsymbol{W}_1^d)), \tag{2}$$

where $\mathcal{F}(\boldsymbol{F}_5, \boldsymbol{W}_1^d)$ consists of a ReLU layer and a 3×3 convolution layer with kernel of $\boldsymbol{W}_1^d$. $\mathcal{U}p_{\times 2}(\cdot)$ denotes the 2 times upsampling operation. It transmits $\boldsymbol{F}_5$ into a $h_4 \times w_4 \times d_4/r$ tensor defined as $\boldsymbol{D}_1$.

Then, shortcuts are incorporated to feed feature maps of all higher encoding stages into every decoding stage. The uniqueness of these shortcuts is that high-level feature maps are progressively compressed and enlarged stage by stage. As shown in Figure 3, $\boldsymbol{F}_i$ is propagated to the j-th $(7-i \leqslant j \leqslant 5)$ decoding stage in a recursive manner, generating $\boldsymbol{F}_i^j$,

$$\boldsymbol{F}_i^j = \mathcal{U}p_{\times 2}(\mathcal{C}_{r^j}(\boldsymbol{F}_i^{j-1}, \boldsymbol{W}_i^j)). \tag{3}$$

$\boldsymbol{F}_i^{6-i} = \boldsymbol{F}_i$, which is the exact origination of the information propagation pathway. $\mathcal{C}_{r^j}(\boldsymbol{F}_i^{j-1}, \boldsymbol{W}_i^j)$ reduces the depth of the input tensor $\boldsymbol{F}_i^{j-1}$ to that of the feature map in the j-th decoding stage. It is implemented with a 1×1 convolution layer with the parameter $\boldsymbol{W}_i^j$ and the compression ratio $r^j = \frac{d_{7-j}}{d_{6-j}}$. The shortcut path without using any nonlinear function benefits the gradient back-propagation, thus helping to relieve the gradient vanishing issue caused by the multi-stage decoding process. On the other hand, compared with reducing the depth into the target values at once, our compression mechanism is more efficient and consumes less parameters.

For the j-th $(2 \leqslant j \leqslant 5)$ decoding stage, the calculation process is composed of two fusion steps. First, we derive an additional feature map from the $(6-j)$-th encoding stage, $\hat{\boldsymbol{F}}_{6-j} = \mathcal{C}_1(\boldsymbol{F}_{6-j}, \boldsymbol{W}_{6-j}^a)$. Here, $\boldsymbol{W}_{6-j}^a$ denotes the parameter involved in the calculation process, and $\hat{\boldsymbol{F}}_{6-j}$ maintains the same depth with $\boldsymbol{F}_{6-j}$. Together with $\hat{\boldsymbol{F}}_{6-j}$, the feature maps from higher-level encoding stages $\{\boldsymbol{F}_i^j\}_{i=7-j}^5$ are concatenated and fused into a new context feature map $\boldsymbol{C}_j$:

$$\boldsymbol{C}_j = \mathcal{C}_j(\{\hat{\boldsymbol{F}}_{6-j}, \boldsymbol{F}_i^j | i = 7-j, \ldots, 5\}, \boldsymbol{W}_j^f). \tag{4}$$

Note that the above fusion operation compresses the concatenated feature maps with the ratio of j, which indicates the depth of $\boldsymbol{C}_j$ is d_{6-j}/r. $\boldsymbol{W}_j^f$ denotes the parameter of the compression operation.

Then, the global feature $\boldsymbol{G}$ is complemented to the j-th stage of the decoder as well,

$$\boldsymbol{G}_j = \mathcal{U}p_{\times 2^{j-1}}(\mathcal{C}_{r_j^g}(\boldsymbol{G}, \boldsymbol{W}_j^g)), \tag{5}$$

where $\boldsymbol{W}_j^g$ denotes the related convolutional parameter, and $r_j^g = \frac{d_g}{d_{6-j}}$. $\boldsymbol{D}_{j-1}$, $\boldsymbol{F}_{6-j}$, $\boldsymbol{C}_j$, and $\boldsymbol{G}_j$ are fused with a pre-placed ReLU and a 3×3 convolution layer, yielding the feature representation of the j-th stage of the decoder,

$$\boldsymbol{D}_j = \mathcal{U}p_{\times 2}(\mathcal{F}(\{\boldsymbol{D}_{j-1}, \boldsymbol{F}_{6-j}, \boldsymbol{C}_j, \boldsymbol{G}_j\}, \boldsymbol{W}_j^d)), \tag{6}$$

where $\boldsymbol{W}_j^d$ denotes the parameter of the 3×3 convolution, and the depth of $\boldsymbol{D}_j$ is transformed into d_{5-j}/r. Reusing $\boldsymbol{F}_{6-j}$ when calculating $\boldsymbol{D}_j$ allows $\boldsymbol{C}_j$ to merely learn information which is complementary to $\boldsymbol{F}_{6-j}$. This helps to decrease the difficulty of learning a comprehensive representation from the multi-scale features of the encoder.

