SAMNet: Stereoscopically Attentive Multi-scale Network for Lightweight Salient Object Detection

Yun Liu*, Xin-Yu Zhang*, Jia-Wang Bian, Le Zhang, Ming-Ming Cheng

***Abstract*—Recent progress on salient object detection (SOD) mostly benefits from the explosive development of Convolutional Neural Networks (CNNs). However, much of the improvement comes with the larger network size and heavier computation overhead, which, in our view, is not mobile-friendly and thus difficult to deploy in practice. To promote more practical SOD systems, we introduce a novel Stereoscopically Attentive Multi-scale (SAM) module, which adopts a stereoscopic attention mechanism to adaptively fuse the features of various scales. Embarking on this module, we propose an extremely lightweight network, namely SAMNet, for SOD. Extensive experiments on popular benchmarks demonstrate that the proposed SAMNet yields comparable accuracy with state-of-the-art methods while running at a GPU speed of 343fps and a CPU speed of 5fps for 336×336 inputs with only 1.33M parameters. Therefore, SAMNet paves a new path towards SOD. The source code is available on the project page https://mmcheng.net/SAMNet/.**

***Index Terms*—Lightweight salient object detection, lightweight saliency detection, multi-scale learning.**

I. Introduction

SALIENT object detection (SOD), also known as saliency detection, aims at detecting the most visually distinctive objects or regions in natural images [1]. The progress in SOD has been beneficial to a wide range of computer vision applications, including image retrieval [2], image segmentation [3], object detection [4], visual tracking [5], scene classification [6], content-aware image editing [7], *etc*. Conventional methods for this task mainly rely on hand-crafted low-level features and heuristic priors [1], [8], [9], but the lack of high-level semantic information usually leads to limited accuracy. Recently, thanks to the unprecedented success of Convolutional Neural Networks (CNNs), especially Fully Convolutional Networks (FCNs), deep learning based methods have refreshed the state-of-the-art performance of SOD [10]–[33].

However, those improvements do not come without cost: they usually rely on large network size and substantial computational overhead [14], [16], [21], [30], [31], [33]–[36]. For example, EGNet [37] with the VGG16 backbone has 108M parameters and needs ∼432 MB disk to store its pretrained model. Moreover, EGNet [37] can only run at 0.09fps on the powerful i7-8700K CPU and 12.7fps on an NVIDIA TITAN XP GPU for 336×336 images. Such ponderousness undoubtedly makes it less practical for real-time and resource-constrained applications such as autonomous driving, robots, augmented reality, and so on. In those scenarios, mobile devices have reduced computational capabilities, restrictive memory constraints, and limited energy overhead.

Designing lightweight CNNs could be definitely one of the solutions for the above problem, and it has been studied for other tasks such as image classification [38]–[41]. Although drawing inspirations from them, we are the first to make an effort in SOD. This is nontrivial due to the following challenges: i) SOD requires both high-level abstract semantic features and low-level fine-grained features to locate salient objects and refine object details, respectively; ii) SOD needs multi-scale information to process salient objects with various sizes and aspect ratios in natural scenes. Since lightweight networks usually have shallow depths and simplified operations, they are less potent in multi-level and multi-scale learning than traditional large networks [42], [43]. Therefore, naively applying existing lightweight backbone networks such as MobileNets [38], [39] and ShuffleNets [40], [41] into SOD leads to suboptimal performance, which will be demonstrated in the experiments.

It is well-known that CNNs can learn high-level semantic information at their top sides and low-level fine details at their bottom sides. This makes different side-outputs of CNNs contain multi-scale information. Hence, to learn multi-level and multi-scale information, current state-of-the-art SOD methods (with large networks) adopt encoder-decoder network architectures [10]–[29] to integrate the multi-level side-output features of backbone networks. Recent development of SOD mainly comes from new strategies and modules for the effective fusion of multi-level backbone features.

Based on the above analyses, the key to lightweight SOD is how to effectively learn multi-level and multi-scale information within limited parameter budgets. Instead of integrating different side-outputs of backbone networks [14], [19], [21], [23], [28], [30], [37] or summarizing the convolutional features of different dilation rates [15], [44] as done in previous studies, we propose a novel Stereoscopically Attentive Multi-scale (SAM) module for multi-scale learning. It adopts a stereoscopic attention mechanism to automatically control the learning at different scales, so it has the capability to effectively learn the necessary information at various levels of deep CNNs. Using SAM module as the basic unit, we build a lightweight encoder-decoder network, namely SAMNet, to integrate the multi-level and multi-scale features learned by SAM modules. SAMNet achieves comparable accuracy with state-of-the-art SOD methods while running at 5fps on an i7-

Y. Liu and M.M. Cheng are with College of Computer Science, Nankai University. X.Y. Zhang is with School of Mathematical Science, Nankai University. J.W. Bian is with the School of Computer Science, The University of Adelaide, Australia. L. Zhang is with the Agency for Science Technology and Research (A*STAR), Singapore.

The first two authors contributed equally to this paper. M.M. Cheng is the corresponding author (cmm@nankai.edu.cn).

8700K CPU and 343fps on an NVIDIA TITAN XP GPU for 336×336 images. Moreover, SAMNet only has 1.33M parameters. These lightweight properties make it possible for practical mobile applications.

In summary, our contributions are threefold:

- We propose a novel **Stereoscopically Attentive Multi-scale** (**SAM**) module that adopts a stereoscopic attention mechanism for effective and efficient multi-scale learning.
- Using the SAM module as the basic unit, we propose SAMNet, a lightweight encoder-decoder architecture for SOD, which is the first lightweight SOD model as we know.
- We empirically evaluate SAMNet on six popular SOD datasets and demonstrate its comparable accuracy, much higher efficiency, and much smaller network size.

II. Related Work

a) Salient Object Detection: Over the past two decades, numerous methods have been proposed to detect salient objects in an image. Traditional methods are mainly based on hand-crafted features, such as image contrast [1], texture [45], central prior [8], background prior [46], *etc*. Despite the efficiency of these approaches, hand-crafted features intrinsically lack the capacity for high-level representation, leading to limited performance.

With the rapid development of deep learning, CNN-based SOD methods have surpassed traditional counterparts by a considerable margin. Early CNN-based methods [24], [47]–[49] include several fully-connected layers, resulting in the loss of essential spatial information of the whole image. Since the seminal work [50] proposed FCNs to predict semantic labels at the pixel level, FCN-based SOD approaches [11]–[16], [19]–[23], [25] have dominated this field by exploring the multi-level and multi-scale deep features, as described above.

Although previous approaches have achieved high accuracy by employing powerful CNNs, they are relatively slow in speed, hungry in energy consumption, and occupy large memory space. These shortcomings make it difficult to deploy state-of-the-art methods into real-world applications. This is our motivation for this work, *i.e.,* towards lightweight SOD that trades off accuracy, model size, and speed. Our technical motivation comes from the fact that most of previous CNN-based methods [10]–[29] improve the performance through the exploration in multi-scale and multi-level deep learning. In this paper, we propose a novel SAM module that adopts stereoscopic attention for lightweight multi-scale learning.

b) Attention Mechanism: Attention mechanism plays an essential role in human perception [51], [52]. Instead of processing the whole image at once, human visual system adaptively filters less essential information like image background to enhance the capture of visual structure. This has inspired the research in deep learning.

Recent computer vision community has witnessed numerous successes of attention mechanism in a wide range of tasks, including SOD [19], [26], [34], sequence learning [53], person re-ID [54], and image recovery [55]. For image classification, Wang *et al.* [56] proposed to use an hourglass module to generate attention maps for hidden features. Furthermore, Hu *et al.* [57] proposed a "Squeeze-and-Excitation" module to explicitly exploit inter-channel relationships and adaptively recalibrate feature maps in a channel-wise manner. Beyond channel-wise attention, CBAM [58] introduces spatial attention in a similar way. These methods fall into the *self-attention* category. Spatial and channel-wise *self-attention* can adaptively emphasize the most informative feature patches and channels, respectively. Different from these methods, we introduce a stereoscopic attention mechanism to adaptively recalibrate information flow from multiple branches based on both channel inter-dependencies and spatial contextual clues. Hence the proposed SAM module can effectively learn multi-scale information under a lightweight setting.

III. Methodology

In this section, we elaborate on the proposed framework for SOD. In Section III-A, we present a simple multi-scale module. In Section III-B, we propose the SAM module for effective multi-scale learning. Finally, in Section III-C, we incorporate the proposed SAM module into the encode-decoder network and elaborate on the full network architecture.

