SAMNet: 用于轻量级显著性物体检测的立体注意多尺度网络

刘云 *, 张鑫禹 *, 边佳旺, 张乐, 程明明

摘要—近来在显著性物体检测 (SOD) 任务上的进展大多得益于卷积神经网络 (CNNs) 的爆炸式发展。但是，这些改进大多会带来更大的网络体积和更重的计算负荷，这在我们看来是移动不友好的，因而会更难在实际中应用这些技术。为了推进更加实用的 SOD 系统，我们提出了一种新颖的立体注意多尺度 (SAM) 模块，它能够利用一种立体注意力的机制来自适应地融合不同尺度的特征。利用这个模块，我们提出了一种极度轻量级的网络，SAMNet ，用于 SOD。在许多流行的数据集上的实验表明了我们提出的 SAMNet 能够在在仅仅使用 1.33M 的参数量的情况下，达到和现在的一流模型 相当的精度，同时对于 336×336 尺寸的输入能够达到 343fps 的 GPU 处理速度和 5fps 的 CPU 处理速度。因此，SAMNet 为 SOD 铺就了一条新的道路。源代码可以在项目页面 https://mmcheng.net/SAMNet/ 上获取。

***Index Terms*—轻量级显著性物体检测，轻量级显著性检测，多尺度学习。**

I. 引言

显著性物体检测，又名显著性检测，的目标是检测出自然图片中视觉上最为特别的物体或者区域 [1]。SOD 的进展能有益于许多计算机视觉的应用，包括图像检索 [2]，图像分割 [3]，目标检测 [4]，视觉跟踪 [5]，场景分类 [6]，内容感知图像编辑 [7] 等。这项任务的传统方法主要依赖于人工设计的低层次特征和启发式的先验假设 [1], [8], [9]，但是高层次语义信息的缺失通常会导致精度的限制。近来，由于卷积神经网络 (CNNs)、尤其是全卷积神经网络 (FCNs) 前所未有的成功，基于深度学习的方法刷新了 SOD 领域一流方法的表现 [10]–[33]。

但是，这些改进并非是无条件的：它们通常依赖于更大的网络体积和庞大的计算负荷 [14], [16], [21], [30], [31], [33]–[36]。举例而言，使用 VGG16 作为骨干网络的 EGNet [37] 有着 108M 的参数量并且需要 ~432 MB 的硬盘空间去储存它的预训练模型。除此之外，EGNet [37] 对于 336×335 尺寸的图片，在强劲的 i7-8700K CPU 上只有 0.09fps 的运行速度，同时在 NVIDIA TITAN XP GPU 上只有 12.7fps 的运行速度。这样的笨重无疑会使得它在如自动驾驶、机器人、增强现实等实时且资源受限的应用场景里具有更低的使用价值。在这些场景下，移动设备会有低下的计算性能、受限的内存限制以及被局限的功率负荷。

设计轻量级的 CNNs 无疑是上述问题的一个解决方案，它也正在诸如图像分类等领域 [38]–[41] 被研究。尽管我们从它们之中得到了灵感，但我们仍旧是首先在 SOD 领域着手相关工作的。这一工作是非平凡的，原因在于它所面临的以下几项挑战：i) SOD 同时需要高层次的抽象语义特征和低层次的细粒度特征来分别定位显著物体和其细化的物体细节。ii) SOD 需要多尺度的信息来处理自然场景下不同大小和横纵比的显著性物体。由于轻量级网络通常会有较浅的深度和简化的操作，它在多层和多尺度学习上和传统的大型网络相比效果较差 [42], [43]。因此，单纯将现有的轻量级骨架网络如 MobileNets [38], [39] 和 ShuffleNets [40], [41] 用于 SOD 会导致不理想的表现，我们将在实验中阐明这一点。

众所周知，CNNs 可以在其高层学习高层次的语义信息，而在其底层学习低层次的细节。这让 CNNs 的不同层输出包含了多尺度的信息。因此，为了学习多层和多尺度的信息，当下的一流 SOD 方法 (大网络) 使用了编码器-解码器的神经网络结构 [10]–[29] 来整合骨干网络多层的输出特征。最近 SOD 领域的进展主要来自于能有效的融合多层骨干网络的特征的新策略和新模块。

基于上述的分析，轻量级 SOD 的关键部分在于如何在有限的参数量下有效地学习多层的和多尺度的信息。不同于融合骨干网络中不同层的输出 [14], [19], [21], [23], [28], [30], [37] 或者是综合不同膨胀率的卷积特征 [15], [44] 这些在之前的研究中已经尝试过的方法，我们提出了一种新颖的 立体注意多尺度 (SAM) 模块来用于多尺度学习。它采用了一种立体注意力机制来自动地控制不同尺度的学习，所以它具有有效地学习 CNNs 不同层的必要信息的能力。在使用 SAM 模块作为基本模块的情况下，我们搭建了一个轻量级的编码器-解码器网络，也就是 SAMNet，来统合

Y. Liu 和 M.M. Cheng 就职于南开大学计算机学院。X.Y. Zhang 就职于南开大学数学学院。J.W. Bian 就职于澳大利亚阿德莱德大学计算机学院。L. Zhang 就职于新加坡科技研究局 (A*STAR)。

两位第一作者在本文中有同等的贡献。M.M. Cheng 是通讯作者 (cmm@nankai.edu.cn).