The final output is produced by a score prediction module consisting of a pre-placed ReLU layer, a 1×1 convolution layer, and a Sigmoid function $\mathcal{S}(\cdot)$,

$$\boldsymbol{P} = \mathcal{S}(\mathcal{U}p_{\times 2}(\mathcal{C}_{d_1/r}(\mathrm{ReLU}(\boldsymbol{D}_5), \boldsymbol{W}^o))), \tag{7}$$

where $\boldsymbol{W}^o$ represents the kernel of the convolution layer, and $\boldsymbol{P}$ $(h \times w \times 1)$ is the final predicted saliency map.

The advantage of our densely nested decoder is that every stage is accessible to all higher-level feature maps of the encoder. This framework greatly improves the utilization of high-level features in the top-down information propagation flow.

3.3 Light-weighted network design

In this paper, we are not stacking piles of convolution layers to build an SOD network with high performance. Our devised model has a light-weighted architecture while preserving high performance.

Without backbones initialized with parameters pre-trained on ImageNet, it is difficult to achieve high performance via training a light-weighted backbone from scratch such as CSNet [18]. However, these initialized parameters are learned for solving the image recognition task. This means that the features extracted by the pre-trained backbone are responsible for jointly locating the discriminative regions and predicting semantic categories. In the SOD task, it is no longer necessary to recognize the category of the salient object. Considering the above point, we can assume that there exists a large amount of redundant information in the features extracted by the backbone. Hence, in our method, a large value is adopted for the compression ratio r in (1). We empirically find out that using $r \in \{2, 4, 8, 16\}$ has little effects on the SOD performance in our method. This will be illustrated in the experimental section. With the help of a large compression ratio, the computation burden in the decoder can be greatly reduced.

Previous high-performance SOD models are usually equipped with decoders having a moderate amount of calculation complexity. For example, Ref. [40] used multiple 3×3 convolutions to construct a pyramid fusion module in every stage of the decoder. Ref. [20] adopted a number of 3×3 convolutions to aggregate inter-level and inter-layer feature maps in the decoder. Cascaded decoders are employed to implement top-down information in [32, 35], which leads to a decoding procedure with a large computation burden. In our proposed model, all convolutions adopted in the progressive compression shortcut paths have the kernel size of 1×1. This makes these complicated shortcuts only cost a few weights and computation resources in fact. Furthermore, benefitted from rich top-down information, employing a single 3×3 convolution in every encoder stage is sufficient to construct a high-performance decoder. The above network designs help us build up an effective and cost-efficient salient object detection model.

3.4 Network training

To make our model pay more attention to the edge of salient object, we adopt the edge weighted binary cross entropy loss [32] during the training stage,

$$L_{\mathrm{wbce}} = \sum_{i=1}^{H}\sum_{j=1}^{W}(1 + \gamma\alpha_{i,j})\mathrm{BCE}\left(P_{i,j}, Y_{i,j}\right), \tag{8}$$

$$\alpha_{i,j} = \left|\frac{\sum_{m=-\delta}^{\delta}\sum_{n=-\delta}^{\delta} Y_{i+m,j+n}}{(2\delta+1)^2} - Y_{ij}\right|, \tag{9}$$

where $\mathrm{BCE}(\cdot,\cdot)$ is the binary cross entropy loss function, and γ is a constant. $P_{i,j}$ and $Y_{i,j}$ are the value at position (i, j) of $\boldsymbol{P}$ and the ground-truth saliency map $\boldsymbol{Y}$, respectively. $\alpha_{i,j}$ measures the weight

assigned to the loss at position (i, j), which receives a relatively large value when (i, j) locates around the boundaries of salient objects. δ represents the radius of window size for calculating $\alpha_{i,j}$, and mirrored padding is adopted to fill positions outside the border of the image. Adam [45] is used to optimize network parameters.

4 Experiments

4.1 Datasets & evaluation metrics

The DUTS [46] is the largest dataset for salient object detection, containing 10553 training images (DUTS-TR) and 5019 testing images (DUTS-TE). Our proposed model is trained with images of DUTS-TR and evaluated on six commonly used salient object detection datasets, including DUTS-TE, HKU-IS [10], ECSSD [47], PASCAL-S [48], DUT-OMRON [25], and SOD [49].

Three metrics are adopted to evaluate the performance of SOD methods, including the maximum of F-measure ($F_{\max}$) [50], mean absolute error (MAE), and S-measure (S) [51].

