探索显著性检测中语义信息的高效模型

程明明, 高尚华, Ali Borji, 谭永强, 林铮, 汪萌

摘要—基于卷积神经网络的显著性检测（SOD）方法已经实现了出色的检测效果。然而，语义信息的编码方式以及这些方法是否与类别无关仍然缺少探索。显著性检测模型是基于ImageNet预训练的骨干网络建立的，这导致了信息泄漏和特征冗余，进而成为了探索上述这些问题的一个主要障碍。为解决这一问题，我们首先提出一个极其轻量化的整体网络用于显著性检测，这一网络可以摆脱分类任务的骨干网络并且从头进行训练。我们之后使用这一网络探索显著性检测模型的语义信息问题。基于我们提出的整体网络，以及通过一种新颖的动态权重衰减机制实现的表征冗余缩减，我们的模型的参数数量仅仅100K（约大模型的0.2%），并且在主流的SOD基准测试上的表现与SOTA模型相当。使用CSNet，我们发现a) 显著性检测方法和分类方法有着不同的机制， b) 显著性检测模型对类别不敏感， c) 基于ImageNet的预训练对于显著性检测模型是不必要的，并且 d) 显著性监测模型需要的参数远少于分类模型。源代码已经开源于https://mmcheng.net/sod100k/。

Index Terms—显著性监测，高效的显著性预测，语义。

1 引言

基于观察到的人类反应时间和信号沿生物路径的传播时间 [12], [82]，认知心理学的研究表明人类的视觉系统（HVS）在识别场景的语义信息之前存在着预先注意、自下而上的注意力机制。计算机视觉社区已经通过类别无关的手工设计的对比特征对这些发现进行了建模 [2], [10]，实现了传统的显著性检测方法 [3]。许多计算机视觉应用，例如图片检索 [5], [9], [24]，视觉追踪 [30]，摄影构图 [8], [22]，图像质量评估 [80]，内容感知图像处理 [60], [92]，以及无监督语义分割 [19] 都利用了显著性检测模型。这些模型都基于假设“显著物体是通用且类别无关的”。

随着基于卷积神经网络的显著性检测方法取得了巨大的进展，这些方法大都通过局部细节和全局特征 [75], [90], [95], [96]，注意力信息 [4] 以及边缘信息 [15], [79], [98] 来提升显著性检测的SOTA效果。现有的基于卷积神经网络的显著性检测模型依赖于ImageNet预训练的骨干网络 [16], [26]，其通过大量的参数提取特征。然而，ImageNet预训练会不可避免地引入类别语义信息，这与显著性区域是类别无关的这一假设会形成潜在的冲突 [3], [38], [87]。这一潜在的冲突又引发了新的问题。语义信息在自底向上的显著性检测任务中扮演着什么样的作用？对于显著性检测模型的训练，ImageNet预训练是否是必要的？

- *程明明 (通讯作者，*cmm@nankai.edu.cn*) 和高尚华为并列第一作者。
- 程明明，高尚华，谭永强，林铮来自于南开大学。
- *A. Borji is with Primer Technologies Inc., San Francisco, USA (aliborji@gmail.com).*
- 汪萌为合肥工业大学教授。

- 本文是*IEEE TPAMI* 论文 *[7]*的中译版。相应的会议版本发表于*ECCV 2020 [20]*。

设计显著性检测模型的两个原则可以被用以回答这些问题。首先也是最重要的，一个显著性检查模型在不依赖ImageNet预训练时应该可以完成训练。现有的显著性检测模型是基于分类任务的骨干网络设计的。这些网络其包含了太多的参数，使其难以从头训练。为了区分成千上万的类别，即使是轻量级的分类模型如ResNet-18 [26] 和 MobileNet v2 [35] 也分别有 11M和4.2M的参数（*vs.*我们整个模型有100k参数）。这项工作的一个重要动机就是验证是否面向类别的特征和相应的大量参数对显著性检测任务是必不可少的。第二，显著性检测任务需要模型有高分辨率的特征以及强大的多尺度能力，这对分类是不必要的 [77], [95]。现有的工作 [15], [31], [78] 通过在骨干网络上增加显著性检测任务相关的模块来缓解这些问题，但不可避免地引入了额外的参数。

为了满足上述要求，我们提出了一个极其轻量化的模型，其整体地考虑特征提取器和显著性检测模块。我们将OctConv [6]一般化，得到gOctConv，其有着更高的灵活性以及额外的自适应能力。我们提出一种动态权重衰减机制，基于此gOctConv可以实现自适应可学习的通道数。这种机制不仅仅帮助我们在SOD模型中分析语义信息，也使模型在损失微不足道的性能的情况下减少～ 80%的参数。利用gOctConv，我们提出了一种极其轻量化的整体的跨阶段跨尺度的网络，即CSNet。受益于这种整体的设计以及动态权重衰减机制，CSNet在仅仅有100K参数(约SOTA模型参数的0.2%)的情况下实现了与SOTA相当的效果。

受益于非常少的参数，我们的CSNet可以直接从头训练而不需要ImageNet预训练，这提供了回答问题（关于基于CNN的显著性检测模型中语义信息）的机会。通过将显著性检测模型迁移到分类任务并且测试其在未知类别上的效果，我们分析了显著性检查任务对类别的敏感性。实验结果说明

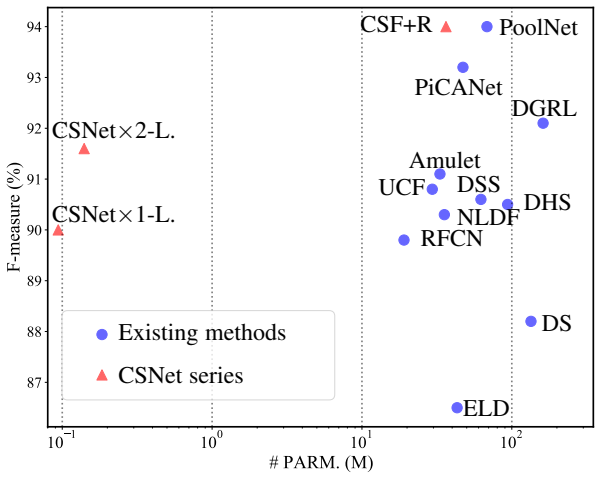

图 1. 显著性检测模型的参数 vs. 准确性。

了显著性检测模型对类别并不敏感，并且检测到的显著性对象是通用的。进一步地，我们观察到显著性模型需要远少于分类模型的参数量，因为分类模型中用于区分类别的表征对于显著性检测模型是不需要的。我们的显著性模型对类别不敏感，不仅提供了一个提高显著性检测模型效率的机会，同时也模仿了类别无关的的自下而上的人类注意力机制。

总的来说，本文由两个主要的贡献：

- 我们对基于卷积神经网络的显著性检测模型中的语义信息进行了彻底的分析，并通过实验验证了显著性检测模型几乎不需要语义信息（即对类别不敏感）。
- 通过整体地设计特征提取器和显著性检测模块，我们抛弃了之前被普遍使用的CNN骨干网络，这些骨干网络包含了大量关于类别信息的参数。与通过动态权重衰减机制引入的稀疏性相配合，我们将模型的参数减小到SOTA模型的约0.2%，同时有相当的效果。

在这些工作中，我们的关注点在于显著性检测的语义信息，然而会议版主要关于搭建轻量级的显著性检测模型。在Sec. 3中，我们介绍了动态权重衰减机制以及gOctConv中可学习的通道数。在Sec. 4中，我们之后介绍了轻量的整体CSNet分析，用于分析显著性检测模型。使用CSNet，我们在Sec. 5中分析了显著性检测模型的语义信息。在Sec. 6中，我们通过性能评估和消融展示了CSNet的效率。

2 相关工作

2.1 显著性物体检测

传统方法 [10], [38], [73], [87], [102] 主要依赖于人工设计的特征来检测显著物体。早期的基于深度学习的方法 [44], [51], [76] 利用卷积神经网络从图片块中提取有更多信息的特征，以此提取显著图的质量。受全连接卷积网络（FCN） [56]启发，最近的方法 [14], [42], [52], [78], [93], [95] 将显著性物体检测视作像素级预测任务。这些方法 [31], [67], [77], [95], [96] 从网络的不同阶段同时捕捉细节信息和全局信息。 [46], [58], [79], [98] 利用边缘信息进一步优化显著图的边界。其他的方法 [83], [95], [99] 也从网络优化的角度优化显著性检测。最近，Wei等人 [81]将原始的显著图分解为一个细节图和一个物体图，以分别更好地学习边缘特征和避免边缘周围像素上的错误。 Pan等人 [65]整合相邻层次的特征以解决显著性检测任务中的多尺度问题。 [100]利用多级门单元来平衡每个编码器块的作用，以抑制非显著区域的特征。尽管有出色的效果，这些基于卷积神经网络的方法需要强大的ImageNet预训练模型作为特征提取器，这通常会导致很高的计算成本。而且，基于卷积神经网络的显著性检测模型背后的语义信息以及预训练的必要性还没有人研究。

2.2 轻量级模型

目前，大多数轻量化模型主要是为分类任务设计的，他们通常利用 inverted block [34], [35]， channel shuffling [59], [97], 和 SE attention module [34], [72] 等模块提升模型的效率。分类任务 [68] 预测一张图片的高层次的语义类别，需要更多的全局的信息和更少的细节信息。因此，为分类任务设计的轻量化模型 [34], [35], [59], [97] 在网络的浅层使用大量的降采样策略来减少乘加运算(MACC)。这些策略使他们不适用于显著性检测任务的特征提取，因为显著性检测任务同时需要粗糙和精细的多尺度信息。而且，显著性检测任务关注于确定显著区域，然而分类任务被用来预测类别信息。为了在有限的计算资源下提供显著性检测的效果，计算资源（如分辨率、通道数）的分配应该被重新考虑。

2.3 网络剪枝

许多网络剪枝算法可以剪去不那么重要的卷积核，特别是在通道上 [29], [45]。为了剪去卷积核，冗余的卷积核可以通过范数 [27], [45]， 下一层的统计信息 [57]，权重的几何中值 [28] 以及批归一化层的缩放因子 [54]来寻找。元剪枝 [55]利用生成的权重来估计剩余卷积核的重要性。大多数剪枝方法仍然依赖于权重衰减等正则化方法增强卷积核的稀疏性。我们提出的动态权重衰减机制可以稳定地增强网络参数的稀疏性，以辅助剪枝算法剪掉冗余的卷积核，这最终使我们提出的gOctConv模块有可学习的通道数。

