隐蔽目标检测

范登平，季葛鹏，程明明，邵岭

摘要—本文提出了第一个针对隐蔽目标检测（Concealed Object Detection, COD）的系统研究，这项任务旨在识别那些视觉上嵌入周围环境的目标。隐蔽目标与其周边环境之间高度的内在相似性使得 COD 任务比传统的目标检测/分割任务更具挑战性。为了更好地理解这项任务，本文收集了一个名为 **COD10K** 的大规模数据集，包由 10,000 张图像组成，隐蔽目标来自各种真实场景中的 78 个类别对象。此外，我们还提供了丰富的标签，包括目标类别、目标边界、具有挑战性的属性、目标级别掩膜和实例级别掩膜。本文构建的 *COD10K* 是迄今为止最大的具有最丰富标签的 COD 数据集，这有利于全面地理解隐蔽对象，甚至可以用于推进其他诸如检测、分割和分类等视觉任务的发展。受到动物在野生环境下狩猎过程的启发，本文进一步设计了一个简单但鲁棒的基线模型，名为搜索识别网络（***SINet***）。没有花里胡哨的技巧，*SINet* 在所有的测试数据集上超过了 12 个前沿的基线对比模型，这种鲁棒且通用的网络结构可成为未来 COD 领域发展中的催化剂。最后，本文总结了许多有趣的发现并强调了多个潜在应用领域以及未来的研究方向。为了激发这一新兴领域的研究，本文的代码、数据集和在线演示系统可以在项目主页上找到：http://mmcheng.net/cod。

Index Terms—隐蔽目标检测、伪装目标检测、COD、数据集、基准。

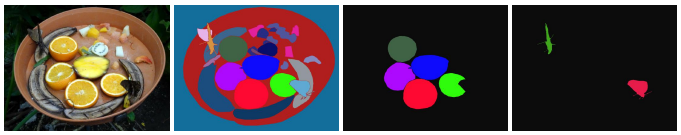
1 前沿

您能够在十秒内找到图. 1 中所对应的每一张图片的隐蔽物体吗？生物学家把这种现象称为背景匹配伪装（*BMC*）[3]，用于表示一个或者多个生物为了防止被发现 [4]，尝试将其颜色与周围环境“无缝地”匹配。感官生态学家 [5] 发现，这种 BMC 策略通过欺骗观察者的视觉感知系统而起作用。自然，解决隐蔽目标检测（**COD**[1]）需要丰富的视觉感知 [7] 知识。理解 COD 任务本身不仅具有科学意义，而且它在众多基础领域中具有应用价值，包括：计算机视觉（例如：搜救工作或者稀有物种的发现）、医学（例如：息肉分割 [8]、肺部感染区域分割 [9]）、农业（例如：蝗虫监测以防止入侵）和艺术（例如：娱乐艺术 [10]）。

图. 2 提供了一般、显著和隐蔽目标检测任务的示例。目标与非目标之间高度的内在相似性使得 COD 任务比传统的目标分割/检测任务 [11], [12], [13] 更具挑战性。近年来，目标检测任务得到了越来越多的关注，但与隐蔽目标检测任务相关的研究却很少见，主要是因为缺乏超大规模的数据集和诸如 Pascal-VOC [14]、ImageNet [15]、MS-COCO [16]、ADE20K [17] 和 DAVIS [18] 这样标准的基准评测。

本文呈现了第一个基于深度学习的隐蔽目标检测任务的完备研究，从隐蔽这一新视角来看待目标检测任务。

图 1. **背景匹配伪装（BMC）示例**。左/右子图中分别隐藏了七/六只小鸟。答案以彩图形式展示在图. 27 中。



(a) 输入图像 (b) 一般目标 (c) 显著目标 (d) 隐蔽目标

图 2. **任务关系图**。给定一张输入图像 (a)，上图分别展示了三种任务的真值图：(b) 全景分割 [6]（检测包括东西和目标在内的**一般**目标 [19], [20]）实例级别的 (c) **显著**目标检测 [12], [21] 和 (d) 本文的**隐蔽**目标检测任务，其目标是检测出那些与自然栖息地中有着相似模式的物体。在图例中，两只蝴蝶的边缘与香蕉混合在一起，使其难以辨识。

1.1 贡献

本文的主要贡献如下：

1) **COD10K 数据集：**确定上述目标后，本文精心地收集了一个用于隐蔽目标检测的大规模数据集 *COD10K*。该数据集包含 10,000 张图像样本，囊括了 78 类目标类别，包括*陆生动物类*、*两栖动物类*、*飞行动物类*、*水生动物类*等。所有的隐蔽图像样本基于类别、边界框、目标级别、实例级别的标签进行分层次的标注，从而可推动众多相关任务的发展，包括目标检测、定位、语义边缘检测、迁移学习 [22] 和域自适应 [23] 等。每幅隐蔽的图像都被赋

- *这项工作的初期版本已经在 CVPR 2020 [1] 上发表。*
- *本文为 TPMAI [2] 论文的中文翻译版本。*
- *该论文的主体工作在南开大学完成。*

- *通讯作者：程明明（cmm@nankai.edu.cn）。*

1. 本文将 COD 任务定义为分割出与周围自然的或者人造的环境具有相似模式（如：纹理、颜色和方向等）的对象或者东西（非定形区域 [6]）。为简便，后续描述本文将伪装目标分割和伪装物体检测等价使用。

