Salient Object Detection via Dynamic Scale Routing

Zhenyu Wu, Shuai Li, Chenglizhao Chen, *Member, IEEE*, Hong Qin, and Aimin Hao

***Abstract*— Recent research advances in salient object detection (SOD) could largely be attributed to ever-stronger multi-scale feature representation empowered by the deep learning technologies. The existing SOD deep models extract multi-scale features via the off-the-shelf encoders and combine them smartly via various delicate decoders. However, the kernel sizes in this commonly-used thread are usually "fixed". In our new experiments, we have observed that kernels of small size are preferable in scenarios containing tiny salient objects. In contrast, large kernel sizes could perform better for images with large salient objects. Inspired by this observation, we advocate the "dynamic" scale routing (as a brand-new idea) in this paper. It will result in a generic plug-in that could directly fit the existing feature backbone. This paper's key technical innovations are two-fold. First, instead of using the vanilla convolution with fixed kernel sizes for the encoder design, we propose the dynamic pyramid convolution (DPConv), which dynamically selects the best-suited kernel sizes w.r.t. the given input. Second, we provide a self-adaptive bidirectional decoder design to accommodate the DPConv-based encoder best. The most significant highlight is its capability of routing between feature scales and their dynamic collection, making the inference process scale-aware. As a result, this paper continues to enhance the current SOTA performance. Both the code and dataset are publicly available at *https://github.com/wuzhenyubuaa/DPNet*.**

***Index Terms*— Dynamic scale routing, scale-aware feature aggregation, salient object detection.**

I. Introduction

SALIENT **o**bject **d**etection (SOD) aims to identify the most visually attractive objects from a given image, which can be applied to video edit [1], photo cropping [2] and semantic segmentation [3]. Early SOD literatures [4], [5], [6], [7] have primarily followed the bottom-up methodology, which typically relies on various hand-crafted saliency cues. However, the common nature of these hand-crafted features is not generic in essence, so they tend to yield unsatisfactory performance in complex scenarios. In the deep learning era, the advent of **c**onvolutional **n**eural **n**etwork**s** (CNNs), which is essentially top-down, significantly improves the conventional hand-crafted approaches with ever stronger feature representation, meanwhile the overall computational speed also gets boosted after the widespread deployment of the seminal **f**ully **c**onvolutional **n**etwork**s** (FCNs) [8], where the time-consuming fully connected layers are replaced by convolutional layers. To date, various FCN-based end-to-end SOD models have been proposed, whose key technical innovations could be coarsely categorized as follows: modifying loss functions [9], [10], [11], utilizing edge labels [11], [12], [13], [14], [15], combining with depth information [16], [17], [18], devising attention-based fusion [19], [20], [21], [22], [23], adopting progressive training [9], [24], [25], and exploring the appropriate multi-scale feature aggregation [9], [10], [26], [27], [28].

Although remarkable progress has been achieved, there remain big challenges for the existing SOD models to meet the requirements in real-world applications, where failure cases keep occurring. Through our extensive and careful studies of existing literature (and also augmented by our most recent experiments), we have observed one critical factor, which has long been ignored in previous literature, could have the potential to continue to enhance the SOD performance. This factor hinges upon the fact that the existing SOD models usually perform their convolutional operations using the pre-defined kernels of fixed sizes for all images, and inappropriate kernel sizes most likely cause failure cases. For example, successful detections of large salient objects tend to appear in those layers using large kernel sizes, while kernels with small sizes are preferable for other images. We believe this is a chicken-and-egg problem since the widely-used training process and network are completely scale-unaware. We attribute this phenomenon as "*scale confusion*", which causes learning ambiguity and encoding difficulty towards a perfect inference, as shown in Fig. 1.

Previous models [9], [10], [26], [27], [28] have attempted to alleviate this dilemma by collecting multi-scale features obtained from each layer of an encoder in a "static." fashion, i.e., kernels' sizes are all fixed during the whole process. The features provided by a static encoder usually have a very the unique characteristic, i.e., each of which respectively represents its own individual scale independently. Thus, given the existing **s**tate-**o**f-**t**he-**a**rt (SOTA) models, these scale-specific features shall be considered equally important in advance, with the hope that their decoders could automatically formulate a series of feature combination rules in pursuit of a near-

Manuscript received 12 March 2022; revised 14 September 2022; accepted 4 October 2022. Date of publication 19 October 2022; date of current version 27 October 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62172437 and Grant 62272021, in part by the Open Project Program of the State Key Laboratory of Virtual Reality Technology and Systems under Grant VRLAB2021A05, in part by the Chinese Academy of Medical Sciences (CAMS) Innovation Fund for Medical Sciences under Grant 2019-I2M-5-016, in part by the National Key Research and Development Program of China under Grant 2018YFB1700603, in part by the Youth Innovation and Technology Support Plan of Colleges and Universities in Shandong Province under Grant 2021KJ062, and in part by the National Science Foundation of the USA under Grant IIS-1715985 and Grant IIS-1812606. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aykut Erdem. *(Corresponding author: Chenglizhao Chen.)*

Zhenyu Wu is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China.

Shuai Li is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China, and also with the Peng Cheng Laboratory, Shenzhen 518066, China.

Chenglizhao Chen is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China, and also with the College of Computer Science and Technology, Qingdao Institute of Software, China University of Petroleum (East China), Dongying 266580, China (e-mail: cclz123@163.com).

Hong Qin is with Department of Computer Science, Stony Brook University, Stony Brook, NY 11794 USA.

Aimin Hao is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China, also with the Peng Cheng Laboratory, Shenzhen 518066, China, and also with the Research Unit of Virtual Human and Virtual Surgery (2019RU004), Chinese Academy of Medical Sciences, Beijing 100006, China.

Digital Object Identifier 10.1109/TIP.2022.3214332

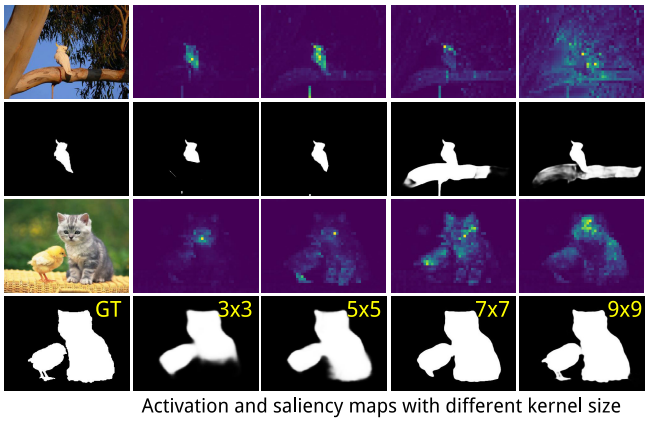


Fig. 1. Two most representative illustrations of activation maps at Conv5_3 of ResNet50. As can be seen in the first two rows, the salient bird, which is a relatively small-sized object, could get massive false-alarms in its surroundings if we use a large kernel size (e.g., 9×9), yet a small kernel size (e.g., 3×3) can easily achieve better detection. Large kernel sizes are preferable to the big objects, as expected.

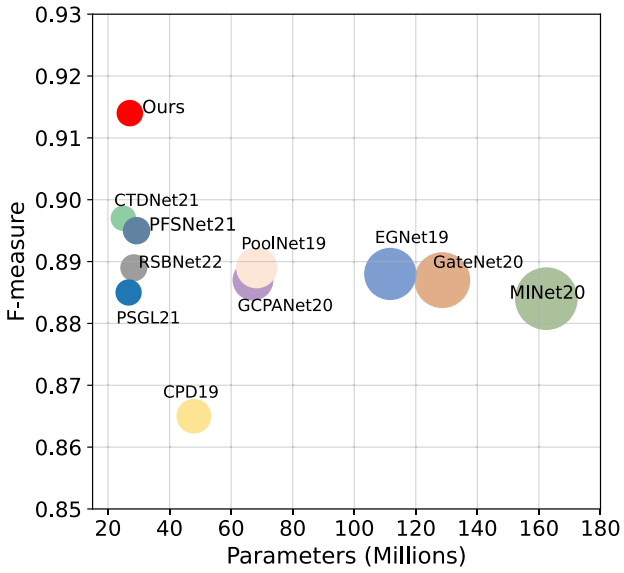


Fig. 2. **F-measure (max) and Parameters** comparisons of our DPNet with other existing SOTA models on the challenging DUTS-TE set, where each circle's size is positively proportional to the model's parameter number. More results can be found in Fig. 11.

optimal full feature complement towards the desired learning objective. In fact, the nature of this naive "feature combining" is a variant of the well-known weighted summation, which selectively integrates all saliency cues revealed in different reception fields, where the weights are formulated during the online learning process. But the problem is that this widely-used learning process is completely static with its own shortcoming since the SOD's problem domain is extremely large, so those implicitly learned feature combination rules are essentially too sparse to have a compact fit. Instead of continuing to seek better feature combination weights in the learning process pertinent to the existing SOTA encoders, our current thought is to replace the widely-used static kernels with possible dynamically-adjusted ones in an encoder to combat this dilemma.

In this paper, we propose a novel yet effective **d**ynamic **p**yramid **net**work (DPNet), which comprises two technical innovations relevant to both the encoder and decoder respectively. As for the encoder's design, we have newly devised the **d**ynamic **p**yramid **conv**olution (DPConv) to address the challenge in the current widely-used paradigm, where the "static" convolutional kernels are replaced by "dynamic" ones. We refer readers to Fig. 3, which illustrates the differences between our DPConv and the widely-used common threads for a better understanding. The primary differences of the proposed DPConv can be highlighted as follows. First, it provides the flexibility of choosing eligible feature scales at the encoding stage, while the existing methods can only resort to their decoders for an appropriate scale combination. Second, the proposed DPConv could directly serve as a generic plug-in for the existing feature backbone with a consistent and enhanced performance (refer to Table II for the quantitative evidence).

As for the decoder design, we present a novel self-adaptive bidirectional decoder (see Fig. 5), which also follows the same "dynamic" rationale, whose goal is to achieve scale-aware feature combination. Most of the existing decoders usually utilize the static one-directional feature aggregation, which either complements low-level tiny details with high-level coarse localization information, or vice versa. Since they have adopted the conventional static encoders (i.e., with fixed kernel sizes), their decoders' input could individually represent different feature scales, making the static one-directional feature aggregation reasonable and feasible. To better match the new attribute of DPConv, it calls for a novel decoder design which could combine them densely and dynamically in both directions, and that's why we devised the self-adaptive bidirectional decoder to perform feature collection in both bottom-up and top-down fashions.