3 GOCTCONV与可学习通道数

为了在不受ImageNet预训练干扰的情况下分析基于卷积神经网络的显著性检测模型的语义信息，我们需要搭建一个轻量的显著性检测模型，且该模型需要有几点特性： 1) 可以从头训练而不依赖于ImageNet的预训练； 2) 足够简单，可以避免来自复杂模块的潜在混淆； 3) 有强大的多尺度表征能力以实现出色的检测效果。为了更好地研究显著性模型的复杂性和特征的需要，自适应的属性如自适应的计算资源分配也是需

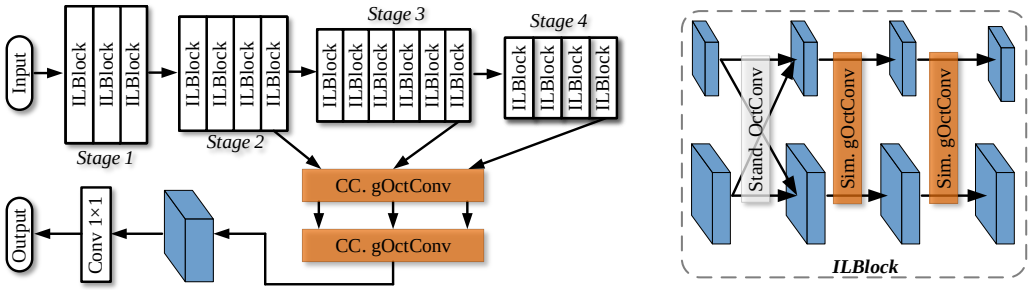

图 2. 我们的显著性检测模型的流程，其以高效的方式利用gOctConv提取阶段内和阶段间的多尺度特征。 Sim. gOctConv和CC. gOctConv分别表示简化的gOctConv实例和跨阶段融合的gOctConv实例。

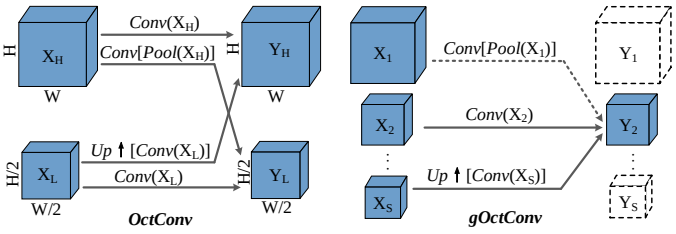

图 3. OctConv [6]最初是原始卷积单元的替代模块，其利用自同一阶段的高低分辨率特征为输入，同时通道数是固定的。我们的gOctConv允许存在任意数量的来自阶段内或跨阶段的不同分辨率的特征，同时通道是可学习的。

要的。然而，通过原生卷积层构建的模型很难满足所有的这些要求。因此，我们需要一个新的基础组件来设计需要的显著性检测模型。

因此我们提出一个灵活的自适应卷积层，即一般化的Octave卷积（gOctConv）。我们通过gOctConv搭建轻量的显著性检测模型，其结构简单且高效。我们提出的gOctConv足够灵活，他可以通过多种不同的形式组成显著性检测模型的不同部分。同时，他的自适应属性可以辅助对显著性模型的分析。

3.1 gOctConv

Fig. 3中原始的OctConv [6]最初是作为传统卷积模块的替代品，他可以实现一层内高低尺度间的卷积操作。然而，仅仅两个尺度不足以得到显著性检测任务需要的多尺度信息 (查看Tab. 6)。原始OctConv的每一个尺度中通道的数量是手动设置的，针对显著性模型调整通道数需要很大的成本。因此，我们提出gOctConv，如Fig. 3所示。该模块可以在阶段内和跨阶段的操作中利用任意数量的尺度，同时该模块有自适应的可学习的通道数。

作为gOctConv的一般化模块，gOctConv通过以下几方面提升原始gOctConv在显著性检测任务上的效果。首先，原始的gOctConv有着固定的结构，而gOctConv表示一类模块，其根据不同的显著性检测需求可以设计不同的具体模块。例如，通过关闭跨尺度的特征交互，gOctConv可以有更好的复杂度灵活性。任意数量的输入和输出尺度可以产生更大范围的多尺度表征。除了阶段内的特征，gOctConv可以处理任意尺度间的跨阶段特征。利用gOctConv的高度灵活性，我们可以设计一个高效但简单的显著性检测模型。其次，gOctConv在每个尺度中支持自适应的通道数。利用这一属性我们可以分析一些显著性模型的性质，例如模型复杂度和多尺度特征的需要。 gOctConv的实现细节和复杂性分析在补充材料中说明。

3.2 自适应的通道

通过利用我们提出的动态权重衰减机制来辅助剪枝算法，我们为gOctConv中的每一个尺度设计自适应可学习的通道。动态权重衰减机制维持通道间稳定的输出分布的同时引入稀疏性，这可以帮助剪枝算法消除冗余的通道而只损失微不足道的精度。

通过权重衰减机制引入稀疏性： 通常使用的权重衰减机制使卷积神经网络有更好的泛化性 [40], [91]。 Mehta等人 [61]说明了权重衰减会向卷积神经网络中引入稀疏性，这可以帮助减去不重要的参数。利用权重衰减机制训练，卷积神经网络中不重要的权重的值会接近于0。因此，权重衰减被广泛应用于剪枝算法以引入稀疏性 [28], [29], [45], [54], [55], [57]。权重衰减的一般实现是在损失函数中加入L2正则化，如下所示：

$$\mathbf{L} = \mathbf{L_0} + \lambda \sum \frac{1}{2} \mathbf{w}_i^2, \tag{1}$$

其中$\mathbf{L_0}$是针对特定任务的损失函数，$\mathbf{w}_i$是第i层的参数，λ是权重衰减机制的权重项。在反向传播期间，参数$\mathbf{w}_i$按以下方式更新：

$$\mathbf{w}_i \leftarrow \mathbf{w}_i - \nabla f_i(\mathbf{w}_i) - \lambda \mathbf{w}_i, \tag{2}$$

其中$\nabla f_i(\mathbf{w}_i)$表示梯度。 $\lambda \mathbf{w}_i$是权重衰减项，其只与参数本身有关。更大的λ会导致更强的稀疏性，同时不可避免地扩大不通道间参数的差异。 Fig. 4说明了差异较大的参数会导致通道间的输出分布不稳定。 Ruan等人 [13]揭示了输出差异更大的通道更有可能包含噪音，最终导致偏置化的表征。利用额外的模块和计算成本，注意力机制被广泛用于重新校准输出 [13], [36]。我们提出缓解通道间输出的差异性，同时在推

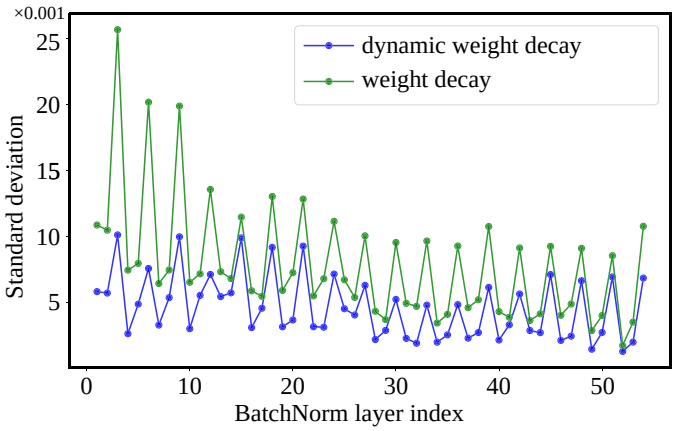

图 4. 使用/不使用动态权重衰减机制时，BatchNorm和激活层的输出特征的通道间的平均标准偏差。

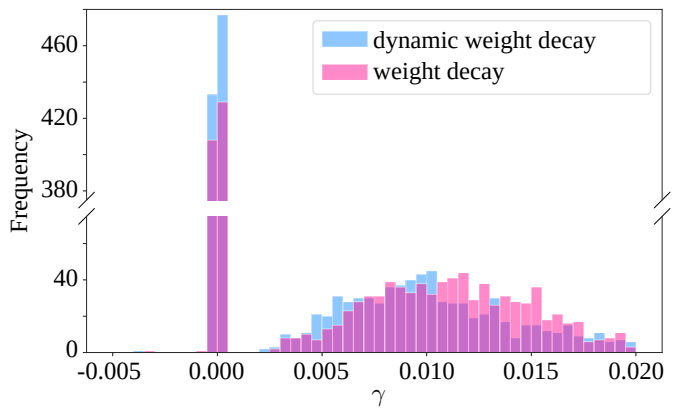

图 5. 使用/不使用动态权重衰减机制时，Eqn. (5)中γ的分布。

理时不产生额外的成本。我们认为，输出差异性大主要是由不加选择地抑制参数的衰减项导致的。因此我们提出通过特定通道的特定特征调整权重衰减。也就是说，在反向传播时，衰减项会根据特征通道的特征动态地变化。利用我们提出的动态权重衰减机制，参数的更新方式如下所示：

$$\mathbf{w}_i \leftarrow \mathbf{w}_i - \nabla f_i(\mathbf{w}_i) - \lambda_d \mathrm{M}(\mathbf{x}_i)\mathbf{w}_i, \tag{3}$$

其中λ_d是动态权重衰减的权重。 $\mathbf{x}_i$表示根据参数$\mathbf{w}_i$计算得到的特征。 $\mathrm{M}(\mathbf{x}_i)$是特征的度量，根据任务其有不同的定义。在本文中，我们的目标是根据特征稳定通道间参数的分布。因此，我们简单地使用全局平均池化（GAP） [49]作为特定通道的度量：

$$\mathrm{M}(\mathbf{x}_i) = \frac{1}{HW}\sum_{h=0}^{H}\sum_{w=0}^{W}\mathbf{x}_i(h,w), \tag{4}$$

其中H和W是特征图$\mathbf{x}_i$的高和宽。 如Fig. 4和Fig. 5， 利用GAP，动态权重衰减可以确保产生较高输出的参数被抑制，因此参数和输出的分布会更加紧凑且稳定。度量函数也可以被定义为其他形式以满足特定的任务，我们会在未来的工作中进行进一步研究。

Algorithm 1 Learning Channels for gOctConv with Dynamic Weight Decay

Require: The initial CSNet in which channels for all scales in gOctConvs are set. Input images X and corresponding label Y.

1: **for** each iteration $i \in [1, MaxIteration]$ **do**
2: Feed input X to the network to get the result $\hat{Y}$
3: Compute $Loss = criterion(\hat{Y}, Y)$
4: Compute metric for each channel using Eqn. (4)
5: Backward with dynamic weight decay using Eqn. (3).
6: **end for**
7: Eliminate redundant channels to get the learnable channels for each scale in gOctConv.
8: Train for several iterations to finetune remaining weights.