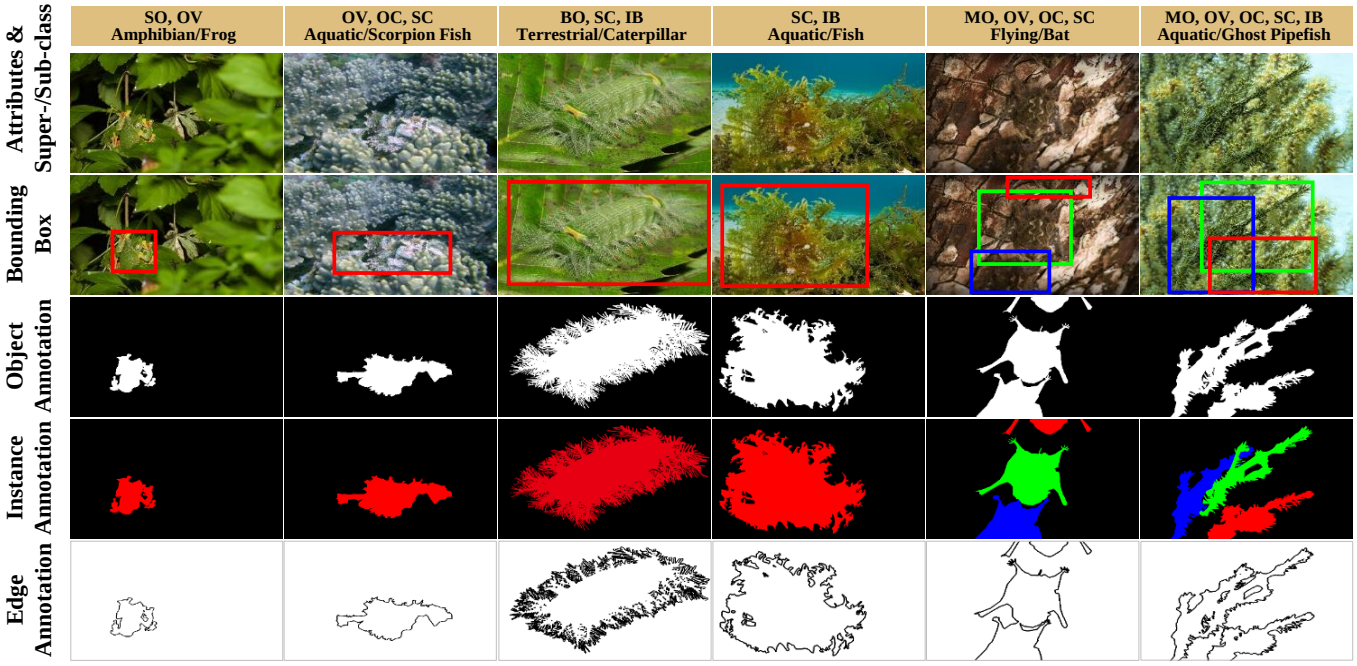


图 3. 本文 *COD10K* 数据集的标签多样性。与以往工作中仅提供粗糙的目标级别标签相比，本文为每一张图片提供了六种不同的标签，包括属性和类别（1^{st} 行）、边界框（2^{nd} 行）、目标级别标签（3^{rd} 行）、实例级别标签（4^{th} 行）和边缘标签（5^{th} 行）。

予了真实世界中挑战性的属性（例如：形状复杂性-SC、边界模糊-IB 和遮挡-OC）和抠图级别 [24] 的标签（每张图像约花费 60 分钟）。这些高质量的标签有助于为模型的性能提供更深入的理解。

2) **COD 框架：**本文提出了一个简单但有效的框架，名为 ***SINet***（搜索识别网络）。值得注意是，*SINet* 的整个训练时间花费四个小时，就在现有的 COD 数据集上取得了最优的性能，表明该模型为检测隐蔽的目标提供了一个潜在的解决方案。该网络同样激发了许多有趣的发现（例如：搜索和识别策略十分契合 COD 任务），使得各式各样的潜在应用成为可能。

3) **COD 基准：**基于所收集的 *COD10K* 和以往的数据集 [25], [26]，本文针对 12 个前沿基线模型进行了详尽的评估，使得本文成为了最全面的 COD 研究工作。本文报告了父类和子类两种场景下的基线性能。同时提供在线评估基准来跟踪研究社区内的发展进程 (http://dpfan.net/camouflage/)。

4) **下游应用：**为进一步促进相关研究，开发了一个在线演示系统 (http://mc.nankai.edu.cn/cod)，以便其他研究人员针对相应场景进行快速测试。此外，本文还阐述了医学、制造业、农业和艺术等相关领域的潜在应用。

5) **未来方向：**基于本文的 *COD10K* 数据集，本文还讨论了未来的十个研究方向，并发现针对隐蔽目标的检测问题还远远没有解决，还具有很大的改进空间。

本文基于会议版本 [1]，在如下角度进行了拓展。首先，本文提供了更为详尽的 *COD10K* 数据集分析，包括分类、统计、标注和分辨率。其次，通过引入近邻连接解码器（NCD）和分组反向注意力（GRA），提升了 *SINet* 模型的性能。第三，进行广泛的实验来验证本文的模型，并针对模型中不同的子模块提供了大量的消融实验。第四，本文提供了一个详尽的父类和子类基准测试，以及针对 COD 这一新颖的任务更深入的讨论。最后，基于基准测评结果得出了多个重要的结论，突出未来有前景的研究方向，例如：隐蔽对象排序、隐蔽对象检测和隐蔽实例分割。

2 相关工作

本节简要回顾密切相关工作。根据 [11]，本节将目标检测分为三大类：一般目标检测、显著目标检测和隐蔽目标检测。

一般目标分割（Generic Object Segmentation，GOS）：一般目标分割是计算机视觉中的一个重要研究方向 [6], [27], [28], [29]。注意，一般目标可以是显著的，也可以是隐蔽的。隐蔽对象可以看作是一般目标的困难案例。典型的 GOS 任务包括语义分割和全景分割 (见图. 2 b)。

显著目标检测 (Salient Object Detection，SOD)：这项任务的目的是识别最引人注目的目标然后分割出其像素级轮廓 [30], [31], [32]。利用 SOD 技术 [33] 的旗舰产品有华为公司的智能手机，使用该技术实现了 “AI Selfies” 自拍技术。

近来，Qin 等人将 SOD 算法 [34] 应用于两个商业应用：AR COPY & PASTE[2]和 OBJECT CUT[3]。这些应用已经引起了广泛的关注（12K+ github stars），并产生了巨大的影响力。

2. https://github.com/cyrildiagne/ar-cutpaste
3. https://github.com/AlbertSuarez/object-cut

表 1
COD 数据集的归纳总结，展示出本文的 *COD10K* 数据集提供了更为丰富的标注，并有助于促进多种任务的发展。 Att.: 属性标签。BBox.: 边界框标签。MI.: 抠图级别标签 [24]。Ins.: 实例级别标签。Cate.: 类别标签。Obj.: 目标级别标签。Loc.: 定位任务。Det.: 检测任务。Cls.: 分类任务。WS.: 弱监督。InSeg. 实例分割。

	统计信息			标注信息						数据划分		任务				
数据集	年份	#Img.	#Cls.	Att.	BBox.	Ml.	Ins.	Cate.	Obj.	#Training	#Testing	Loc.	Det.	Cls.	WS.	InSeg.
CHAMELEON [25]	2018	76	N/A	×	×	×	×	×	✓	0	76	✓	✓	×	×	×
CAMO-COCO [26]	2019	2,500	8	✓	×	×	×	×	✓	1,250	1,250	✓	✓	×	×	×
***COD10K* (OUR)**	2020	**10,000**	**78**	✓	✓	✓	✓	✓	✓	**6,000**	**4,000**	✓	✓	✓	✓	✓