A. Multi-scale Learning

Based on the above analyses, the multi-scale feature representations of CNNs are of great importance for SOD [14], [19], [23], [28], [30], [37]. Inspired by this, we first propose a simple multi-scale module to process visual information at different scales. Lightweight is central to our design. We adopt dilated convolutions with different dilation rates to capture multi-scale information and use the depthwise separable convolution to reduce floating-point operations and model parameters. We call it *dilated depthwise separable convolution* and use it as the basic convolutional operator to scale up the network dimensionality in terms of depth and width.

Formally, let $\boldsymbol{I} \in \mathbb{R}^{C\times H\times W}$ be the input feature map whose number of channels, height, and width are C, H, and W, respectively. With the input $\boldsymbol{I}$, we first apply a single depthwise separable $\mathrm{conv}3\times 3$ ($\mathrm{DSConv}3\times 3$ for short) to extract common information $\boldsymbol{F}_0$ for each branch, namely,

$$\boldsymbol{F}_0 = \mathcal{K}_0(\boldsymbol{I}), \tag{1}$$

where $\mathcal{K}_0$ denotes a $\mathrm{DSConv}3\times 3$ operation. At different branches, dilated $\mathrm{DSConv}3\times 3$ with different dilation rates are applied to $\boldsymbol{F}_0$, *i.e.,*

$$\boldsymbol{F}_i = \mathcal{K}_i(\boldsymbol{F}_0),\ i = 1, 2, \cdots, N, \tag{2}$$

where $\mathcal{K}_i$ denotes the dilated $\mathrm{DSConv}3\times 3$ operation at branch i, and N is the number of branches. Then, contextual information at different scales is aggregated by a single element-wise summation with a residual connection, namely,

$$\boldsymbol{F} = \sum_{i=0}^{N} \boldsymbol{F}_i. \tag{3}$$

Here, we use element-wise summation instead of concatenation, because concatenation will greatly increase the number

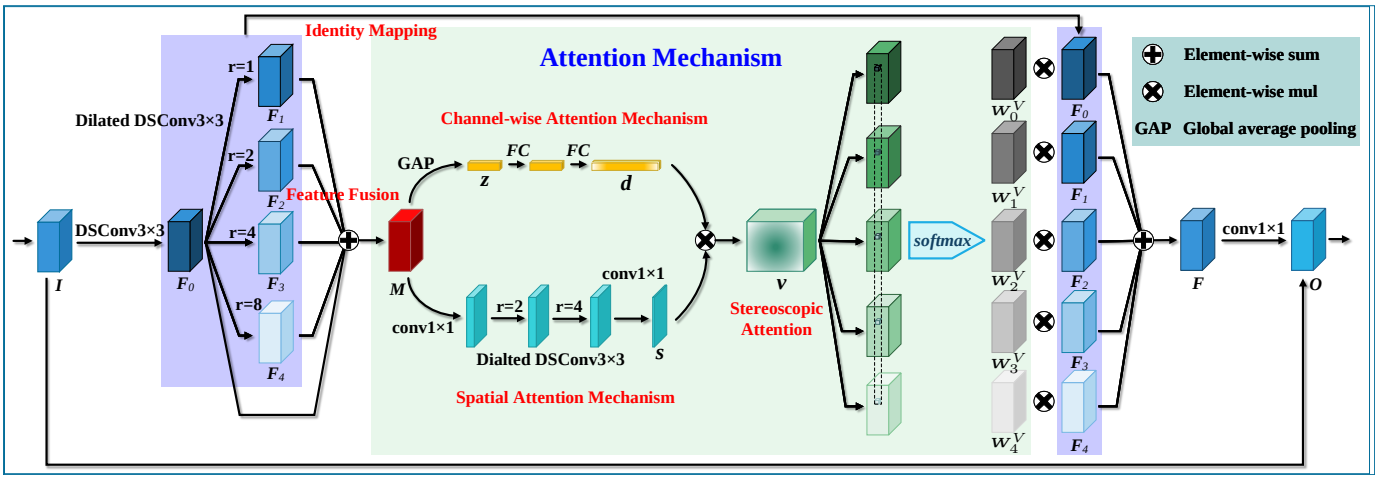

Fig. 1. Illustration of the proposed SAM module. Given the input feature map $\boldsymbol{I}$, SAM first extracts multi-scale features $\boldsymbol{F}_i$ $(i = 0, 1, \cdots, N)$ through multiple branches. Multi-scale features $\boldsymbol{F}_i$ are aggregated via element-wise summation to produce $\boldsymbol{M}$. Then, we calculate channel-wise attention logits $\boldsymbol{d}$ and spatial attention logits $\boldsymbol{s}$ based on $\boldsymbol{M}$. $\boldsymbol{d}$ and $\boldsymbol{s}$ are multiplied to get stereoscopic attention logits $\boldsymbol{v}$ that are transformed to stereoscopic attention weights $\boldsymbol{W}_i^V$ $(i = 0, 1, \cdots, N)$ using a *softmax* function. $\boldsymbol{W}_i^V$ $(i = 0, 1, \cdots, N)$ are used as weights to aggregate $\boldsymbol{F}_i$ to obtain $\boldsymbol{F}$. At last, a residual connection [43] from $\boldsymbol{I}$ is added to get the output $\boldsymbol{O}$. In the figure, $\otimes$ indicates element-wise multiplication, and the two multiplied feature maps are replicated to the same shape before multiplication. The symbol r denotes the dilation rate of the dilated DSConv3 $\times$ 3. Best viewed in color.

of channels, leading to heavier computational complexity and more network parameters. Finally, the aggregated features are further rearranged by a vanilla conv1 $\times$ 1, *i.e.,*

$$\boldsymbol{O} = \mathcal{K}_{fuse}(\boldsymbol{F}) + \boldsymbol{I}, \tag{4}$$

where $\mathcal{K}_{fuse}$ denotes the conv1 $\times$ 1 operation to fuse contextual information at various scales. The summation of $\boldsymbol{I}$ denotes a residual connection [43], which has demonstrated to be effective in CNN training.

The dilation rates and the number of branches are hyper-parameters involved in our multi-scale module. Empirically, larger dilation rates and more branches are desired when the input features $\boldsymbol{I}$ are of high resolution, as large feature maps usually possess contextual information at various scales.

B. Stereoscopically Attentive Multi-scale Module

A potential drawback of the multi-scale module in Section III-A lies in the element-wise summation. When contextual information from different branches is directly summarized together, informative branches may be weakened or even overwhelmed by non-informative ones. On the other hand, the layers at different network depths may prefer the information from different scales, while the element-wise summation assigns equal importance to all scales. To alleviate this problem, we propose a novel stereoscopic attention mechanism that allows each channel at each spatial location to adaptively adjust the weight of each branch with a soft attention mechanism. Fig. 1 provides an illustration of the proposed SAM module with four branches.

a) Stereoscopic Attention Mechanism: Without loss of generality, we consider the input feature is a 4-dimensional tensor $F \in \mathbb{R}^{(N+1)\times C\times H\times W}$, in which each branch $i \in \{0, \cdots N\}$ generates features $F_i \in \mathbb{R}^{C\times H\times W}$ of different scale and semantic level. It is widely believed that layers from different depth levels in a network may prefer the information from different scales, and thus blindly feeding all features F may lead to severe overfitting. A natural solution is to attentively suppress non-informative branches and promote discriminative ones automatically, and we achieve this by the well-established attention mechanism.

In our setting, an ideal attention module should have the following features. i) Due to the independence of each channel, the final attention should have a strong intra-channel dependency. More specifically, different feature channels are usually obtained by convolving independent filters with input features. In this case, if a feature from a particular channel is informative for the final prediction, the features in the same channel of the same branch are likely to be informative as well. ii) The final attention should have a strong spatial-wise dependency. The reason could be that, as a mid-level task, SOD requires certain level of reasoning in the local neighbourhood for each pixel. iii) The computation should be efficient. In this case, the naïve solution of independently learning a group (*e.g.,* $C \times H \times W$) of branch-wise attention weights is suboptimal due to its heavy computational overload.