自适应可学习的通道: 现在，我们利用动态权重衰减机制配合剪枝算法消除冗余的参数，以此在gOctConv的每一个尺度中实现自适应的通道。与 [54]类似，我们使用BatchNorm层的权重指示通道的重要性。 BatchNorm [37]操作为：

$$y = \frac{x - \mathrm{E}(x)}{\sqrt{\mathrm{Var}(x) + \epsilon}}\gamma + \beta, \tag{5}$$

其中x和y是输入输出特征，$\mathrm{E}(x)$和 $\mathrm{Var}(x)$是均值和方差。ϵ则是一个小超参数，作用是避免方差为0。 γ和β则是可学习的参数。在训练期间，我们对γ使用动态权重衰减。BatchNorm和激活层后的输出特征被用来计算Eqn. (4)中的度量。 Fig. 5说明了重要的参数和冗余的参数间有明显的差异，不重要的参数会被几乎抑制到0（$\mathbf{w}_i < 1e-20$）。因此，我们可以简单地移除那些γ小于某一阈值的通道， gOctConv中可学习的通道因此可以实现。 gOctConv实现可学习通道的算法如算法1所示。

4 用于研究显著性检测模型语义信息的整体的轻量级模型

4.1 概述

目前有一些有着出色准确性和效率的基于卷积神经网络的显著性检测模型 [31], [46], [58], [67], [77], [79], [83], [95], [95], [96], [98], [99] 已经被提出。然而，显著性检测社区通常基于ImageNet的预训练骨干网络搭建模型，这限制了设计模型的空间并继承了大量参数（包含面向类别的表征）。甚至轻量化的分类骨干网络本身，如ResNet-18和MobileNet v2就包括11M和4.2M的参数。受益于极低的参数量（100K）和整体的设计，我们的CSNet可以直接从头训练而不需要ImageNet预训练。这样的设计使CSNet从ImageNet预训练模型中不必要的面向类别的信息中解放 [63], [88]。

为了分析模型中的每一个模块，我们使用gOctConv的不同实例构建一个简单但高效的显著性模型。如Fig. 2，我们整

体地设计特征提取器以及一个跨阶段融合模块以满足显著性检测任务的需要。他同时处理多个尺度的特征。特征提取器由我们提出的层内多尺度块（ILBlocks）组成。跨阶段融合部分处理来自特征提取器多个阶段的特征，以此获得高分辨的输出。 ILBlock和跨阶段融合部分都是由gOctConv的实例组成，强化了阶段内和跨阶段多尺度的表征能力，这种能力是显著性检测任务需要的。 CSNet的简单结构避免了复杂模块的潜在影响。而且，受益于gOctConv的自适应属性，我们可以更好地研究显著性检测模型的复杂性和对特征的要求。因此，CSNet是一个研究显著性检测模型语义信息的合适工具。

4.2 层内多尺度模块

ILBlock在一个阶段内强化特征的多尺度表征能力。 ILBlock利用gOctConvs引入多尺度的表征能力。借用分类模型的通用定义 [34], [35], [69], [97]，一个高度轻量化的显著性检测模型需要至少比现有的模型小10倍。而原始的OctConv需要至少60% MACC [6] 以实现和标准卷积相似的性能，这不足以设计一个高度轻量化的模型。为了节省计算成本，在每一层内集成不同尺度的特征是不必要的。因此，我们这里使用的gOctConv实例消除了跨尺度的操作，同时保留尺度内的操作，即简化版的gOctConv。在简化版的gOctConv中，每个输入通道对应到一个相同分辨率的输出通道，同时在每个尺度内利用depthwise操作进一步节省计算成本。相较于原始的OctConv，简化版的gOctConv仅仅需要$1/channel$ MACC。ILBlock由一个原始的OctConv和两个3 × 3简化gOctConv组成，如Fig. 2。原始的OctConv融合两个尺度的特征，然而简化的gOctConv在每个尺度内提取特征。一个模块内对每个尺度的特征的单独处理和互相融合会交替地进行。每个gOctConv的输出紧接着被BatchNorm [37]以及PReLu [25]处理。

4.3 跨阶段融合

显著性检测模型一般会通过通过在特征提取器的深层保持高分辨率的特征以保持高分辨率的输出，这不可避免地增加了计算冗余。另一个解决方案是通过搭建复杂的多层次聚合模块以融合高层次特征的语义信息和低层次特征的细节信息。多层次聚合模块的价值已经在许多任务上被广泛认可，例如边缘检测 [85]，目标检测 [23]，分类 [71]，以及显著性检测任务 [31]。然而，这些工作利用了很大的骨干网络。在显著性检测模块上，如何高效简洁地实现跨尺度融合仍然具有挑战性。本文中，我们旨在设计一个简单但高效的整体模型，其与显著性检测任务紧密相关。我们也需要一个简单但高效的多层次融合策略以分析显著性模型的语义信息。因此，我们简单地使用gOctConv的跨阶段融合的实例，以融合来自特征提取器不同阶段的特征并产生高分辨率的输出。一个跨阶段融合的1 × 1 gOctConv以每一个阶段最后一个卷积层的输出为输入（有着不同的尺度），并且构建一个跨尺度卷积以产生有着不同尺度的输出特征。为了细粒度地提取多尺度的特征，每个尺度的特征由一组不同扩张率的卷积并行处理。特征之后作为另一个跨阶段1 × 1 gOctConv的输入，产生有着最高分辨率的特征。另一个标准的1 × 1卷积层产生预测结果。

4.4 CSNet的实现细节

CSNet包括一个特征提取器和一个跨尺度融合部分。如Tab. 2，特征提取器由ILBlocks组成，同时根据特征的分辨率被分为4阶段。每个阶段分别有3、4、6、4个ILBlocks。最初，我们将第一个阶段的通道数设置为20，当分辨率降低时将通道数扩大一倍（除了最后两个阶段有相同的通道数）。每个gOctConv的通道可以进一步增大以扩大模型的容量。通道数扩大k倍的模型用CSNet-×k表示。可学习的通道之后通过自适应通道学习策略实现。给定一张(H, W)大小的图片，第一个ILBlock中的第一个gOctConv输出(H, W)和$(H/2, W/2)$两种分辨率的特征。两种尺度的特征被特质提取器并行地处理。我们使用“split-ratio”表示gOctConv中不同尺度的特征的通道数之比。通过调整ILBlocks中的split-ratio，我们可以构建有不同MACC的模型，用C_H/C_L表示。除非特别说明， ILBlock中不同尺度的通道数是均匀设置的。对于跨阶段的融合，仅仅每个阶段的输出特征被利用。每个阶段最后的ILBlock将不同尺度的特征融合到高分辨率的特征。跨尺度融合部分处理特征提取器每个阶段的特征，以获得高分辨率的输出。通过在效率和精度间权衡，来自最后三个阶段的特征被利用。在这一部分中我们同样利用gOctConv可学习的通道。跨阶段融合的具体配置见Tab. 1。

5 分析显著性检测模型

本节中，我们利用我们提出的CSNet回答关于基于卷积神经网络的显著性检测模型的语义信息的问题： 1) 显著性检测模型对类别信息敏感吗？ 2) 显著性检测模型的哪一部分与定位显著物体最相关？ 3) 显著性检测模型能否检测到未知类别（训练时没有见过的类别）的显著性物体？ 4) 显著性检测模型是否有与分类模型相同的复杂性？ 5) 显著性检测任务中特征提取器需要什么样的特征？ 6) ImageNet预训练在显著性检测模型训练中扮演着什么作用？

5.1 类别敏感性

在卷积神经网络出现之前，显著性检测模型被视为类别抽象的 [3], [38], [87]。研究者已经使用这些类别抽象的显著性检测模型来支持下游任务 [5], [22], [24], [32], [60], [80], [92]，例如弱监督的分割。随着卷积神经网络的流行，研究者提出了一些新的显著性检测模型，利用强大的ImageNet预训练网络提取特征。这些模型是否仍然对类别不敏感，可以被作为通用的特征？类别信息在这些模型中扮演什么样的角色。我

表 1
使用CSNet×1四个阶段的跨尺度融合部分的结构。

name	output feature size	config
gOctConv	[224×224×20, 112×112×40, 56×56×80, 28×28×80]	gOctConv, kernel size 1×1, dilation 1
Parallel DilatedConvs	[224×224× (1+1+1+1+1), 112×112×(2+2+2+2+5), 56×56×(5+5+5+5+6),28×28×(5+5+5+5+6)]	[DilatedConvs, kernel size 3×3 dilations [1, 2, 4, 8, 16]]× scales
gOctConv	224×224×70	gOctConv, kernel size 1×1, dilation 1
StandardConv	224×224×1	StandardConv, kernel size 1×1, dilation 1

表 2
CSNet×1中特征提取器的结构。

stage	output feature size	config [op, kernel size, stride]
stage1	[224×224×10, 112×112×10]	[OctConv 3×3, 1 gOctConv 3×3, 1 gOctConv 3×3, 1] ×1
		[OctConv 1×1, 1 gOctConv 3×3, 1 gOctConv 3×3, 1] ×2
stage2	[112×112×20, 56×56×20]	[OctConv 3×3, 2 gOctConv 3×3, 1 gOctConv 3×3, 1] ×1
		[OctConv 1×1, 1 gOctConv 3×3, 1 gOctConv 3×3, 1] ×3
stage3	[56×56×40, 28×28×40]	[OctConv 3×3, 2 gOctConv 3×3, 1 gOctConv 3×3, 1] ×1
		[OctConv 1×1, 1 gOctConv 3×3, 1 gOctConv 3×3, 1] ×5
stage4	[28×28×40, 14×14×40]	[OctConv 3×3, 2 gOctConv 3×3, 1 gOctConv 3×3, 1] ×1
		[OctConv 1×1, 1 gOctConv 3×3, 1 gOctConv 3×3, 1] ×3

们能否使用显著性作为通用的信息，以此减少特定域上数据的标注（如弱监督语义分割）？这些是非常重要的问题，但这方面的探索还很少。为了研究显著性检测模型的类别敏感性，我们使用相同的数据集在显著性物体检测和分类任务上训练CSNet模型。

5.1.1 数据准备

在表征学习中，数据的分布起着重要的作用。为了消除不同数据集不同数据分布的影响，我们在相同的数据集上训练显著性检测模型和分类模型，但两个任务分别使用显著物体掩膜和类别标签作为监督。因为图像类别标签相较于像素标注更容易获取，我们为现有的显著性物体检测数据集标注类别标签。也就是说，我们利用数据集DUTS-TR [74]，DUTS-TE [74] 和ECSSD [86]作为源数据集，将ImageNet的类别标签赋给每一张图片。这些显著性检测模型有不平衡的类别分布。因此，他们不能直接用于分析类别敏感性。在分类指标和显著性检测指标下分析模型需要不同的数据分布。因此我们针对两种任务分别使用不同的数据子集用于评测。