虽然“显著的”一词本质上是“隐蔽的”的反义词 (突出与沉浸), 然而显著的目标可以为 COD 任务提供重要的信息，例如，含有显著物体的图像可作为负样本。针对 SOD 任务进行完备的回顾已经超出了本文工作的研究范围。推荐读者参考近期的综述和基准文献 [12], [35], [36], [37] 中的更多细节。在线基准评测已经公开于：http://dpfan.net/socbenchmark/。

隐蔽目标检测（Concealed Object Detection，COD）： 关于隐蔽/伪装目标检测的研究在生物学和艺术领域中有着悠久而丰富的历史，这对人类视觉感知的知识发展产生了深远的影响。Abbott Thayer [38] 和 Hugh Cott [39] 的两项关于隐蔽动物的杰出研究工作仍具有巨大的影响力。关于这段历史的更多细节，读者可以参考 Stevens 等人 [5] 的综述。在本文提交后，又出现了一些同期的工作 [40], [41], [42]。

COD 数据集： CHAMELEON [25] 是一个未公开发表的数据集，只有 76 张具有手动标注的目标级别的真值图 (GTs)。这些图片是在谷歌搜索引擎上以“隐蔽的动物”作为搜索关键词收集而来的。另一个现代的数据集是 CAMO [26]，它有 2.5K 图像（2K 训练集和 0.5K 测试集），涵盖 8 个类别。含有两个子数据集（CAMO 和 MS-COCO），每一个子集包含 1.25K 张数据样本。与现有的数据集不同的是，*COD10K* 数据集的目标是提供一个更具挑战性、高质量且具备多样化标注的数据集。*COD10K* 是迄今为止最大的隐蔽目标检测数据集，囊括了 10K 数据样本（6K 训练集和 4K 测试集）更多细节详见表. 1 。

伪装类型：伪装大致可以分为两种类型：自然伪装和人造伪装。自然伪装是动物 (例如：昆虫、海马和头足类动物) 避免被捕食者识别的生存技能。相反地，人造伪装通常在艺术设计/游戏之中用于隐藏信息、产品制造过程中（又称为表观缺陷 [43] 和缺陷检测 [44], [45]），或出现在日常生活之中（例如：透明目标 [46], [47], [48]）。

COD 定义：与语义分割等类别相关任务不同的是，隐蔽目标检测是一个类别无关的任务。因此，COD 任务的数学表达更简洁且方便定义。给定一幅图像，该任务需要隐蔽目标检测算法为每一个像素点 i 分配一个标签 $Label_i \in \{0,1\}$，其中 $Label_i$ 代表每一个二值化像素点 i 的值。标签值为 0 代表该像素点不属于隐蔽目标，而像素值为 1 代表该点完全属于隐蔽目标。本文聚焦于对象级别的隐蔽目标检测任务，实例级别的检测任务留作未来工作。

3 COD10K 数据集

新任务和新数据集 [17], [49], [50] 的出现促进了计算机视觉各个领域的蓬勃发展。例如，ImageNet 数据集 [51] 彻底地改变了深度模型在视觉识别中的应用。出于这种考虑，本文研究并构建 COD 数据集的目的是：(1) 从隐蔽的角度提供一个全新且富挑战性的检测任务、(2) 促进领域中多个新主题的研究以及 (3) 激发新颖的想法。*COD10K* 的图例见图. 1 。本节将呈现 *COD10K* 数据集的三个重要的方面，包括图像搜集、专业标注和数据集特征及统计。

3.1 图像搜集

如 [12], [18], [52] 所述，标注质量和数据集规模是基准评测寿命的关键指标。鉴于此，*COD10K* 数据集包含了从多个摄影网站搜集而来的 10,000 张图像样本（5,066 张隐蔽、3,000 张背景和 1,934 张非隐蔽），分为 10 个父类（即：飞行动物类、水生动物类、陆生动物类、两栖动物类、其他类、天空类、植被类、室内类、海洋类和沙地类）以及 78 个子类（69 个隐蔽类和 9 个非隐蔽类）。

大部分隐蔽图像源于 Flickr 网站并仅限于学术研究用途，在搜索图像时使用如下关键词：*隐蔽动物、不明显的动物、隐蔽的鱼、隐蔽的蝴蝶、隐藏的狼蛛、竹节虫、枯叶螳螂、鸟、海马、猫、侏儒海马*等 (详见图. 4)。

其余的隐藏图像（约 200 张）来自其他网站，即：Visual Hunt、Pixabay、Unsplash 和 Free-images 等用于发表公开领域的归档图像且不具有版权和归属问题的网站。为了避免选择上的偏差 [12]，我们同时在 Flickr 上搜集了 3,000 张显著图像。为了进一步使负样本更为丰富，1,934 张非隐蔽图像也从网络中选择出来，包括*森林、雪地、草原、天空、海水*和其他类别的背景场景。有关图像挑选方案的更多细节，本文借鉴了 Zhou 等人的工作 [53]。