The first two requirements mentioned above regularize the attention mechanism in a global and local manner, respectively. This naturally motivates us to factorize the final attention weights $\boldsymbol{v}$ into two individual weights as follows:

$$\boldsymbol{v} = \boldsymbol{d} \otimes \boldsymbol{s}, \tag{5}$$

where $\boldsymbol{d}$ and $\boldsymbol{s}$ denote the channel-wise attention and the spatial-wise attention, respectively. $\otimes$ indicates element-wise multiplication, and $\boldsymbol{d}$ and $\boldsymbol{s}$ are replicated to the same shape of $(N+1)\times C\times H\times W$ before multiplication. More specifically, channel-wise attention $\boldsymbol{d} \in \mathbb{R}^{(N+1)\times C}$ shrinks feature spatially and attentively suppress non-informative features and promote discriminative ones in a channel-wise global manner. In the same way, the spatial-wise attention $\boldsymbol{s} \in \mathbb{R}^{(N+1)\times H\times W}$ absorbs the features at a particular spatial location across different channels. Finally, $\boldsymbol{d}$ and $\boldsymbol{s}$ are broadcasted into the

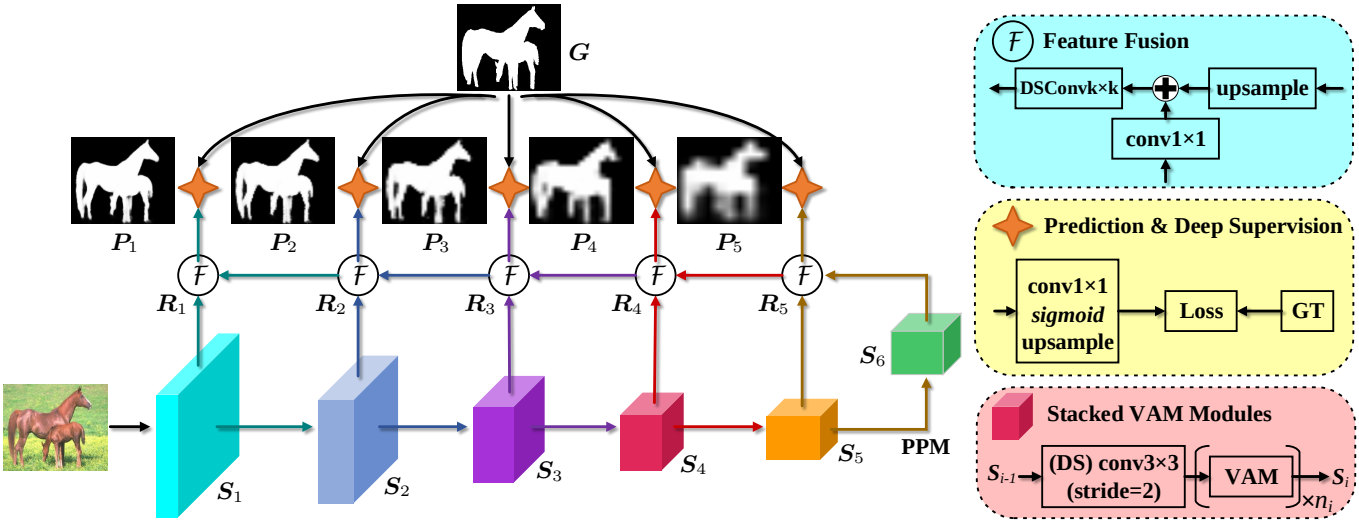

Fig. 2. Overall encoder-decoder architecture of the proposed SAMNet. $\boldsymbol{S}_i$ and $\boldsymbol{R}_i$ represent the output feature maps of the encoder and the decoder at the i^{th} stage, respectively. $\boldsymbol{P}_i$ is the predicted saliency map at the i^{th} stage, and $\boldsymbol{P}_1$ is the final prediction of SAMNet. $\boldsymbol{G}$ is the ground-truth saliency map. PPM denotes the well-known Pyramid Pooling Module [59]. Best viewed in color.

same dimensionality of $(N+1)\times C\times H\times W$ to obtain the final attention weight by Eq. (5).

b) Preprocessing by scale fusion: In order to reduce the computational overload, we aggregate multi-scale contextual information with element-wise summation, namely,

$$\boldsymbol{M}=\sum_{i=0}^{N}\boldsymbol{F}_i, \tag{6}$$

The fused features $\boldsymbol{M}$ is used to compute the gating statistics for each branch. Different from *self-attention*, the attention for each module branch is decided by the learning of all branches, instead of a single branch as in *self-attention* mechanisms. Therefore, the SAM module can extract the informative features from all branches within a "global" view. As a result, the SAM module has the capability to learn the features at its preferred scales.

c) Channel-wise Attention Mechanism: The channel attention mechanism aims at calculating a channel-wise attention vector $\boldsymbol{W}_i^D\in\mathbb{R}^C$ for each branch i, $i=0,1,\cdots,N$. To explore the inter-channel relationship among different channels [57], we embed the global information using global average pooling (GAP) on the fused feature map $\boldsymbol{M}$, *i.e.,*

$$\begin{aligned}\boldsymbol{z}_c=\mathcal{F}_{\text{GAP}}(\boldsymbol{M})=\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\boldsymbol{M}_{c,i,j},\\ c=0,1,\cdots,C-1,\end{aligned} \tag{7}$$

where $\boldsymbol{z}\in\mathbb{R}^C$ is the latent vector encoding channel-wise information of $\boldsymbol{M}$. Then, we apply a multi-layer perceptron (MLP) with two layers[1] on the latent vector, and extract channel-wise information at different scales as follows:

$$\boldsymbol{d}=\mathcal{F}_{\text{MLP}}(\boldsymbol{z}), \tag{8}$$

[1]We insert batch normalization and ReLU activation function between the two linear transformations.

where $\boldsymbol{d}\in\mathbb{R}^{(N+1)C}$ is further reshaped to $\mathbb{R}^{(N+1)\times C}$. The *softmax* function is applied to $\boldsymbol{d}$ on the branch-wise dimension to obtain the channel-wise attention, *i.e.,*

$$\begin{aligned}\boldsymbol{W}_{i,c}^D=\frac{e^{\boldsymbol{d}_{i,c}}}{\sum_{j=0}^{N}e^{\boldsymbol{d}_{j,c}}},\\ i=0,1,\cdots,N;c=1,2,\cdots,C.\end{aligned} \tag{9}$$

With channel-wise attention incorporated, the feature aggregation in Eq. (3) is rewritten as

$$\boldsymbol{F}^D=\sum_{i=0}^{N}\boldsymbol{W}_i^D\otimes\boldsymbol{F}_i, \tag{10}$$

where $\boldsymbol{W}_i^D$ is replicated to the same shape as $\boldsymbol{F}_i$ (*i.e.,* $\mathbb{R}^{C\times H\times W}$) before element-wise multiplication. The fusion of Eq. (4) is applied to $\boldsymbol{F}^D$ to serve as the output.

d) Spatial Attention Mechanism: The spatial attention mechanism aims at computing a spatial attention map $\boldsymbol{W}_i^S\in\mathbb{R}^{H\times W}$ to highlight or suppress the activation at specific locations. It is well-acknowledged that large receptive fields can capture contextual information better, which is crucial for learning location-wise attention [58]. Based on this, we adopt dilated DSConv3 $\times$ 3 to enlarge the receptive fields while maintaining low computational complexity. Specifically, the fused features $\mathbf{M}$ is first projected to a low-dimensional space $\mathbb{R}^{C/4\times H\times W}$ by a conv1 $\times$ 1 for reducing parameters and computational cost. Then, two dilated DSConv3 $\times$ 3 are applied to the reduced features for efficient contextual information aggregation. Finally, the features are again reduced to $\mathbb{R}^{(N+1)\times H\times W}$ using a conv1 $\times$ 1. Mathematically, we have

$$\boldsymbol{s}=\mathcal{F}_4^{1\times1}(\mathcal{F}_3^{3\times3}(\mathcal{F}_2^{3\times3}(\mathcal{F}_1^{1\times1}(\boldsymbol{M})))), \tag{11}$$

where $\mathcal{F}_i^{k\times k}$ denotes the i^{th} (depthwise separable or vanilla) $k\times k$ convolution. Similar to channel-wise attention mecha-

nism, we can formulate the fusion of multiple branches using spatial attention as follows:

$$\begin{aligned} \boldsymbol{W}^S_{i,h,w} &= \frac{e^{\boldsymbol{s}_{i,h,w}}}{\sum_{j=0}^{N} e^{\boldsymbol{s}_{j,h,w}}},\ i = 0, 1, \cdots, N, \\ \boldsymbol{F}^S &= \sum_{i=0}^{N} \boldsymbol{W}^S_i \otimes \boldsymbol{F}_i, \end{aligned} \tag{12}$$

in which we have $0 \leq h < H$ and $0 \leq w < W$. $\boldsymbol{W}^S_i$ is replicated to the shape of $\mathbb{R}^{C\times H\times W}$ before multiplication. Eq. (4) is applied to $\boldsymbol{F}^S$ to produce the output.

e) Final Normalization: With $\boldsymbol{d}$ and $\boldsymbol{s}$ defined, the *softmax* function is applied to the stereoscopic attention $\boldsymbol{v}$ on the branch-wise dimension, namely,

$$\boldsymbol{W}^V_{i,c,h,w} = \frac{e^{\boldsymbol{v}_{i,c,h,w}}}{\sum_{j=0}^{N} e^{\boldsymbol{v}_{j,c,h,w}}},\ i = 0, 1, \cdots, N. \tag{13}$$

After this, $\boldsymbol{W}^V_i \in \mathbb{R}^{C\times H\times W}$ serves as the stereoscopic weighting scalars for the i^{th} branch. The feature fusion in Eq. (3) is rewritten as

$$\boldsymbol{F}^V = \sum_{i=0}^{N} \boldsymbol{W}^V_i \otimes \boldsymbol{F}_i, \tag{14}$$

Eq. (4) can also be applied to $\boldsymbol{F}^V$ to generate the output of a SAM module. We display the computation process of $\boldsymbol{F}^V$ in Fig. 1.