SAM 模块所学到的多层和多尺度的特征。SAMNet 能够与现在的一流 SOD 方法达到相当的精度，同时对于 336×336 尺寸的图片，能够在 i7-8700K CPU 上达到 5fps 的处理速度以及在 NVIDIA TITAN XP GPU 上达到 343fps 的处理速度。除此之外，SAMNet 只有 1.33M 的参数量。这些轻量级的特征使得它在实际移动端上的应用成为了可能。

总结来说，我们的贡献主要有三点：

- 我们提出了一种新颖的**立体注意多尺度（SAM）**模块。它使用了一种立体注意力的机制来有效并且高效率地进行多尺度学习。
- 通过使用 SAM 模块作为基本单元，我们提出了 SAM-Net，一个轻量级的编码器-解码器结构的用于 SOD 的网络。这也是在我们所知范围内的第一个轻量级 SOD 网络。
- 我们实际评估了 SAMNet 在六个流行的 SOD 数据集上的效果并且展现了其有竞争力的精度水平，更高的效率以及更小的网络体积。

II. 相关工作

a) 显著性物体检测： 在过去的二十年里，为了检测图像中的显著物体，大量的方法被提出。传统的方法主要基于人工设计的特征，比如图像对比度 [1], 纹理 [45], 中心先验 [8], 背景先验 [46] 等。尽管这些上述的方法效率很高，但人工设计的特征在本质上欠缺高层表征的能力，导致了性能受限。

随着深度学习的飞速发展，基于 CNNs 的 SOD 方法已经很大地超越了传统的方法。早期基于 CNNs 的方法 [24], [47]–[49] 包括了一些全连接层，导致了整幅图像中的关键区域信息的缺失。自从影响深远的 [50] 提出了将 FCN 用于预测逐像素的语义信息，基于 FCN 的 SOD 方法 [11]–[16], [19]–[23], [25] 依靠探索多层和多尺度的深度特征，正如上面所提到的，已经主宰了这个领域。

尽管先前的方法已经通过采用强大的 CNNs 来达到很高的精度，但是这些方法都有相对来说较低的速度，较大的能量损耗并且会占据很大的存储空间。这些缺点使得在现实应用中采用这些一流模型十分困难。这也是我们进行这项工作的动机，也就是完成一个轻量级的 SOD，让它能够在精度、模型大小和速度之间进行权衡。我们的技术动机来自于这样一项事实，即已有的大部分基于 CNN 的方法 [10]–[29] 通过探索多尺度和多层的深度学习来提升其表现。在本文中，我们提出了一种新颖的 SAM 模块，它能够利用立体注意力来进行轻量级的多尺度学习。

b) 注意力机制： 注意力机制在人类的认知上扮演着重要的角色 [51], [52]。不同于一次性处理整张图片，人类的视觉系统会自适应地过滤掉如图像背景等相对不重要的信息来加强对于视觉结构的捕获。这启发了深度学习的研究。

最近的计算机视觉社群已经见证了注意力机制在许多任务上的大量成功，包括 SOD [19], [26], [34]，序列学习 [53]，行人重检测 [54]，和图像恢复 [55]。在图像分类领域，Wang *et al.* [56] 提出了使用一种沙漏模块来生成隐特征的注意力图像。在此之上，Hu *et al.* [57] 提出了一种被称为“压缩和激励”的模块来显式地发掘通道内的联系并且自适应地以逐通道的方式调整特征图像。在逐通道的注意力之上，CBAM [58] 以一种类似的方法提出了空间的注意力。这些方法均属于自注意力的类别。空间的和逐通道的自注意力分别可以自适应地强调信息量最大的特征图像块和通道。不同于这些方法，我们提出了一种立体的注意力机制，同时基于通道内的依赖关系和空间上的上下文线索来自适应地调整不同分支的信息流。因此，我们所提出的 SAM 模块可以在一种轻量级的设置下学习多尺度的信息。

III. 方法

在这个部分，我们会介绍我们所提出的用于 SOD 的框架。在 Section III-A中，我们展示了一种简单的多尺度模块。在 Section III-B，我们提出了 SAM 模块来有效地进行多尺度学习。最后，在 Section III-C中，我们把所提出的 SAM 模块合并到编码器-解码器网络中并且介绍整体的网络结构。

A. 多尺度学习

基于上述的分析，CNNs 的多尺度特征表征对于 SOD 有着很大的重要性 [14], [19], [23], [28], [30], [37]。受此启发，我们首先提出了多尺度模块来处理不同尺度的视觉信息。轻量是我们设计的中心想法。我们使用不同膨胀率的膨胀卷积来获取多尺度信息并且使用深度可分离卷积来减少浮点数运算量和模型的参数量。我们把这称为膨胀深度可分离卷积并且把它作为基础的卷积操作来提升网络的维度——也就是深度和宽度。

正式地，设 $\boldsymbol{I} \in \mathbb{R}^{C\times H\times W}$ 为输入特征图像，其中 C，H，W 分别为通道数、高和宽。对于这个输入 $\boldsymbol{I}$，我们首先对其采用一个深度可分离3×3 卷积 (指3×3 膨胀卷积) 来提取所有通道共通的信息 $\boldsymbol{F}_0$，也就是说，

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

其中 $\mathcal{K}_0$ 表示一次3×3 膨胀卷积 操作。在不同的分支上，不同膨胀率的3×3 膨胀卷积 被用于处理 $\boldsymbol{F}_0$，也就是说，