3.2 专业标注

最近发表的数据集 [52], [54], [55] 已证明在创建一个大规模数据集时建立一个分类系统是十分重要的。受 [56] 的启发，本

图 4. 子类的样本展示。其他子类请参考补充材料。

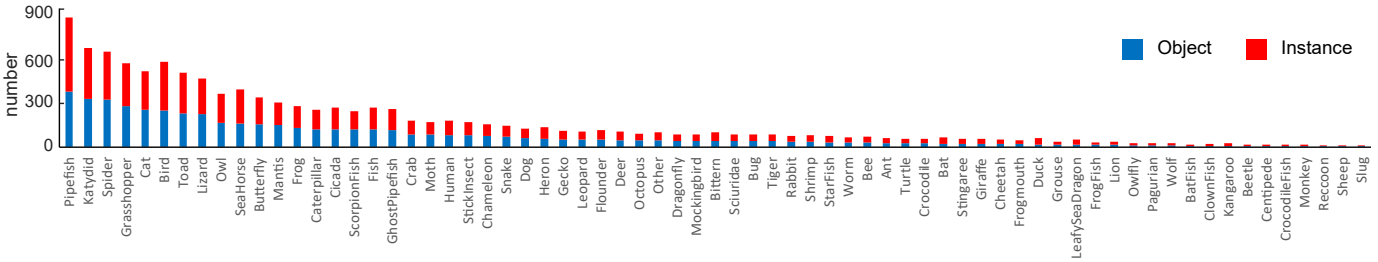


图 5. *COD10K* 中各个隐蔽类别的目标与实例分布。 *COD10K* 包含 69 个类别的 5,066 张隐蔽图像。放大阅读视觉效果最佳。

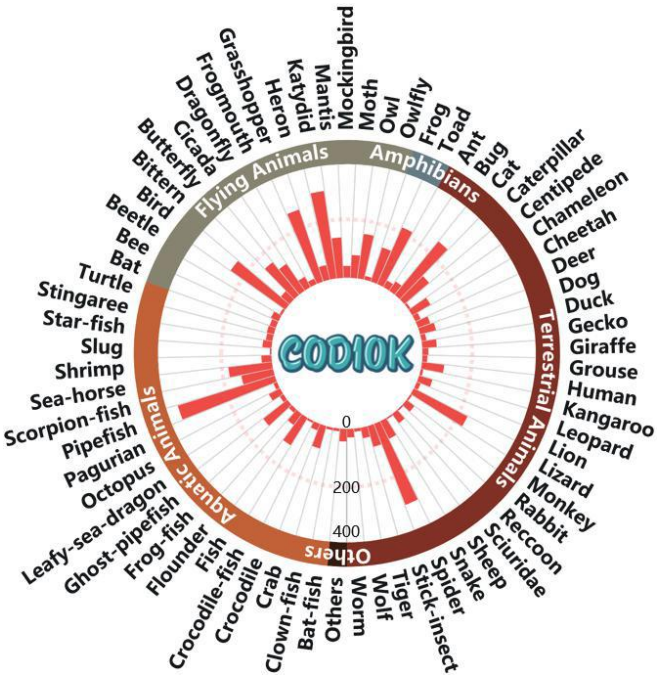
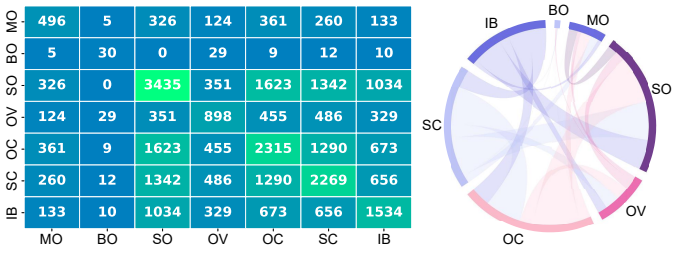


图 6. 分类系统。 *COD10K* 数据集中 69 个隐蔽类别的直方图分布。



属性	描述
MO	多目标。图像中有至少两个目标。
BO	大目标。目标区域与图像区域的比率 (τ_{bo}) $\geq$0.5。
SO	小目标。目标区域与图像区域的比率 (τ_{so}) $\leq$0.1。
OV	超出视野。目标被图像的边界裁切。
OC	遮挡。部分目标被遮挡。
SC	复杂形状。目标包含细长的形状（例如：动物的脚）。
IB	模糊边界。前背景与目标的边界具有相近的颜色（RGB 直方图中 χ^2 距离 τ_{gc} 小于 0.9）。

图 7. 属性分布。左上：*COD10K* 中共现属性的分布。每个网格中的数字代表图像的总数。**右上**：属性的多重依赖。越大的弧长代表该属性与其他属性具有越大的关联。**底部**：属性描述。示例见图. 3 的第一行。

工作的数据标注（通过众包平台完成）是层次化的（类别 ↣ 边界框 ↣ 属性 ↣ 目标/实例）。

• 类别：如图. 6 所示，本文首先建立 5 个父类类别，然后，根据所搜集的数据总结 69 个最常见的子类类别。最后，标注子类和父类中的每一张图像。若候选图像无法归类到任一类别，则将其划分为 '其他' 类别。

• 边界框：为拓展 *COD10K* 在隐蔽目标中的相关任务，我们也为每张图像标注了精确的边界框。

• 属性：与文献 [12], [18] 一致，我们为每幅图像提供了在自然场景中经常遇见的极具挑战的属性标签，例如：遮挡、模糊边界。属性描述和共同属性分布请见图. 7 。

• 目标/实例。值得注意的是，现有的 COD 数据集仅专注于目标级的标注（表. 1 ）。但是，将目标解析为可区分的实例，对计算机视觉中需要对场景进行编辑和理解的研究至关重要。因此，本文进一步以实例级别来标注目标，类似 COCO [16] 数据集，本文最终得到了 5,069 个目标和 5,930 个实例。

3.3 数据集特征和统计

接下来，讨论本文的数据集并呈现他们的统计数据。

• 分辨率分布：如 [57] 所述，高分辨率数据能为模型训练提供更多的目标边界细节并在测试时获得更好的性能。图. 8 提供了 *COD10K* 数据集的分辨率分布，其包含了大量 1080p 的高清图像。

• 目标尺寸：根据 [12]，在图. 9 （左上）中绘制了归一化后的目标尺寸，其尺寸分布从 0.01%~80.74% (平均：8.94%)，相较于 CAMO-COCO 和 CHAMELEON 数据集展现出了更为广泛的分布区间。

• 全局/局部对比：为了衡量一个目标是否容易被检出，本文以全局/局部对比策略 [58] 来进行描述。如图. 9 （右上）所示，*COD10K* 数据集中的目标比其他数据集更具挑战性。

• 中心偏差：这个情况通常发生在拍照时人会自然地倾向关注场景的中心。本文采用 [12] 中描述的策略来分析这个偏差。图. 9 （底部左/右）表明 *COD10K* 数据集相较于其他数据集具有更小的中心偏差。

• 质量控制：为确保高质量的标注，邀请了三位被试来参与标注的过程中的十折交叉验证。图. 10 给出了被接受和被拒绝的示例。这种抠图级别的标注平均每张约花费 60 分钟。