4.2 Implementation details

In our experiments, our proposed top-down flow mechanism is integrated with two kinds of backbone models, including ResNet50 [37] and EfficientNet [38]. For ResNet50, we adopt the knowledge distillation strategy in [52] to initialize network parameters. The other models, EfficientNet-B0 and EfficientNet-B3, are pretrained on ImageNet [53]. The trainable parameters of the decoder are initialized as in [54]. Random horizontal flipping and multi-scale training strategy (0.8, 0.9, 1.0, 1.1, and 1.2 times geometric scaling) are applied for data augmentation. All models are trained with 210 epochs and the batch size is set as 1. The learning rate is initially set to 1.0×10^{-5} and 4.5×10^{-4} respectively for ResNet50 and EfficientNet, and divided by 10 at the 168-th epoch. Hyper-parameters in (8) are set as $\gamma = 3$ and $\delta = 10$. A variety of values $\{2, 4, 8, 16, 32\}$ are tested for the compression ratio r. Without specification, r is set as 2. Our proposed model is implemented with PyTorch, and one 11 GB NVIDIA GTX 1080Ti GPU is used to train all models.

4.3 Comparison with state-of-the-arts

As presented in Table 1 [55–58], we compare our method against various existing SOD methods. Three versions of our method, which use a backbone of ResNet50 (Ours+R50), EfficientNet-B0 (Ours+E0), and EfficientNet-B3 (Ours+E3), respectively, are reported. For a fair comparison, we reimplemented very recently proposed SOD algorithms, including ITSD [35], MINet [20], CSF [18], and F^{3}Net [32], via replacing their original backbones with EfficientNet-B3. This forms 4 new SOD methods: MINet+E3, ITSD+E3, CSF+E3, and F^{3}Net+E3. When using EfficientNet-B3 as the backbone, our method achieves the best performance in most scenarios. For example, in contrast to F^{3}Net+E3, 0.01 higher S-measure is achieved by our method on the DUTS-TE dataset. On the basis of ResNet50, our method also achieves overall better performance than F^{3}Net, e.g., the $F_{\max}$ of our method is 0.004 higher than that of F^{3}Net on the DUTS-TE dataset. FLOPs (evaluated with a 288×288 input image) and numbers of parameters are presented in Figure 4. Our method achieves high performance with a relatively small number of memory costs and parameters.

In addition, we follow [58] to compare the precision-recall curves of our approach against the state-of-the-art methods on six datasets (see Figure 5). A gallery of SOD examples is also visualized in Figure 6 for qualitative comparisons. Our method performs clearly better than other methods, across small and large salient objects.

4.4 Ablation study

Efficacy of main components. In this experiment, we first verify the efficacy of main components in our proposed model, including the progressive compression shortcut path (PCSP) and the PPM for global feature extraction. ResNet50 is used to construct the backbone of our proposed model and SOD performance is evaluated on the DUTS-TE dataset. The experimental results are presented in Table 2. As more top-down feature maps are used to complement high-level semantic information in bottom convolution layers via multiple PCSPs, the performance of our method increases consistently. The adoption

Table 1 Quantitative comparison of our method against other SOD methods on DUST-TE and HKU-IS datasets. All these models are trained on DUTS-TR. The performances ranked first, second, and third are marked by bold, underline, and italic, respectively.

Model	DUTS-TE			HKU-IS			ECSSD			PASCAL-S			DUT-O			SOD		
	$F_{\max}$	MAE	S	$F_{\max}$	MAE	S	$F_{\max}$	MAE	S	$F_{\max}$	MAE	S	$F_{\max}$	MAE	S	$F_{\max}$	MAE	S
CSNet [14]	0.819	0.074	0.822	0.899	0.059	0.880	0.914	0.069	0.888	0.835	0.104	0.813	0.792	0.080	0.803	0.827	0.139	0.747
MLMSNet [55]	0.854	0.048	0.861	0.922	0.039	0.907	0.926	0.048	0.905	0.858	0.074	0.844	0.793	0.064	0.809	0.862	0.108	0.786
BASNet [56]	0.860	0.047	0.866	0.929	0.032	0.909	0.939	0.040	0.910	0.858	0.076	0.838	0.811	0.056	0.836	0.851	0.114	0.769
CPD [57]	0.865	0.043	0.869	0.925	0.034	0.906	0.936	0.040	0.913	0.861	0.071	0.848	0.797	0.056	0.825	0.860	0.112	0.767
SCRN [16]	0.888	0.039	0.885	0.934	0.034	0.916	0.944	0.041	0.920	0.879	*0.063*	*0.869*	0.811	0.056	0.837	0.870	0.100	0.797
GateNet [19]	0.889	0.040	0.885	0.935	0.034	0.915	0.942	0.043	0.914	0.877	0.068	0.858	0.831	0.055	0.837	–	–	–
EGNet [58]	0.893	0.039	0.885	0.938	0.031	0.918	0.943	0.041	0.918	0.869	0.074	0.852	0.842	0.053	0.838	*0.889*	0.099	0.802
PoolNet [23]	0.894	0.036	0.886	0.938	0.030	0.918	*0.945*	0.038	0.919	0.884	0.065	0.865	0.830	0.054	0.830	0.879	0.106	0.787
ITSD [35]	0.883	0.041	0.885	0.934	0.031	0.917	0.944	0.037	0.919	0.871	0.066	0.859	0.824	0.061	0.840	0.880	0.095	0.806
ITSD+E3	0.894	0.041	0.894	*0.939*	0.034	*0.924*	0.939	0.034	*0.924*	0.877	0.065	**0.872**	0.834	0.058	0.854	0.882	0.096	0.815
MINet [20]	0.888	0.037	0.884	0.936	0.029	0.919	*0.945*	0.036	0.920	0.874	0.064	0.856	0.826	0.056	0.833	–	–	–
MINet+E3	0.879	0.044	0.875	0.929	0.036	0.909	0.936	0.043	0.912	0.873	0.070	0.855	0.813	0.067	0.821	0.858	0.101	0.795
CSF [14]	0.893	0.037	0.890	0.936	0.030	0.921	0.947	0.036	*0.924*	0.876	0.069	0.862	0.833	0.055	0.837	0.870	0.100	0.797
CSF+E3	0.892	*0.032*	0.894	0.936	0.027	0.921	0.944	*0.034*	0.921	0.872	0.061	0.860	0.826	*0.052*	*0.844*	0.881	0.089	0.808
F^3Net [32]	0.897	0.035	0.888	*0.939*	*0.028*	0.917	0.944	0.036	0.919	0.878	0.061	0.861	*0.839*	0.053	0.838	–	–	–
F^3Net+E3	0.906	0.033	0.898	**0.944**	**0.025**	0.926	0.947	**0.032**	0.925	**0.888**	**0.058**	0.871	**0.844**	0.056	*0.844*	0.890	**0.083**	**0.821**
Ours+R50	*0.901*	0.031	*0.895*	0.941	0.027	0.921	0.945	0.035	0.918	*0.882*	0.061	0.861	0.832	0.051	0.833	0.875	0.101	0.784
Ours+E0	0.891	0.035	0.890	0.936	0.030	0.920	0.942	0.038	0.918	0.872	*0.063*	0.858	0.827	*0.052*	0.841	0.873	0.099	0.795
Ours+E3	**0.907**	**0.030**	**0.905**	**0.944**	0.027	**0.928**	**0.950**	0.033	**0.927**	**0.888**	**0.058**	**0.872**	**0.844**	**0.047**	**0.857**	**0.893**	*0.091*	*0.811*