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

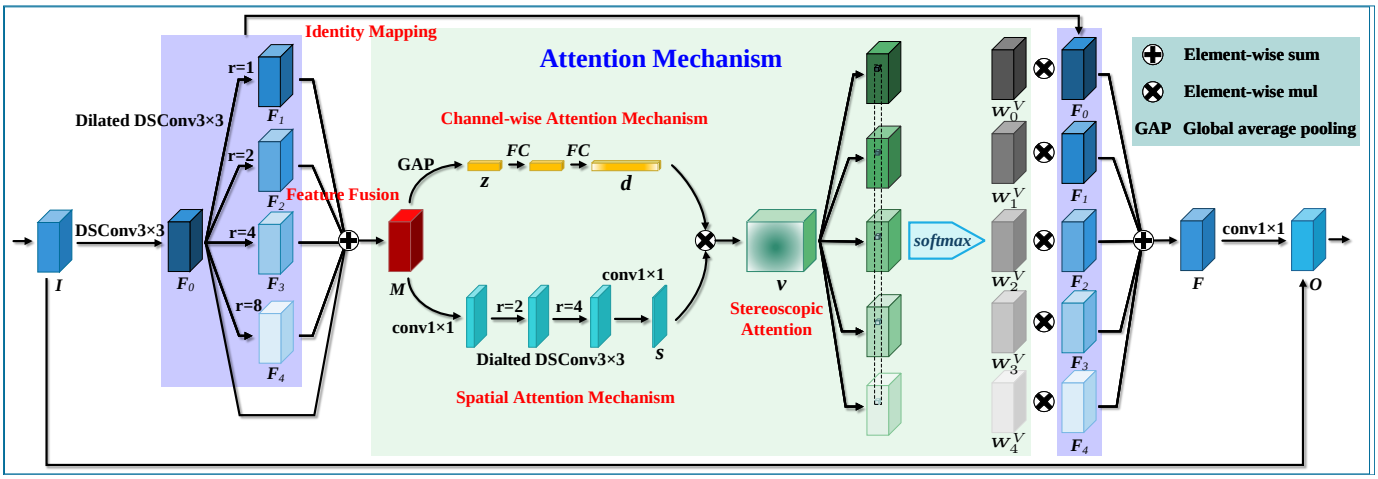

图 1. 对我们提出的 SAM 模块的图示。给出输入特征图像 $\boldsymbol{I}$ 的情况下，SAM 首先会通过多个分支来提取多尺度的特征 $\boldsymbol{F}_i(i=0,1,\cdots,N)$。多尺度特征 $\boldsymbol{F}_i$ 会通过逐元素相加的方法整合并得到 $\boldsymbol{M}$。然后，我们基于 $\boldsymbol{M}$ 计算逐通道的注意力 logits $\boldsymbol{d}$ 以及空间注意力 logits $\boldsymbol{s}$。把 $\boldsymbol{d}$ 和 $\boldsymbol{s}$ 相乘获得空间注意力 logits $\boldsymbol{v}$，然后使用 $softmax$ 函数来获得空间注意力权值 $\boldsymbol{W}_i^V(i=0,1,\cdots,N)$。$\boldsymbol{W}_i^V(i=0,1,\cdots,N)$ 是被用于整合 $\boldsymbol{F}_i$ 为 $\boldsymbol{F}$ 的权值。最后，一条残差连接 [43] 从 $\boldsymbol{I}$ 连接到最后并相加获得输出 $\boldsymbol{O}$。图中，$\otimes$ 表示逐元素乘法，并且两个相乘的特征图像会在相乘前复制为同样的形状。符号 r 表示膨胀卷积 3×3 膨胀卷积的膨胀率。在彩色下观看以获得最好效果。

其中 $\mathcal{K}_i$ 表示第 i 个分支的3×3 膨胀卷积 操作，并且 N 表示分支的数量。然后，我们使用一个逐元素的加法和一个残差连接来整合不同的尺度，也就是，

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

这里，我们使用逐元素加法而非拼接，是因为拼接会极大地增加通道的数量，导致更大的计算复杂度和更多的网络参数。最后，被整合的特征会在1×1 卷积 的作用下进一步重新排列，也就是，

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

其中 $\mathcal{K}_{fuse}$ 表示用来融合不同尺度的前后文信息的1×1 卷积 操作。和 $\boldsymbol{I}$ 的求和操作表示一个残差连接 [43]，这一操作的在 CNNs 训练中的有效性已经被证明。

膨胀率和分支数作为超参数参与到多尺度模块之中。经验显示，分辨率更高的输入特征 $\boldsymbol{I}$ 会需要更大的膨胀率和更多的分支，因为更大的特征往往拥有不同尺度的上下文信息。

B. 立体注意多尺度 模块

在 Section III-A中描述的多尺度模块存在着一个可能的短板，那就是逐元素相加的部分。当不同分支的上下文信息被直接相加在一起时，提供有用信息的分支可能会被没有提供有用信息的分支弱化甚至直接被完全压过。除此之外，不同深度的层可能会倾向需要不同尺度的信息，但逐元素相加会给所有的尺度赋予相同的重要性。为了缓解这一问题，我们提出了一种新颖的立体注意力机制，它可以允许不同空间位置的不同通道能够借助一种软注意力机制，自适应地调整不同分支的权重。Fig. 1提供了我们所提出的 SAM 模块在四个分支情况下的图示。

a) 立体注意力机制: 不失一般性，我们把输入特征看作是一个四维的张量 $F\in\mathbb{R}^{(N+1)\times C\times H\times W}$，其中每个分支 $u\in\{0,\cdots,N\}$ 生成不同尺度和语义等级的特征 $F_i\in\mathbb{R}^{C\times H\times W}$。人们普遍认为，网络中不同层倾向于需要不同尺度的信息，因此盲目地把所有特征 F 喂入会导致严重的过拟合。一个自然的解决方案是注意压制不提供有用信息的分支而自动促进具有辨别性的分支，而我们使用一种完善的注意力机制来实现了它。

在我们的设置中，一种理想的注意力模块需要有以下的特征。i) 由于各个通道的独立性，最终的注意力需要有很强的通道内依赖性。更具体来说，不同的特征通道通常是由独立的滤波器对输入特征卷积而来。在这种情况下，如果一个来自于特定通道的特征对最后的预测结果有何更加有用的信息，那么同样分支并且同样通道的特征应该也会提供同样有价值的信息。ii) 最后的注意力需要有一个强的区域间依赖。原因在于，作为一个中层任务，SOD 需要对特定层的每一个像素的相邻像素进行一定程度的推理。iii) 计算过程需要有效率。这种情况下，朴素地独立学习一组 (比如 $C\times H\times W$) 逐分支的注意力权重是次优的，因为它会有很重的计算负荷。

上述的前两个需求分别在全局和局部两个方面规范了注意力机制。这自然地驱使我们把最终的注意力权重 $\boldsymbol{v}$ 分解为如下的两个独立的权重:

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

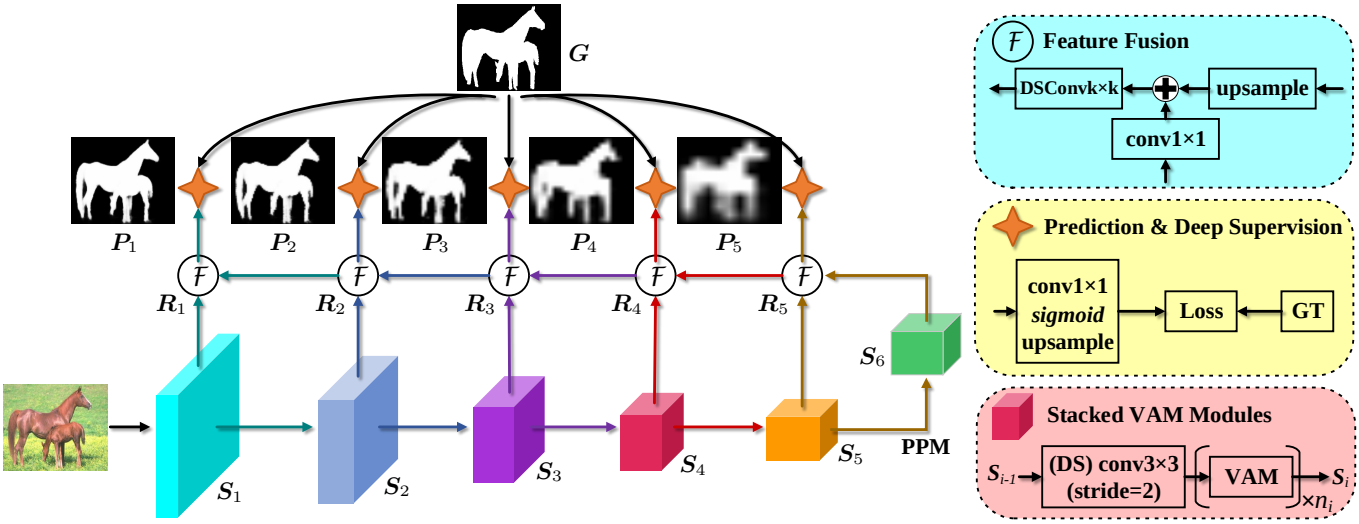

图 2. SAMNet 的整体编码器-解码器结构。$\boldsymbol{S}_i$ 和 $\boldsymbol{R}_i$ 分别表示编码器输出的特征图像和第 i^{th} 阶段的解码器。$\boldsymbol{P}_i$ 是在 i^{th} 阶段预测的显著性图像，而 $\boldsymbol{P}_1$ 就是 SAMNet 的最终预测结果。$\boldsymbol{G}$ 是正确标注的显著性图像。PPM 表示著名的金字塔池化模块 [59]。最好在彩色下观看。

其中 $\boldsymbol{d}$ 和 $\boldsymbol{s}$ 分别表示逐通道的注意力和逐区域的注意力。$\otimes$ 表示逐元素乘法，并且 $\boldsymbol{d}$ 和 $\boldsymbol{s}$ 会在相乘前被复制为同样的形状，也就是 $(N+1)\times C\times H\times W$。更具体来说，逐通道的注意力 $\boldsymbol{d}\in\mathbb{R}^{(N+1)\times C}$ 从空间上压缩特征，并且有注意地以一种全局逐通道的方式压制不能提供有效信息的特征而提升有辨析度的特征。同样的，逐区域的注意力 $\boldsymbol{s}\in\mathbb{R}^{(N+1)\times H\times W}$ 从特定区域的不同通道中吸取特征。最终，$\boldsymbol{d}$ 和 $\boldsymbol{s}$ 被广播到同样的维度 $(N+1)\times C\times H\times W$，通过 Eq. (5) 以获得最终的注意力权重。

b) 作为预处理的尺度融合: 为了减少计算负担，我们通过逐元素的加法整合了多尺度的上下文信息，也就是：

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

融合后的特征 $\boldsymbol{M}$ 被用来计算每个分支的门控的统计数据。不同于自注意力，我们的方法中每个模块分支被所有的分支的学习情况所决定，而不是自注意力机制中的单个分支。因此，我们的 SAM 模块可以以一种“全局”的视角来提取所有分支中能提供有效信息的特征。因此，SAM 模块具有能学习其偏好尺度特征的能力。

c) 逐通道注意力机制: 通道注意力机制的目标是为每个分支 i，$i=0,1,\cdots,N$ 计算一个逐通道的注意力向量 $\boldsymbol{W}_i^D\in\mathbb{R}^C$。为了探索不同通道间的关系 [57]，我们利用全局平均池化 (GAP) 来往融合的特征图像 $\boldsymbol{M}$ 中嵌入全局的信息，也就是，

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

其中 $\boldsymbol{z}\in\mathbb{R}^C$ 是把逐通道的信息 $\boldsymbol{M}$ 编码后的隐向量。然后，我们对这个隐向量采用一个两层的多层感知机 (MLP)[1]，同时按照如下的方式提取了不同尺度的逐通道信息：

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

其中 $\boldsymbol{d}\in\mathbb{R}^{(N+1)C}$ 的形状会被修改为 $R^{(N+1)\times C}$。*softmax* 函数会在逐分支的维度上作用于 $\boldsymbol{d}$ 来获得逐通道的注意力，也就是，

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

加上上述的逐通道注意力后，Eq. (3) 中的特征整合会被重写为：

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

其中 $\boldsymbol{W}_i^D$ 会在进行逐元素乘法前被复制为和 $\boldsymbol{F}_i$ 一样的形状 (也就是 $\mathbb{R}^{C\times H\times W}$)。Eq. (4) 中的融合会被用到 $\boldsymbol{F}^D$ 上来获得最后的结果。

d) 区域注意力机制: 区域注意力机制的目标是计算一个区域注意力图像 $\boldsymbol{W}_i^S\in\mathbb{R}^{H\times W}$ 来强调或压制特定位置的活动。众所周知，更大的感知域可以更好地捕获上下文信息，这对于学习逐位置的注意力十分关键 [58]。基于此，我们使用了3×3 膨胀卷积 来增大感知域并同时保持较低的计算复杂度。具体来说，融合后的特征 $\mathbf{M}$ 会首先通过1×1 卷积 映射到一个低维空间 $\mathbb{R}^{C/4\times H\times W}$ 来减少参数量和计算消耗。然后，两个3×3 膨胀卷积 被作用于这个降维后的特征来有效地进行上下文信息的整合。最后，