So far, we have designed four modules for multi-scale learning, *i.e.,* $\boldsymbol{F}$, $\boldsymbol{F}^D$, $\boldsymbol{F}^S$, and $\boldsymbol{F}^V$. Among them, $\boldsymbol{F}^V$ is the default SAM module in this paper, because it achieves better performance with the consideration of both channel-wise attention and spatial attention. With the above design, each SAM module can automatically decide what information it wants through the control of multi-scale branches.

C. Network Architecture

a) Backbone Architecture: Following previous studies [30]–[32], [35], [60], we build an FCN structure [50] using the proposed SAM module as the basic unit. Within the first five stages, we use $\text{DSConv3} \times 3$[2] with the stride of 2 to downsample the input and adjust the number of channels. Then, we apply the proposed SAM module to learn multi-scale contextual information. For the first two stages, as the input feature maps are of relatively high resolution, we merely employ one SAM module to avoid the heavy computational overhead. On the contrary, from the third stage to the fifth stage, we stack multiple SAM modules to enlarge the receptive fields and enrich deep convolutional representations. After the final fifth stage, we adopt the Pyramid Pooling Module (PPM) [59] to further enhance global feature learning. The default configuration for dilation rates and the number of branches in SAM modules is shown in Table I. Please refer to Table V for detailed ablation studies about different network configurations.

[2]For the first stage, we just use regular $\text{conv3} \times 3$.

TABLE I
BACKBONE SETTINGS OF THE PROPOSED SAMNET.

Stage	Resolution	Module	#M	#F	Stride	Dilation rates
1	224×224	$\text{conv3} \times 3$	1	16	2	-
	112×112	SAM	1	16	1	1,2,3
2	112×112	$\text{DSConv3} \times 3$	1	32	2	-
	56×56	SAM	1	32	1	1,2,3
3	56×56	$\text{DSConv3} \times 3$	1	64	2	-
	28×28	SAM	3	64	1	1,2,3
4	28×28	$\text{DSConv3} \times 3$	1	96	2	-
	14×14	SAM	6	96	1	1,2,3
5	14×14	$\text{DSConv3} \times 3$	1	128	2	-
	7×7	SAM	3	128	1	1,2
6	7×7	PPM	1	128	1	-

* "#M" means the number of modules whose types are specified in the column of "Module". "#F" means the number of convolution filters (*i.e.,* channels).

b) Encoder-Decoder Network: Based on the above backbone, we can build a lightweight encoder-decoder network, as illustrated in Fig. 2. Let $\{\boldsymbol{S}_i : i = 1, 2, \cdots, 6\}$ denote the output feature maps of each stage of the backbone. For the fusion of top features, we apply a single $\text{conv1} \times 1$ to $\boldsymbol{S}_5$ to adjust the number of channels and fuse $\boldsymbol{S}_5$ and $\boldsymbol{S}_6$ via element-wise summation. Then, we adopt a dilated $\text{DSConv}k \times k$ to further integrate the fused activation. Formally, we have

$$\boldsymbol{R}_5 = \mathcal{G}^{k\times k}_5(\mathcal{G}^{1\times 1}_5(\boldsymbol{S}_5) + \boldsymbol{S}_6), \tag{15}$$

where $\mathcal{G}^{k\times k}_5$ denotes the (depthwise separable) $k \times k$ convolution at the fifth stage, and $\boldsymbol{R}_5$ denotes the fused feature map at the fifth stage. Similarly, for the fusion of bottom features, we upsample the fused features from the top stages to match the spatial resolution of the feature maps at the bottom stages. In summary, we have

$$\boldsymbol{R}_i = \mathcal{G}^{k\times k}_i(\mathcal{G}^{1\times 1}_i(\boldsymbol{S}_i) + \text{Up}(\boldsymbol{R}_{i+1})),\quad i = 1, 2, 3, 4, \tag{16}$$

in which Up represents the upsampling operation with an upsampling rate of 2.

c) Deep Supervision & Loss Function: We employ *deep supervision* [61] to improve the transparency of the learning process for hidden layers. Concretely, for the fused features $\{\boldsymbol{R}_i,\ i = 1, 2, 3, 4, 5\}$, we sequentially apply a $\text{conv1} \times 1$ with a single output channel and the *sigmoid* activation function to derive several predictions $\{\boldsymbol{P}_i,\ i = 1, 2, 3, 4, 5\}$. We adopt the standard *binary cross-entropy loss* for training, which can be formulated as follows:

$$L = \mathcal{L}_{\text{BCE}}(\boldsymbol{P}_1, \boldsymbol{G}) + \lambda \sum_{i=2}^{5} \mathcal{L}_{\text{BCE}}(\boldsymbol{P}_i, \boldsymbol{G}), \tag{17}$$

where $\mathcal{L}_{\text{BCE}}$ is the standard *binary cross-entropy loss* function, and $\boldsymbol{G}$ denotes the ground-truth saliency map. λ denotes the weighting scalar for loss balance, and we follow [59] to empirically set λ to 0.4 in this paper.

IV. EXPERIMENTS

A. Experimental Setup

a) Implementation Details: The proposed method is implemented using the PyTorch [68] library. The training of

TABLE II
COMPARISON WITH EXISTING METHODS IN TERMS OF THE NUMBER OF PARAMETERS (#PARAM), GPU MEMORY USAGE, F_β, AND MAE.

Methods	#Param (M)	Memory (M)	ECSSD		DUT-OMRON		DUTS-TE		HKU-IS		SOD		THUR15K	
			$F_\beta\uparrow$	MAE$\downarrow$	$F_\beta\uparrow$	MAE$\downarrow$	$F_\beta\uparrow$	MAE$\downarrow$	$F_\beta\uparrow$	MAE$\downarrow$	$F_\beta\uparrow$	MAE$\downarrow$	$F_\beta\uparrow$	MAE$\downarrow$
DRFI [8]	-	-	0.777	0.161	0.652	0.138	0.649	0.154	0.774	0.146	0.704	0.217	0.670	0.150
DCL [62]	66.24	3737	0.895	0.080	0.733	0.095	0.785	0.082	0.892	0.063	0.831	0.131	0.747	0.096
DHSNet [24]	94.04	1899	0.903	0.062	-	-	0.807	0.066	0.889	0.053	0.822	0.128	0.752	0.082
RFCN [10]	134.69	4501	0.896	0.097	0.738	0.095	0.782	0.089	0.892	0.080	0.802	0.161	0.754	0.100
NLDF [12]	35.49	11707	0.902	0.066	0.753	0.080	0.806	0.065	0.902	0.048	0.837	0.123	0.762	0.080
DSS [30]	62.23	3209	0.915	0.056	0.774	0.066	0.827	0.056	0.913	0.041	0.842	0.122	0.770	0.074
Amulet [23]	33.15	3359	0.913	0.061	0.743	0.098	0.778	0.085	0.897	0.051	0.795	0.144	0.755	0.094
UCF [11]	23.98	5317	0.901	0.071	0.730	0.120	0.772	0.112	0.888	0.062	0.805	0.148	0.758	0.112
SRM [35]	43.74	1927	0.914	0.056	0.769	0.069	0.826	0.059	0.906	0.046	0.840	0.126	0.778	0.077
PiCANet [21]	32.85	1541	0.923	0.049	0.766	0.068	0.837	0.054	0.916	0.042	0.836	0.102	0.783	0.083
BRN [14]	126.35	5477	0.919	0.043	0.774	0.062	0.827	0.050	0.910	0.036	0.843	0.103	0.769	0.076
C2S [16]	137.03	1455	0.907	0.057	0.759	0.072	0.811	0.062	0.898	0.046	0.819	0.122	0.775	0.083
RAS [34]	20.13	2417	0.916	0.058	0.785	0.063	0.831	0.059	0.913	0.045	0.847	0.123	0.772	0.075
DNA [31]	20.06	2071	0.935	0.041	0.799	0.056	0.865	0.044	0.930	0.031	0.853	0.107	0.793	0.069
CPD [63]	29.23	761	0.930	0.044	0.794	0.057	0.861	0.043	0.924	0.033	0.848	0.113	0.795	0.068
BASNet [18]	87.06	1103	0.938	0.040	0.805	0.056	0.859	0.048	0.928	0.032	0.849	0.112	0.783	0.073
AFNet [27]	37.11	1123	0.930	0.045	0.784	0.057	0.857	0.046	0.921	0.036	0.848	0.108	0.791	0.072
PoolNet [28]	53.63	1087	0.934	0.048	0.791	0.057	0.866	0.043	0.925	0.037	0.863	0.111	0.800	0.068
EGNet [37]	108.07	1177	0.938	0.044	0.794	0.056	0.870	0.044	0.928	0.034	0.859	0.110	0.800	0.070
BANet [64]	55.90	3275	0.940	0.038	0.803	0.059	0.872	0.040	0.932	0.031	0.865	0.105	0.796	0.068
MobileNet [38]	4.27	633	0.906	0.064	0.753	0.073	0.804	0.066	0.895	0.052	0.809	0.136	0.767	0.081
MobileNetV2 [39]	2.37	609	0.905	0.066	0.758	0.075	0.798	0.070	0.890	0.056	0.801	0.138	0.766	0.085
ShuffleNet [40]	1.80	585	0.907	0.062	0.757	0.069	0.811	0.062	0.898	0.050	0.816	0.130	0.771	0.078
ShuffleNetV2 [41]	1.60	579	0.901	0.069	0.746	0.076	0.789	0.071	0.884	0.059	0.789	0.147	0.755	0.086
ICNet [65]	6.70	633	0.918	0.059	0.773	0.072	0.810	0.067	0.898	0.052	0.802	0.134	0.768	0.084
BiSeNet R18 [66]	13.48	719	0.909	0.062	0.757	0.072	0.815	0.062	0.902	0.049	0.821	0.128	0.776	0.080
BiSeNet X39 [66]	1.84	665	0.901	0.070	0.755	0.078	0.787	0.074	0.888	0.059	0.792	0.147	0.756	0.090
DFANet [67]	1.83	749	0.896	0.073	0.750	0.078	0.791	0.075	0.884	0.061	0.802	0.148	0.757	0.089
SAMNet (OURS)	1.33	599	0.925	0.053	0.797	0.065	0.835	0.058	0.915	0.045	0.833	0.123	0.785	0.077