分类任务的数据： DUTS-TR数据集被用来作为训练集。DUTS-TE数据集的类别分布与DUTS-TR数据集非常相似，我们使用DUTS-TE数据集评测模型。在类别不平衡数据上训练的分类器可能过拟合到某些类别，使分类指标没有意义。为了解决这一问题，我们在DUTS-TR和DUTS-TE上选择10个类别，这些类别的图片数量是平衡的。训练集和测试集分别包含644张图片和150张图片。分类数据集的分布见 Fig. 6。

显著性检测的数据： 消除某个小类别对原始的显著性检测数据集几乎没有影响。为了加强类别在显著性检测数据集上的影响，我们根据ImageNet的WordTree [68]将图片合并到主类别，最终选取12个合并后的类别以及对应的大量图片。本文中大多数显著性检测实验结果是在ECSSD [86]数据集上得到的。因此，我们使用ECSSD数据集评测显著性检测模型。我们在显著性检测数据集上选择的类别的图片数量分布如Fig. 7。在图Fig. 8中，我们说明了即使是不平衡的类别分布，显著性检测模型仍然可以实现出色的效果。

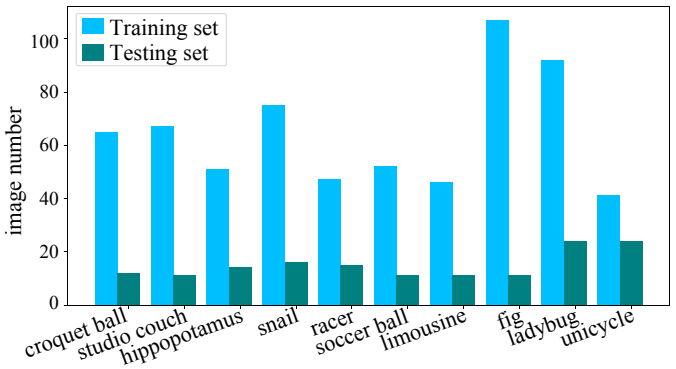

图 6. 分类数据集图像的分布。

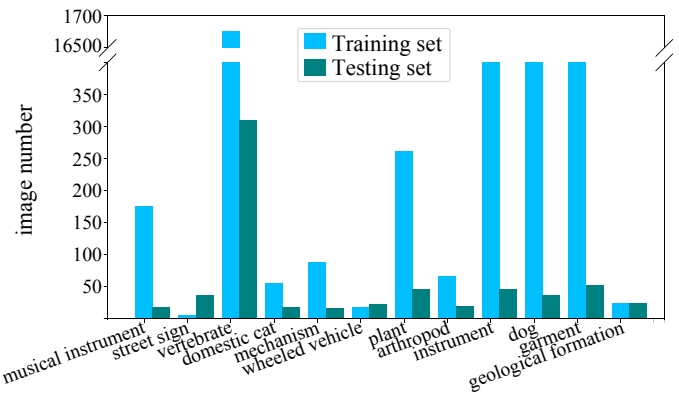

图 7. 显著性检测模型中被选中的图片的分布。

5.1.2 从显著性模型迁移到分类模型

设置： 迁移学习的基本思想是在源任务上预先训练模型，之后再目标任务上微调预训练模型，以此提升性能或者使收敛更快 [26]。

从ImageNet预训练模型迁移到许多下游任务已经被证明非常有效，例如深度估计，人群计数，边界框回归。并且在两个有着相似语义需求的任务间迁移，要比在两个有着不同语义的任务间迁移更有效。分类任务是对类别敏感的任务，并且分类准确率可以被用来衡量模型对类别的敏感性。通过将语义分割模型迁移到分类任务，我们已经证明了这一假设。为了研究显著性检测模型对类别的敏感性，我们首先在显著性检测任务上预训练模型，之后再分类任务上进行微调。通过修改显著性检测任务的预训练模型的不同部分，并且微调剩余的部分，我们可以发现显著性检测的哪一部分与分类最相关而哪一部分与显著性检测更相关。

CSNet是专门为显著性检测任务设计的，我们修改其跨阶段融合部分使其更适合分类任务。跨阶段融合部分以不同

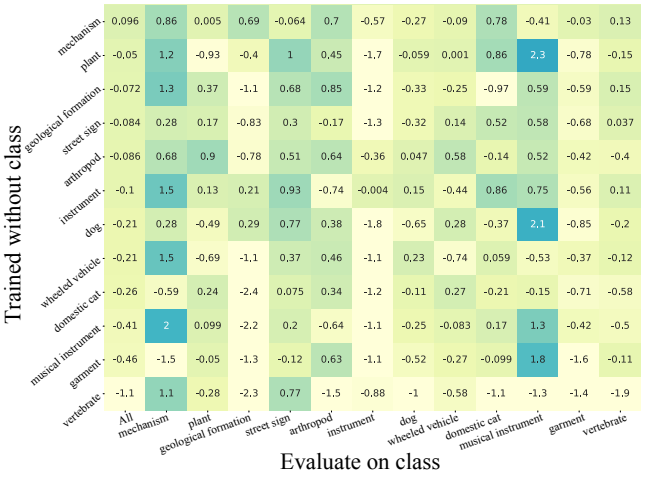

图 8. 训练显著性检测模型时移去某一类别时模型F-measure的相对变化。每一行显示了一个没有使用某一类别图像训练得到的模型在所有类别上的F-measure。在训练时移除某一类别并不会明显影响那一类的测试精度，这说明显著性检测模型对类别信息不敏感。

阶段的特征为输入，并且输出分辨率最低的特征，因为低分辨率的特征更适合分类任务 [26]。 CSNet最后的 1×1卷积层使用一个全局平均池化层和全连接层代替。通过这种修改，除了最后的预测层，CSNet的其余参数在训练显著性检测模型和分类模型时可以共享参数。我们使用top-1分类准确率作为分析类别敏感性的指标。我们进行如下两组实验： (1) *Cls-scratch*指模型仅仅使用图片类别标签从头训练分类网络。 (2) *Finetune-SOD* 指模型使用显著性标注预训练之后再在分类任务上微调。

如果*Finetune-SOD*模型实现了较*Cls-scratch*更差的结果，我们可以得出结论：“显著性检测模型对类别信息不敏感”。在微调期间， *Finetune-SOD*模型的一部分被微调而其他部分的参数保持不变。通过这种操作，我们可以发现显著性检测模型的哪些部分分别对分类任务和显著性检测任务更通用。例如，相较于只微调全连接层，如果微调显著性检测模型的阶段1只能得到很有限的提升，这说明阶段1的特征更通用而不是只和显著性检测任务密切相关。如Tab. 3，我们对阶段1到4以及跨阶段融合部分都进行了相应的研究。

通过将显著性检测模型迁移到分类任务，我们可以回答以下问题： 1) 是否显著性检测模型对类别信息敏感。 2) 显著性模型中哪些部分与任务更相关，哪些更加通用。

实验结果： Tab. 3展示了从显著性检测任务迁移到分类任务时模型的准确性。从头训练的分类模型实现了61.1%的准确率，然而仅仅微调显著性模型的全连接层实现了18.1%的准确率。这说明为显著性检测训练的模型几乎不需要根据类别信息决定显著性区域。为了说明模型的哪些部分与任务更相关以及哪些部分更通用，我们微调显著性检测模型的一部分，查看分类准确率的相对提升。微调跨阶段融合部分实现了明显的提升（30.2%），说明分类和显著性检测任务在模型这一

表 3
从显著性检测任务迁移到分类任务是，分类模型的top-1准确率。s1到s4和fuse分别表示Fig. 2中低阶段1到阶段4和融合部分。✓表示相应的参数被微调。

Setups	s1	s2	s3	s4	fuse	top1 acc.	gain
Finetune-SOD						18.1	-
	✓					21.5	3.4
		✓				30.9	12.8
			✓			32.9	14.8
				✓		36.2	18.1
					✓	48.3	30.2
	✓	✓	✓	✓	✓	58.4	40.3
Cls-scratch	✓	✓	✓	✓	✓	61.1	43.0

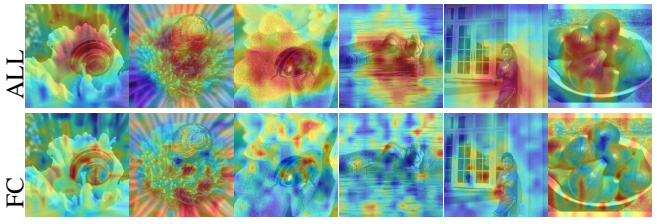

图 9. 微调所有阶段的显著性检测模型（ALL）和只微调全连接层的模型（FC）的类激活图的比较。微调所有阶段的模型的类激活图更准确地关注了分类需要的物体，只微调全连接层的模型没有关注到与类别有关的物体。

部分的特征差异更大。微调阶段1到4实现的提升依次逐渐增大，说明高层次的特征更加与任务相关， 2 低层次的特征更通用。

为了给出显著性检测模型对类别敏感性的直接证据，如Tab. 9，我们给出微调所有阶段（准确率58.4）和只微调全连接层（准确率18.1）的模型的类激活图（CAM）的比较[101]。微调所有阶段的模型的类激活图更准确地关注到了分类需要的物体，然而只微调全连接层的模型的类激活图并没有关注到与类别相关的物体。这种比较直接证明了显著性检测模型抛弃了类别相关的特征，因为他的类激活图不能定位到类别相关的区域。

5.1.3 当显著性模型遇上未知类别

设置： 我们现在从显著性检测任务的评价指标的角度，研究显著性检测模型的类别敏感性。能泛化到模型未知类别的物体是显著性检测模型的一个关键特性，因为这种特征是许多下游视觉任务的基础，例如图像检索 [24]，视觉追踪 [30]，摄影构图 [22]，以及图像质量评估 [80]。为了验证显著性检测模型的类别敏感性，我们在没有见过的类别上测试显著性模型的效果。给定一个没有使用某类图片训练得到的模型，如果该模型仍然可以在该类别的图像上监测到显著性物体，可以视为该模型对类别信息不敏感。也就是说，基于显著性检测的数据，我们可以得到一系列模型，其中每个模型的训练中某一类的图片都被移除。之后我们使用全部类别的数据分别测试每一个模型。我们比较这些模型和在全部数据上训练的基线模型的相对性能。移除训练数据会不可避免地导致性能的降低。如果在那些类别不在训练集中的图像上，模型的精度的降低不是最大的，则显著性检测模型可以视为对类别不敏感。