[1]我们在这两层线性变换中插入了批标准化和 ReLu 激活函数。

特征通过一个1×1 卷积 被再次降维到 $R^{(N+1)\times H\times W}$。数学上，我们有：

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

其中 $\mathcal{F}_i^{k\times k}$ 表示 i^{th} (深度可分离或者一般) $k\times k$ 卷积. 类似于逐通道的注意力机制，我们可以公式化利用空间注意力的多分支融合如下：

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

其中我们有 $0 \le h < H$ 以及 $0 \le w < W$。$\boldsymbol{W}^S_i$ 在进行乘法前被复制为 $\mathbb{R}^{C\times H\times W}$ 的形状。Eq. (4) 被用在 $\boldsymbol{F}^S$ 上来计算结果。

e) 最终标准化: 当 $\boldsymbol{d}$ 和 $\boldsymbol{s}$ 决定了之后，*softmax* 函数会在分支的维度上作用于立体注意力，也就是，

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

在这之后，$\boldsymbol{W}^V_i \in \mathbb{R}^{C\times H\times W}$ 就会作为第 i^{th} 个分支的立体权重值。在 Eq. (3) 中的特征融合可以被重写为：

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

Eq. (4) 也可以被用于 $\boldsymbol{F}^V$ 来生成 SAM 模块的结果。我们在 Fig. 1中展示了 $\boldsymbol{F}^V$ 的计算过程。

到目前位置，我们已经设计了四种用于多尺度学习的模块，也就是 *i.e.,* $\boldsymbol{F}$, $\boldsymbol{F}^D$, $\boldsymbol{F}^S$ 和 $\boldsymbol{F}^V$。在它们之中，$\boldsymbol{F}^V$ 是本文的默认 SAM 模块，因为它能够同时考虑逐通道和空间的注意力从而获得更好的表现。有了上述的设计之后，每个 SAM 模块可以自动地决定在多尺度分支中哪些信息是它所需要的。

C. 网络结构

a) 骨干网络结构: 根据已有的研究 [30]–[32], [35], [60]，我们搭建了一个使用我们提出的 SAM 模块作为基础单元的 FCN [50] 结构。在前五个阶段中，我们使用步长为 2 的3×3 膨胀卷积[2]来对输入进行降采样并且调整通道的数量。然后我们使用提出的 SAM 模块来学习多尺度上下文信息。对于前两个阶段，因为输入的特征图像会有相对来说较高的分辨率，我们只会使用一个 SAM 模块从而减少繁重的计算负担。与此相对的，从第三个阶段到第五个阶段，我们把多个 SAM 模块叠加到一起来增大感知域并丰富深度卷积表征。在最后第五个阶段之后，我们使用金字塔池化模块 (PPM) [59] 来进一步提高全局特征的学习。在 Table I中展示了 SAM 模块中膨胀率和分支数的默认设置。关于不同网络设置中的详细消融研究，请参考 Table I。

[2]在第一个阶段，我们只使用了普通的3×3 卷积。

表 I
所提出 SAMNet 的骨干网络设置。

阶段	分辨率	模块	#M	#F	步长	膨胀率
1	224×224	3×3 卷积	1	16	2	-
	112×112	SAM	1	16	1	1,2,3
2	112×112	3×3 膨胀卷积	1	32	2	-
	56×56	SAM	1	32	1	1,2,3
3	56×56	3×3 膨胀卷积	1	64	2	-
	28×28	SAM	3	64	1	1,2,3
4	28×28	3×3 膨胀卷积	1	96	2	-
	14×14	SAM	6	96	1	1,2,3
5	14×14	3×3 膨胀卷积	1	128	2	-
	7×7	SAM	3	128	1	1,2
6	7×7	PPM	1	128	1	-

* "#M" 指 "模块" 列中被指定了类型的模块的数量。"#F" 指卷积核 (即通道) 的数量。

b) 编码器-解码器网络: 基于上述的骨干网络，我们可以搭建一个轻量级的编码器-解码器网络，如图 Fig. 2所示。令 $\boldsymbol{S}_i : i = 1, 2, \cdots, 6\}$ 表示骨干网络每个阶段输出的特征图像。为了融合顶层的特征，我们对 $\boldsymbol{S}_5$ 使用了一个1×1 卷积 来调整其通道数，并且通过逐元素加法来结合 $\boldsymbol{S}_5$ 和 $\boldsymbol{S}_6$。然后，我们使用了一个膨胀卷积 $k\times k$ 来进一步整合融合后的激活结果。正式地，我们有

$$\boldsymbol{R}_5 = \mathcal{G}_5^{k\times k}(\mathcal{G}_5^{1\times1}(\boldsymbol{S}_5) + \boldsymbol{S}_6), \tag{15}$$

其中，$\mathcal{G}_5^{k\times k}$ 表示第五阶段的 (深度可分离)$k\times k$ 卷积，并且 $\boldsymbol{R}_5$ 表示第五阶段的合成特征图像。类似的，为了合成底层的特征，我们对顶层的合成后的特征进行上采样来保证空间分辨率能够对应上底层阶段的分辨率。总结起来，我们有

$$\boldsymbol{R}_i = \mathcal{G}_i^{k\times k}(\mathcal{G}_i^{1\times1}(\boldsymbol{S}_i) + \mathrm{Up}(\boldsymbol{R}_{i+1})),\quad i = 1, 2, 3, 4, \tag{16}$$

其中 Up 表示上采样率为 2 的上采样操作。

c) 深监督 & 损失函数: 我们采用深监督 [67] 来提升隐层的学习过程的透明度。具体而言，对于合成的特征 $\{\boldsymbol{R}_i,\ i = 1, 2, 3, 4, 5\}$，我们依次使用一个单输出通道的$1\times1$ 卷积 和 *sigmoid* 激活函数来生成许多的预测 $\{\boldsymbol{P}_i,\ i = 1, 2, 3, 4, 5\}$。我们使用标准的交叉熵损失来训练，它可以被公式化如下：