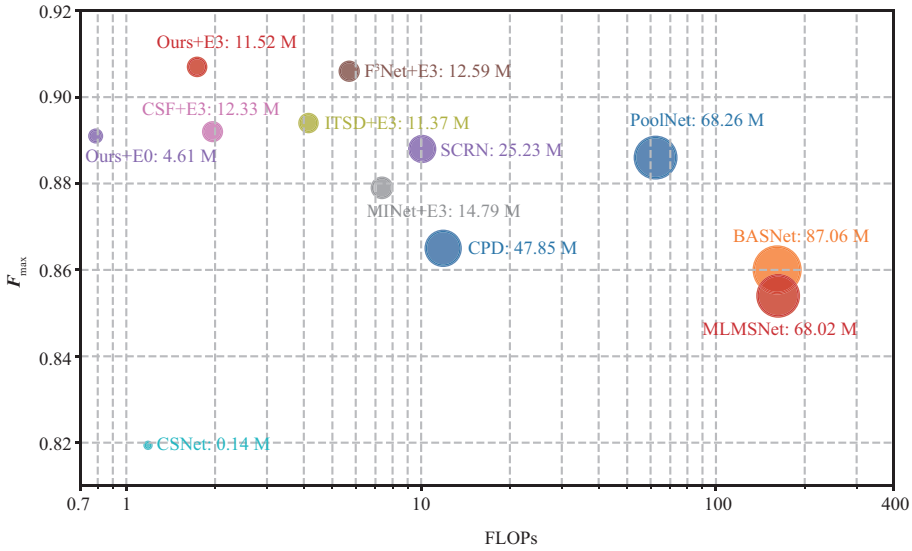

Figure 4 (Color online) Comparisons between our proposed method and other methods: CSNet [14], CSF+E3, F^3Net+E3, SCRN [16], ITSD+E3, CPD [57], PoolNet [23], BASNet [56] and MLMSNet [55] on the DUTS-TE dataset. $F_{\max}$ and FOLPs reflect the detection accuracy and computational complexity, respectively. The diameter of circles indicates the number of parameters.

of 4 PCSPs, induces performance gains of 0.005 (when PPM is not used) and 0.011 (when PPM is used) on the $F_{\max}$ metric. Besides, we can observe that the global information provided by the PPM and high-level semantic information provided by the PCSP can complement each other. Without using any of the two modules, performance degradation is caused.

Different variants of our method. We further provide inner comparisons between variants of our proposed model in Table 3. To validate whether more complicated convolution blocks are effective in the decoder of our method, we replace the 3×3 convolution layer with the FAM block used in [23] to build the fusion module of each decoding stage. However, no obvious performance gain is obtained. To validate the effectiveness of reusing $\boldsymbol{F}_{6-j}$ in (6), we attempt to remove $\boldsymbol{F}_{6-j}$ from the input when calculating $\boldsymbol{D}_j$. The resulted $\boldsymbol{F}_{\max}$ metric is 0.894, which is 0.004 lower than that of our final model.