all experiments is conducted using the Adam optimizer [69] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay of 10^{-4}, and batch size of 20. Our model is pretrained on the ImageNet dataset [70] as in [43]. We adopt *poly* learning rate scheduler so that the learning rate for the n^{th} epoch is $\textit{init\_lr} \times \left(1 - \frac{n}{\#\textit{epochs}}\right)^{\textit{power}}$, where $\textit{init\_lr} = 5 \times 10^{-4}$ and $\textit{power} = 0.9$. We train the proposed model for 50 epochs, *i.e.,* $\#\textit{epochs} = 50$.

b) Datasets: We extensively evaluate the proposed method on six datasets, including DUTS [71], ECSSD [72], SOD [73], HKU-IS [47], THUR15K [74], and DUT-OMRON [45] datasets. These six datasets consist of 15572, 1000, 300, 4447, 6232, and 5168 natural images with corresponding pixel-level labels, respectively. Following recent studies [14], [21], [31], [35], [75], we train the proposed model on the DUTS training set and evaluate it on the DUTS testing set (DUTS-TE) and other five datasets.

c) Evaluation Criteria: We evaluate the accuracy of SAMNet against previous state-of-the-art methods with regard to four widely-used metrics, *i.e.,* F_β-measure score (F_β), mean absolute error (MAE), weighted F_β^ω-measure (F_β^ω) [76], and structure similarity measure (S_β) [77]. Given a threshold in the range of $[0, 1)$, we can binarize the predicted saliency probability map and then calculate the precision and recall values by comparing the binarized prediction map with the binary ground-truth map. With precision and recall calculated, F_β-measure is the weighted harmonic mean of precision and recall, *i.e.,*

$$F_\beta = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}, \tag{18}$$

where we set $\beta^2 = 0.3$ to emphasize the importance of precision, as in previous works [21], [23], [28], [30], [31]. Note that each threshold will correspond to a F_β score here, and we report the maximum F_β score across all thresholds. MAE measures the difference between the predicted saliency map $\boldsymbol{P}$ and the ground-truth saliency map $\boldsymbol{G}$, which can be computed as

$$\text{MAE}(\boldsymbol{P}, \boldsymbol{G}) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} |\boldsymbol{P}_{ij} - \boldsymbol{G}_{ij}|, \tag{19}$$

where H and W denote the height and width of the saliency map, respectively. The weighted F_β^ω-measure [76] is designed to amend the interpolation flaw, the dependency flaw, and the equal-importance flaw in traditional evaluation metrics, and we use it with default settings to evaluate SAMNet and other competitors. As for S_β [77], it is also a widely-used saliency evaluation metric as it can measure the structure similarity between predictions and ground truths. S_β consists of region-aware and object-aware structural similarity measures, in which the former is achieved using well-known SSIM [78], and the latter is based on the probability theory. We adopt the official code with default settings in our experiments.

d) Efficiency Measures: This paper targets a lightweight yet powerful solution for SOD, so we also evaluate the efficiency and flexibility of various methods, including the number of model parameters (#Param), GPU memory usage, the number of floating-point operations (FLOPs), and the inference speed (FPS). GPU memory usage measures the amount of memory required for a model to test an image. The number of FLOPs measures the computational cost of a

TABLE III
COMPARISON WITH EXISTING METHODS IN TERMS OF THE NUMBER OF FLOPS, SPEED, F_β^ω, AND S_β.

Methods	FLOPs (G)	Speed (FPS)	ECSSD		DUT-OMRON		DUTS-TE		HKU-IS		SOD		THUR15K	
			$F_\beta^\omega \uparrow$	$S_\beta \uparrow$	$F_\beta^\omega \uparrow$	$S_\beta \uparrow$	$F_\beta^\omega \uparrow$	$S_\beta \uparrow$	$F_\beta^\omega \uparrow$	$S_\beta \uparrow$	$F_\beta^\omega \uparrow$	$S_\beta \uparrow$	$F_\beta^\omega \uparrow$	$S_\beta \uparrow$
DRFI [8]	-	0.1	0.548	0.727	0.424	0.697	0.378	0.676	0.504	0.739	0.450	0.616	0.444	0.711
DCL [62]	224.9	1.4	0.782	0.869	0.584	0.762	0.632	0.803	0.770	0.871	0.669	0.756	0.624	0.794
DHSNet [24]	15.8	10.0	0.837	0.880	-	-	0.705	0.820	0.816	0.870	0.685	0.746	0.666	0.802
RFCN [10]	102.8	0.4	0.725	0.856	0.562	0.774	0.586	0.793	0.707	0.858	0.591	0.716	0.591	0.793
NLDF [12]	263.9	18.5	0.835	0.870	0.634	0.770	0.710	0.816	0.838	0.879	0.708	0.753	0.676	0.801
DSS [30]	114.6	7.0	0.864	0.879	0.688	0.790	0.752	0.826	0.862	0.881	0.711	0.746	0.702	0.805
Amulet [23]	45.3	9.7	0.839	0.891	0.626	0.781	0.657	0.804	0.817	0.886	0.674	0.750	0.650	0.796
UCF [11]	61.4	12.0	0.805	0.881	0.573	0.760	0.595	0.782	0.779	0.875	0.673	0.759	0.613	0.785
SRM [35]	20.3	12.3	0.849	0.890	0.658	0.798	0.721	0.836	0.835	0.887	0.670	0.739	0.684	0.818
PiCANet [21]	37.1	5.6	0.862	0.909	0.691	0.826	0.745	0.860	0.847	0.905	0.721	0.787	0.687	0.823
BRN [14]	24.1	3.6	0.887	0.898	0.709	0.806	0.774	0.842	0.875	0.894	0.738	0.768	0.712	0.813
C2S [16]	20.5	16.7	0.849	0.891	0.663	0.799	0.717	0.831	0.835	0.889	0.699	0.757	0.685	0.812
RAS [34]	35.6	20.4	0.855	0.889	0.695	0.812	0.739	0.838	0.849	0.889	0.718	0.761	0.691	0.813
DNA [31]	82.5	25.0	0.897	0.909	0.729	0.823	0.797	0.863	0.889	0.908	0.755	0.780	0.723	0.824
CPD [63]	59.5	68.0	0.889	0.905	0.715	0.818	0.799	0.866	0.879	0.904	0.718	0.765	0.731	0.831
BASNet [18]	127.3	36.2	0.898	0.910	0.751	0.836	0.802	0.865	0.889	0.909	0.728	0.766	0.721	0.823
AFNet [27]	38.4	21.6	0.880	0.907	0.717	0.826	0.784	0.867	0.869	0.905	0.726	0.773	0.719	0.829
PoolNet [28]	123.4	39.7	0.875	0.909	0.710	0.829	0.783	0.875	0.864	0.908	0.731	0.781	0.724	0.839
EGNet [37]	270.8	12.7	0.886	0.913	0.727	0.836	0.796	0.878	0.876	0.912	0.736	0.781	0.727	0.836
BANet [64]	121.6	12.5	0.901	0.918	0.736	0.832	0.810	0.878	0.889	0.915	0.765	0.788	0.730	0.834
MobileNet [38]	2.2	295.8	0.829	0.884	0.656	0.802	0.696	0.828	0.816	0.884	0.653	0.735	0.675	0.814
MobileNetV2 [39]	0.8	446.2	0.820	0.885	0.651	0.806	0.676	0.823	0.799	0.879	0.657	0.742	0.660	0.811
ShuffleNet [40]	0.7	406.9	0.831	0.884	0.667	0.808	0.709	0.834	0.820	0.885	0.670	0.743	0.683	0.819
ShuffleNetV2 [41]	0.5	452.5	0.812	0.878	0.637	0.797	0.665	0.816	0.788	0.871	0.621	0.715	0.652	0.806
ICNet [65]	6.3	75.1	0.838	0.895	0.669	0.813	0.694	0.830	0.812	0.885	0.663	0.743	0.668	0.812
BiSeNet R18 [66]	25.0	120.5	0.829	0.886	0.648	0.803	0.699	0.835	0.819	0.889	0.669	0.751	0.675	0.818
BiSeNet X39 [66]	7.3	165.8	0.802	0.877	0.632	0.799	0.652	0.813	0.784	0.875	0.620	0.720	0.641	0.802
DFANet [67]	1.7	91.4	0.799	0.872	0.627	0.794	0.652	0.811	0.778	0.868	0.617	0.718	0.639	0.802
SAMNet (OURS)	0.5	343.2	0.855	0.902	0.699	0.830	0.729	0.849	0.837	0.898	0.686	0.756	0.693	0.825