• 父/子类分布： *COD10K* 数据集中隐藏图像有五个父类（即：陆生动物类、飞行动物类、水生动物类、两栖动物类、

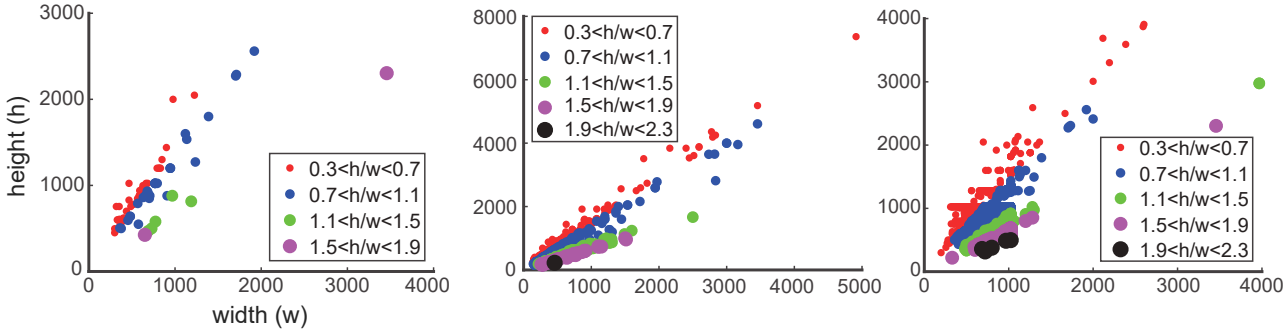


图 8. **COD 数据集的图像分辨率（轴单位：像素）分布。**由左到右：CHAMELEON [25]、CAMO-COCO [26] 和 *COD10K* 数据集。

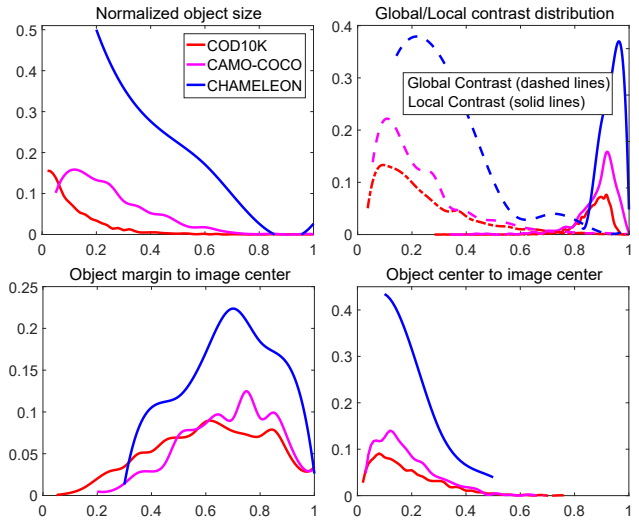


图 9. **本文的 *COD10K* 和现有数据集的比较。** *COD10K* 具有更小的目标（左上），包含更难的隐蔽度（右上）和更小的中心偏差（底部左/右）。

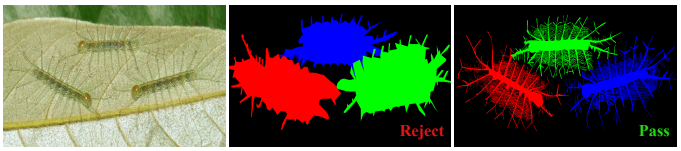


图 10. **高质量标注。**标注质量与现有抠图级别的标注 [24] 相当接近。

其他类）和 69 个子类（即：蝙蝠鱼、狮子、蝙蝠、青蛙等）。不同类别的词云和目标/实例数目的示例分别展示在图. 5 和图. 11 中。

- 数据集划分：为了给深度学习算法提供大量的训练数据，*COD10K* 数据集从各个子类中随机切分成 6,000 张训练图像和 4,000 张测试图像。
- 多样的隐蔽目标：除了图. 1 中所示的一般隐蔽模式，本文数据集还包含了其他种类的隐蔽目标，如：人体彩绘隐蔽和日常中的隐蔽目标（请见图. 12 ）。

4 COD 框架

4.1 模型概览

图. 13 展示了隐蔽目标检测 *SINet*（搜索识别网络）模型的整体框架。下面将阐述动机并介绍模型概览。

图 11. **词云分布。**各个词的体积大小是与该词所占比例成正比。

图 12. ***COD10K* 数据集中不同种类的隐蔽物体。**例如：艺术中隐蔽的人体（第一列）和日常中隐蔽的动物（第二列）。

动机： 生物学研究 [59] 表明，狩猎时捕食者会先判断是否存在潜在的猎物，即：它会先搜寻猎物。接着，目标猎物会被识别；最后，它会被捕获。

介绍： 诸多方法 [60], [61] 采用由多个子步骤组合而成的二次优化策略（即从粗糙到精细）展现出令人满意的效果。这同时说明了将复杂任务的目标进行解耦可以突破性能上的瓶颈。*SINet* 包含了狩猎的前两个阶段，即搜索和识别。具体而言，搜索阶段（第4.2节）负责搜索隐蔽目标，而识别阶段（第4.3节）则以联级的方式来准确地检测出隐蔽目标。