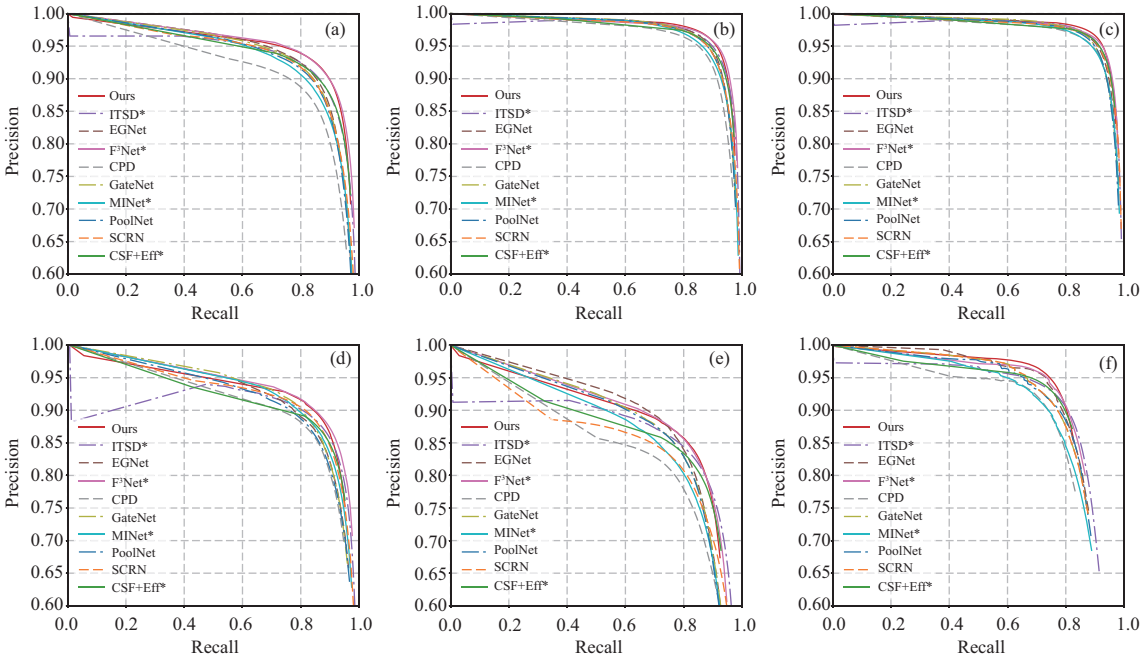

Figure 5 (Color online) Precision-recall curves on six salient object datasets. The EfficientNet-B3 is adopted as the backbone in our method and existing methods marked by $*$. (a) DUTS; (b) HKU-IS; (c) ECSSD; (d) PASCAL-S; (e) DUT-OMRON; (f) SOD.

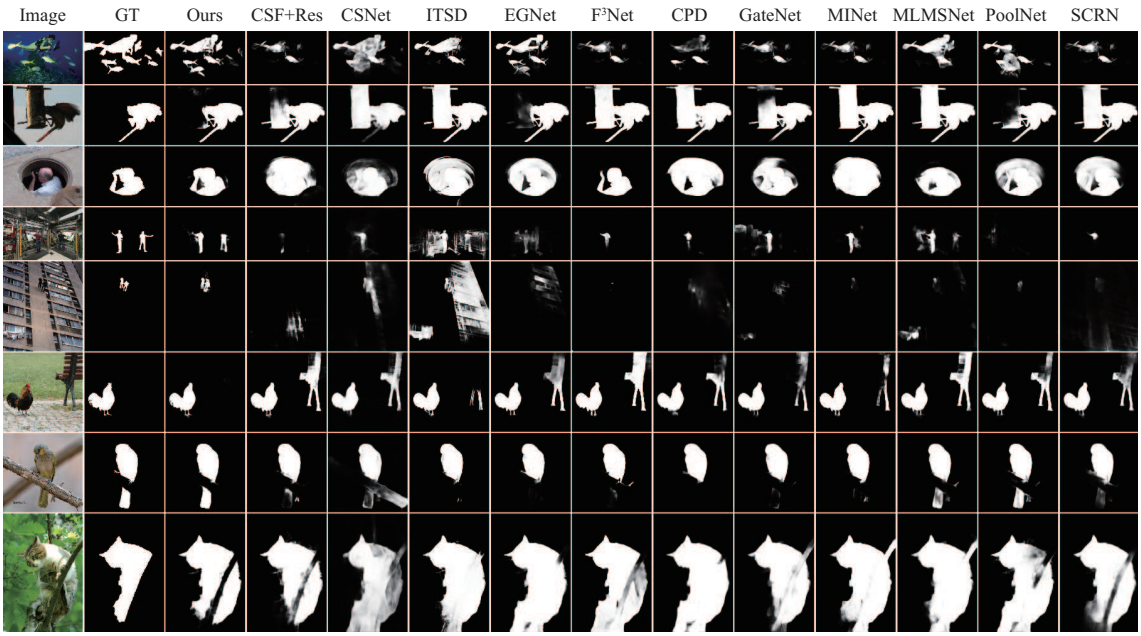

Figure 6 (Color online) Qualitative comparison of our method against other SOD methods.