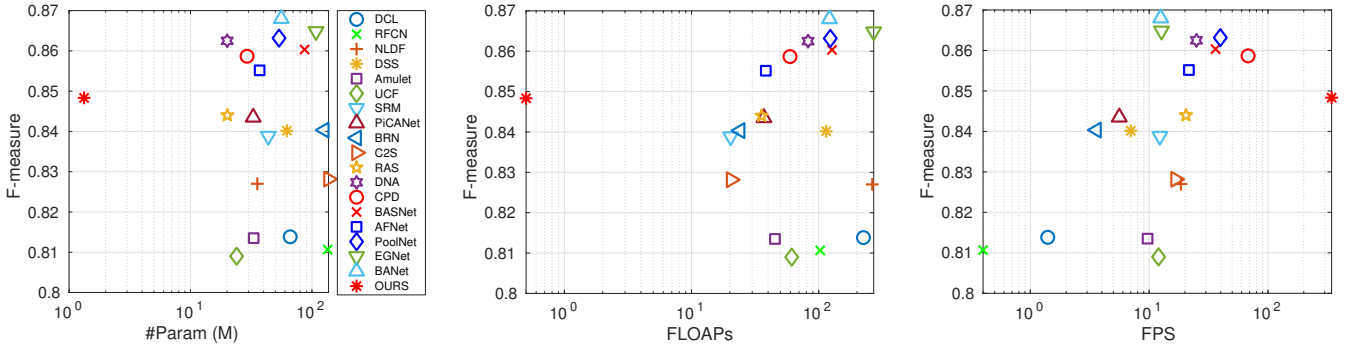

Fig. 3. Illustration of the trade-off between performance and computational cost. The F-measure (F_β) is averaged over six datasets. Note that the horizon axis is logarithmic.

model, and a smaller number of FLOPs leads to lower energy consumption. Following lightweight backbone networks [38]–[41] and efficient semantic segmentation [65]–[67], the speed stands for the number of images that a model can do network inference per second, tested on a single NVIDIA TITAN Xp GPU. Since deep supervision is only used in training, we omit the calculation of $\{\boldsymbol{P}_i, i = 2, 3, 4, 5\}$ in the speed testing of SAMNet. For saliency detectors, memory usage, FLOPs, and speed are tested using a 336×336 input image except that a method specifies its input dimensions. Since efficient semantic segmentation methods [65]–[67] are usually designed for high-resolution images, we use a 672×672 input to ensure accuracy; otherwise, we would get very low accuracy. For reformed baselines based on lightweight backbones [38]–[41], we use a 336×336 input. Here, #Param and GPU memory usage are measured in million (M), and the number of FLOPs is measured in giga (G).

B. Performance Analysis

In this part, we compare the proposed SAMNet with 20 state-of-the-art SOD methods, including DRFI [8], DCL [62], DHSNet [24], RFCN [10], NLDF [12], DSS [30], Amulet [23], UCF [11], SRM [35], PiCANet [21], BRN [14], C2S [16], RAS [34], DNA [31], CPD [63], BASNet [18], AFNet [27], PoolNet [28], EGNet [37], and BANet [64]. Other than existing SOD methods, we also compare with several state-of-the-art lightweight backbone networks that are widely used for image classification, including MobileNet [38], MobileNetV2 [39], ShuffleNet [40], and ShuffleNetV2 [41]. For adapting them to the SOD task, we add the same decoder in the proposed SAMNet to them. The resulting baselines are trained with the same settings for comparison. Besides, we compare with some efficient semantic segmentation methods as well, including ICNet [65], BiSeNet [66], and DFANet [67]. We reform them for SOD by replacing their final *softmax* acti-

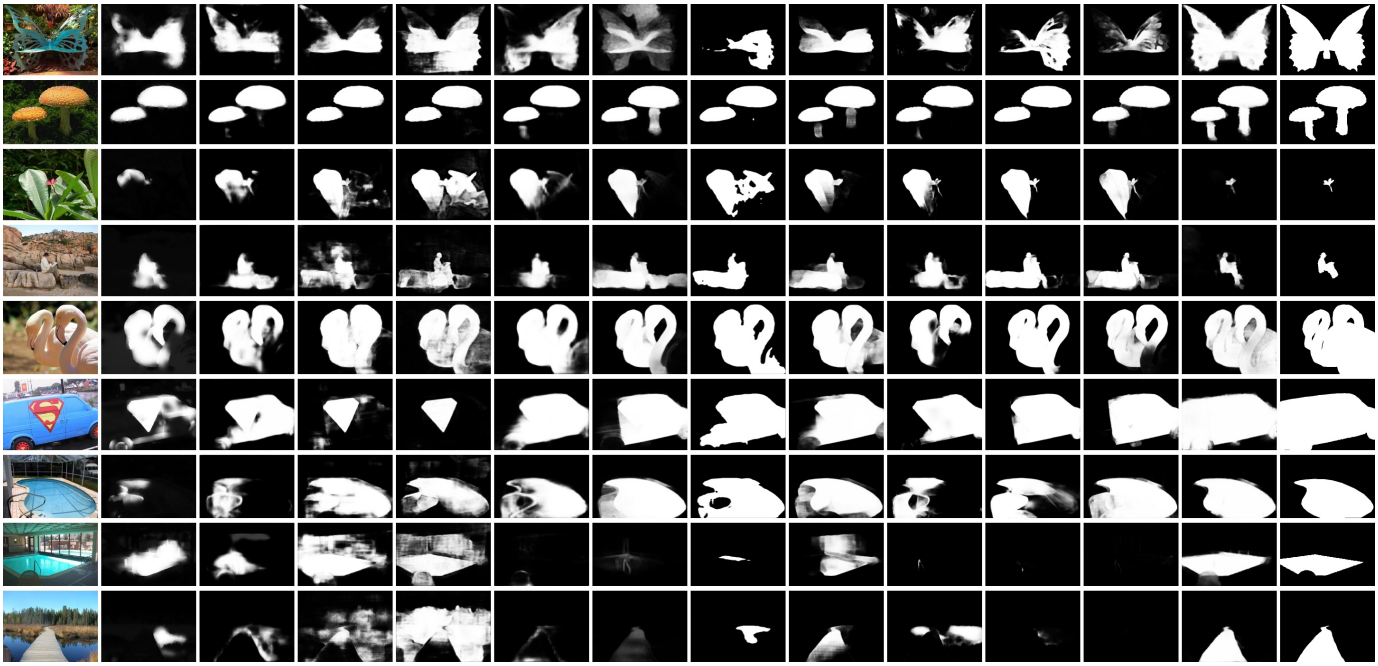

Fig. 4. Qualitative comparison with state-of-the-art SOD methods.