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

表 II
与现有方法在参数量、GPU 用量、F_β 和 MAE 上的比较。

方法	# 参数量 (M)	存储 (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 [61]	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 [62]	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 [63]	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 [64]	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 [65]	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 [65]	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 [66]	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

其中 $\mathcal{L}_{\text{BCE}}$ 是标准的交叉熵损失函数，而 $\boldsymbol{G}$ 表示标注好的的显著性图像。λ 表示为了平衡损失的权重值，本文中，我们按照 [59] 将其经验性地设置为 0.4。

IV. 实验

A. 实验设置

a) 实现细节： 我们使用 PyTorch [68] 库来实现我们提出的方法。所有实验的训练都是使用的 Adam 优化器 [69]，参数为 $\beta_1 = 0.9$，$\beta_2 = 0.999$，10^{-4} 的权值衰减以及 20 的批大小。我们的方法在 ImageNet 数据集上进行了预训练，如同 [43]。我们使用多项式学习率衰减策略，这样的话，第 n^{th} 个迭代的学习率为：$init\_lr \times \left(1 - \frac{n}{\#epochs}\right)^{power}$，其中 $init\_lr = 5 \times 10^{-4}$ 并且 $power = 0.9$。我们对所提出的模型进行了 50 个迭代的训练，即 $\#epochs = 50$。

b) 数据集： 我们在六个数据集上广泛验证了我们提出的方法，包括 DUTS [70]，ECSSD [71]，SOD [72]，HKU-IS [47] 和 DUT-OMRON [45] 数据集。六个数据集分别包括 15572、1000、300、4447、6232 以及 5168 张自然图像以及它们分别对应的像素级标注。根据最近的研究 [14], [21], [31], [35], [73]，我们把提出的模型在 DUTS 的训练集上训练并在 DUTS 的测试集 (DUTS-TE) 和其它五个数据集上进行测试。

c) 评估标准： 为了评估 SAMNet 和先前的一流方法的精度，我们考虑了四种广泛使用的指标，也就是 F_β 指标分数 (F_β)，平均绝对误差 (MAE)，加权 F_β^ω 指标 (F_β^ω) [74] 和结构相似性指标 (S_β) [75]。给定范围在 $[0, 1)$ 的阈值之后，我们可以对预测出的显著性概率图像进行二值化，并通过比较二值化后的概率图像和二值的正确标注的显著性图像计算其精度和召回值。计算出精度和召回值后，F_β 指标就是精度和召回值的加权平均数，即，

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

其中，根据之前的工作 [21], [23], [28], [30], [31]，我们设置 $\beta^2 = 0.3$ 来强调精度的重要性。注意每个阈值会对应一个这里的 F_β，而我们会报告所有阈值中最大的 F_β。MAE 会测量预测的显著性图像 $\boldsymbol{P}$ 和标注好的显著性图像 $\boldsymbol{G}$ 的

表 III
与现有方法在 FLOPS，速度，F_β^ω 和 S_β 等方面的比较。

Methods	FLOPs (G)	速度 (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 [61]	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 [62]	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 [63]	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 [64]	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 [65]	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 [65]	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 [66]	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

区别，他可以被这样计算，

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

其中 H 和 W 分别表示显著性图像的高和宽。加权的 F_β^ω 指标 [74] 旨在修正传统的评估指标中的插值缺陷、依赖缺陷以及等重要性缺陷等重要缺陷，而我们使用它作为评估 SAMNet 和其它竞争者的默认设置。对于 S_β [75]，它同样是一个广泛使用的评估指标，因为它可以测量预测值和被标注好的标签的结构相似性。S_β 由区域感知和对象感知的结性相似性指标构成，其中前者由著名的 SSIM [76] 实现，而后者基于概率论。在我们的实验中，我们使用官方代码的默认设定。

d) 效率指标: 本文的目标是为 SOD 提供一种轻量而且强大的解决方案，所以我们同样评估了不同方法的效率和灵活性，包括模型的参数量 (#Param)，GPU 的内存用量，浮点运算数 (FLOPs) 和推理速度 (FPS)。GPU 内存用量衡量模型在测试单张图片时对于内存的需求。FLOPS 衡量模型的计算量消耗，同时更少的 FLOPS 可以有更低的能量消耗。在以下轻量级骨干网络 [38]–[41] 和高效语义分割 [64]–[66] 中，速度代表在使用一张 NVIDIA TITAN XP GPU 时，模型每秒可以推理的图像数量。因为深监督只用于训练，因此在测试 SAMNet 的速度时，我们忽视 $\{\boldsymbol{P}_i, i=2,3,4,5\}$ 的计算。对于显著性检测来说，我们通常使用内存用量，FLOPS 以及速度这几个指标，并且是在在处理 336×336 大小的输入图像时，除非该方法特定了输入的维度。因为高效语义分割方法 [64]–[66] 通常为高分辨率图像所设计，我们使用 672×672 的输入来保证其精度，否则我们会得到很低的精度。对于基于轻量级骨干网络重新设计的基准方法 [38]–[41]，我们使用 336×336 的输入。本文中，#Param 和 GPU 内存用量以兆 (M) 为单位衡量，而 FLOPS 以千兆 (G) 为单位衡量。

B. 表现分析

在这个部分，我们比较所提出的 SAMNet 和其它 20 种一流 SOD 方法，包括 DRFI [8]，DCL [61]，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 [62]，BASNet [18]，AFNet [27]，