Extensive quantitative experiments have verified that the proposed DPNet significantly outperforms the existing SOTA models on six popular datasets under various evaluation metrics. As shown in Fig. 2, on the DUTS-TE [32] dataset, our DPNet shows 1.6% improvement in terms of max F-measure over the second best CTDNet21 [33], while the computational costs are essentially the same. In addition, our DPNet achieves higher than GateNet20 [34] (88.7→91.6%), and the number of parameters of DPNet is only 27.1 million, which is 4.8× fewer than that of GateNet20. In summary, this paper makes the following salient contributions:

- We have conducted an in-depth investigation into the relationship between kernel sizes and the model's SOD performance, which rise to a possible technical solution of using dynamic kernel sizes with more potential to break the current SOTA performance bottleneck.
- Motivated by this dynamic rationale, we have also devised the new DPConv, which could serve as a generic plug-in. This solution addresses the challenge of the widely-used paradigm built upon static convolutional kernels, making

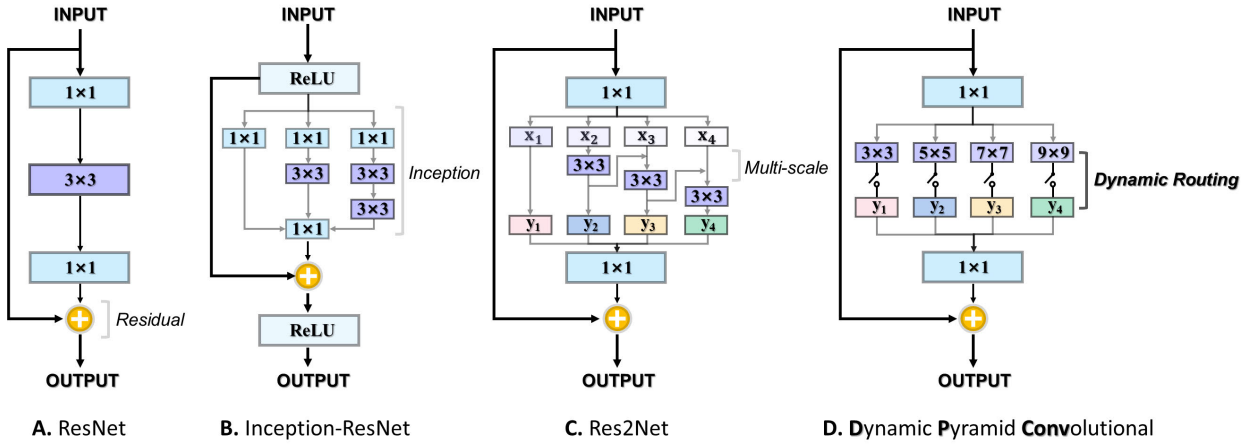


Fig. 3. Comparing (A) ResNet [29], (B) Inception-v4 [30], and (C) Res2Net [31] with (D) our DPConv, where "+" represents the addition operation. The major advantage of our DPConv is its dynamic routing capability of between scales.

the features obtained from the encoder completely scale-aware.

- We have also afforded a delicate solution for the decoder design, which turns the static feature aggregation into a dynamic one. Collectively, we present a feasible roadmap with the potential to guide future research in the improved design of DPConv-compatible decoders. We also conducted extensive quantitative evaluations to confirm the effectiveness of each participating component adopted in our new paradigm.

II. Related Works

A. Deep SOD Model

Existing SOD approaches can be divided into two groups: conventional handcrafted and deep learning-based. Early non-deep learning SOD works usually follow the bottom-up methodology, which collects low-level handcrafted features (e.g., color contrast [5], background prior [35]) to formulate the final detection. However, these works tend to show poor generalization ability, and their results in complex scenarios are extremely worse. More content regarding this topic can be found in [36].

After entering the deep learning era, more and more CNNs based works achieved impressive results in various computer vision tasks, and the deep learning based SOD has received extensive research attentions [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48]. Although these methods achieved promising performances against the handcrafted ones, they still have various limitations, e.g., both computation and memory costs are relatively high. To improve it, FCNs [8] based works formulate the SOD task as a pixel-wise end-to-end binary classification, and ever since then, this methodology has dominated the whole SOD research field in terms of both accuracy and efficiency. Subsequent works mainly focused on appropriately aggregating multi-scale features toward the given learning objective. Most of them [26], [27], [49], [50], [51], [52] followed the encoder-decoder structure. The encoder, also known as the feature backbone, extracts features in a "static" manner — each of its convolutional layers strictly correlates to a fixed scale, and thus these features together could be able to span a multi-scale feature space. Thus, the primary target of the decoder is to formulate a series of decision-making rules in this feature space for the given SOD task. We shall review some most representative works.

Hou et al. [53] proposed to integrate both high-level and low-level features via one-directional cross-scale short connections. Different to [53], which only uses features in specific levels, Zhang et al. [26] performed the bi-directional feature integration, which could interact with low-level features and high-level features, achieving the mutual complementary status. Similarly, Wang et al. [25] proposed a novel multi-scale feature refinement scheme, which could be able to polish its features progressively. Pang et al. [10] proposed an interactive integration network which fuses multi-scale features to deal with the prevalent scale variation issue. Chen et al. [54] proposed a global context-aware progressive aggregation network, which integrates low-level details, high-level semantics, and global contexts in an interweaved way. As pointed by the [55], exploring dynamic network structures in SOD is promising for improving efficiency and effectiveness. Wang et al. [52] integrated both top-down and bottom-up saliency inferences in an iterative and cooperative manner.

The works most relevant to us are [52] and [56]. Our DPNet has three main differences with the [52]: **1)** From the perspective of motivation, our DPNet was designed to solve the "scale confusion" phenomenon while the [52] was proposed to mimic the human visual information processing, i.e., bottom-up and top-down processing. **2)** From the perspective of function, our DPNet focus on "dynamically" generating better feature representation according to the content of the given input image while the [52] was proposed to learn the complementary features by aggregating low-level and high-level features. **3)** From the perspective of decoder, our DPNet adopt the Bidirectional Cross-Fusion Module (BiCFM) while the [52] use employ RNN for top-down/bottom-up inference. In addition, our BiCFM has two main differences with the BiFPN [56] in network structure. **1)** Each node of our BiCFM module has two sets of features while the top and bottom node of BiFPN has one input edge with no feature fusion. **2)** Moreover, our BiCFM and the BiFPN has different cross-scale feature fusion module, i.e., Cross Feature Module (CFM) v.s. Weighted Feature Fusion (WFF). Formally, give a list

of cross-scale features $\{F_{l_1}^{in}, F_{l_2}^{in}, \ldots\}$, where $F_{l_i}^{in}$ represents the feature at level l_i. Compared with WFF, our CFM avoids redundant information introduced to $F_{l_i}^{in}$ and $F_{l_{i+1}}^{in}$, which may pollute the original features and bring adverse effect to the generation of saliency maps. By multiple feature crossings, $F_{l_i}^{in}$ and $F_{l_{i+1}}^{in}$ will gradually absorb useful information from each other to complement themselves, i.e., noises of $F_{l_i}^{in}$ will be suppressed and boundaries of $F_{l_{i+1}}^{in}$ will be sharpened.

B. Dynamic Network

In contrast to the standard paradigm which uses fixed kernel sizes, dynamic CNNs [57], [58], [59], [60], [61], [62], [63], [64] use dynamic modules, kernels, width or depth conditioned on the given input.

One representative line is to learn an auxiliary controller to determine which network's components should be skipped. As one of the pioneering works, Bolukbasi et al. [57] devised an adaptive network evaluation scheme which adaptively chooses the most valuable network's components. In an associative reinforcement learning setting, SkipNet [58] and BlockDrop [59], which adopted very similar ideas to that of [57], proposed to learn policy networks to dynamically determine their networks' architectures.

Another line is to determine the model's capacity dynamically. DCNet [60] proposed to generate convolution kernels dynamically via a linear layer. DFNet [61] introduced a new framework, the dynamic filter network, where spatial filters are generated dynamically to suppress spatial redundancy. CondConv [62] and DCNet [63] proposed to replace the static convolution kernels by a linear combination of n experts with an identical kernel size. Theoretically, CondConv and DCNet belong to linear aggregation, which is insufficient to provide powerful adaptation ability to fit all its given learning objectives. In contrast, we propose the DPConv, which aggregates information from multiple different kernel sizes to provide the flexibility of choosing eligible feature scales, ensuring a more effective multi-scale feature representation. SKNet [64] proposed a novel Selective Kernel convolution to improve the efficiency and effectiveness of object recognition by adaptive kernel selection in a soft-attention manner. Our DPConv has two main differences from the SKNet. **1)** From the perspective of motivation, our DPConv was designed to solve the "scale confusion" phenomenon, while the SKNet was proposed to mimic the receptive fields of neurons in the primary visual cortex of the human visual system. **2)** From the implementation perspective, our DPConv consists of a set of pyramid convolution $\{3 \times 3, \ldots, N \times N\}$ while SK convolution only consists of kernel size 3×3 and 5×5. Thus, the SK convolution is only a special case of our DPCconv. In addition, our DPNet is more lightweight than SKNet (parameters: 27.1M v.s 27.5M).

III. Dynamic Pyramid Convolution

To address the "scale confusion" as shown in Fig. 1, we present a novel plug-and-play module, i.e., dynamic pyramid convolution (DPConv), which can deal with the scale confusion by "dynamically" generating better feature representation according to the content of the given input image. The proposed DPConv is flexible in choosing eligible feature scales while retaining a very small parameter size.

Our rationale is mainly based on the following insights: the receptive fields of neurons in the primary visual cortex are not "static" but "dynamically" modulated by various stimuli. Analogously, the receptive field sizes of one convolutional layer should also be dynamically adjusted accordingly. The proposed DPConv consists of a group of "pyramidal convolution (Sec. III-A)" and "dynamic routing (Sec. III-B)". A detailed illustration of DPConv's structure can be seen in Fig. 4.

A. Pyramid Convolution

In a classic convolutional layer, a series of pre-defined kernels with fixed sizes (e.g., 3×3) is used for all input examples. In our pyramidal convolution block, we replace the 3×3 filters with a group of pyramidal kernels (e.g., $3 \times 3, \ldots, K_m \times K_m$), and the exact kernel number is defined by the hyperparameter m.

For any given input feature tensor $\mathbf{X}$, we apply a series of convolutions $\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_m\}$ to each individual input feature, then the intermediate feature representation of each convolution group is $\mathbf{y}_i = \mathbf{W}_i * \mathbf{X}$, and $i \in \{1, 2, \ldots, m\}$. By using these "pyramidal" convolutions, the learning capacity can be increased. Besides, the proposed pyramidal convolution using multiple kernels with different sizes has a unique advantage, i.e., ***it enables each convolution block to be inner multi-scale***.

However, a plain implementation of pyramid convolution could lead to additional parameters. To make the pyramid convolution block more efficient, we resort to the group convolution [65], where the total parameter number of a standard convolution ($\mathcal{S}_{para}$) is:

$$\mathcal{S}_{para} = C_{out} \times C_{in} \times K^2, \tag{1}$$

While after being replaced by group convolution, the parameter number becomes:

$$\mathcal{G}_{para} = \sum_{i=1}^{m} \left(C_{out} \times \frac{C_{in}}{g_i} \times K_i^2 \right), \tag{2}$$

where K represents kernel size, g is the group size, C_{in} and C_{out} are the number of input and output channels, respectively.

Here we shall explore the potential ranges of g and K to ensure a lightweight implementation. For simplicity, in our pyramid convolution block, we constraint the output channel numbers of each convolution level to be the same, thus the $\mathcal{G}_{para}$ can be expanded to:

$$\mathcal{G}_{para} = \frac{C_{out}}{m} \times \frac{C_{in}}{g_1} \times (K_1)^2 + \ldots + \frac{C_{out}}{m} \times \frac{C_{in}}{g_m} \times (K_m)^2. \tag{3}$$

Based on Eq. 1 and Eq. 2, we can establish the condition to ensure a lightweight design of the proposed pyramidal group convolution in the following Proposition.