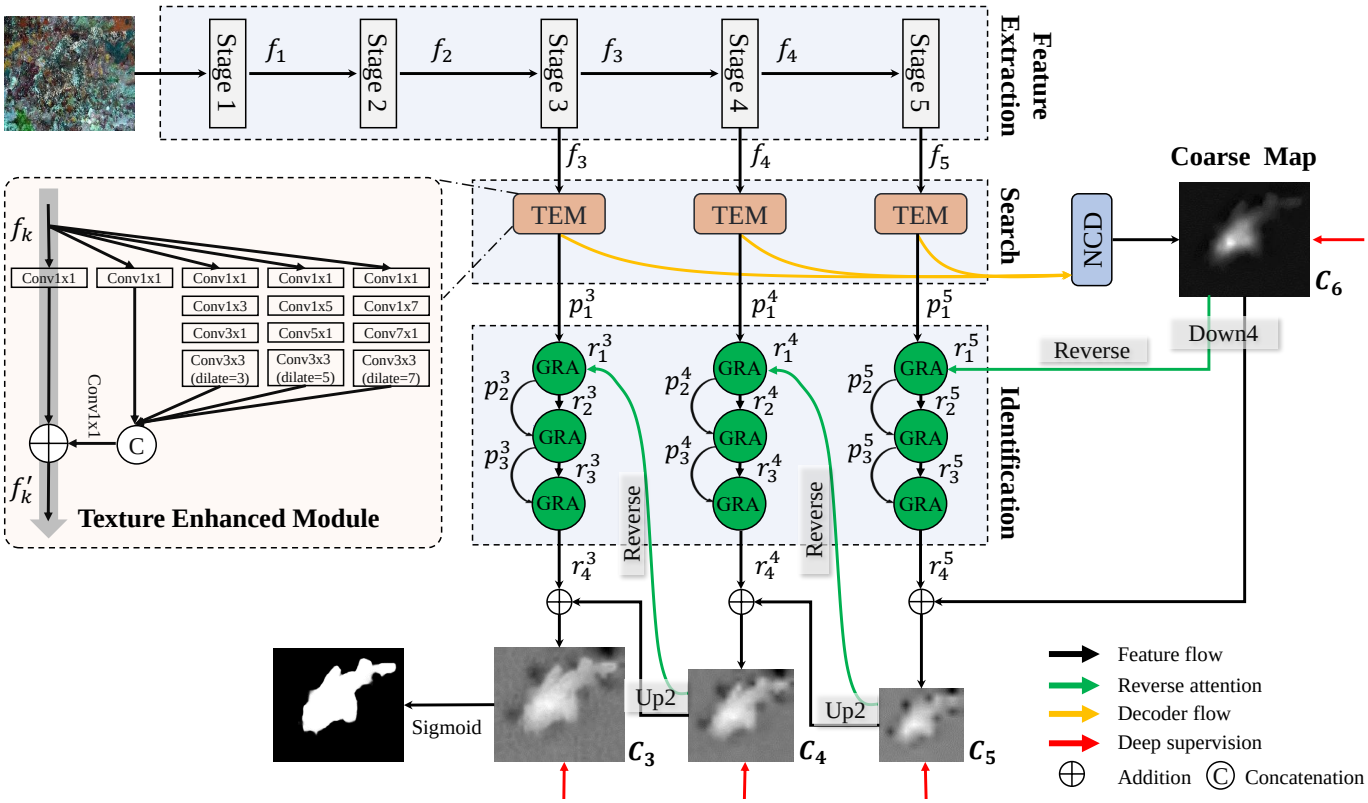


图 13. ***SINet* 框架的流程图**。其包含了三个主要的部件：纹理增强模块（TEM）、近邻连接解码器（NCD）和分组反向注意力（GRA）。TEM 用于模仿人类视觉系统中感受野的纹理结构。NCD 则在 TEM 的协助下负责定位候选区域。GRA 模块则会重现动物狩猎的识别阶段。注意：$f'_k = p^k_1$。

下面将详细描述三个主要模块，包括 *a)* 纹理增强模块 (**TEM**), 其以拓增的上下文线索来捕捉细粒度的纹理；*b)* 近邻连接解码器 (**NCD**), 能够提供定位信息；而 *c)* 分组反向注意力 (**GRA**) 模块可以协同式地提纯来自深层的粗糙预测图。

4.2 搜索阶段

特征提取： 给定一张输入图像 $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$，由 Res2Net-50 [62]（移除了平均池化层、全连接层以及 softmax 层）提取出一组特征 $f_k, k \in \{1,2,3,4,5\}$。因此，每个特征的分辨率 f_k 大小是 $H/2^k \times W/2^k, k \in \{1,2,3,4,5\}$，涵盖了从高分辨率低语义到低分辨率高语义的特征金字塔。

纹理增强模块 (TEM)： 神经科学的相关实验证实，在人类视觉系统中，有一组大小不同的感受野它们对微小的空间变化较为敏感 [63]，并对靠近视网膜中央凹的区域起到了增强作用。这促使我们在搜索阶段（通常在小的/局部空间）使用 TEM [64] 模块以便融合更多具有辨识度的特征。如图. 13 所示，每个 TEM 包含四个平行的残差分支 $\{b_i, i = 1,2,3,4\}$ 相对应着不同的膨胀率 $d \in \{1,3,5,7\}$ 和一个短链接（灰箭头）。在每个分支中 b_i，第一层卷积使用一个 1×1 卷积核将（Conv1×1）通道数降至 32。接下来的另外两层：一个 $(2i-1) \times (2i-1)$ 大小的卷积层和一个具有 $(2i-1)$ 膨胀率大小为 3×3 的卷积层（当 $i > 1$ 时）。接着，前四个分支 $\{b_i, i = 1,2,3,4\}$ 被拼接在一起并通过一个 3×3 卷积操作将通道数降至 C。注意到本文默认设置 $C = 32$ 是为了权衡模型的时间消耗。最终，加上恒等映射后将整个模块传入 ReLU 函数来获得输出特征 f'_k。此外，诸多工作（例如：Inception-V3 [65]）建议可以将大小为 $(2i-1) \times (2i-1)$ 的标准卷积操作分解成由 $(2i-1) \times 1$ 和 $1 \times (2i-1)$ 的卷积核所组成的连续两步，借此加速测试的效率，同时不会降低表征能力。这些想法都建立在秩为 1 的二维核与一系列的一维核是等同的事实基础之上 [66], [67]。简单来说，相较于标准的感受野模块结构 [63]，TEM 新增了一个具有更大膨胀率的分支来扩张感受野，并进一步以非对称卷积取代标准卷积。更多细节请参见图. 13 。

近邻连接解码器 (NCD)： 根据 Wu 等人 [64] 的观察，由于低层特征具有更大的分辨率，会导致其消耗更多的计算资源，然而对性能提升的帮助并不大。受到该观察的启发，本文决定只融合最高三层的特征（即：$\{f_k \in \mathbb{R}^{W/2^k \times H/2^k \times C}, k = 3,4,5\}$）来获取更高效的学习能力，而非采用所有的特征金字塔。具体而言，从三个 TEM 模块获得候选特征后，用于搜索阶段中隐蔽目标的定位。

除此之外，在聚合多个特征金字塔时仍有两个关键问题：那就是如何保持层内语义一致性和如何桥接层间的上下文内容。这里提出 近邻连接解码器（NCD）来解决这些问题。具

体而言，通过近邻连接函数修改了部分解码器 [64]（PDC）模块并得到三个提纯后的特征：$f_k^{nc} = F_{NC}(f_k'; \mathbf{W}_{NC}^u)$，其中 $k \in \{3,4,5\}$ 以及 $u \in \{1,2,3\}$，整个过程定义如下：

$$\begin{cases} f_5^{nc} = f_5' \\ f_4^{nc} = f_4' \otimes g[\delta_{\uparrow}^2(f_5'); \mathbf{W}_{NC}^1] \\ f_3^{nc} = f_3' \otimes g[\delta_{\uparrow}^2(f_4^{nc}); \mathbf{W}_{NC}^2] \otimes g[\delta_{\uparrow}^2(f_4'); \mathbf{W}_{NC}^3] \end{cases} \tag{1}$$