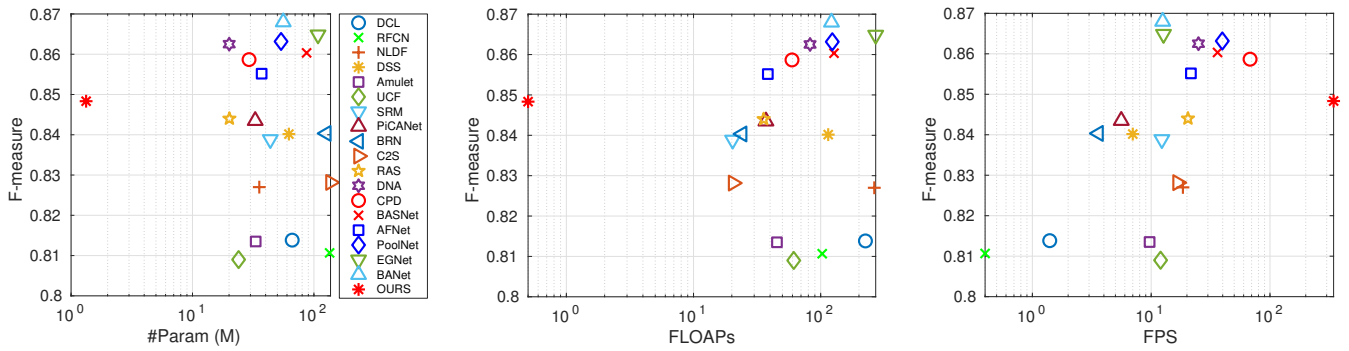

图 3. 性能与计算量损失之间权衡的图示。F 指标 (F_β) 为六个数据集上结果的平均值。注意横坐标为对数。

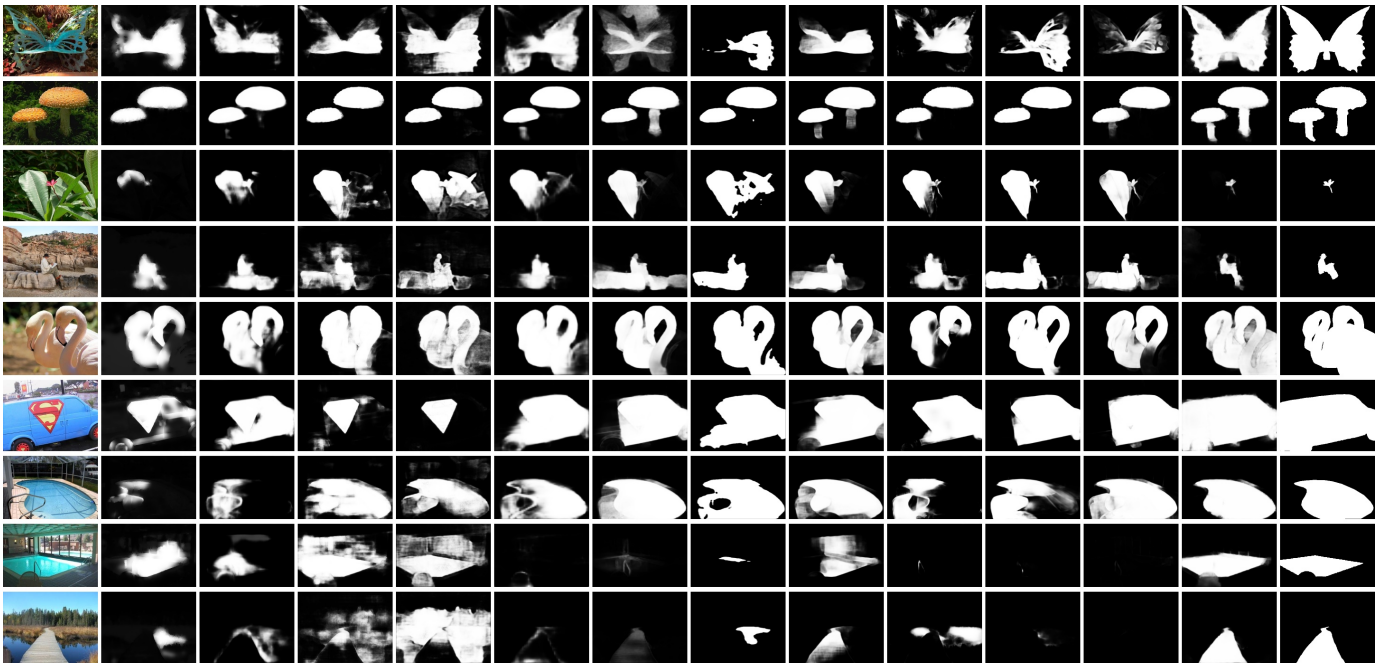

图 4. 和一流 SOD 方法的质量比较

PoolNet [28]，EGNet [37] 和 BANet [63]。除了这些已有的 SOD 方法，我们同样对比了几种被广泛应用于图像分类的轻量级骨干网络，包括 MobileNet [38]，MobileNetV2 [39]，ShuffleNet [40] 和 ShuffleNetV2 [41]。为了将其用于 SOD 任务上，我们在这些网络上增加了一个和提出的 SAMNet 一样的解码器。为了对比，得到的基础模型会使用同样的设置进行训练。除此之外，我们同样比较了一些高效的语义分割方法，包括 ICNet [64]，BiSeNet [65]，和 DFANet [66]。为了把这些基础模型用于 SOD，我们将其最后的 *softmax* 激活函数换成标准的 *sigmoid* 激活函数。对于 BiSeNet [65]，我们会分别报告其使用 ResNet-18 [43] (也就是 BiSeNet R18) 骨干网络和 Xception-39 [77] (也就是 BiSeNet X39) 骨干网络的结果。

a) 和现有 SOD 方法的对比: Table II展示了提出的方法 SAMNet 和先前的一流替代方法在参数量、GPU 内存用量、F_β 和 MAE 方面的评估结果。Table III展示了在 FLOPs、速度、F_β^ω 和 S_β 方面的评估结果。注意 FLOPs 的大小和网络推理时的能量的消耗高度相关。结果清楚地展示了 SAMNet 得到了与一流的 SOD 解决方案相同的精度，特别是在 F_β、MAE 和 S_β 方面。而 SAMNet 需要 1 到 2 个数量级的更少的计算资源。举例而言，和表现最好的 BANet [63] 相比，SAMNet 在六个数据集上表现出了稍低的平均 F_β (0.848 对比 0.868)，但 SAMNet 比 BANet [63] 的参数量少了 42×，GPU 内存消耗少了 5.5×，FLPOs 少了 243×，速度快了 27×。这在移动设备上会有显著的影响。在这些设备上，有限的计算资源、减少的能源供应、被限制的运行内存和存储空间无法承担传统繁琐的 SOD 方法所带来的沉重开销。