4.5 Efficiency discussions

Analysis of compression ratio. The influence of using different ratios to compress the features of the encoder as in (1) is illustrated in Table 4. As discussed in Subsection 3.3, there are large amounts of redundant information in the features extracted by the backbone since it is pre-trained for image recognition. Hence, using a moderately large ratio (up to 16) to compress features of the network backbone has no significant effect on the SOD performance, according to the results reported in Table 4. The benefit of using a large compression ratio is achieving the goal of light-weighted network design in our method while not causing an unbearable performance decrease.

Complexity of decoder. As shown in Figure 7, the decoder of our proposed model costs significantly

Table 2 Ablation study on DUTS-TE dataset, using backbone ResNet50. ✓/✗ indicates whether the module is used or not. The number of PCSP denotes the number of most top feature maps of the encoder which is propagated to bottom convolutional blocks of the decoder.

PCSP	PPM	w/ $\boldsymbol{F}_{6-j}$ in (6)	$F_{\max}$	MAE	S
✗	✗	✓	0.887	0.039	0.883
✗	✓	✓	0.891	0.037	0.886
1	✓	✓	0.891	0.034	0.890
2	✓	✓	0.894	0.034	0.892
3	✓	✓	0.896	0.034	0.891
4	✗	✓	0.891	0.037	0.886
4	✓	✗	0.894	0.033	0.892
4	✓	✓	**0.898**	**0.033**	**0.891**

Table 3 Inner comparisons of different variants of our method based on EfficientNet-B3. DCB indicates the convolutional block adopted in every stage of the decoder. FAM means the module containing a single 3×3 convolution operation is replaced with the FAM [23] in every stage of the decoder.

DCB	$F_{\max}$	MAE	S	FLOPs (G)	Param (M)
3×3	0.907	0.0305	0.905	0.108	0.825
FAM	0.904	0.0312	0.902	0.123	1.906

Table 4 Comparisons of performance, parameters, and FLOPs which are based on ResNet50 using different compression scales. Param and FLOPs denote the parameters and FLOPs of the decoder.

Scale	$\boldsymbol{F}_{\max}$	MAE	S	Param	FLOPs
32	0.884	0.036	0.880	379.50 K	63.77 M
16	0.890	0.035	0.888	840.92 K	154.83 M
8	0.893	0.034	0.887	2.01 M	420.96 M
4	0.898	0.033	0.891	5.33 M	1.29 G
2	0.901	0.031	0.895	15.90 M	4.37 G

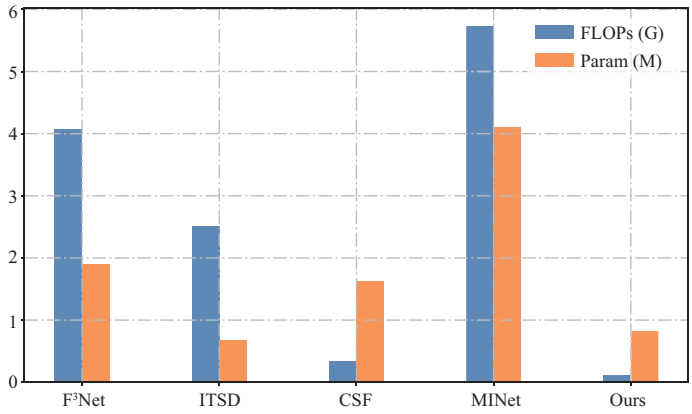

Figure 7 (Color online) Comparisons of the parameters and FLOPs between different decoders based on EfficientNet-B3.

fewer FLOPs than the decoders of recent SOD models, including F^{3}Net, ITSD, CSF, and MINet. The parameters and FLOPs are counted using the backbone of EfficientNet-B3. As can be observed from Table 1, our method outperforms these methods on most datasets and metrics. This indicates that our method achieves better performance even if less memory is consumed.

4.6 Failure cases

In Figure 8, we present a gallery of examples on which our method cannot produce high-quality predictions. They can be categorized into three typical kinds: parts of the foreground objects are missed (namely incomplete objects); some objects, especially small objects, are overlooked; the background around the target object is mistaken as salient content.