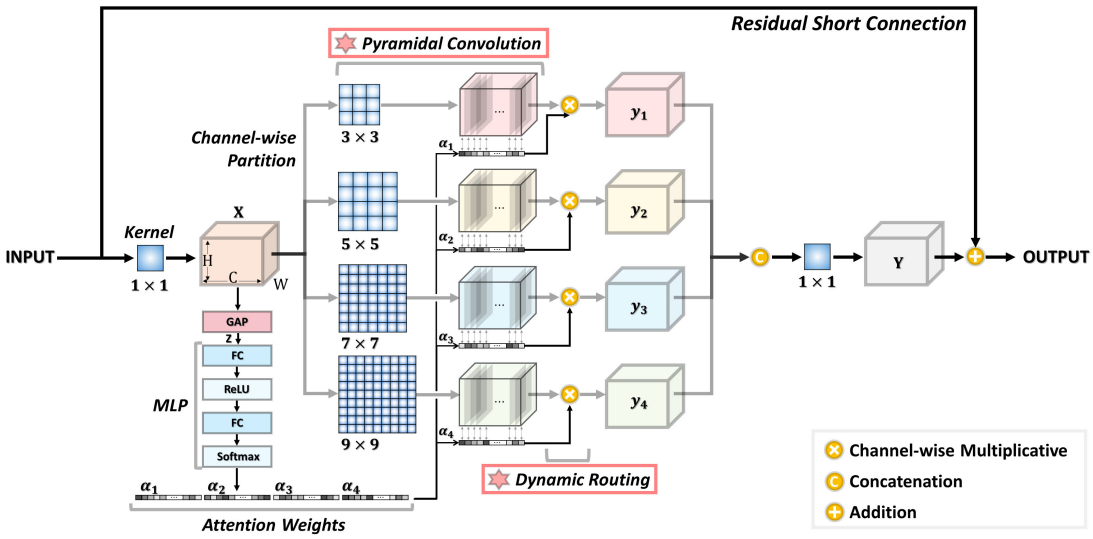


Fig. 4. Network detail of the proposed DPConv block, where the key technical innovations include 1) pyramid convolution (Sec. III-A) and 2) dynamic routing (Sec. III-B).

Proposition 1: The total parameter number of the proposed pyramidal group convolution can be more lightweight than that of the standard convolution in most cases.

Proof: The problem can be interpreted as:

$$\mathcal{G}_{para} \leq \mathcal{S}_{para}. \tag{4}$$

We can expand both sides of Eq. 4 by using Eq. 1 and Eq. 2, and it leads to

$$\left\{\frac{C_{out}}{m} \times \frac{C_{in}}{g_1} \times (K_1)^2 + \ldots + \frac{C_{out}}{m} \times \frac{C_{in}}{g_m} \times (K_m)^2\right\} \leq C_{out} \times C_{in} \times K^2, \tag{5}$$

which can be reduced to

$$\left(\frac{K_1}{K}\right)^2 \times \frac{1}{g_1} + \ldots + \left(\frac{K_m}{K}\right)^2 \times \frac{1}{g_m} \leq m, \tag{6}$$

and, finally, it can be reached to the following condition:

$$g_i > \left(\frac{K_i}{K}\right)^2, \quad i \in \{1, 2, \ldots, m\}. \tag{7}$$

It is easy to see that Eq. 7 is a very generic condition which can be ensured easily, showing the proposed pyramidal group convolution can be more lightweight than that of the standard convolution yet with better learning capacity. □

B. Dynamic Routing

To enable the proposed DPConv to have similar capabilities as humans—we humans could automatically adjust the focused range of our **f**ield **o**f **v**iew (FOV), we devise the dynamic routing, which routings between different kernel sizes, aiming for a finer control towards the receptive fields.

As shown in Fig. 4, the key idea of our dynamic routing is simple and straightforward, where we utilize the channel-wise attention of the grouped pyramidal convolution's original input $\mathbf{X}$ to dynamically control the routing process, which can be detailed as follows.

We gather the global contextual information with channel-wise statistics by using **g**lobal **a**verage **p**ooling (GAP). This process embeds the input $\mathbf{X} \in \mathbb{R}^{1 \times C_{in} \times H \times W}$ to a learnable latent vector $\mathbf{Z} \in \mathbb{R}^{1 \times C_{in}}$ by performing GAP on $\mathbf{X}$ over the spatial dimension, where C_{in} denotes the channel number of the input tensor $\mathbf{X}$. Thus, the c-th component of $\mathbf{Z}$ can be detailed as follows:

$$\mathbf{Z}(1, c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{X}(1, c, i, j). \tag{8}$$

One could apply $\mathbf{Z}$ as the channel-wise attention to the grouped pyramidal convolution directly since the value of each element in $\mathbf{Z}$ represents the importance of the corresponding feature slice in $\mathbf{X}$. However, we shall perform an additional embedding regarding $\mathbf{Z}$ via an MLP consisting of two fully-connected layers, a non-linearity ReLU, and a softmax operation.

There are two reasons for this implementation: 1) this MLP enables the latent vector $\mathbf{Z}$ to be learnable, making our routing process to be dynamical and completely different to the widely-used static attention mechanism; 2) to ensure a generic nature, this MLP can convert $\mathbf{Z}$ to an arbitrary dimension, since there exist cases that, if the grouping process allows overlap between channels, the channel numbers of the grouped pyramidal convolutions is different to the original input.

The dynamic weights $\boldsymbol{\alpha}$ for the grouped pyramidal convolution can be formulated as follows:

$$\boldsymbol{\alpha} = Softmax\Big(FC\Big(ReLU\big(FC(\mathbf{Z})\big)\Big)\Big), \tag{9}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{1 \times C_{group}}$ and, in our implementation, we set $C_{group} = C_{in}$ for simplicity.

In sharp contrast to the classic residual bottlenecks, which are limited to fixed kernel sizes, we can dynamically route between kernels with different sizes by using $\boldsymbol{\alpha}$ as the sole indicator. The output $\mathbf{Y}$ of this dynamic routing process can

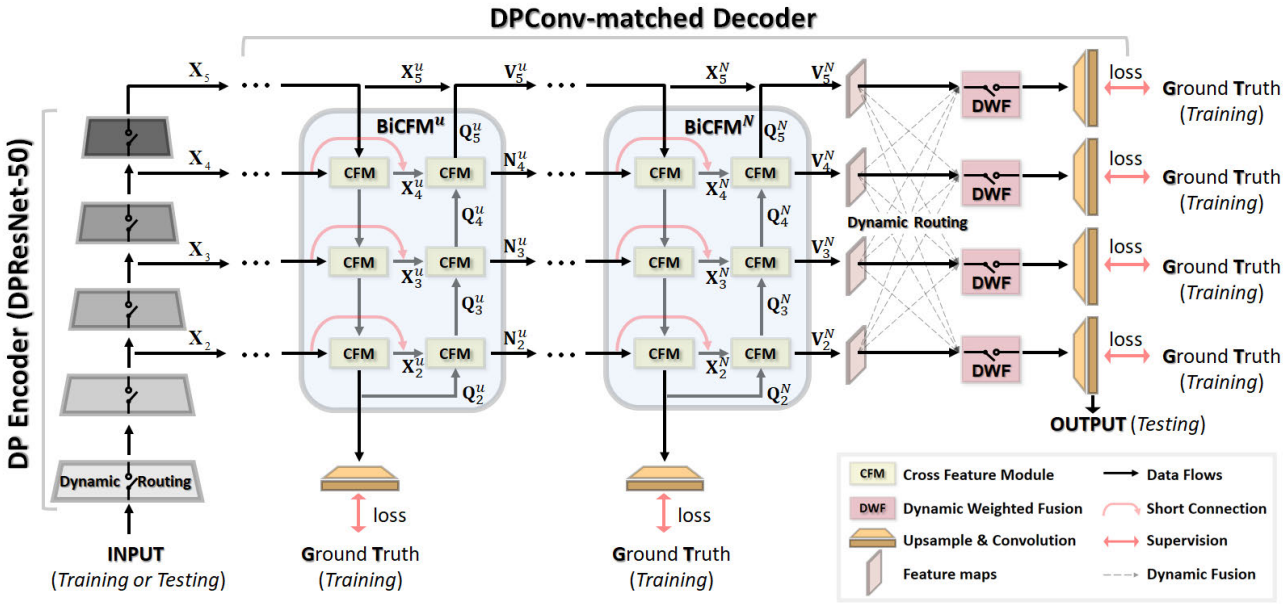


Fig. 5. The architecture of our DPNet consists of three key modules: 1) DPRestNet-50 generates customized scale-aware features, 2) N BiCFMs are stacked to fuse these features bidirectionally, and 3) the DWF module is applied for dynamic feature collection.

be represented as follows:

$$\mathbf{Y} = \mathbf{X} + Concat\big(\boldsymbol{\alpha}_1 \times (\mathbf{W}_1 * \mathbf{X}), \boldsymbol{\alpha}_2 \times (\mathbf{W}_2 * \mathbf{X}), \ldots, \boldsymbol{\alpha}_m \times (\mathbf{W}_m * \mathbf{X})\big), \quad (10)$$

where $*$ represents the convolution operation, and $Concat$ denotes the concatenation operation.

In summary, the final version of our DPConv has three distinguished merits: 1) it provides the flexibility of adaptively choosing different feature scales; 2) it consumes less memory and computational resources; 3) it can replace the conventional convolutional layer directly without additional network modification.

As a plug-and-play module, the DPConv can be easily combined with ResNet, coined DPResNet. The most relevant one is Res2Net [31], Fig. 3 shows an overall comparison, where Res2Net explores multi-scale information of each convolution layer. Our DPResNet is different from Res2Net in both motivation and internal structure. In contrast, our goal is to generate feature representation according to the input image, and thus this could be more consistent with the real human attention mechanism.

IV. DPConv-Matched Decoder

As previously mentioned, the proposed DPConv addresses the "scale confusion" problem (Fig. 1) partly by generating scale-aware feature representation. Another issue that deserves further investigation is how to use these scale-aware features for the SOD task appropriately.

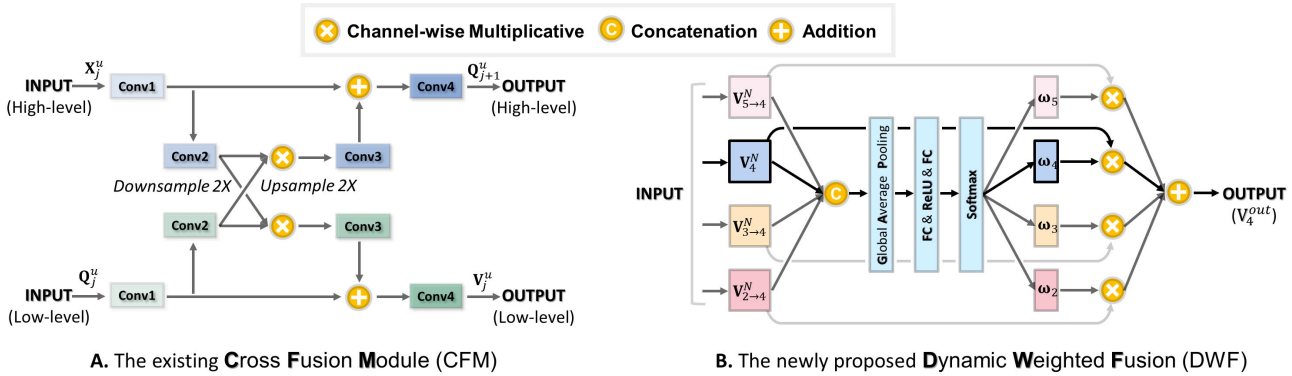
The main difference between features respectively obtained from DPResNet and other SOTA backbones are that, in SOTA backbones, each convolution layer has its distinct scale information. In contrast, in our DPResNet, multi-scale information of each convolution layer might get mixed. Therefore, the widely-used network design — delicately interacting between multi-scale information via static short-connections, is not an optimal choice for the DPConv-based DPResNet. Instead, we shall consider the feature collection process densely and dynamically.