我们同样在 Fig. 3种用图像展示了这种比较，在这里，各种方法在效率和准确度上的权衡被更加清晰地展示。在展示 F_β *vs.* #Param 和 F_β *vs.* FLOPs 的子图上，SAMNet 位于最左上的角落。在展示 F_β *vs.* FPS 的子图上，

表 IV
关于 SAMNet 设计选项的消融实验。

No.	成分					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**

* 我们使用一般的单分支模块作为基础模型 (No. 0)。这里，“MB”，“CA”，“SA”，“PP” 和 “IP” 分别指代简单的单分支模块 ($\boldsymbol{F}$)，逐通道注意力 ($\boldsymbol{F}^D$)，空间注意力 ($\boldsymbol{F}^S$)，金字塔池化和 ImageNet [78] 预训练。当同时使用通道注意力和空间注意力时，我们就得到了立体注意力机制 ($\boldsymbol{F}^V$)。

表 V
关于 SAMNet 配置的消融实验。

	阶段	配置	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
膨胀率	$1-4$	$1,2$	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” 代表每个阶段 SAM 模块的数量 “#Filters” 表示卷积核的数量 (也就是通道数)，同时 “$\times k$” 表示我们把 SAMNet 中卷积核的数量乘了 k。注意所有的实验全都从零开始训练。

SAMNet 位于最右上的角落。这意味着 SAMNet 以更少的参数量和 FLOPs 以及更快的速度，达到了和先前的一流模型相当的准确度。因此，我们可以得出结论：SAMNet 在精度、参数量、GPU 内存用量、FLOPs 量和速度之间达成了一个很好的权衡。

b) 和其它轻量级网络的比较： Table II和 Table III 同样展示了 SAMNet 和其它基于轻量级骨干网络的基础模型，也就是 MobileNet [38]，MobileNetV2 [39]，ShuffleNet [40]，和 ShuffleNetV2 [41] 之间的比较。尽管这些基础方法有着比起 SAMNet 相似甚至更快的的速度，但 SAMNet 达到了在所有的指标上都远超它们的精度。这表明，将现有的轻量级骨干网直接应用于 SOD 是不理想的。这同样展示了为轻量级 SOD 任务精心设计网络结构的重要性以及所提出的多尺度注意力方法的优势。

c) 与高效语义分割方法的比较： 从 Table II和 Table III中，我们可以看到 SAMNet 的性能大大胜过了高效语义分割方法 [64]–[66] ，并且还有更少的参数，更少的 FLOPs 和更快的速度。有趣的是，高效语义分割方法比起基于轻量级骨干网络的基础模型表现更差。这显示了高效语义分割方法为了针对语义分割任务而进行了大量调整，而不适合直接用于 SOD 任务。因此，轻量级 SOD 是一个重要的问题并且需要得到社群的更多的关注。

d) 质量对比： 在 Fig. 4中，我们提供了一些可视化的例子来展现 SAMNet 的优越性。尽管 SAMNet 的表现略逊于繁琐的传统 SOD 方法，它仍旧可以在许多很具挑战的场景下分割出显著的物体以及其边界，比如复杂的情景 (第 1 和第 3 行)、前景和背景对比度较低 (第 2 和第 4 行)，大型物体 (第 5 和第 6 行)，有不自然光照的场景 (第 7 和第 8 行) 和迷惑性自然场景 (第 9 行)。结合 SAMNet 的轻量级和高效性质。它有可能促进现实世界的 SOD 应用。

C. 消融实验

在这个部分里，我们进行了消融研究来展示在 SAMNet 中所提出模块各个组件的效果以及参数的配置。实验

的设置遵循 Section IV-B中的设置。

a) 所提出模块的组件: Table IV展示了所提出模块的组件的消融研究结果。所提出的 SAM 模块被设计用于仔细组合这些基本的组件于一个不平凡的模块来进行有效而又高效率的多尺度学习。Table IV说明了随着更多组件加入框架，其效果也在逐渐变好。除此之外，No. 0 和 No. 5 之间的比较说明了提出的解决方案比起基础模型的优越性，其中表现的差距完全来自于我们的贡献因为两个模型均从零开始训练而没有 ImageNet [78] 的预训练。

b) SAMNet 的配置: Table V展示了不同网络配置的消融研究结果。有趣的是，所提出的 SAMNet 对于配置的微小变动有着鲁棒性、引入更多的参数会得到更好的表现，如提升卷积核的数量，但这与我们的轻量级 SOD 的目标是正交的。我们的 SAMNet 的默认设置是在考虑了有效性和轻量级限制之间的权衡后设定的。

V. 总结

不同于仅仅考虑精度，本文专注于轻量级 SOD，它在精度、效率、参数量和 FLOPs 间进行权衡。我们提出了一种新颖的 SAM 模块，它能够让小网络有效地编码高层次特征和低层次细节。运用 SAM 模块，所提出的 SAMNet 能够达成和一流 SOD 方法相当的表现，同时省下几个数量级的开销。这种性能和效率之间出色的权衡使得 SAMNet 可以在资源被限制的环境下，比如移动设备，提供更高精度的 SOD。SAMNet 同样明显超越了其它著名的在图像分类领域的轻量级网络 [38]–[41] 以及语义分割领域的轻量级网络 [64]–[66]，这展现了轻量级的 SOD 是一项值得研究的工作并且需要被设立为一个单独的研究领域。据我们所知，SAMNet 是首个轻量级 SOD 方法，可以期待其为 SOD 铺就一条新的道路。在这项工作中，我们希望能够唤起对轻量级 SOD 的研究，这将促进更多实际的 SOD 应用。在未来，我们计划将权值量化和网络压缩技术用于加快 SAMNet 的 5fps 的 CPU 速度，以实现实时的性能。

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).

参考文献

[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 classifica-

tion," 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] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 478–487.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artif. Intell. Stat.*, 2015, pp. 562–570.

[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] 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.

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

[72] 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.

[73] 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.

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

[75] 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.

[76] 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.

[77] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1251–1258.

[78] 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.