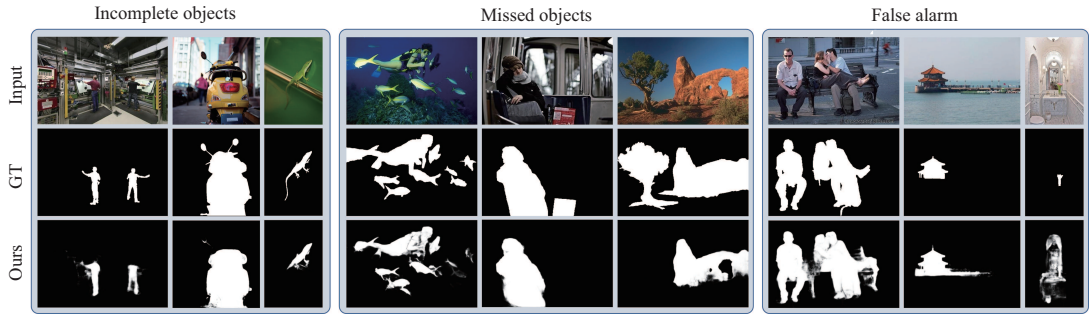

Figure 8 (Color online) Examples on which our method fails to generate sufficiently accurate results.

5 Conclusion

In this paper, we first revisit existing CNN-based top-down flow architectures, including bottom-up encoding flow-based, side information fusion-based, and top-down decoding flow-based frameworks. Then, to make full usage of multi-scale high-level feature maps and to avoid the gradient vanishing issues caused by non-linear operations in the decoding phase, progressive compression shortcut paths are devised to propagate higher-level features of the encoder to bottom convolutional blocks of the decoder, forming the novel densely nested top-down flow-based framework. Extensive experiments indicate that the proposed SOD model can achieve state-of-the-art performance on six widely-used benchmark datasets, including DUTS-TE, HKU-IS, ECSSD, PASCAL-S, DUT-OMRON, and SOD. Notably, thanks to the efficacy of the densely nested top-down flows in exploring high-level features, applying a lightweight design for the decoding architecture does not cause much performance degradation. Ablation study on the progressive compression shortcut paths demonstrates its effectiveness in exploring high-level features for every decoding stage. However, the computing resources in our method are mainly consumed by the encoding stage, instead of the decoding stage. It deserves further research to devise an effective and lightweight encoder for SOD. The other directions for improving SOD models are to improve the semantic understanding capacity and enhance the robustness for detecting tiny salient objects.

Acknowledgements This work was supported in part by Key-Area Research and Development Program of Guangdong Province (Grant No. 2021B0101200001), National Natural Science Foundation of China (Grant Nos. 62003256, 61876140, 62027813, U1801265, U21B2048), and Open Research Projects of Zhejiang Lab (Grant No. 2019kD0AD01/010).

References

1 Han J W, Zhang D W, Cheng G, et al. Advanced deep-learning techniques for salient and category-specific object detection: a survey. IEEE Signal Process Mag, 2018, 35: 84–100
2 Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 2117–2125
3 Zhang D W, Han J W, Yang L, et al. SPFTN: a joint learning framework for localizing and segmenting objects in weakly labeled videos. IEEE Trans Pattern Anal Mach Intell, 2020, 42: 475–489
4 Zhang D W, Han J W, Zhao L, et al. Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework. Int J Comput Vis, 2019, 127: 363–380
5 Cheng G, Li R M, Lang C B, et al. Task-wise attention guided part complementary learning for few-shot image classification. Sci China Inf Sci, 2021, 64: 120104
6 Zhang D, Tian H, Han J. Few-cost salient object detection with adversarial-paced learning. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 12236–12247
7 Zhang D, Wang B, Wang G, et al. Onfocus detection: identifying individual-camera eye contact from unconstrained images. Sci China Inf Sci, 2022, 65: 160101
8 Wang Z H, Liu X, Lin J W, et al. Multi-attention based cross-domain beauty product image retrieval. Sci China Inf Sci, 2020, 63: 120112
9 Liu N, Han J, Zhang D, et al. Predicting eye fixations using convolutional neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015. 362–370
10 Li G, Yu Y. Visual saliency based on multiscale deep features. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015. 5455–5463
11 Zhao R, Ouyang W, Li H, et al. Saliency detection by multi-context deep learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015. 1265–1274

12 Li X, Zhao L M, Wei L N, et al. Deepsaliency: multi-task deep neural network model for salient object detection. IEEE Trans Image Process, 2016, 25: 3919–3930

13 Wang L Z, Wang L J, Lu H C, et al. Saliency detection with recurrent fully convolutional networks. In: Proceedings of European Conference on Computer Vision. Springer, 2016. 825–841

14 Hou Q B, Cheng M M, Hu X W, et al. Deeply supervised salient object detection with short connections. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 3203–3212

15 Zhao T, Wu X Q. Pyramid feature attention network for saliency detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 3085–3094

16 Wu Z, Su L, Huang Q M. Stacked cross refinement network for edge-aware salient object detection. In: Proceedings of IEEE International Conference on Computer Vision, 2019. 7264–7273

17 Su J M, Li J, Zhang Y, et al. Selectivity or invariance: boundary-aware salient object detection. In: Proceedings of IEEE International Conference on Computer Vision, 2019. 3799–3808

18 Gao S H, Tan Y Q, Cheng M M, et al. Highly efficient salient object detection with 100k parameters. In: Proceedings of European Conference on Computer Vision, 2020