Here we present the DPConv-matched decoder, whose technical innovations include: 1) **bi**directional **c**ross-scale **f**usion **m**odule (BiCFM, Sec. IV-A) and 2) **d**ynamic **w**eighted **f**eature (DWF, Sec. IV-B). Fig. 5 shows the architecture overview, and we call this entire network as **d**ynamic **p**yramid **net**work (DPNet), and, clearly, DPNet is made of DPRestNet encoder and DPConv-matched decoder.

A. *Bi*directional *C*ross-Scale *F*usion *M*odule (BiCFM)

Generally, w.r.t. our human visual system, high-level neurons are usually applied for the salient object localization, while those low-level ones are more likely to be activated by local textures and patterns [66]. Following this rationale, previous decoders [51], [67] have adopted the top-down path, where the cross fusion module (CFM [9], see Fig. 6-A) is one of the most representatives. However, the conventional one-directional feature collection process is not suitable for the DPConv-based DPResNet, since features generated by DPConv are no longer belonging to single scales, where cross-scale information exists in each encoder layer. Besides, some previous studies (e.g., [67]) have also indicated that low-level features could also be able to help the localization process. Thus, we shall consider collecting multi-scale features in a bi-directional manner.

We propose the bidirectional cross-scale fusion module (BiCFM), and its structure can also be found in Fig. 5. Unlike the existing top-down fusion modules (e.g., CFM), the proposed BiCFM can collect multi-scale information in bottom-up aggregation paths. To be more specific, it starts from the



A. The existing **C**ross **F**usion **M**odule (CFM)

B. The newly proposed **D**ynamic **W**eighted **F**usion (DWF)

Fig. 6. The architecture details of CFM and DWF, where the existing CFM was designed to mitigate the discrepancy between high-level and low-level features, and the newly devised DWF aims to make the final inference period suitable towards those scale-aware features provided by the DPConv-based DPResNet.

penultimate lowest level ($\mathbf{X}_2^u$) and gradually approaches $\mathbf{X}_5^u$, where u denotes the u-th bidirectional structure.

We use $\mathbf{V}_i^u$, and $\mathbf{Q}_{i+1}^u$ to denote the newly generated lateral and vertical features respectively corresponding to $\mathbf{X}_i^u$. In the bottom-up path, each CFM block takes a lateral feature $\mathbf{X}_i^u$ and a vertical feature $\mathbf{Q}_i^u$ to generate the new feature $\mathbf{V}_i^u$. Moreover, we introduce shortcut connections from the original input (e.g., $\mathbf{X}_5^u$) to BiCFM's output, which requires no additional computational cost yet leads to better feature representation. We treat each bidirectional (top-down and bottom-up) path as one building block and repeat the same block multiple times to enable more efficient feature fusion. Thus, there are N BiCFMs stacked sequentially in our decoder, and the exact choice of N will be explored in Table VII.

B. Dynamic Weighted Fusion (DWF)

A common way for saliency inference could be directly upsampling the multi-scale features to the same size as the input image and then combining them via element-wise summation or concatenation.

Previous decoders [25], [26] have treated all multi-scale features equally without considering their characteristics. In our DPResNet, those multi-scale features generated by our DPConv are already scale-aware, some of which are more helpful than others. Thus, the final inference stage should follow the dynamic routing rationale also, i.e., to automatically select those really valuable features. Therefore, we propose the **d**ynamic **w**eighted **f**usion (DWF) module, which learns the importance of each of its input.

As shown in Fig. 6-B, we take the 4-th level, for instance, all features in other levels (e.g., $\mathbf{V}_j^N$, where $j \neq 4$) are resized to an identical size as that of the 4-th level. Next, a dynamic weighted fusion operation is introduced to fuse features from different levels selectively, and this process can be detailed as follows:

$$\mathbf{V}_4^{out} = \boldsymbol{\omega}_1 \times \mathbf{V}_{2\to4}^N + \boldsymbol{\omega}_2 \times \mathbf{V}_{3\to4}^N + \boldsymbol{\omega}_3 \times \mathbf{V}_4^N + \boldsymbol{\omega}_4 \times \mathbf{V}_{5\to4}^N, \quad (11)$$

where $\mathbf{V}_{i\to j}^N$ denotes the features resized from level i to level j, and $\boldsymbol{\omega} \in \mathbb{R}^{1\times C_{out}}$ is a learnable weight vector, which follows an identical rationale to $\boldsymbol{\alpha}$ mentioned in Eq. 9, and $\boldsymbol{\omega}$ is defined as:

$$Softmax\Big(FC\Big(ReLU\big(FC(\mathbf{P})\big)\Big)\Big) \to \boldsymbol{\omega}, \quad \overbrace{GAP\big(Concat(\mathbf{V}_{2\to4}^N, \mathbf{V}_{3\to4}^N, \mathbf{V}_4^N, \mathbf{V}_{5\to4}^N)\big)} \quad (12)$$

where GAP is the **g**lobal **a**verage **p**ooling operation; $\boldsymbol{\omega}_i$ is normalized via softmax, which represents the importance of each input channel. Features in other levels proceed in a similar way. This method dynamically aggregates the features at all levels toward the given SOD objective.

C. DPNet Vs. SOTA SOD Models

To highlight our innovations, we shall provide an in-depth discussion regarding the relationship between our DPNet and the existing SOTA SOD models from the perspective of encoder and decoder designs.

W.r.t. the encoder, most of the existing SOTA models have directly adopted the off-the-shelf ResNet as their feature backbones, where the ResNet was designed for image recognition tasks rather than dense prediction tasks (e.g., our SOD task). Instead of using the off-the-shelf ResNet directly, we propose the DPConv-based DPResNet, which could alleviate the scale confusion problem by providing scale-aware feature representation.

In view of the decoder's design, ITSD20 [52] and BMP18 [51] are the two most relevant works to our DPNet. Beyond using a bidirectional architecture simply as BMP18, our DPNet has additionally considered the discrepancy between features of different scales via the proposed BiCFM, which simultaneously takes advantage of low-level features in detail retaining and high-level features in localization.

On the other hand, unlike ITSD20, which treats all its intermediate features equally in the final inference period, our model has designed a novel dynamic weighted fusion module to route between the scale-aware features provided by DPResNet selectively. And this dynamic process makes our encoder suit the proposed DPResNet well while being more consistent with the real human visual system.

V. Experiments

A. Implementation Details

Following the existing SOTA methods [9], [15], [20], [22], we use the DUTS-TR [32] as the training set, and other sets are utilized to evaluate the proposed network. The proposed DPResNet first trained on the ImageNet, and then finetuned on the DUTS-TR dataset. During the training stage, random horizontal flipping, random crop, and multi-scale input images were used as data augmentation to avoid over-fitting. During the testing stage, images are resized to 352×352 and then fed into the proposed DPNet to obtain predictions "*without*" any other post-processing (e.g., CRF). We use Pytorch to implement our model. Mini-batch stochastic gradient descent is used to optimize the whole network, and the batch size, momentum, and weight decay are assigned to {32, 0.9, 5e-4} respectively. We use the warm-up and linear decay strategies with a maximum learning rate 0.005 for the DPResNet backbone and 0.05 for other parts to train our model and stop training after 32 epochs on an NVIDIA GTX 2080 GPU.

Following the previous work [9], we use weighted cross entropy loss ($\mathcal{L}_{wbce}$) and weighted IoU loss ($\mathcal{L}_{wiou}$) to measure the residuals between predicted saliency maps and ground truths. Besides, the widely-used multi-level supervision is applied as auxiliary losses. Our decoder consists of N BiCFMs with M levels in total, thus the total loss function ($\mathcal{L}_{total}$) is calculated as follows:

$$\mathcal{L}_{total} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{L}_{wbce}^{i} + \mathcal{L}_{wiou}^{i}) + \sum_{j=2}^{M}\frac{1}{2^{j-1}}(\mathcal{L}_{wbce}^{j} + \mathcal{L}_{wiou}^{j}), \tag{13}$$

where $M = 5$ and the exact value of N will be determined via ablation study.

B. Datasets and Evaluation Metrics

We have evaluated the performance of the proposed method on six commonly-used benchmark datasets, including DUTS-TE [32], DUT-OMRON [76], ECSSD [77], HKU-IS [78], and PASCAL-S [79].

Meanwhile, five widely-used metrics have been adopted to evaluate our method, including the **P**recision-**R**ecall (PR) curves, the F-measure curves, **M**ean **A**bsolute **E**rror (MAE), weighted F-measure [80], S-measure [81] and E-measure [82].

C. Comparison With the SOTA Models

We have compared our proposed DPNet with SOTA models, including RSBNet22 [68], CTDNet21 [33], PSGL21 [70], PFSNet21 [69], SAMNet21 [71], GCPANet20 [54], MINet20 [10], F3Net20 [9], RANet20 [73], MRNet20 [83], GateNet20 [34], EGNet19 [14], CPD-19 [23], PoolNet19 [74], AFNet19 [12]. Instead of copying and pasting the performance score from the original paper, we evaluate these SOTA models by using the same Code with the authors provided saliency maps (except for the MRNet20).

1) Quantitative Comparisons: To demonstrate the effectiveness of the proposed DPNet, we compare the DPNet with SOTA methods in terms of five metrics in Table I. As can be seen, the proposed DPNet consistently outperforms all other competitors in terms of different metrics on all datasets, even though some methods [14], [33] have used edge information as the extra supervision. In particular, in terms of F_β and MAE, the performances are improved by 1.9% and 5% respectively over the second best method PFSNet21 [69] on DUTS-TE.

Further, we have also provided the standard PR and F-measure curves in Fig. 7. From these curves, we can observe that DPNet consistently outperforms all other models under different thresholds, showing that the proposed method is more robust than other models even on challenging datasets such as DUT-OMRON [76].

2) Visual Comparison: To further illustrate the advantages of the proposed DPNet, we provide some representative examples of our model along with other SOTA models for visual comparisons. As shown in Fig. 8, our DPNet can handle various challenging scenarios, including cluttered background (1st row), occlusion (2nd row), small objects (3rd row) and underwater objects (4th row). Compared with other competitors, the saliency maps generated by our DPNet are more accurate.