其中 $g[\cdot; \mathbf{W}_{NC}^u]$ 表示一个 3×3 卷积层接一个批归一化操作。为了确保候选特征之间的尺寸是匹配的，在元素级别的相乘 $\otimes$ 之前运用上采样操作（例如两倍上采样）$\delta_{\uparrow}^2(\cdot)$。接着，将 $f_k^{nc}, k \in \{3,4,5\}$ 传入近邻连接解码器（NCD）并生成粗糙的定位图 $\mathbf{C_6}$。

4.3 识别阶段

反向引导： 如第4.2节所讨论，全局定位图 C_6 由最高三层特征所生成，它仅仅捕捉了相对粗略的隐蔽物体的位置，而忽略结构和纹理细节（参见图. 13 ）。为了解决上述问题，本文提出了一个原则性的策略，通过抹除目标 [8], [68], [69] 来提取具鉴别性的隐蔽区域。如图. 14 （b）所示，通过 sigmoid 函数和反向操作获得输出的反向引导。准确来说，采用一个反向操作生成反向引导 r_1^k，可以被定义为：

$$r_1^k = \begin{cases} \ominus \left[\sigma(\delta_{\downarrow}^4(C_{k+1})), \mathbf{E}\right], k = 5, \\ \ominus \left[\sigma(\delta_{\uparrow}^2(C_{k+1})), \mathbf{E}\right], k \in \{3,4\}, \end{cases} \tag{2}$$

其 $\delta_{\downarrow}^4$ 和 $\delta_{\uparrow}^2$ 分别表示 ×4 下采样和 ×2 上采样操作。$\sigma(x) = 1/(1+e^{-x})$ 是 *sigmoid* 函数，可将掩膜转换至 $[0,1]$ 区间。$\ominus$ 是一个反向操作，是利用一个全 1 矩阵减去输入矩阵 $\mathbf{E}$。

分组引导操作（GGO）： 如 [8] 所示，反向注意力被用以抹除侧输出特征中预测的目标区域来提取互补的区域和细节。

受到 [70] 的启发，本文提出了一个新颖的分组操作来更有效地利用反向引导先验。如图. 14 （a）所示，分组引导操作主要包含两个步骤：首先，在特征维度上将候选特征 $\{p_i^k \in \mathbb{R}^{H/2^k \times W/2^k \times C}, k = 3,4,5\}$ 切分至 $m_i = C/g_i$ 组，其中 $i = 1,2,3$，g_i 代表处理后特征的组大小。然后，引导先验 r_1^k 被周期性地插入切分的特征 $p_{i,j}^k \in \mathbb{R}^{H/2^k \times W/2^k \times g_i}$ 中，其中 $i \in \{1,2,3\}, j \in \{1,\ldots,m_i\}, k \in \{3,4,5\}$。因此，这个操作（即：$q_{i+1}^k = \mathbf{F}^{GGO}[p_i^k, r_i^k; m_i]$）可以被解耦成如下两个步骤：

$$\begin{aligned} &\textit{Step I: } \{p_{i,1}^k, \ldots, p_{i,j}^k, \ldots, p_{i,m_i}^k\} \leftarrow \mathbf{F}^S(p_i^k) \\ &\textit{Step II: } q_{i+1}^k \leftarrow \mathbf{F}^C(\{p_{i,1}^k, r_1^k\}, \ldots, \{p_{i,j}^k, r_1^k\}, \ldots, \{p_{i,m_i}^k, r_1^k\}), \end{aligned} \tag{3}$$

其中 $\mathbf{F}^S$ 和 $\mathbf{F}^C$ 表示通道维度的切分和候选特征的拼接函数。注意，$\mathbf{F}^{GGO} : p_i^k \in \mathbb{R}^{H/2^k \times W/2^k \times C} \rightarrow q_{i+1}^k \in \mathbb{R}^{H/2^k \times W/2^k \times (C+m_i)}$，其中 $k \in \{3,4,5\}$。相比之下，[8] 更加强调保留候选特征，因此直接与先验信息相乘，这可能导致两个问题：*a*）由于模型缺乏识别能力而引起的特征混淆和 *b*）单纯的特征相乘将同时导致正确和错误的引导先验并容易累积误差。与 [8] 不同，本文的 GGO 能够在后续的提纯过程之前，显式地将引导先验和候选特征隔离开。

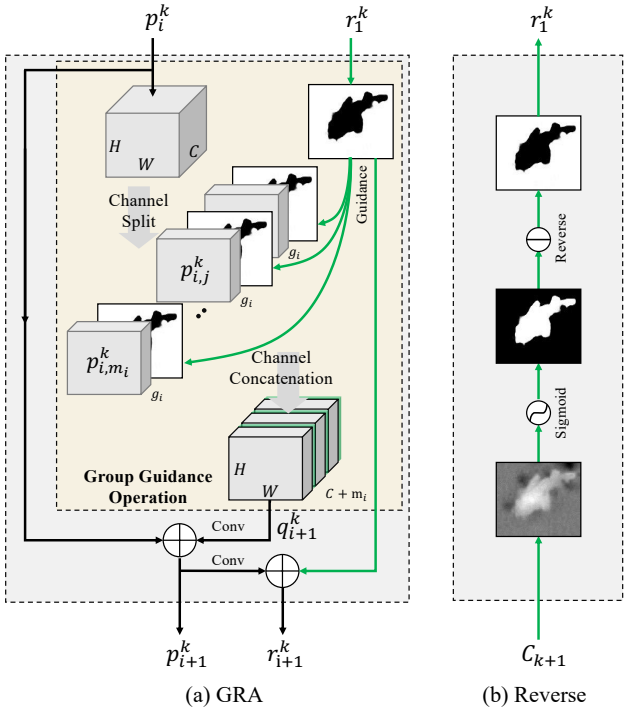


图 14. 模块细节。识别阶段中，分组反向注意力（b）模块 G_i^k 的细节，其中，i 表示在 k-th 特征金字塔中了 GRA 的数目。请注意 $m_i = C/g_i$。

分组方向注意力（GRA）： 最后，本工作提出一个残差学习过程，名为 GRA 模块，其借助于反向引导和分组引导操作。根据之前的研究 [60], [61]，多阶段优化过程可以提升性能。因此，通过组合多个 GRA 模块来逐步提纯（例如：G_i^k，$i \in \{1,2,3\}, k \in \{3,4,5\}$）来自不同特征金字塔的粗糙预测结果。整体而言，每个 GRA 模块包含如下三个残差学习的过程：

i) 通过分组引导操作来融合候选特征 p_i^k 和 r_1^k，再使用残差阶段来生成提纯特征 p_{i+1}^k。定义为：

$$p_{i+1}^k = p_i^k + g[\mathbf{F}^{GGO}[p_i^k, r_1^k; m_i]; \mathbf{W}_{GRA}^v], \tag{4}$$