实验结果： 现在，我们测试显著性检测模型在未知类别上的效果。 Fig. 8说明了在训练时忽略某一类图片时F-measure的相对变化。所有的结果是三次运行结果的平均。在训练时移除某一类图片并不会严重影响在这一类上的测试精度，这说明显著性检测模型对类别不敏感。例如，移除类别vertebrate后，测试精度损失最大的是类别geological而不是vertebrate。如Fig. 7，显著性检测数据集中每一类的图片数量并不是均匀分布的。减少训练图片的数量对精度有明显的影响。例如，移除类别vertebrate导致了ECSSD数据集上最大的精度损失，因为该类别的训练图像是最多的。相反，移除street sign和plant这样的类别对精度的影响很有限，因为他们对应的图片数量相对更小。因此，我们认为显著性检测模型需要更多的多样化的训练图片来提升效果，而不是需要类别信息。

5.2 模型复杂性

设置： 不同任务的模型复杂性不需要一样。目前，基于卷积神经网络的显著性检测模型主要是基于分类骨干网络，例如VGGNet [70]和ResNet [26]。尽管这些模型在分类任务上是有效的，但这些模型较大的复杂性对显著性检测任务是不必要的。分类任务需要类别相关的特征，而显著性检测模型却不需要。我们从模型压缩的角度分析显著性检测任务和分类任务中模型的复杂性。因为我们提出的gOctConv配合动态权重衰减机制可以消除冗余的参数，我们可以对每一个任务的模型复杂性有清晰的认识。

实验结果： 我们使用Sec. 5.1.2中测试分类准确性的数据集作为训练集。因为训练集仅仅包含644张图片，标准的300轮的训练不足以是模型收敛。我们因此将训练轮数增加到3000。Tab. 4展示了不同任务基于CSNet×1.5的模型复杂性。相较于原始模型，显著性检测模型需要36%的参数，而分类任务需要65%的参数，这说明显著性检测模型需要更少的参数，因为他们几乎不需要类别信息。基于该观察，我们相信可以专门为显著性检测任务设计一个紧凑的显著性检测模型。

5.3 提取器需要的特征

特征提取器不同阶段的特征： 显著性检测作为像素预测任务，其应该产生高分辨率的输出。因此，利用特征提取器不同阶段的特征是现有显著性检测模型 [50], [52], [94]的共

表 4
基于CSNet×1.5，不同模型的复杂性。显著性检测模型相较于分类模型复杂性更低。

Setups	Full	SOD model	CLS model
Parms.	455K	167K	296K
MACC.	1.17G	0.70G	0.85G

表 5
在CSNet×2-L中使用提取器不同阶段的特征作为跨阶段融合部分的输入。

Stages	4	3 to 4	2 to 4	1 to 4
MACC	0.58G	0.66G	0.72G	1.29G
Parms.	134k	139k	141k	177k
F_β	90.0	91.0	91.6	91.8

识。利用更浅的阶段的特征会产生更高的输出分辨率，但会增大计算成本。这里我们探索是否使用更多更浅阶段的特征会提升模型的效果。如Fig. 2，跨阶段融合部分利用不同阶段的特征。我们现在使用不同阶段的特征进行消融实验。为了最小化初始通道数的影响，我们使用CSNet×2-L模型以及gOctConv已经学习到的通道。如Tab. 5，使用来自特征提取器更多阶段的特征会产生更好的效果和更大的计算成本。相较于使用阶段2到4的特征， CSNet×2-L使用阶段1到4的特征可以实现0.2%的提升(91.8 vs 91.6)同时多出 37K的参数（177k vs 141k）。因此，使用更多的浅层特征可以提升显著性检测模型的效果。在本文中，为了在精度和效率间达到平衡，如果没有另外的说明，我们选择使用最后三个阶段的特征作为跨阶段融合的输入。

特征尺度需求的可视化： 因为CSNet的特征提取器由gOctConv组成，我们可以利用gOctConv自适应通道的属性研究特征提取器对特征尺度的要求。我们在Fig. 10可视化了gOctConv学习到的通道数。可以看到随着网络越深，特征提取器物倾向于利用更多的低分辨率特征。在同一阶段内，模型在阶段的中部需要更多的高分辨率特征。而且，利用动态权重衰减训练得到的模型模型在不同层内的通道数量更稳定。相较于浅层，更深的层有更多的冗余特征。

5.4 ImageNet预训练

ImageNet预训练已经被证明对很多下游任务非常有效，例如深度估计、人群计数和边界框回归。在基于卷积神经网络的显著性检测模型中，由于其有效性，利用ImageNet的预训练模型作为特征提取器已经成为了一个默认的设置。如Fig. 11，我们针对轻量模型和大模型，比较了在是否有ImageNet预训练时模型的收敛速度。对于大模型即CSF+ResNet， ImageNet预训练模型帮助模型更快地收敛。如Sec. 5.1.2，模型的浅层包含更多的低层次特征。显著性

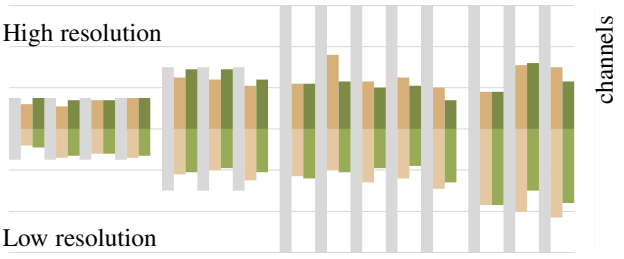

图 10. CSNet特征提取器的通道数可可视化。 Gray表示通道数固定的CSNet。黄色和绿色分别表示使用标准/动态权重衰减机制时的CSNet-L。水平轴表示特征提取器从浅层开始的ILBlock。

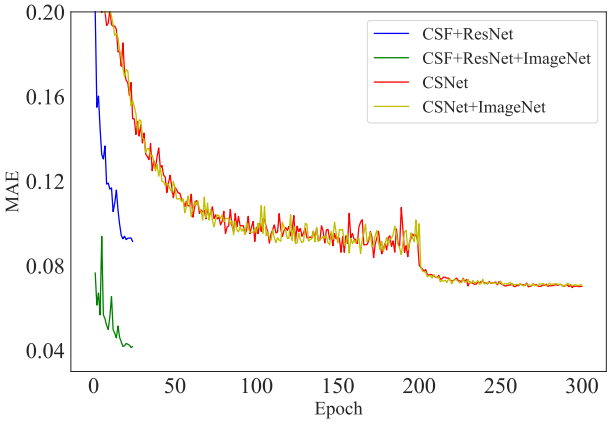

图 11. 使用/不使用ImageNet预训练时模型的测试MAE。

检测的大型SOTA模型需要ImageNet的预训练，因为大模型有很多的参数而ImageNet预训练模型浅层的低层次通用特征可以帮助大模型收敛。对于轻量化的的模型CSNet，基于ImageNet模型训练的模型的收敛速度和从头训练几乎没有差异。轻量化的模型太小了以至于ImageNet预训练不能帮助其加速收敛。我们在ImageNet上预训练CSNet的特征提取器，以观察预训练能否进一步体现模型的性能。如Tab. 7，ImageNet预训练的CSNet×1.5-L的精度与从头训练的模型相似。因此，ImageNet预训练对轻量化的显著性检测模型的作用是有限的。然而ImageNet预训练仍然可以帮助大模型收敛。

6 分析与消融

6.1 实现

Training: 我们的模型使用PyTorch实现。我们使用Adam优化器 [39]训练轻量模型，批大小为24，训练300轮。即使没有ImageNet预训练模型， CSNet仍然可以实现与基于ImageNet预训练 [26], [70]的大模型相当的效果。学习率初始为1e-4，在第200、250轮时除以10。在最后的20轮训练时，我们消除冗余的参数并微调模型以压缩模型并且实现可学习的通道。我们仅仅利用随机翻转作为数据增强。 gOctConv之后BatchNorm的权重衰减项替换为我们的提出的动态权重衰