3) Integrating DPConv Into SOTA Models: Since the proposed DPConv block doesn't need a specific network structure, we can easily integrate it into the existing SOTA SOD networks without special adjustment, where we take four most representative ones here, i.e., F3Net20 [9], MINet20 [10], CPD19 [23], and PoolNet19 [74]. The corresponding models are referred to as F3Net20-DP, MINet20-DP, CPD19-DP and PoolNet19-DP, respectively. For a fair comparison, we retrain those models by using the original codes with the same configurations instead of directly using the generated saliency maps provided by the authors.

As shown in Table II, these SOD models equipped with our DPConv could achieve persistent performance gains over the raw versions. In particular, F3Net20-DP has an improvement of 1.5% in terms of F_β on the DUT-OMRON dataset. Similarly, MINet20-DP outperforms MINet20 [10] by 1.3% on DUT-OMRON in terms of F_β. We conclude that the proposed DPConv is more effective than the standard residual block when processing multi-scale information, which also demonstrates the effectiveness of the DPConv block.

To further illustrate the advantages of the proposed DPResNet, we newly conducted experiments on polyp segmentation with the SOTA model, i.e., SANet [84]. As shown in Fig. 10, our SANet with DPResNet can produce more accurate and complete segmentation maps in various scenarios. These experimental results further demonstrate the versatility of the DPResNet.

To understand the multi-scale representation capability of DPResNet, we introduce the commonly-used **c**lass **a**ctivation **m**apping (CAM) to explain why the predicted saliency maps of our DPResNet are better than that of the ResNet based one. Concretely, we visualize the CAM by using Grad-CAM [85], which is a simple yet effective method to visualize image

TABLE I

QUANTITATIVE COMPARISONS BETWEEN THE PROPOSED DPNET AND SOTA MODELS. INSTEAD OF COPYING AND PASTING THE PERFORMANCE SCORE FROM THE ORIGINAL PAPER, WE EVALUATE THESE SOTA MODELS BY USING THE SAME [CODE] WITH THE AUTHORS PROVIDED SALIENCY MAPS (EXCEPT FOR THE MRNET20). THESE SALIENCY MAPS ARE AVAILABLE AT OUR [GITHUB]

Methods	DUT-OMRON					DUTS-TE					ECSSD					HKU-IS					PASCAL-S				
	F_β	MAE	W-F_β	S-m	E-m	F_β	MAE	W-F_β	S-m	E-m	F_β	MAE	W-F_β	S-m	E-m	F_β	MAE	W-F_β	S-m	E-m	F_β	MAE	W-F_β	S-m	E-m
DPNet(Ours)	.837	.049	.770	.853	.876	.916	.029	.849	.912	.921	.958	.028	.935	.937	.944	.954	.023	.920	.934	.942	.912	.072	.817	.856	.867
RSBNet22 [68]	.801	.046	.754	.835	.861	.889	.035	.828	.879	.917	.944	.033	.915	.922	.951	.938	.027	.908	.919	.955	.897	.077	.795	.836	.851
PFSNet21 [69]	.823	.055	.742	.843	.813	.895	.036	.819	.892	.873	.952	.031	.914	.930	.919	.943	.026	.903	.924	.919	.899	.079	.798	.843	.848
CTDNet21 [33]	.824	.052	.746	.844	.825	.897	.034	.824	.892	.888	.950	.032	.910	924	.924	.941	.027	.903	.921	.928	.901	.080	.796	.841	.846
PSGL21 [70]	.811	.052	.742	.833	.846	.885	.036	.819	.883	.902	.949	.032	.915	.925	.939	.938	.027	.903	.920	.939	.896	.081	.793	.837	.851
SAMNet21 [71]	.803	.065	.648	.830	.688	.836	.058	.682	.848	.712	.928	.050	.834	.907	.801	.915	.045	.811	.898	.786	.857	.113	.724	.803	.765
JCS21 [72]	.822	.051	.752	.843	.864	.894	.032	.832	.899	.915	.945	.030	.919	.930	.940	.936	.026	.904	.923	.913	.900	.082	.795	.850	.854
GCPANet20 [54]	.812	.056	.708	.839	.779	.887	.038	.787	.890	.835	.949	.035	.890	.926	.887	.938	.031	.875	.921	.876	.898	.077	.790	.847	.833
GateNet20 [34]	.812	.056	.698	.838	.767	.887	.038	.771	.885	.816	.949	.035	.882	.920	.881	.938	.031	.866	.915	.873	.898	.077	.760	.831	.818
MINet20 [10]	.810	.056	.719	.833	.798	.884	.038	.797	.883	.852	.943	.034	.903	.925	.906	.935	.029	.888	.919	.901	.889	.083	.779	.833	.835
F3Net20 [9]	.813	.053	.720	.839	.803	.891	.036	.801	.888	.864	.945	.033	.902	.924	.915	.937	.028	.890	.917	.912	.895	.080	.789	.840	.843
RANet20 [73]	.799	.058	.671	.825	.742	.874	.044	.743	.874	.776	.941	.042	.866	.917	.844	.928	.036	.846	.908	.841	.866	.078	.757	.847	.812
CPD19 [23]	.797	.056	.705	.825	.786	.865	.044	.769	.868	.838	.939	.037	.889	.918	.902	.925	.034	.866	.906	.888	.884	.089	.771	.828	.827
PoolNet19 [74]	.805	.054	.696	.831	.775	.889	.037	.775	.886	.819	.949	.035	.890	.926	.877	.936	.030	.873	.919	.870	.902	.081	.781	.847	.826
AFNet19 [12]	.797	.057	.690	.826	.760	.862	.046	.747	.866	.785	.935	.042	.867	.914	.849	.923	.036	.848	.905	.839	.885	.086	.772	.833	.810
EGNet19 [14]	.815	.053	.701	.841	.760	.888	.040	.769	.886	.802	.947	.037	.887	.925	.870	.935	.031	.870	.918	.860	.891	.090	.777	.835	.821
BASNet19 [28]	.805	.057	.752	.836	.857	.859	.048	.793	.865	.886	.942	.037	.904	.916	.938	.929	.032	.889	.909	.936	.876	.092	.776	.819	.834
PFAN19 [75]	.808	.051	.753	.841	.864	.870	.041	.787	.874	.825	.922	.047	.856	.904	.857	.926	.035	.892	.913	.938	.891	.077	.781	.841	.838

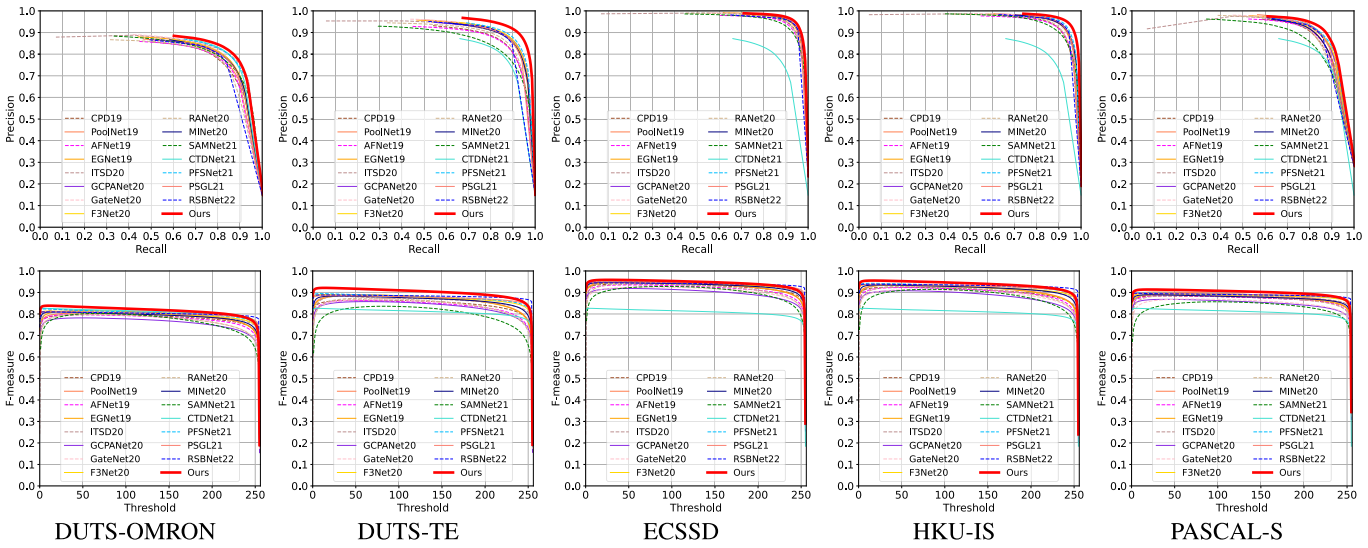


Fig. 7. The 1st and 2nd columns show the performance of the proposed DPNet with other SOTA models in terms of PR and F-measure curves respectively. The last column shows the average precision, recall, and F-measure scores.

regions with the strongest responses, i.e., the salient objects in our case.

As shown in Fig. 9, compared with ResNet-50, the CAM generated by DPResNet-50 tends to cover the whole objects, while the CAM of ResNet-50 only covers the objects partially. In addition, the proposed DPResNet-50 can simultaneously localize multiple objects with accurate activation maps, and such ability in localizing the most discriminative regions makes it potentially suitable for the pixel-wise SOD task.

4) Going Deeper With DPResNet: Previous works [29], [87] indicated that deeper networks have more powerful feature representation capability. To validate such ability of our model, we compare the SOD performance of the DPResNet with SOTA models, including ResNet [29], DenseNet [86], Inception-V4 [30], and Res2Net [31]. Three groups of results are shown in Table III.

In the 1st group, our DPResNet-50 shows 1.8% better performance than ResNet-50 in terms of F_β averagely without using additional *parameters* and *FLOPs*. Also, our DPResNet-50, which could obtain a better balance between accuracy and efficiency, achieves 1.4% higher F-measure than Res2Net-50. In the 2nd group, our DPResNet-101 achieves better performance than the Inception-V4 (82.6→84.7% on DUT-OMRON in terms of F_β) while reducing the

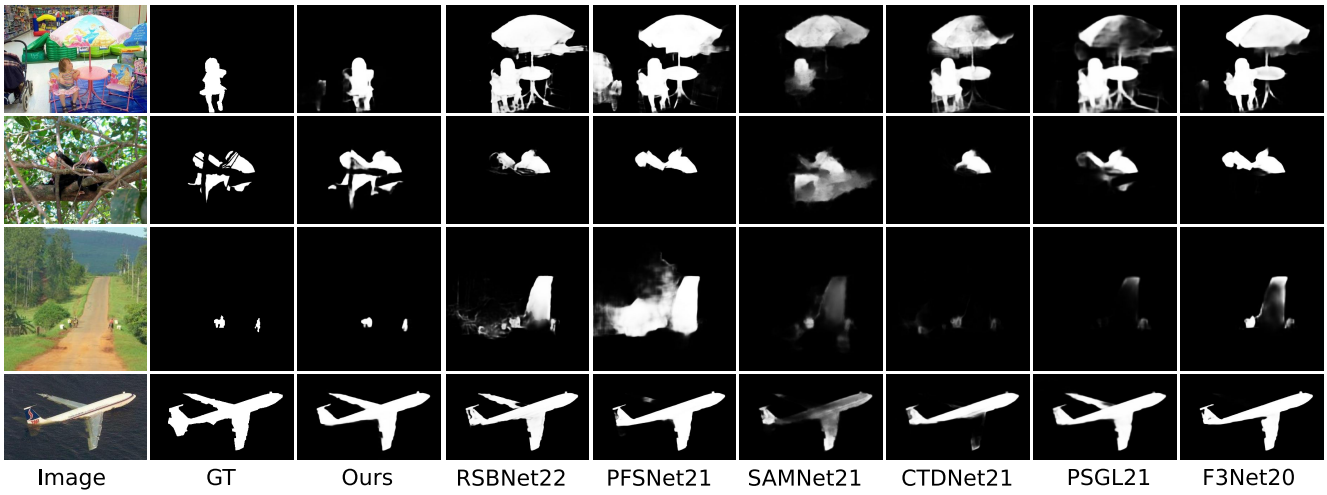


Fig. 8. Visual comparison of the proposed DPNet with other SOTA SOD models, where saliency maps produced by our model are more accurate and complete in various challenging scenarios.