vation function with the standard *sigmoid* activation function. For BiSeNet [66], we report its results with ResNet-18 [43] (*i.e.,* BiSeNet R18) and Xception-39 [79] (*i.e.,* BiSeNet X39) backbones, respectively.

a) Comparison with Existing SOD Methods: Table II shows the evaluation results of the proposed SAMNet compared with previous state-of-the-art alternatives in terms of the number of parameters, GPU memory usage, F_β, and MAE. Table III shows the evaluation results in terms of the number of FLOPs, speed, F_β^ω, and S_β. Note that the number of FLOPs is highly related to energy consumption for the network inference. The results clearly show that SAMNet achieves on par accuracy with state-of-the-art SOD solutions, especially in terms of F_β, MAE, and S_β. However, SAMNet requires one or two orders of magnitude fewer computational resources. For example, compared with the best performing BANet [64], SAMNet shows a slightly lower average F_β (0.848 vs. 0.868) over six datasets, but SAMNet has 42$\times$ fewer parameters, 5.5$\times$ less GPU memory consumption, 243$\times$ fewer FLOPs, 27$\times$ faster running speed than BANet [64]. This has significant impacts on mobile devices, where the limited computational resource, reduced energy supply, restrictive running memory and storage space cannot afford the heavy overhead of traditional cumbersome SOD methods.

We also illustrate the comparison in Fig. 3, where the trade-offs between the accuracy and efficiency of various methods are more clearly shown. In the sub-figures of F_β *vs.* #Param and F_β *vs.* FLOPs, SAMNet lies at the top-left corner; in the sub-figure of F_β *vs.* FPS, SAMNet lies at the top-right corner. This implies that SAMNet achieves comparable accuracy to previous state-of-the-art methods with much fewer parameters and FLOPs, and much faster speed. Therefore, we can come to the conclusion that SAMNet achieves a good trade-off among accuracy, the number of parameters, GPU memory usage, the number of FLOPs, and speed.

b) Comparison with Lightweight Backbones: Table II and Table III also present the comparison of SAMNet to lightweight backbone based baselines, *i.e.,* MobileNet [38], MobileNetV2 [39], ShuffleNet [40], and ShuffleNetV2 [41]. Although these baselines have similar or even faster speeds than SAMNet, SAMNet achieves substantially better accuracy in terms of all metrics. This suggests that it is suboptimal to apply existing lightweight backbones for SOD directly. This also demonstrates the importance of carefully designing network architectures for the lightweight SOD task and the advantage of the proposed multi-scale attention mechanism.

c) Comparison with Efficient Semantic Segmentation Methods: From Table II and Table III, we can see that SAMNet outperforms efficient semantic segmentation methods [65]–[67] by a large margin, with fewer parameters, fewer FLOPs, and faster speed. It is interesting to find that efficient semantic segmentation methods perform worse than baselines based on lightweight backbones. This implies that efficient semantic segmentation methods are heavily tuned for semantic segmentation and are thus unsuitable for directly applying to SOD. Hence, lightweight SOD is an essential problem and should get more attention from this community.

d) Qualitative comparison: In Fig. 4, we provide some visualization examples to exhibit the superiority of SAMNet. Although SAMNet performs slightly worse than traditional cumbersome SOD methods, it can segment salient objects with coherent boundaries in many challenging circumstances, such as complicated scenarios (1^{st} and 3^{rd} rows), low contrast between foreground and background (2^{nd} and 4^{th} rows), large objects (5^{th} and 6^{th} rows), scenarios with abnormal brightness (7^{th} and 8^{th} rows), and confusing natural scenarios

TABLE IV
ABLATION STUDY FOR THE DESIGN CHOICES OF SAMNET.

No.	Component					ECSSD		DUT-OMRON		DUTS-TE		HKU-IS		SOD		THUR15K	
	MB	CA	SA	PP	IP	$F_\beta\uparrow$	MAE $\downarrow$	$F_\beta\uparrow$	MAE $\downarrow$	$F_\beta\uparrow$	MAE $\downarrow$	$F_\beta\uparrow$	MAE $\downarrow$	$F_\beta\uparrow$	MAE $\downarrow$	$F_\beta\uparrow$	MAE $\downarrow$
0						0.891	0.075	0.750	0.079	0.779	0.077	0.877	0.062	0.791	0.149	0.749	0.089
1	✓					0.904	0.068	0.767	0.076	0.792	0.074	0.888	0.058	0.797	0.142	0.757	0.090
2	✓	✓				0.903	0.066	0.773	0.074	0.795	0.071	0.892	0.056	0.803	0.138	0.759	0.086
3	✓		✓			0.902	0.065	0.769	0.072	0.797	0.071	0.889	0.056	0.807	0.138	0.758	0.087
4	✓	✓	✓			0.905	0.063	0.766	0.072	0.802	0.068	0.892	0.055	0.795	0.135	0.761	0.085
5	✓	✓	✓	✓		0.909	0.060	0.769	0.072	0.802	0.069	0.894	0.053	0.810	0.134	0.761	0.086
6	✓	✓	✓	✓	✓	**0.925**	**0.053**	**0.797**	**0.065**	**0.835**	**0.058**	**0.915**	**0.045**	**0.833**	**0.123**	**0.785**	**0.077**

* We use the vanilla single branch module as the baseline (No. 0). Here, "MB", "CA", "SA", "PP", and "IP" refer to the simple multi-branch module ($\boldsymbol{F}$), channel-wise attention ($\boldsymbol{F}^D$), spatial attention ($\boldsymbol{F}^S$), pyramid pooling, and ImageNet [70] pretraining, respectively. When incorporating both channel attention and spatial attention, we reach the stereoscopic attention mechanism ($\boldsymbol{F}^V$).

TABLE V
ABLATION STUDY FOR THE CONFIGURATIONS OF SAMNET.

	Stage	Configuration	ECSSD		DUT-OMRON		DUTS-TE		HKU-IS		SOD		THUR15K	
			$F_\beta\uparrow$	MAE $\downarrow$	$F_\beta\uparrow$	MAE $\downarrow$	$F_\beta\uparrow$	MAE $\downarrow$	$F_\beta\uparrow$	MAE $\downarrow$	$F_\beta\uparrow$	MAE $\downarrow$	$F_\beta\uparrow$	MAE $\downarrow$
Default Configuration			0.909	0.060	0.769	0.072	0.802	0.069	0.894	0.053	0.810	0.134	0.761	0.086
Dilation rates	$1-4$	$1,2$	0.907	0.063	0.769	0.071	0.802	0.068	0.893	0.054	0.803	0.136	0.761	0.085
	$1-4$	$1,2,3,4$	0.908	0.063	0.777	0.071	0.804	0.069	0.893	0.055	0.792	0.145	0.765	0.083
	$1-4$	$1,2,4,8$	0.905	0.064	0.777	0.070	0.807	0.068	0.892	0.055	0.803	0.141	0.765	0.084
	5	$1,2,3$	0.904	0.064	0.772	0.074	0.800	0.072	0.890	0.056	0.800	0.144	0.765	0.086
	5	$1,2,3,4$	0.904	0.065	0.772	0.073	0.798	0.072	0.893	0.054	0.801	0.139	0.762	0.084
#Modules	3	2	0.907	0.063	0.770	0.070	0.798	0.068	0.890	0.055	0.798	0.142	0.759	0.086
	4	5	0.908	0.063	0.774	0.071	0.801	0.068	0.894	0.054	0.808	0.133	0.763	0.084
	4	7	0.910	0.062	0.767	0.073	0.801	0.069	0.894	0.053	0.812	0.138	0.760	0.086
	5	2	0.910	0.061	0.770	0.071	0.798	0.069	0.894	0.053	0.804	0.136	0.761	0.084
	5	4	0.903	0.064	0.776	0.070	0.803	0.067	0.893	0.054	0.790	0.139	0.763	0.083
#Filters	ALL	$\times 0.75$	0.899	0.069	0.765	0.074	0.784	0.074	0.883	0.059	0.786	0.144	0.753	0.089
	ALL	$\times 1.25$	0.911	0.061	0.779	0.070	0.809	0.067	0.897	0.053	0.808	0.132	0.765	0.084

* "#Modules" represents the number of SAM modules in each stage. "#Filters" denotes the number of convolution filters (*i.e.,* channel), and "$\times k$" means that we multiply the number of filters in SAMNet by a factor of k. Note that all experiments here are trained from scratch.

(9^{th} row). Combined with the lightweight and efficient nature of SAMNet, it has the potential to promote real-world SOD applications.