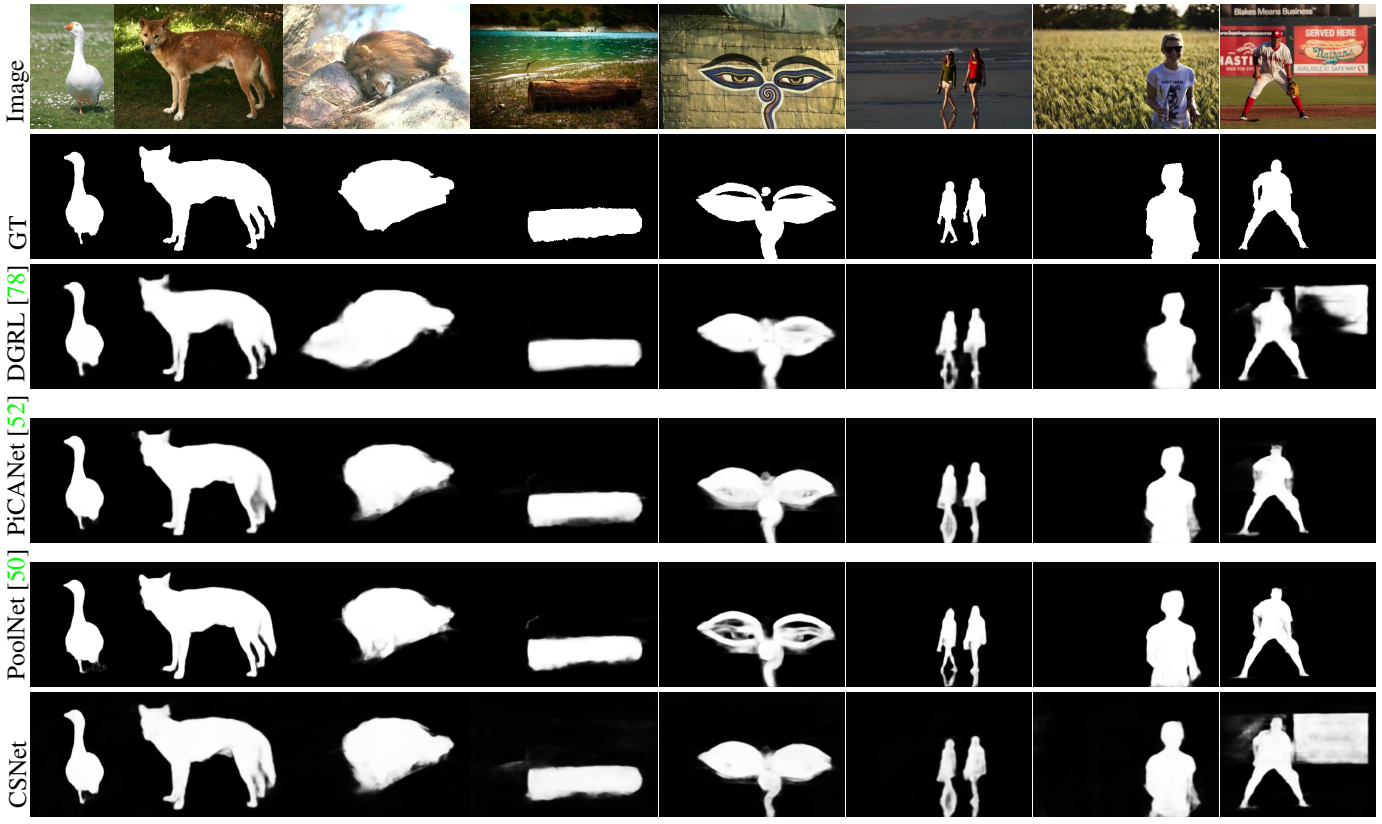

图 12. 我们提出的CSNet和现有的SOTA模型的可视化比较。最后一列显示了一个 CSNet未能成功处理的例子，我们认为这是极轻量模型的有限表征能力导致的。

减（默认权重为3，其他的参数权重默认设为5e-3）。对于基于预训练骨干网络的大模型，我们使用 [50]的实现训练模型。

数据集： MSRA 10K [10]， MSRA-B [53]， DUT-O [87]和HKU-IS [43] 是早期工作 [21], [44], [47]训练显著性检测器时使用的数据集，这些数据集要么太小要么缺乏多样性。我们仿照最近的工作 [50], [52], [77], [78], [96], [98] 的设置使用DUTS-TR [74]数据集训练我们的模型，并且在ECSSD [86], PASCAL-S [48], DUT-O [87]， HKU-IS [43]， SOD [64]，和DUTS-TE [74]数据集上测试我们的模型。在消融实验中，如果没有另外说明我们报告ECSSD数据集上的测试结果。在分析显著性模型的语义信息时，我们使用Sec. 5.1.1中描述的两个数据集。

评价指标： 我们使用maximum F-measure (F_β) [1] 和 MAE (M) [11] 作为我们的评价指标。轻量化模型的MACC根据224 × 224大小的图片计算。

6.2 性能分析

在本节中，我们首先使用固定的通道数测试我们提出的轻量化模型CSNet的效果。之后，利用我们测试可学习通道的CSNet的性能。我们说明了对基于卷积神经网络的显著性检测模型来说， ImageNet预训练不是不可避免的。 Fig. 12 显示了利用我们提出的轻量化模型实现的显著性检测的可视化结果。而且，我们将我们提出的跨阶段融合模块迁移到常用的大小骨干网络 [26] 以证明其跨阶段特征融合能力。

通道固定时的CSNet： 特征提取器仅仅由ILBlock组成。如Tab. 6， 当使用原始的OctConv代替gOctConv，特征提取器物的参数量和MACC分别增大8倍和7倍，同时精度提升很有限。巨大的模型复杂性差异说明了ILBlock中简化版的gOctConv的效率。 Tab. 6说明了不同高低分辨率特征通道比例下split-ratio下，模型的测试结果。多亏gOctConv的简化版本，提取器实现了很低的复杂度。受益于阶段内的多尺度表征和ILBlock中的低分辨率特征， extractor-3/1实现了0.4%的F-measure提升，而MACC为extractor-1/0的80%。跨阶段融合模块中的gOctConv强化了跨阶段的多尺度融合能力同时通过利用不同阶段的特征保证了高分辨率的输出。如Tab. 6， CSNet-5/5较extractor-3/1实现了1.4%的F-measure提升，同时MACC更少。即使在极端的情况下，在特征提取器中仅仅使用低分辨率特征的CSNet-0/1 实现的效果也与使用了高分辨率输出的extractor-1/0相当，而MACC只有extractor-1/0的 44%。然而，手动地设置不同分辨率特征通道数的比例split-ratio 可能只能在精度和计算成本间实现次优的平衡。为了进一步验证跨阶段融合（CSF）在大模型上的作用，我们将该部分加在常用的骨干网络之上，即ResNet [26]和 Res2Net [16]。 Tab. 7说明了 the ResNet+CSF实

表 6
在gOctConv中设置固定通道划分比例split-ratio时和使用可学习通道时，CSNet实现的效果。 CSNet：在gOctConv中设置固定通道划分比例split-ratio的模型。 Extractor：仅仅由ILBlocks组成的模型。 Vanilla：仅仅由OctConv组成的提取器（Extractor）。 CSNet-L：根据算法1实现可学习通道的模型。

Method		PARM.	MACC	$F_\beta \uparrow$	$M \downarrow$
Vanilla	5/5	1457K	3.31G	88.4	0.088
Extractor	1/0	180K	0.80G	88.2	0.088
	3/1	180K	0.64G	**88.6**	**0.085**
	5/5	180K	0.45G	88.1	0.086
	1/3	180K	0.30G	87.4	0.090
	0/1	180K	0.20G	86.4	0.095
CSNet	1/0	211K	0.91G	90.0	**0.076**
	3/1	211K	0.78G	89.9	0.077
	5/5	211K	0.61G	**90.1**	0.077
	1/3	211K	0.47G	89.2	0.082
	0/1	211K	0.35G	88.2	0.089
CSNet-L	×2	141K	0.72G	**91.6**	**0.066**
	×1	94K	0.43G	90.0	0.075

现了与ResNet+PoolNet相似的性能，然而参数和MACC只有 53%和 21%。其他模型如PoolNet消除骨干网络深层的降采样操作以产生高分辨率的输出。与其不同，gOctConv同时利用骨干网络不同阶段的高低分辨输出，产生高分辨率输出的同时节省许多计算成本。

利用可学习通道时CSNet： 我们进一步利用我们的动态权重衰减机制训练模型，同时如算法1得到可学习的通道。这样得到的模型我们称作CSNet-L。 Tab. 11说明了在剪枝算法的辅助下，我们提出的动态权重衰减机制可以将模型压缩到原始模型大小的18%，同时精度的损失是微不足道的。相较于手动调整的通道比例split-ratio， gOctConv通过模型压缩实现的可学习通道实现了更好的效率。如Tab. 6，压缩后的CSNet×2-L较CSNet-5/5实现了1.6%的提升，同时参数更少而MACC相当。 CSNet×1-L的效果与CSNet-5/5相当，而只有45%的参数和70%的MACC。表Tab. 7 说明了相较于参数量巨大的模型如SRM [77]和 Amulet [95]， CSNet-L系列模型可以实现与其相当的性能而只有～ 0.2%的参数。注意我们的轻量模型是从头训练的，而那些大型模型是基于ImageNet预训练模型的。我们的轻量模型和有着大量参数和MACC的SOTA模型的差距为～ 2%。利用新的技术，如representative batch normalization [17]和感受野搜索 [18]，可以进一步减小这种差距。

与轻量模型的比较： 据我们所知，我们是第一个设计极其轻量的显著性检测模型的工作。为了进行更详尽的分析，我们使用几个为其他任务（如分类和语义分割）设计的轻量模型进行显著性检测。所以模型的训练策略和我们的配置是一样的。当将分类模型迁移到显著性检测任务，全连接层使用1 ×

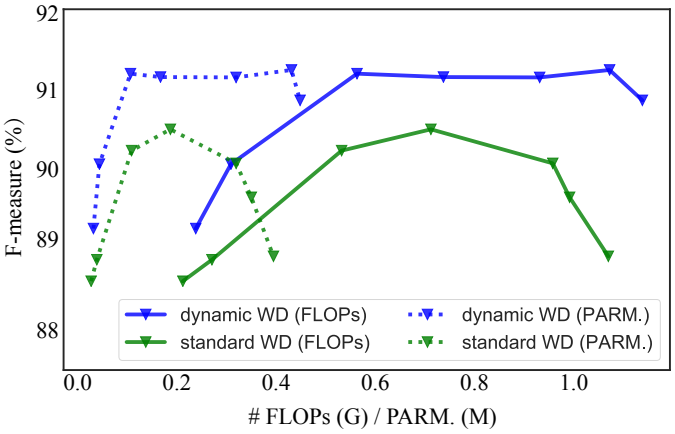

图 13. 在Eqn. (1)中不同的λ下，使用动态/标准的权重衰减时，压缩后的模型的精度和复杂性。应用不同程度的权重衰减会产生模型精度和稀疏性间的权衡。

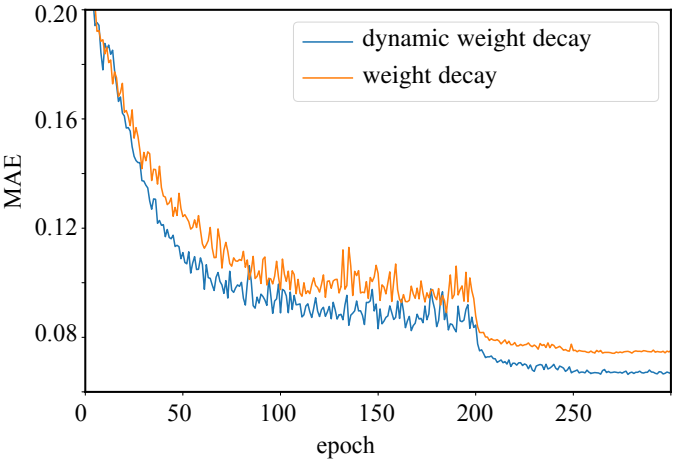

图 14. 使用/不使用动态权重衰减时模型的测试MAE。

1卷积代替。对于分割模型，输出层的输出通道由类别个数改为1。 Tab. 7说明了我们提出的模型相较于这些轻量模型有巨大的优势。

运行时间： CSNet是为显著性检测任务设计的轻量、高效率的模型。如Tab. 8，我们比较我们提出的模型和Tab. 7中其他模型的运行时间。运行时间在单核i7-8700K CPU上测试，图片大小224 × 224。我们提出的CSNet较大型模型快10倍。在相似的速度下，相较于为其他任务设计的模型，CSNet实现了6%的F-measure提升。然而，MACC和运行时间之间仍有差距，因为当前的深度学习框架并没有为原始的vanilla和gOctConv进行优化。

6.3 消融

动态权重衰减： 本节中，我们测试我们提出的动态权重衰减的效果。我们利用使用不同的标准权重衰减的权重项以平衡

表 7
与SOTA模型的复杂性和精度的比较。 +R和+R2分别指使用了ImageNet预训练的ResNet50 [26]和Res2Net50 [16]。之前的方法需要ImageNet的预训练模型，我们的方法是从头训练的。