19 Zhao X Q, Pang Y W, Zhang L H, et al. Suppress and balance: a simple gated network for salient object detection. In: Proceedings of European Conference on Computer Vision, 2020

20 Pang Y W, Zhao X Q, Zhang L H, et al. Multi-scale interactive network for salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020. 9413–9422

21 Liu N, Han J W, Yang M H. PiCANet: learning pixel-wise contextual attention for saliency detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 3089–3098

22 Feng M Y, Lu H C, Ding E R. Attentive feedback network for boundary-aware salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019

23 Liu J J, Hou Q B, Cheng M M, et al. A simple pooling-based design for real-time salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 3917–3926

24 Zhang L, Dai J, Lu H C, et al. A bi-directional message passing model for salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 1741–1750

25 Yang C, Zhang L H, Lu H C, et al. Saliency detection via graph-based manifold ranking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013. 3166–3173

26 Zhang J M, Sclaroff S, Lin Z, et al. Minimum barrier salient object detection at 80 fps. In: Proceedings of IEEE International Conference on Computer Vision, 2015. 1404–1412

27 Cheng M M, Mitra N J, Huang X L, et al. Global contrast based salient region detection. IEEE Trans Pattern Anal Mach Intell, 2015, 37: 569–582

28 Zhu W J, Liang S, Wei Y C, et al. Saliency optimization from robust background detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014. 2814–2821

29 Jiang H Z, Wang J D, Yuan Z J, et al. Salient object detection: a discriminative regional feature integration approach. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013. 2083–2090

30 Klein D A, Frintrop S. Center-surround divergence of feature statistics for salient object detection. In: Proceedings of IEEE International Conference on Computer Vision, 2011. 2214–2219

31 Xie S N, Tu Z W. Holistically-nested edge detection. In: Proceedings of IEEE International Conference on Computer Vision, 2015. 1395–1403

32 Wei J, Wang S H, Huang Q M. F^3Net: fusion, feedback and focus for salient object detection. In: Proceedings of AAAI Conference on Artificial Intelligence, 2020. 12321–12328

33 Zhang D W, Han J W, Zhang Y, et al. Synthesizing supervision for learning deep saliency network without human annotation. IEEE Trans Pattern Anal Mach Intell, 2020, 42: 1755–1769

34 Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015. 234–241

35 Zhou H J, Xie X H, Lai J H, et al. Interactive two-stream decoder for accurate and fast saliency detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2020. 9141–9150

36 Huang G, Liu Z, van der Maaten L, et al. Densely connected convolutional networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 4700–4708

37 He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778

38 Tan M X, Le Q V. EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of International Conference on Machine Learning, 2019. 6105–6114

39 Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of International Conference on Artificial Intelligence and Statistics, 2011. 315–323

40 He K M, Zhang X Y, Ren S Q, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell, 2015, 37: 1904–1916

41 Zhao H S, Shi J P, Qi X J, et al. Pyramid scene parsing network. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 2881–2890
42 Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis, 2015, 115: 211–252
43 Zhou B L, Khosla A, Lapedriza A, et al. Learning deep features for discriminative localization. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2921–2929
44 Li G, Yu Y. Deep contrast learning for salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 478–487
45 Kingma D P, Ba J. Adam: a method for stochastic optimization. 2014. ArXiv:1412.6980
46 Wang L J, Lu H C, Wang Y F, et al. Learning to detect salient objects with image-level supervision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 136–145
47 Yan Q, Xu L, Shi J P, et al. Hierarchical saliency detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013. 1155–1162
48 Li Y, Hou X D, Koch C, et al. The secrets of salient object segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014. 280–287
49 Movahedi V, Elder J H. Design and perceptual validation of performance measures for salient object segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2010. 49–56
50 Achanta R, Hemami S, Estrada F, et al. Frequency-tuned salient region detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009. 1597–1604
51 Fan D P, Cheng M M, Liu Y, et al. Structure-measure: a new way to evaluate foreground maps. In: Proceedings of IEEE International Conference on Computer Vision, 2017
52 Shen Z, Savvides M. Meal V2: boosting vanilla ResNet-50 to 80%+ top-1 accuracy on ImageNet without tricks. 2020. ArXiv:2009.08453
53 Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–255
54 He K M, Zhang X Y, Ren S Q, et al. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of IEEE International Conference on Computer Vision, 2015. 1026–1034
55 Wu R M, Feng M Y, Guan W L, et al. A mutual learning method for salient object detection with intertwined multi-supervision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 8150–8159
56 Qin X B, Zhang Z C, Huang C Y, et al. BasNet: boundary-aware salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 7479–7489
57 Wu Z, Su L, Huang Q M. Cascaded partial decoder for fast and accurate salient object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 3907–3916
58 Zhao J X, Liu J J, Fan D P, et al. EGNet: edge guidance network for salient object detection. In: Proceedings of IEEE International Conference on Computer Vision, 2019. 8779–8788