TABLE II

PERFORMANCE OF DPCONV BASED SOTA MODELS IN TERMS OF F-MEASURE, S-MEASURE AND MAE. NOTE THAT THE PROPOSED DPCONV IS ONLY EQUIPPED WITH THE ENCODER NETWORK. WE USE POSTFIX "-DP" TO REPRESENT THAT THE MODEL HAS BEEN EQUIPPED WITH OUR DPCONV. THE BETTER PERFORMANCES HIGHLIGHTED VIA **BOLD** FONT

Method	DUT-OMRON			DUTS-TE		
	maxF_β	S-m	MAE	maxF_β	S-m	MAE
F3Net20 [9]	.813	.839	**.053**	.891	.888	.036
F3Net20-DP	**.828**	**.845**	.055	**.899**	**.896**	**.035**
MINet20 [10]	.803	.830	.056	.887	.885	.037
MINet20-DP	**.816**	**.839**	**.051**	**.896**	**.890**	**.034**
PoolNet19 [74]	.796	.825	.055	.874	.876	.041
PoolNet19-DP	**.809**	**.836**	**.054**	**.887**	**.878**	**.038**
CPD19 [23]	.787	.799	.054	.854	.839	.045
CPD19-DP	**.790**	**.816**	**.053**	**.864**	**.853**	**.044**

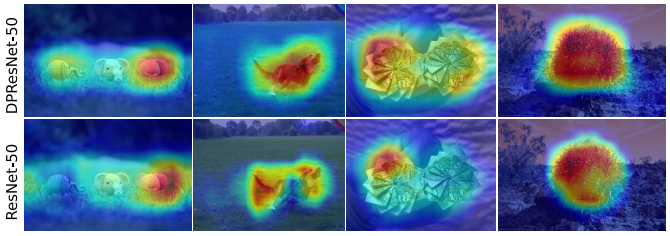


Fig. 9. Visualization of class activation mapping [85], using DPResNet-50 and ResNet-50 as backbone networks.

TABLE III

QUANTITATIVE COMPARISONS OF DEEPER DPRESNET WITH RELATED MULTI-SCALE REPRESENTATIVE METHODS. NOTICING THAT THE OFFICIAL CODE OF RES2NET DOESN'T PROVIDE THE 152 LAYER IMPLEMENTATION

Backbone	DUT-OMRON			DUTS-TE			#Params	#FLOPs
	F_β	S-m↑	MAE	F_β	S-m	MAE		
DPResNet-50(Ours)	**.837**	**.853**	**.049**	**.916**	**.912**	**.029**	27.1M	9.2G
ResNet-50 [29]	.819	.834	.060	.892	.889	.035	27.8M	9.4G
DenseNet-121 [86]	.825	.844	.057	.893	.889	.039	11.1M	8.2G
Res2Net50 [31]	.823	.844	.055	.900	.895	.036	27.9M	9.6G
DPResNet-101(Ours)	**.847**	**.861**	**.048**	**.918**	**.914**	**.028**	44.7M	12.6G
ResNet-101 [29]	.823	.841	.058	.894	.890	.036	46.7M	13.2G
Inception-V4 [30]	.826	.850	.049	.908	.905	.030	59.7M	16.5G
DenseNet-169 [86]	.825	.845	.057	.896	.892	.037	16.7M	8.7G
Res2Net101 [31]	.836	.852	.049	.907	.903	.031	47.4 M	13.4G
DPResNet-152(Ours)	**.847**	**.864**	**.047**	**.920**	**.915**	**.027**	59.1M	16.0G
ResNet-152 [29]	.827	.848	.053	.901	.898	.033	62.4M	16.9G
DenseNet-201 [86]	.837	.855	.050	.906	.899	.033	22.4M	9.7G

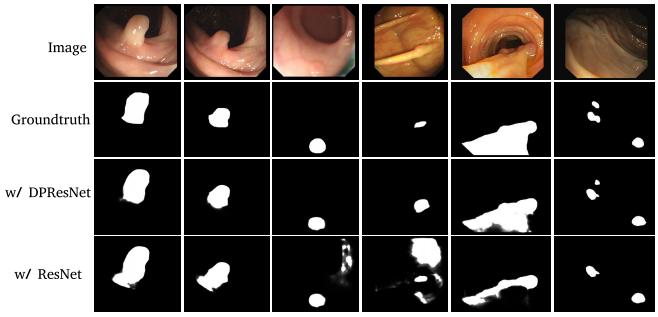


Fig. 10. Visual comparisons of the proposed DPResNet with the ResNet, where segmentation maps produced by our DPResNet are more accurate and complete in various scenarios.

computational cost by almost 25%. In the 3rd group, the DPResNet-152 also outperforms the ResNet-152 while costing 5% less computation. When compared with the DenseNet, our DPResNet achieves better performance than DenseNet at the expense of more parameters. Noticing that, breaking the upper limit of performance is often more important than using fewer parameters. Thus, our method aims atbreaking such performance upper bound by using as few parameters as possible.

In a word, the proposed DPResNet achieves significant performance gains over other multi-scale representation methods, suggesting the effectiveness of our proposed DPConv for the SOD task.

5) Efficiency Analysis: In the previous Sec. III, we have shown the computational cost of DPConv over the vanilla convolution. Here we extensively compare the proposed method with others in terms of F-measure, parameters,

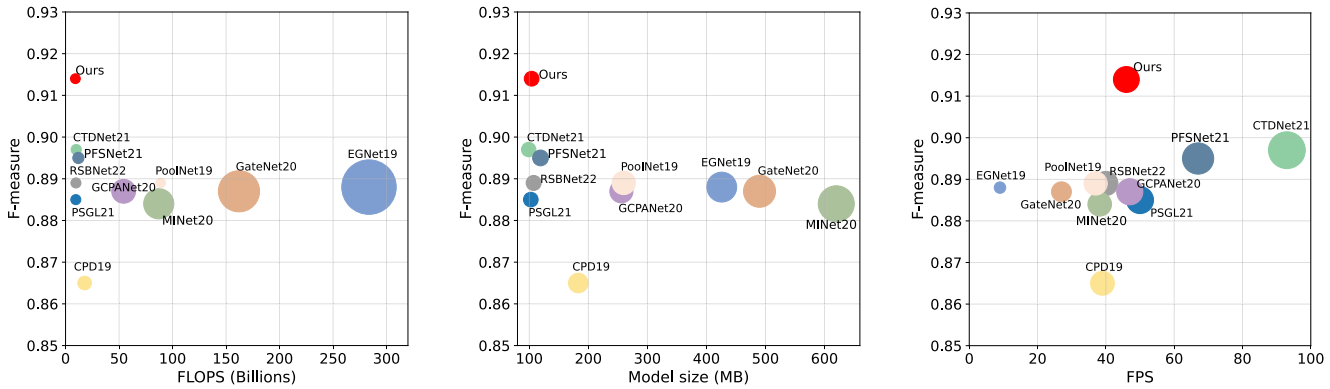


Fig. 11. Comparisons in terms of **maxF**, **FLOPs**, **Model-size**, and **FPS** towards our DPNet vs. RSBNet22 [68], CTDNet21 [33], PSGL21 [70], PFSNet21 [69], GateNet20 [34], GCPANet20 [54], MINet20 [10], CPD19 [23], PoolNet19 [74], EGNet19 [14] on the challenge DUTS-TE set [76]. All methods are based on ResNet [29] network. Circles' sizes are proportional to their FLOPs/Model-sizes/FPS.

TABLE IV
QUANTITATIVE COMPARISONS AMONG THE PROPOSED DPNET, CONDCONV AND DCNET

Methods	DUT-OMRON					DUTS-TE				
	F_β	MAE	W-F_β	S-m	E-m	F_β	MAE	W-F_β	S-m	E-m
DPNet(Ours)	**.837**	**.049**	**.770**	**.853**	**.876**	**.916**	**.029**	**.849**	**.912**	**.921**
DCNet [63]	.828	.050	.762	.847	.864	.906	.031	.837	.903	.917
CondConv [62]	.821	.0502	.758	.840	.860	.894	.033	.843	.895	.913

FLOPs, model size and FPS with other popular methods which have publicly released their codes, and these methods are RSBNet22 [68], CTDNet21 [33], PSGL21 [70], PFSNet21 [69], GateNet20 [34], GCPANet20 [54], MINet20 [10], CPD19 [23], PoolNet19 [74], and EGNet19 [14]. The input sizes are set according to either their papers or the default implementations, and we employ the *torchstat*[1] as the analyzer tool to compute the total number of network parameters and FLOPs. As shown in Fig. 2 and Fig. 11, our DPNet achieves better performance and efficiency than most previous methods on the DUTS-TE [76]. In particular, our proposed DPNet achieves comparable computational cost with RSBNet22 [68], CTDNet21 [33] and PFSNet21 [69], while it improves the F-measure score by 1.7% averagely. Our DPNet model also significantly outperforms GateNet20 [34], GCPANet20 [54], MINet20 [10], CPD19 [23], EGNet19 [14], and PoolNet19 [74], in both F-measure and computational cost. For instance, MINet20 [10] achieves 88.4% F-measure with 162.4 millions of parameters and 87.1 GFLOPs. Compared to MINet20, our DPNet improves the F-measure by 3.2%, using about 6x fewer parameters and 9.5× fewer FLOPs.

Furthermore, we have also conducted a comparison regarding FPS in the 3rd column of Fig. 11. To ensure a fair comparison, all compared SOTA models were evaluated on the same machine. Among the compared real-time models, our approach also achieves competitive performance with 45 FPS on an NVIDIA GTX 2080 GPU.

[1]https://github.com/Swall0w/torchstat

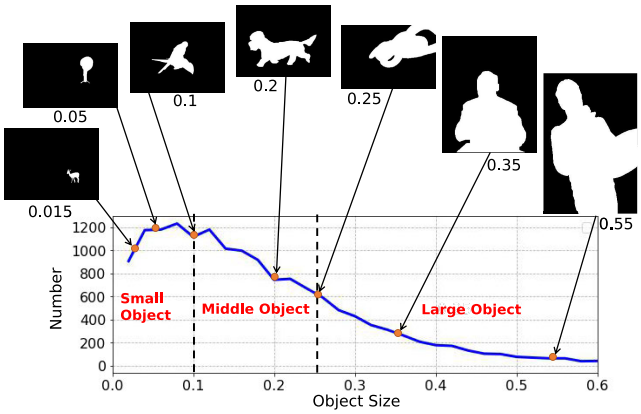


Fig. 12. The statistics of object size of the DUTS-TE, DUT-OMRON, ECSSD, HKU-IS, and PASCAL-S datasets.