Model	Complexity		ECSSD		PASCAL-S		DUT-O		HKU-IS		SOD		DUTS-TE	
	#PARM.	MACC	F_β	M	F_β	M	F_β	M	F_β	M	F_β	M	F_β	M
ELD [21]$_{CVPR'16}$	43.15M	17.63G	.865	.981	.767	.121	.719	.091	.844	.071	.760	.154	-	-
DS [47]$_{TIP'16}$	134.27M	211.28G	.882	.122	.765	.176	.745	.120	.865	.080	.784	.190	.777	.090
DCL [44]$_{CVPR'16}$	-	-	.896	.080	.805	.115	.733	.094	.893	.063	.831	.131	.786	.081
RFCN [76]$_{ECCV'16}$	19.08M	64.95G	.898	.097	.827	.118	.747	.094	.895	.079	.805	.161	.786	.090
DHS [51]$_{CVPR'16}$	93.76M	25.82G	.905	.062	.825	.092	-	-	.892	.052	.823	.128	.815	.065
MSR [42]$_{CVPR'17}$	-	-	.903	.059	.839	.083	.790	.073	.907	.043	.841	.111	.824	.062
DSS [31]$_{PAMI'19}$	62.23M	276.37G	.906	.064	.821	.101	.760	.074	.900	.050	.834	.125	.813	.065
NLDF [58]$_{CVPR'17}$	35.48M	57.73G	.903	.065	.822	.098	.753	.079	.902	.048	.837	.123	.816	.065
UCF [95]$_{CVPR'17}$	29.47M	146.42G	.908	.080	.820	.127	.735	.131	.888	.073	.798	.164	.771	.116
Amulet [94]$_{ICCV'17}$	33.15M	40.22G	.911	.062	.826	.092	.737	.083	.889	.052	.799	.146	.773	.075
GearNet [33]$_{CoRR'17}$	-	-	.923	.055	-	-	.790	.068	.934	.034	.853	.117	-	-
PAGR [96]$_{CVPR'18}$	-	-	.924	.064	.847	.089	.771	.071	.919	.047	-	-	.854	.055
SRM [77]$_{ICCV'17}$	53.14M	36.82G	.916	.056	.838	.084	.769	.069	.906	.046	.840	.126	.826	.058
DGRL [78]$_{CVPR'18}$	161.74M	191.28G	.921	.043	.844	.072	.774	.062	.910	.036	.843	.103	.828	.049
PiCANet [52]$_{CVPR'18}$	47.22M	54.05G	.932	.048	.864	.075	.820	.064	.920	.044	.861	.103	.863	.050
PoolNet [50]$_{CVPR'19}$	68.26M	88.89G	.940	.042	.863	.075	.830	.055	.934	.032	.867	.100	.886	.040
Light-weight models designed for other tasks:														
Eff.Net [72]$_{ICML'19}$	8.64M	2.62G	.828	.129	.739	.158	.696	.129	.807	.116	.712	.199	.687	.135
Sf.Netv2 [59]$_{ECCV'18}$	9.54M	4.35G	.870	.092	.781	.127	.720	.100	.853	.078	.779	.163	.743	.096
ENet [66]$_{CoRR'16}$	0.36M	0.40G	.857	.107	.770	.138	.730	.109	.839	.094	.741	.183	.730	.111
CGNet [84]$_{CoRR'18}$	0.49M	0.69G	.868	.099	.784	.130	.727	.108	.849	.088	.772	.168	.742	.106
DABNet [41]$_{BMVC'19}$	0.75M	1.03G	.877	.091	.790	.123	.747	.094	.862	.078	.778	.157	.759	.093
ESPNetv2 [62]$_{CVPR'19}$	0.79M	0.31G	.889	.081	.795	.119	.760	.088	.872	.069	.780	.157	.765	.089
BiseNet [89]$_{ECCV'18}$	12.80M	2.50G	.894	.078	.817	.115	.762	.087	.872	.071	.796	.148	.778	.084
Ours:														
CSF+R	36.37M	18.40G	.940	.041	.866	.073	.821	.055	.930	.033	.866	.106	.881	.039
CSF+R2	36.53M	18.96G	.947	.036	.876	.068	.833	.055	.936	.030	.870	.098	.893	.037
CSNet×1-L	94K	0.43G	.900	.075	.819	.110	.777	.087	.889	.065	.809	.149	.799	.082
CSNet×1.5-L	118K	0.63G	.912	.070	.831	.105	.783	.082	.893	.062	.808	.139	.809	.076
CSNet×1.5-L$_{ImageNet}$	124K	0.63G	.911	.070	.835	.103	.781	.084	.898	.060	.818	.141	.810	.077
CSNet×2-L	141K	0.72G	.916	.066	.835	.102	.792	.080	.899	.059	.825	.137	.819	.074

表 8
使用224 × 224大小的图片为输入时，模型在单核i7-8700K CPU上的运行时间。

Method	MACC (G)	Run-time (ms)
PiCANet [51]	54.06	2850.2
PoolNet [50]	88.89	997.3
ENet [66]	0.40	89.9
ESPNetv2 [62]	0.31	186.3
CSNet×1	0.61	135.9
CSNet×1-L	0.43	95.3

模型的精度和稀疏性，同时使动态权重衰减的权重保持不变。我们将我们提出的动态权重衰减机制插入BatchNorm层的参数，同时在剩余参数上使用标准的权重衰减以实现公平的比较。设置Eqn. (1)中不同的λ，Fig. 13 显示了使用动态/标准权重衰减后，压缩后的模型的精度和复杂性。对BatchNorm层的动态权重衰减， Eqn. (3)的λ_d默认设置为3。通过动态权重衰减训练得到的模型在相同的复杂性下有更好的效果。而且，基于动态权重衰减的模型的性能对模型复杂性更不敏感。按Sec. 3.2描述的方法，根据Eqn. (5)中γ的绝对值，我们消除了冗余的通道。 Fig. 5展示了在有/没有动态权重衰减时，模型中γ的分布。通过根据特征抑制参数，动态权重衰减强化了模型的稀疏性。 Fig. 4 显示了使用/不使用动态权重衰减训练的模型在BatchNorm层和激活层后输出的通道间的标准偏差。由于参数产生了稳定的输出分布，基于动态权重衰减的模型的特征更加稳定。 Fig. 14展示了使用/不使用动态权重衰减时每个模型的测试MAE。使用动态权重衰减机制训练会产生更

表 9
将动态权重衰减机制集成进剪枝算法。 Standard/Dynamic指标准/动态权重衰减机制。

	PARM.	MACC	F_β	M
Pruning Filters [45]				
Standard	227K	0.69G	88.7	0.080
Dynamic	226K	0.69G	89.4	0.078
Geometric-Median [28]				
Standard	227K	0.70G	88.7	0.083
Dynamic	226K	0.68G	89.6	0.082

低的MAE。

固定的剪枝率/阈值： 与 [54]一致，我们使用BatchNorm的参数作为通道重要性的度量。我们修改剪枝算法 [54]，使用固定的阈值消除通道而不是使用固定的剪枝率（pruning ratio）。Tab. 10展示了使用固定的阈值会实现更好的效果，相较于固定的剪枝率有更少的参数。产生这种结果的原因是不同的层需要不同数量的通道。因此使用阈值修建在每一层会得到不同的通道数。如Fig. 5，模型中较大的参数和接近于0的参数间有明显的距离。在这之间使用任意阈值对模型的最终结果都几乎没有影响。

将动态权重衰减机制集成进剪枝算法： 我们默认使用 [54]中的剪枝算法消除冗余的参数。因为我们提出的动态权重衰减机制关注于在引入稀疏性的同时保持通道间稳定和紧凑的参数分布，他可以插入常用的关注于寻找不重要参数的剪枝算法。因此，我们将动态权重衰减机制集成进一些剪枝算法，如Tab. 9。所有的配置保持不变，除了使用动态权重衰减机制代替标准的动态权重衰减。剪枝算法 [28], [45]配合动态动态权重衰减，可以在相似的参数量下实现更好的效果。

剪枝率 & 通道宽度： 为了学习更多有用的特征，一个通道较宽的模型是被需要的。我们线性地扩展gOctConv的通道数来强化初始模型的容量。剪枝率被定义为被修剪掉的部分和整个模型的复杂度之比。 Tab. 11是初始通道宽度不同时CSNet的修剪率。初始模型中gOctConv的通道比例split-ratio设置为5/5。更大的初始通道宽度会产生更好的效果，这与预期一致。因为随着着初始宽度增加，修剪后的模型的复杂性仅仅有很小的增长。修剪后的模型的质量取决于初始模型的大小。而且，受益于动态权重衰减带来的稳定的分布，相较于初始模型。压缩后的模型有相似甚至更好的效果。

7 结论和讨论

本文中，通过抛弃分类骨干网络以及通过动态权重衰减机制减少冗余的表征，我们为显著性检测任务提出了一个极其轻量化的整体模型。动态权重衰减机制保持通道间稳定的参数分布，并且在训练时强化参数的稀疏性，最终减少了80%的参数而损失的精度是微不足道的。在主流显著目标检测数据集上，我们提出的CSNet实现了与大型模型相似的性能而只有～ 0.2%的参数。基于我们提出的CSNet，我们揭示了基于卷积神经网络的显著性检测模型的一些属性，包括 1) 显著性检测模型对类别不敏感，并且其检测到的显著性物体是通用的且与类别无关的， 2) ImageNet预训练对于显著性检测模型的训练是不必要的以及 3) 相较于分类模型，显著性检测模型需要更少的参数。

我们有两个主要的贡献：分析显著性检测模型的语义信息和轻量化的整体显著性检测模型是互相依赖的。我们有意设计的整体的CSNet可以被用来分析显著性检测模型的语义信息。 CSNet是从头训练的，因此可以摆脱ImageNet预训练模型的潜在影响，这是我们分析显著性检测模型对类别依赖的基础。而且，自适应的属性和简单但高效的CSNet结构也有利于分析显著性检测模型的复杂性和对特征的需求。从另一个角度，对显著性检测模型的分析支持了显著性检测模型的设计原理。我们的分析证明了显著性检测模型不需要类别信息，因此我们可以抛弃ImageNet预训练的骨干网络以减少大量的冗余。并且我们可以针对显著性检测任务设计整体的网络，而不是在分类骨干网络上增加额外的模块来弥补两种任务的差异。

未来的研究应该关注于用其他角度分析显著性检测模型，特别是更多样的显著性检测模型结构，以及搭建甚至更高效率的模型。为了促进进一步的工作，我们的代码开源于https://mmcheng.net/sod100k/。

参考文献

[1] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1597–1604, 2009.

[2] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li. Salient object detection: A survey. *Computational Visual Media*, 5(2):117–150, 2019.

[3] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE transactions on image processing*, 24(12):5706–5722, 2015.

[4] S. Chen, X. Tan, B. Wang, and X. Hu. Reverse attention for salient object detection. In *European Conference on Computer Vision*, 2018.