C. Ablation Study

In this section, we conduct ablation study to demonstrate the effectiveness of the proposed module components and the parameter configurations of SAMNet. The experimental settings follow those in Section IV-B.

a) Proposed Module Components: Table IV shows the ablation study results for the proposed module components. The proposed SAM module is designed by carefully combining these basic components into a nontrivial module for effective and efficient multi-scale learning. Table IV suggests that the performance is gradually boosted by adding each component into the framework. Besides, the comparison between No. 0 and No. 5 demonstrates the superiority of the proposed solution compared with the baseline, where the performance gap comes solely from our contributions because two models are both trained from scratch without the ImageNet [70] pretraining.

b) Configurations of SAMNet: Table V shows the ablation study results for different network configurations. It is interesting to find that the proposed SAMNet is robust to the slight changes in configurations. Introducing more parameters will lead to better performance, such as increasing the number of convolution filters, but this is orthogonal to our goal of lightweight SOD. Our default setting of SAMNet is set by considering the trade-off between effectiveness and lightweight restriction.

V. CONCLUSION

Instead of only considering accuracy, this paper focuses on lightweight SOD that trades off among accuracy, efficiency, parameter numbers, and FLOPs. We propose a novel SAM module, which enables small networks to effectively encode both high-level features and low-level details. Incorporating the SAM module, the proposed SAMNet achieves comparable performances with state-of-the-art SOD methods that use significantly more parameters, while saving several orders of magnitude overhead. Such excellent trade-off between performance and efficiency makes SAMNet possible to provide high-accuracy SOD in resource-limited environments, *e.g.,* mobile devices. SAMNet also clearly outperforms other well-known lightweight networks in image classification [38]–[41] and semantic segmentation [65]–[67], suggesting that lightweight SOD is worth studying and should be set up as a separate research field. To the best of our knowledge, SAMNet is the first lightweight SOD method that is expected to pave a new path for SOD. Through this work, we want to arouse the research for lightweight SOD that would promote more practical SOD applications. In the future, we plan to apply the weight quantization and network compression techniques to boost SAMNet's CPU speed of 5fps for real-time performance.

ACKNOWLEDGMENT

This research was supported by Major Project for New Generation of AI under Grant No. 2018AAA0100400, NSFC (61922046), S&T innovation project from Chinese Ministry of Education, and Tianjin Natural Science Foundation (17JCJQJC43700).

REFERENCES

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

[2] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu, "Visual-textual joint relevance learning for tag-based social image search," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 363–376, 2012.

[3] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof, "Saliency driven total variation segmentation," in *Int. Conf. Comput. Vis.*, 2009, pp. 817–824.

[4] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. 37–44.

[5] V. Mahadevan and N. Vasconcelos, "Saliency-based discriminant tracking," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1007–1013.

[6] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang, "Region-based saliency detection and its application in object recognition," *IEEE Trans. Circ. Syst. Video Technol.*, vol. 24, no. 5, pp. 769–779, 2013.

[7] M.-M. Cheng, F.-L. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "RepFinder: Finding approximately repeated scene elements for image editing," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 83:1–83:8, 2010.

[8] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 2083–2090.

[9] D. A. Klein and S. Frintrop, "Center-surround divergence of feature statistics for salient object detection," in *Int. Conf. Comput. Vis.*, 2011, pp. 2214–2219.

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

[11] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Int. Conf. Comput. Vis.*, 2017, pp. 212–221.

[12] Z. Luo, A. K. Mishra, A. Achkar, J. A. Eichel, S. Li, and P.-M. Jodoin, "Non-local deep features for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 6609–6617.

[13] P. Hu, B. Shuai, J. Liu, and G. Wang, "Deep level sets for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2300–2309.

[14] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji, "Detect globally, refine locally: A novel approach to saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3127–3135.

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

[16] X. Li, F. Yang, H. Cheng, W. Liu, and D. Shen, "Contour knowledge transfer for salient object detection," in *Eur. Conf. Comput. Vis.*, 2018, pp. 355–370.

[17] N. D. Bruce, C. Catton, and S. Janjic, "A deeper look at saliency: Feature contrast, semantics, and beyond," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 516–524.

[18] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 7479–7489.

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

[20] W. Wang, J. Shen, X. Dong, and A. Borji, "Salient object detection driven by fixation prediction," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1711–1720.

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

[22] M. A. Islam, M. Kalash, and N. D. Bruce, "Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7142–7150.

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

[24] N. Liu and J. Han, "DHSNet: Deep hierarchical saliency network for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 678–686.

[25] S. He, J. Jiao, X. Zhang, G. Han, and R. W. Lau, "Delving into salient object subitizing and detection," in *Int. Conf. Comput. Vis.*, 2017, pp. 1059–1067.

[26] W. Wang, S. Zhao, J. Shen, S. C. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 1448–1457.

[27] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 1623–1632.

[28] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3917–3926.

[29] R. Wu, M. Feng, W. Guan, D. Wang, H. Lu, and E. Ding, "A mutual learning method for salient object detection with intertwined multi-supervision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 8150–8159.

[30] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, 2019.

[31] Y. Liu, M.-M. Cheng, X.-Y. Zhang, G.-Y. Nie, and M. Wang, "DNA: Deeply-supervised nonlinear aggregation for salient object detection," *IEEE Trans. Cybernetics*, 2021.

[32] Y. Liu, Y.-C. Gu, X.-Y. Zhang, W. Wang, and M.-M. Cheng, "Lightweight salient object detection via hierarchical visual perception learning," *IEEE Trans. Cybernetics*, 2020.

[33] Y. Qiu, Y. Liu, H. Yang, and J. Xu, "A simple saliency detection approach via automatic top-down feature fusion," *Neurocomputing*, vol. 388, pp. 124–134, 2020.

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

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

[36] Y. Qiu, Y. Liu, X. Ma, L. Liu, H. Gao, and J. Xu, "Revisiting multi-level feature fusion: A simple yet effective network for salient object detection," in *IEEE Int. Conf. Image Process.*, 2019, pp. 4010–4014.

[37] J.-X. Zhao, J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "EGNet: Edge guidance network for salient object detection," in *Int. Conf. Comput. Vis.*, 2019, pp. 8779–8788.

[38] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint*, 2017.

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4510–4520.

[40] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 6848–6856.

[41] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet v2: Practical guidelines for efficient CNN architecture design," in *Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2015.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[44] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.

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

[46] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *Int. Conf. Comput. Vis.*, 2013, pp. 2976–2983.

[47] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5455–5463.

[48] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1265–1274.

[49] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3183–3192.
[50] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3431–3440.
[51] R. A. Rensink, "The dynamic representation of scenes," *Visual Cognition*, vol. 7, no. 1-3, pp. 17–42, 2000.
[52] M. Corbetta and G. L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature Reviews Neuroscience*, vol. 3, no. 3, pp. 201–215, 2002.
[53] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," *arXiv preprint*, 2017.
[54] D. Chen, S. Zhang, W. Ouyang, J. Yang, and Y. Tai, "Person search via a mask-guided two-stream CNN model," in *Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.
[55] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Eur. Conf. Comput. Vis.*, 2018, pp. 286–301.
[56] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 3156–3164.
[57] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7132–7141.
[58] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.
[59] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2881–2890.
[60] X. Chen, A. Zheng, J. Li, and F. Lu, "Look, perceive and segment: Finding the salient objects in images via two-stream fixation-semantic CNNs," in *Int. Conf. Comput. Vis.*, 2017, pp. 1050–1058.
[61] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artif. Intell. Stat.*, 2015, pp. 562–570.
[62] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 478–487.
[63] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3907–3916.
[64] J. Su, J. Li, Y. Zhang, C. Xia, and Y. Tian, "Selectivity or invariance: Boundary-aware salient object detection," in *Int. Conf. Comput. Vis.*, 2019, pp. 3799–3808.
[65] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Eur. Conf. Comput. Vis.*, 2018, pp. 405–420.
[66] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *Eur. Conf. Comput. Vis.*, 2018, pp. 325–341.
[67] H. Li, P. Xiong, H. Fan, and J. Sun, "DFANet: Deep feature aggregation for real-time semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 9522–9531.
[68] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inform. Process. Syst.*, 2019, pp. 8026–8037.
[69] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learn. Represent.*, 2015.
[70] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.
[71] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 136–145.
[72] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 1155–1162.
[73] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2010, pp. 49–56.
[74] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu, "SalientShape: Group saliency in image collections," *The Vis. Comput.*, vol. 30, no. 4, pp. 443–453, 2014.
[75] Y. Zeng, H. Lu, L. Zhang, M. Feng, and A. Borji, "Learning to promote saliency detectors," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1644–1653.
[76] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps?" in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 248–255.
[77] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Int. Conf. Comput. Vis.*, 2017, pp. 4548–4557.
[78] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.
[79] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1251–1258.