6) Comparision With SOTA Dynamic Models: As shown in Table IV, we also compared our DPNet with the recent proposed CondConv [62] and DCNet [63]. As we can see, the proposed DPNet consistently outperforms the CondConv and DCNet. Concretely, our DPNet has an improvement of 1.1% in terms of F_β on ECSSD dataset. Similarly, the proposed DPNet outperforms DCNet by 1.6% on the ECSSD in terms of F_β. These experiments demonstrate that the proposed DPNet is more effective than the previous dynamic networks.

7) Statistics of Object Size: As shown in Fig. 12, we calculate the object size statistics of the DUTS-TE, DUT-OMRON, ECSSD, HKU-IS, and PASCAL-S datasets. Formally, we define the object size (S) by computing the proportion of foreground (F) against the whole image (I).

$$S = \frac{F}{I} \tag{14}$$

We define the **Small Object** with $S < 0.1$, **Middle Object** with $0.1 \leq S \leq 0.25$, and **Large Object** with $0.25 < S$. According to the statistics, small objects, middle objects and large objects account for 34.1%, 43.4%, and 22.5% of the entire dataset, respectively. The results show that the Fig. 1 of our original manuscript is not a special case. And small and large objects account for more than 56% of the entire dataset.

TABLE V

COMPONENT EVALUATION IN TERMS OF MAX F-MEASURE, S-MEASURE AND MAE ON FIVE COMMONLY USED DATASET. THE BEST RESULTS ARE SHOWN IN **BOLD** FONT

ResNet50	CFM	BiCFM	DPConv	DWF	DUT-OMRON			DUTS-TE		
					F_β	S-m	MAE	F_β	S-m	MAE
✓	✓				.810	.835	.057	.887	.885	.041
✓		✓			.818	.841	.053	.895	.892	.034
✓	✓		✓		.823	.842	.054	.896	.894	.035
✓		✓	✓		.829	.850	.050	.907	.905	.032
✓	✓		✓	✓	.831	.848	.052	.909	.908	.031
✓	✓	✓		✓	.825	.844	.051	.903	.898	.032
✓		✓	✓	✓	**.837**	**.853**	**.049**	**.916**	**.912**	**.029**
DPNet (w/o dynamic routing)					.828	.847	.052	.907	.903	.032
DPNet (w/ dynamic routing)					**.837**	**.853**	**.049**	**.916**	**.912**	**.029**

TABLE VI

RESULTS OF DPRESNET-50 WITH DIFFERENT COMBINATIONS OF KERNELS

K3	K5	K7	K9	K11	DUT-OMRON			DUTS-TE		
					F_β	S-m	MAE	F_β	S-m↑	MAE
✓					.824	.848	.053	.891	.892	.034
	✓				.823	.850	.049	.907	.903	.037
		✓			.821	.848	.050	.905	.902	.032
			✓		.816	.842	.051	.900	.896	.034
				✓	.806	.835	.053	.899	.895	.033
✓	✓				.827	.850	.048	.909	.905	.030
✓	✓	✓			.831	.848	.050	.913	.911	.029
✓	✓	✓	✓		**.837**	**.853**	**.049**	**.916**	**.912**	**.029**
✓	✓	✓	✓	✓	.832	.849	.051	.913	.907	.031

D. Component Evaluation

Since our model consists of multiple components (i.e., DPConv, BiCFM, and DWF), we shall explore the exact contribution of each of these components. Table V shows the component evaluation results. We start from the {ResNet50+CFM} as the baseline and then progressively extend it with different combinations. The {ResNet50+CFM} is inherently limited by its weak multi-scale feature representation and one-directional feature propagation, and thus it performs the worst. By replacing CFM with our BiCFM (in the 2nd row), we achieve 0.8% improvement in F-measure on the DUT-OMRON. We then replace the vanilla convolution with the proposed DPConv (in the 4th row), improving about 1.3% in F-measure with fewer parameters and FLOPs. Finally, we add the DWF to the model, achieving the best results.

As we can be seen, the model with all components has achieved the best performance, which demonstrates the necessity of each component for the proposed model to obtain the best SOD results. Besides, we have also evaluated the effect of the dynamic routing mechanism in the last two rows of Table V. With the dynamic routing, our DPNet achieves consistent performance enhancement.

E. Combination of Different Kernels in DPConv

We investigate how the combination of different kernels influences saliency performance. Here, we adopt five commonly-used kernels 3×3, 5×5, 7×7, 9×9, and 11×11. For simplicity, we use "Kn" to represent the $n\times n$ kernel. The quantitative results are shown in Table VI. We make the following observations.

TABLE VII

ABLATION STUDY FOR DIFFERENT BICFM NUMBERS (N, SEC. IV-A). CLEARLY, WHEN $N=2$, THE MODEL ACHIEVES THE BEST PERFORMANCE

Setting	DUT-OMRON			DUTS-TE		
	F_β	S-m	MAE	F_β	S-m	MAE
N=1	.825	.849	.051	.912	.907	.029
N=2	**.837**	**.853**	.049	**.916**	.912	.029
N=3	.831	.853	.050	.915	**.914**	.030
N=4	.829	.850	**.047**	.914	.911	**.027**

First, for the case with single-scale kernels, the trade-off between kernel size and performance is a concave curve, where performance rises up and then slowly goes down. In particular, K5 achieves its best performance, i.e., 1.6% higher than K3 in terms of F-measure on the DUTS-TE set. We attribute this improvement to the enlarged receptive field. When increasing the kernel size, the model detects non-salient regions, resulting in performance degradation.

Second, when the adopted kernel types are increased, the performance usually improves, except for the K11. This can be attributed to the dynamic routing mechanism, and we shall give an intuitive explanation by visualizing the activation maps at Conv5_3 level for different scale kernels. As shown in Fig. 14, each activation map has its own detecting pattern, yet these maps generally complement each other, indicating the optimal features representation can be captured via the proposed scale routing.

Third, further integrating K11 into a model, there will be slight performance degradation. One possible reason is that the proposed DPConv can already capture the multi-scale feature representation well by using kernel size from $K1$ to $K9$. If we further integrate the $K11$ into PPConv, it will introduce unnecessary feature perturbation, resulting in slight performance degradation. To strike the performance trade-off, we eventually choose the {K3, K5, K7, K9} combination.

F. BiCFM Number (N)

The BiCFM number N (Sec. IV-A) is a important hyperparameters for our model. We conduct a series of experiments on DUT-OMRON and DUTS-TE to study their effects. We gradually increase N from 1 to 4 and evaluate their performance scores of F-measure, S-measure and MAE. As shown in Table VII, we empirically find that the optimal setting for the number of BiCFM is $N=2$ (the 2nd column).

G. Analysis and Interpretation

To further reveal how DPConv works, we analyze the weight score by training our model in different image scales. Concretely, we take all images from the DUTS-TR set and then progressively enlarge the image size from $1.0\times$ to $2.0\times$. Fig. 13 illustrates the learned weights for different input scales in DPConv layers.

We first calculate the averaged weights for different kernels (K3, K5, K7, and K9) in each channel in the DPConv layer. Fig. 13 shows the weight scores in all channels for Conv2_3, Conv3_4, Conv4_6, and Conv5_3. As we can see, when the

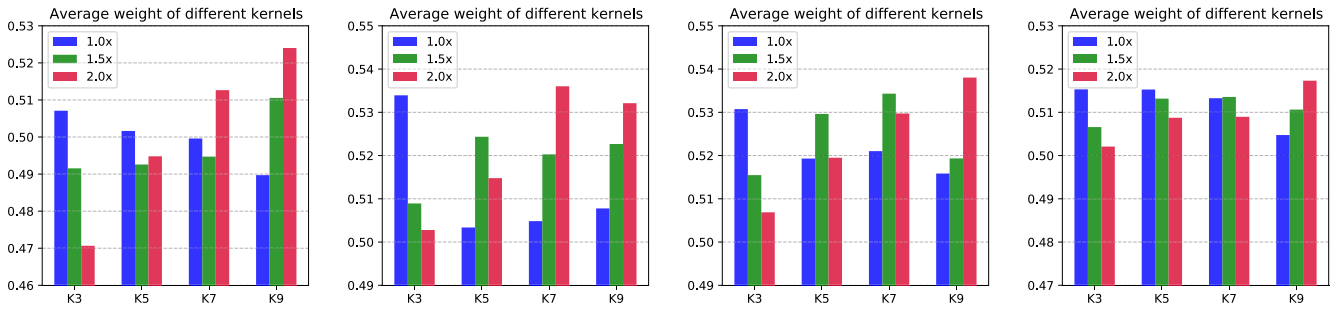


Fig. 13. Average kernels' scores for three different image sizes:1.0×, 1.5× and 2.0×.

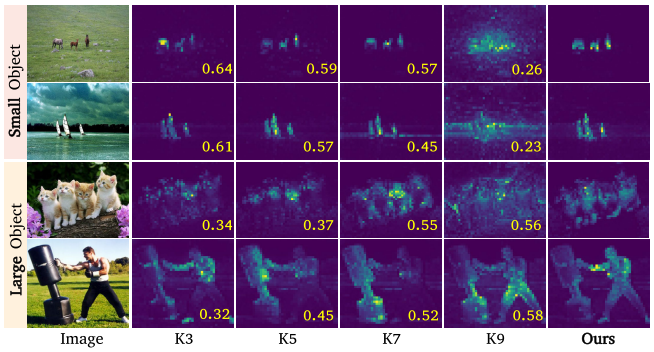


Fig. 14. Visual interpretation of our proposed DPConv at the $Conv5\_3$ stage. K3-K9 denote the models equipped with different kernel sizes, e.g., K3 denotes the response maps using a 3×3 kernel. "Ours" represent the response maps using the proposed DPConv.

selected images' sizes are enlarged, the weights for the small kernel (K3) tend to decrease, yet the weights for the large kernel (K9) get increased, which suggests that the dynamic routing can adaptively select appropriate feature representation according to the input images. We can easily find that the larger the input image is, the more attention will be assigned to kernels with large sizes (e.g., K7 and K9). In stage 5, such differences start to shrink.

In addition, a series of experiments have been conducted to verify our claim: the optimal feature representation is conditioned on the network's input. To this end, we begin with using the popular ResNet-50 as the baseline and replacing the regular K3 convolution with K5, K7, and K9, respectively. We present some typical activation maps at Conv5_3 level in Fig. 14. We have the following observations. First, feature representation with one type of kernel cannot achieve optimal performance. For instance, the horses (in the 1st row) are falsely activated in large kernel size (K9), but they can be predicted better in small kernels. In the last row, the large salient regions, which prefer larger receptive fields, are better activated with a large kernel size (K9). Second, different kernels tend to focus on different feature scales, which are complementary to each other. Third, compared with the conventional settings, the proposed model with dynamic routing can simultaneously highlight salient regions and compress non-salient nearby surroundings.

This evidence suggested that generating feature representation conditioned on the input sample is promising.