[5] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: Internet image montage. *ACM T. Graph.*, 28(5):124:1–10, 2009.

[6] Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, and J. Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Int. Conf. Comput. Vis.*, 2019.

[7] M.-M. Cheng, S.-H. Gao, A. Borji, Y.-Q. Tan, Z. Lin, and M. Wang. A highly efficient model to study the semantics of salient object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[8] M.-M. Cheng, Q.-B. Hou, S.-H. Zhang, and P. L. Rosin. Intelligent visual media processing: When graphics meets vision. *Journal of Computer Science and Technology*, 32(1):110–121, 2017.

[9] M.-M. Cheng, N. Mitra, X. Huang, and S.-M. Hu. Salientshape: group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014.

表 10
使用固定的阈值/剪枝率时修剪CSNet×2-L模型。

	Fixed threshold										Fixed ratio	
threshold/ratio	1e-2	8e-3	6e-3	5e-2	3e-2	1e-3	1e-5	1e-10	1e-15	1e-20	38%	51%
Parms. (K)	55.7	79.4	105.0	118.5	135.9	136.2	139.9	140.5	140.8	140.8	300.0	400.0
F_β	37.8	39.3	57.1	82.5	91.2	91.2	91.5	91.5	91.5	91.6	87.7	91.1

表 11
使用不同初始通道宽度时， CSNet的压缩率。剪枝率被定义为被裁减的部分和整个CSNet模型的复杂度之比。

Width	Prune	×1	×1.2	×1.5	×1.8	×2.0
Parms	N	211K	298K	455K	645K	788K
	Y	94K	109K	118K	134K	141K
Ratio		55%	63%	74%	79%	82%
MACC	N	0.61G	0.82G	1.17G	1.58G	1.87G
	Y	0.43G	0.52G	0.63G	0.71G	0.72G
Ratio		30%	37%	46%	55%	61%
F_β	N	90.0	90.7	91.1	91.2	91.5
	Y	90.0	90.7	91.2	91.3	91.6

[10] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE T. Pattern Anal. Mach. Intell.*, 37(3):569–582, 2015.

[11] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook. Efficient salient region detection with soft image abstraction. In *Int. Conf. Comput. Vis.*, pages 1529–1536, 2013.

[12] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995.

[13] R. Dongsheng, W. Jun, and Z. Nenggan. Linear context transform block. *arXiv preprint arXiv:1909.03834*, 2019.

[14] D.-P. Fan, M.-M. Cheng, J.-J. Liu, S.-H. Gao, Q. Hou, and A. Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *Eur. Conf. Comput. Vis.*, September 2018.

[15] M. Feng, H. Lu, and E. Ding. Attentive feedback network for boundary-aware salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[16] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr. Res2net: A new multi-scale backbone architecture. *IEEE T. Pattern Anal. Mach. Intell.*, 2020.

[17] S.-H. Gao, Q. Han, D. Li, P. Peng, M.-M. Cheng, and P. Peng. Representative batch normalization with feature calibration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[18] S.-H. Gao, Q. Han, Z.-Y. Li, P. Peng, L. Wang, and M.-M. Cheng. Global2local: Efficient structure search for video action segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[19] S.-H. Gao, Z.-Y. Li, M.-H. Yang, M.-M. Cheng, J. Han, and P. Torr. Large-scale unsupervised semantic segmentation, 2021.

[20] S.-H. Gao, Y.-Q. Tan, M.-M. Cheng, C. Lu, Y. Chen, and S. Yan. Highly efficient salient object detection with 100k parameters. In *Eur. Conf. Comput. Vis.*, 2020.

[21] L. Gayoung, T. Yu-Wing, and K. Junmo. Deep saliency with encoded low level distance map and high level features. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[22] Q. Han, K. Zhao, J. Xu, and M.-M. Cheng. Deep hough transform for semantic line detection. In *Eur. Conf. Comput. Vis.*, 2020.

[23] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015.

[24] J. He, J. Feng, X. Liu, C. Tao, and S. F. Chang. Mobile product search with bag of hash bits and boundary reranking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Int. Conf. Comput. Vis.*, pages 1026–1034, 2015.

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

[27] Y. He, G. Kang, X. Dong, Y. Fu, and Y. Yang. Soft filter pruning for accelerating deep convolutional neural networks. In *Int. Jt. Conf. Artif. Intell.*, 2018.

[28] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4340–4349, 2019.

[29] Y. He, X. Zhang, and J. Sun. Channel pruning for accelerating very deep neural networks. In *Int. Conf. Comput. Vis.*, pages 1389–1397, 2017.

[30] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *International Conference on Machine Learning (ICML)*, 2015.

[31] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. *IEEE T. Pattern Anal. Mach. Intell.*, 41(4):815–828, 2019.

[32] Q. Hou, P.-T. Jiang, Y. Wei, and M.-M. Cheng. Self-erasing network for integral object attention. In *NeurIPS*, 2018.

[33] Q. Hou, J. Liu, M.-M. Cheng, A. Borji, and P. H. Torr. Three birds one stone: a unified framework for salient object segmentation, edge detection and skeleton extraction. *arXiv preprint arXiv:1803.09860*, 2018.

[34] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al. Searching for mobilenetv3. *arXiv preprint arXiv:1905.02244*, 2019.

[35] 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 arXiv:1704.04861*, 2017.

[36] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[37] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

[38] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE T. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, 1998.

[39] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2014.

[40] A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. In *Adv. Neural Inform. Process. Syst.*, pages 950–957, 1992.

[41] G. Li and J. Kim. Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. In *Brit. Mach. Vis. Conf.*, 2019.

[42] G. Li, Y. Xie, L. Lin, and Y. Yu. Instance-level salient object segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, July 2017.

[43] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2015.

[44] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2016.

[45] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. In *Int. Conf. Learn. Represent.*, 2016.

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

[47] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE T. Image Process.*, 25(8):3919 – 3930, Aug 2016.

[48] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2014.

[49] M. Lin, Q. Chen, and S. Yan. Network in network. In *Int. Conf. Learn. Represent.*, 2013.

[50] 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.

[51] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2016.

[52] N. Liu, J. Han, and M.-H. Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2018.

[53] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Y. Shum. Learning to detect a salient object. *IEEE Trans Pattern Anal Mach Intell*, 33(2):353–367, 2011.

[54] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning efficient convolutional networks through network slimming. In *Int. Conf. Comput. Vis.*, pages 2736–2744, 2017.

[55] Z. Liu, H. Mu, X. Zhang, Z. Guo, X. Yang, T. K.-T. Cheng, and J. Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *Int. Conf. Comput. Vis.*, 2019.

[56] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3431–3440, 2015.

[57] J.-H. Luo, J. Wu, and W. Lin. Thinet: A filter level pruning method for deep neural network compression. In *Int. Conf. Comput. Vis.*, pages 5058–5066, 2017.

[58] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin. Non-local deep features for salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, July 2017.

[59] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Eur. Conf. Comput. Vis.*, pages 116–131, 2018.

[60] R. Margolin, L. Zelnik-Manor, and A. Tal. Saliency for image manipulation. *The Visual Computer*, 29(5):381–392, 2013.

[61] D. Mehta, K. I. Kim, and C. Theobalt. On implicit filter level sparsity in convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 520–528, 2019.

[62] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi. Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9190–9200, 2019.

[63] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks, 2015.

[64] 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.*, pages 49–56, June 2010.

[65] Y. Pang, X. Zhao, L. Zhang, and H. Lu. Multi-scale interactive network for salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9413–9422, 2020.

[66] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[67] Y. Piao, W. Ji, J. Li, M. Zhang, and H. Lu. Depth-induced multi-scale recurrent attention network for saliency detection. In *Int. Conf. Comput. Vis.*, October 2019.

[68] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

[69] 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.*, pages 4510–4520, 2018.

[70] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, 2014.

[71] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5693–5703, 2019.

[72] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

[73] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng. Salient object detection: A discriminative regional feature integration approach. *Int. J. Comput. Vis.*, 123(2):251–268, 2017.

[74] 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.

[75] L. Wang, H. Lu, R. Xiang, and M. H. Yang. Deep networks for saliency detection via local estimation and global search. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[76] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Eur. Conf. Comput. Vis.*, pages 825–841, 2016.

[77] 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.*, Oct 2017.

[78] 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.*, June 2018.

[79] W. Wang, S. Zhao, J. Shen, S. C. H. Hoi, and A. Borji. Salient object detection with pyramid attention and salient edges. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[80] X. Wang, X. Liang, B. Yang, and F. W. Li. No-reference synthetic image quality assessment with convolutional neural network and local image saliency. *Computational Visual Media*, 5(2):193–208, 2019.

[81] J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian. Label decoupling framework for salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13025–13034, 2020.

[82] J. M. Wolfe and T. S. Horowitz. What attributes guide the deployment of visual attention and how do they do it? *Nature reviews neuroscience*, 5(6):495–501, 2004.

[83] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[84] T. Wu, S. Tang, R. Zhang, and Y. Zhang. Cgnet: A light-weight context guided network for semantic segmentation. *arXiv preprint arXiv:1811.08201*, 2018.

[85] S. Xie and Z. Tu. Holistically-nested edge detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1395–1403, 2015.

[86] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2013.

[87] 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.*, pages 3166–3173, 2013.

[88] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8715–8724, 2020.

[89] 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.*, pages 325–341, 2018.

[90] Y. Zeng, P. Zhang, J. Zhang, Z. Lin, and H. Lu. Towards high-resolution salient object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[91] G. Zhang, C. Wang, B. Xu, and R. Grosse. Three mechanisms of weight decay regularization. In *Int. Conf. Learn. Represent.*, 2019.

[92] G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin. A shape-preserving approach to image resizing. *Computer Graphics Forum*, 28(7):1897–1906, 2009.

[93] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang. A bi-directional message passing model for salient object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[94] 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.*, Oct 2017.

[95] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin. Learning uncertain convolutional features for accurate saliency detection. In *Int. Conf. Comput. Vis.*, pages 212–221. IEEE, 2017.

[96] 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.*, June 2018.

[97] 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.*, pages 6848–6856, 2018.

[98] J.-X. Zhao, J.-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.*, October 2019.

[99] K. Zhao, S.-H. Gao, W. Wang, and M.-M. Cheng. Optimizing the f-measure for threshold-free salient object detection. In *Int. Conf. Comput. Vis.*, October 2019.

[100] X. Zhao, Y. Pang, L. Zhang, H. Lu, and L. Zhang. Suppress and balance: A simple gated network for salient object detection. In *Eur. Conf. Comput. Vis.*, pages 35–51. Springer, 2020.

[101] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2921–2929, 2016.

[102] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2814–2821, 2014.