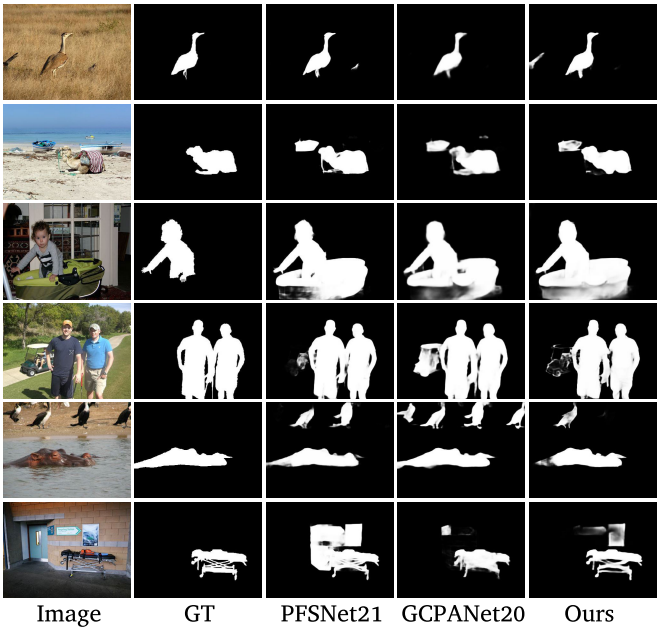


Fig. 15. Failure cases. For a better demonstration, the 3d and 4th columns show the saliency maps produced by the current SOTA models, i.e., PFSNet21 [69] and GCPANet20 [54].

H. Limitations

After taking a deeper look at the predicted saliency maps, we find that our DPNet tends to detect all possible saliency regions. The main reasons are two-fold: **1)** there are many controversial annotations scenes (people have different opinions about which is the salient object), which also confuses our DPNet as well as other SOTA models (e.g., PFSNet21 and GCPANet20). **2)** our DPNet contains multi-scale kernel size convolution layer, which can capture saliency objects with arbitrary size. For instance, in the 1st row, our model can highlight the salient object completely, but it was confused by the tree trunk behind the bird. In the 2nd row, all methods falsely detect the boat behind the camel as the salient object.

VI. Conclusion

In this paper, we attempt to tackle the "scale confusion" problem that is ubiquitous in all SOD-relevant tasks. We sought a novel design of a DPConv layer, which tries to revolutionize existing static convolutions with a novel

dynamic routing mechanism. The newly-proposed DPConv is a generic plug-in for any existing standard convolutional layer without much modification. The dynamic nature of the proposed DPConv ensures its features are scale-aware, thus formulating implicit decision rules to automatically bias towards those most useful scales. Moreover, to better match the DPConv-based encoder, we focused on a novel decoder's design inspired by the same dynamic rationale. The new decoder could dynamically collect scale-aware features in a bidirectional cross-scale fashion, significantly improving the SOTA SOD performance. Extensive experiments on six benchmarks confirm the superiority of our proposed model with a much better trade-off between performance and model efficiency.

REFERENCES

[1] W. Wang, J. Shen, X. Dong, A. Borji, and R. Yang, "Inferring salient objects from human fixations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 1913–1927, Aug. 2019.

[2] W. Wang, J. Shen, X. Lu, S. C. H. Hoi, and H. Ling, "Paying attention to video object pattern understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2413–2428, Jul. 2021.

[3] W. Wang, G. Sun, and L. Van Gool, "Looking beyond single images for weakly supervised semantic segmentation learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 19, 2022, doi: 10.1109/TPAMI.2022.3168530.

[4] C. Chen, S. Li, H. Qin, and A. Hao, "Structure-sensitive saliency detection via multilevel rank analysis in intrinsic feature space," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2303–2316, Aug. 2015.

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

[6] Y.-F. Ma and H.-J. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *Proc. 11th ACM Int. Conf. Multimedia*, 2003, pp. 374–381.

[7] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1597–1604.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[9] J. Wei, S. Wang, and Q. Huang, "F^3Net: Fusion, feedback and focus for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12321–12328.

[10] Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9413–9422.

[11] H. Zhou, X. Xie, J.-H. Lai, Z. Chen, and L. Yang, "Interactive two-stream decoder for accurate and fast saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1449–1457.

[12] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1623–1632.

[13] J. Su, J. Li, Y. Zhang, C. Xia, and Y. Tian, "Selectivity or invariance: Boundary-aware salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3799–3808.

[14] J. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "EGNet: Edge guidance network for salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8779–8788.

[15] J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian, "Label decoupling framework for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13025–13034.

[16] C. Li, R. Cong, Y. Piao, Q. Xu, and C. C. Loy, "RGB-D salient object detection with cross-modality modulation and selection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 225–241.

[17] J. Zhang et al., "Uncertainty inspired RGB-D saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5761–5779, Sep. 2022.

[18] K. Fu, D.-P. Fan, G.-P. Ji, Q. Zhao, J. Shen, and C. Zhu, "Siamese network for RGB-D salient object detection and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5541–5559, Sep. 2022.

[19] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 234–250.

[20] N. Liu, J. Han, and M.-H. Yang, "PiCANet: Learning pixel-wise contextual attention for saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3089–3098.

[21] W. Wang, S. Zhao, J. Shen, S. C. H. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1448–1457.

[22] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 714–722.

[23] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3907–3916.

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

[25] B. Wang, Q. Chen, M. Zhou, Z. Zhang, X. Jin, and K. Gai, "Progressive feature polishing network for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12128–12135.

[26] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 202–211.

[27] X. Hu, L. Zhu, J. Qin, C.-W. Fu, and P.-A. Heng, "Recurrently aggregating deep features for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6943–6950.

[28] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7479–7489.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[31] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.

[32] L. Wang et al., "Learning to detect salient objects with image-level supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 136–145.

[33] Z. Zhao, C. Xia, C. Xie, and J. Li, "Complementary trilateral decoder for fast and accurate salient object detection," in *Proc. 11th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 4967–4975.

[34] X. Zhao, Y. Pang, L. Zhang, H. Lu, and L. Zhang, "Suppress and balance: A simple gated network for salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 35–51.

[35] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 29–42.

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

[37] B. Xu, H. Liang, R. Liang, and P. Chen, "Locate globally, segment locally: A progressive architecture with knowledge review network for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 4, pp. 3004–3012.

[38] J.-J. Liu, Z.-A. Liu, P. Peng, and M.-M. Cheng, "Rethinking the U-shape structure for salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 9030–9042, 2021.

[39] J. Li, J. Su, C. Xia, M. Ma, and Y. Tian, "Salient object detection with purificatory mechanism and structural similarity loss," *IEEE Trans. Image Process.*, vol. 30, pp. 6855–6868, 2021.

[40] Z. Wu, L. Su, and Q. Huang, "Decomposition and completion network for salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 6226–6239, 2021.

[41] L. T. Bo Li, Y. Zhong, S. Ding, and M. Song, "Disentangled high quality salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3580–3590.

[42] M. Zhuge, D.-P. Fan, N. Liu, D. Zhang, D. Xu, and L. Shao, "Salient object detection via integrity learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 6, 2022, doi: 10.1109/TPAMI.2022.3179526.

[43] Z. Wu et al., "Synthetic data supervised salient object detection," in *Proc. 11th ACM Int. Conf. Multimedia*, 2022, pp. 5557–5565.

[44] Z. Wu, S. Li, C. Chen, A. Hao, and H. Qin, "Deeper look at image salient object detection: Bi-stream network with a small training dataset," *IEEE Trans. Multimedia*, vol. 24, pp. 73–86, 2022.

[45] Z. Wu, S. Li, C. Chen, A. Hao, and H. Qin, "Recursive multi-model complementary deep fusion for robust salient object detection via parallel sub-networks," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108212.

[46] C. Chen, S. Li, Y. Wang, H. Qin, and A. Hao, "Video saliency detection via spatial–temporal fusion and low-rank coherency diffusion," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3156–3170, Jul. 2017.

[47] X. Wang, S. Li, C. Chen, Y. Fang, A. Hao, and H. Qin, "Data-level recombination and lightweight fusion scheme for RGB-D salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 458–471, 2021.

[48] C. Chen, S. Li, H. Qin, and A. Hao, "Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis," *Pattern Recognit.*, vol. 52, pp. 410–432, Apr. 2016.

[49] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu, "A stagewise refinement model for detecting salient objects in images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4019–4028.

[50] T. Wang et al., "Detect globally, refine locally: A novel approach to saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3127–3135.

[51] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, "A bi-directional message passing model for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1741–1750.

[52] W. Wang, J. Shen, M.-M. Cheng, and L. Shao, "An iterative and cooperative top-down and bottom-up inference network for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5968–5977.

[53] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5300–5309.

[54] Z. Chen, Q. Xu, R. Cong, and Q. Huang, "Global context-aware progressive aggregation network for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10599–10606.

[55] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3239–3259, Jun. 2022.

[56] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[57] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama, "Adaptive neural networks for efficient inference," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 527–536.

[58] X. Wang, F. Yu, Z.-Y. Dou, T. Darrell, and J. E. Gonzalez, "SkipNet: Learning dynamic routing in convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 409–424.

[59] Z. Wu et al., "BlockDrop: Dynamic inference paths in residual networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8817–8826.

[60] B. Klein, L. Wolf, and Y. Afek, "A dynamic convolutional layer for short range weather prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4840–4848.

[61] X. Jia, B. De Brabandere, T. Tuytelaars, and L. Van Gool, "Dynamic filter networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–9.

[62] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "CondConv: Conditionally parameterized convolutions for efficient inference," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[63] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11030–11039.

[64] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[65] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.

[66] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[67] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[68] Y. Y. Ke and T. Tsubono, "Recursive contour-saliency blending network for accurate salient object detection," in *Proc. WACV*, Jan. 2022, pp. 2940–2950.

[69] M. Ma, C. Xia, and J. Li, "Pyramidal feature shrinking for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2311–2318.

[70] S. Yang, W. Lin, G. Lin, Q. Jiang, and Z. Liu, "Progressive self-guided loss for salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 8426–8438, 2021.

[71] Y. Liu, X.-Y. Zhang, J.-W. Bian, L. Zhang, and M.-M. Cheng, "SAMNet: Stereoscopically attentive multi-scale network for lightweight salient object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 3804–3814, 2021.

[72] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, "Uncertainty-aware joint salient object and camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10071–10081.

[73] S. Chen, X. Tan, B. Wang, H. Lu, X. Hu, and Y. Fu, "Reverse attention-based residual network for salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 3763–3776, 2020.

[74] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3917–3926.

[75] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3085–3094.

[76] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3166–3173.

[77] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1155–1162.

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

[79] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 280–287.

[80] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 248–255.

[81] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4548–4557.

[82] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *Proc. Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 698–704.

[83] L. Zhang, J. Wu, T. Wang, A. Borji, G. Wei, and H. Lu, "A multistage refinement network for salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 3534–3545, 2020.

[84] J. Wei, Y. Hu, R. Zhang, Z. Li, S. Zhou, and S. Cui, "Shallow attention network for polyp segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Strasbourg, France: Springer, 2021, pp. 699–708.

[85] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[86] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[87] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, May 2015, pp. 1–14.