其中 $\mathbf{W}^v$ 代表具有 3×3 核大小的卷积层和批归一化层，用来将通道数从 $C + m_i$ 减少至 C。注意，默认将第一个 GRA 模块设置为使用反向引导先验操作（即当 $i = 1$）。详细的讨论参见第5.3节。

ii) 然后，得到单通道的残差引导图：

$$r_{i+1}^k = r_1^k + g[p_{i+1}^k; \mathbf{W}_{GRA}^w], \tag{5}$$

其通过可学习的权重 $\mathbf{W}_{GRA}^w$ 进行参数化。

iii) 最后，只输出优化后的引导图，可视为残差预测图。其定义为：

$$C_k = r_{i+1}^k + \delta(C_{k+1}), \tag{6}$$

其中 $\delta(\cdot)$ 当 $k = \{3,4\}$ 时为 $\delta_{\uparrow}^2$，而当 $k = 5$ 时为 $\delta_{\downarrow}^4$。

表 2
在三个数据集上的量化结果。最佳的分数会以**粗体**标识。注意，ANet-SRM（只在 CAMO 上训练）没有开源的代码，因此无法获得其他数据集的跑分。↑ 代表越高分越好。E_ϕ 代表 E 指标的平均值 [71]。

基线模型	CHAMELEON [25]				CAMO-Test [26]				COD10K-Test (OUR)			
	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$
FPN [72]	0.794	0.783	0.590	0.075	0.684	0.677	0.483	0.131	0.697	0.691	0.411	0.075
MaskRCNN [73]	0.643	0.778	0.518	0.099	0.574	0.715	0.430	0.151	0.613	0.748	0.402	0.080
PSPNet [74]	0.773	0.758	0.555	0.085	0.663	0.659	0.455	0.139	0.678	0.680	0.377	0.080
UNet++ [75]	0.695	0.762	0.501	0.094	0.599	0.653	0.392	0.149	0.623	0.672	0.350	0.086
PiCANet [76]	0.769	0.749	0.536	0.085	0.609	0.584	0.356	0.156	0.649	0.643	0.322	0.090
MSRCNN [77]	0.637	0.686	0.443	0.091	0.617	0.669	0.454	0.133	0.641	0.706	0.419	0.073
PFANet [78]	0.679	0.648	0.378	0.144	0.659	0.622	0.391	0.172	0.636	0.618	0.286	0.128
CPD [64]	0.853	0.866	0.706	0.052	0.726	0.729	0.550	0.115	0.747	0.770	0.508	0.059
HTC [79]	0.517	0.489	0.204	0.129	0.476	0.442	0.174	0.172	0.548	0.520	0.221	0.088
ANet-SRM [26]	-	-	-	-	0.682	0.685	0.484	0.126	-	-	-	-
EGNet [13]	0.848	0.870	0.702	0.050	0.732	0.768	0.583	0.104	0.737	0.779	0.509	0.056
PraNet [8]	0.860	0.907	0.763	0.044	0.769	0.824	0.663	0.094	0.789	0.861	0.629	0.045
SINet_cvpr [1]	0.869	0.891	0.740	0.044	0.751	0.771	0.606	0.100	0.771	0.806	0.551	0.051
***SINet* (OUR)**	**0.888**	**0.942**	**0.816**	**0.030**	**0.820**	**0.882**	**0.743**	**0.070**	**0.815**	**0.887**	**0.680**	**0.037**

4.4 实现细节

4.4.1 学习策略

损失函数定义为 $L = L_{IoU}^W + L_{BCE}^W$，其中 L_{IoU}^W 和 L_{BCE}^W 代表带权重的交并比（IoU）损失和二值交叉熵（BCE）损失分别用来计算全局约束和局部（像素级别）约束。标准的 IoU 损失已被广泛地使用在分割任务上，而与其不同的是带权重的交并比损失增加了较难像素的权重，用来突出其重要性。此外，相较于标准的二值交叉熵损失，L_{BCE}^W 更注重较难的像素而非赋予每个像素相同的权重。这些损失函数的定义与 [60], [80] 相同并已经被证实在显著性目标检测领域中是有效的。此处对三个旁路输出以及全局特征图 C_6 采用深度监督（即：C_3，C_4，和 C_5）。每个预测图会经上采样（例如：C_3^{up}）成为与真值图 G 相同的尺寸大小。因此，*SINet* 模型完整的损失函数定义为：$L_{total} = L(C_6^{up}, G) + \sum_{i=3}^{i=5} L(C_i^{up}, G)$。

4.4.2 超参数设定

SINet 采用 PyTorch 框架实现并使用 Adam 优化器 [81] 进行训练。在训练的过程中，批大小设定为 36，初始学习率为 1e-4，每 50 个周期再除以 10。整个训练经历 100 个周期并耗时约 4 小时。基于 Intel® i9-9820X CPU @3.30GHz × 20 和单张 NVIDIA TITAN RTX 显卡上进行运行时间的测量。在测试阶段，每张图像被缩放至 352×352 并传入本文的框架中来获得最终的预测，并没有采用任何后处理技术。测试的帧率在去除读/写时间后在单张 GPU 上达到约 45 fps。该模型的 PyTorch [82] 版本和 Jittor [83] 版本的源码将会公开。

5 COD 的评价基准

5.1 实验设定

5.1.1 评价指标

平均绝对误差（MAE）被广泛地用于 SOD 任务。参照 Perazzi 等人 [85] 的设定，本文也采用 MAE（M）指标来评估预测图与真值图之间像素级精度。然而，尽管 MAE 指标可以评估错误的出现和数量，但无法判定错误出现的位置。近来，Fan 等人提出了基于人类视觉感知的 E 指标（E_ϕ）[71]，其同时评测像素级别的匹配度和图像级别的统计量。这个指标自然地适合来评估隐蔽目标检测的整体和局部精确度。注意在本文的实验中报告了 E_ϕ 的平均值。由于隐蔽目标经常包含复杂的形状，COD 也需要一个指标来衡量结构的相似度。因此本文中采用 S 指标（S_α）[84] 来衡量结构的相似度。最后，近期的研究 [71], [84] 表明了带有权重的 F 指标（F_β^w）[86] 比传统的 F_β 更能提供可靠的评价结果。因此，本文考虑将此作为 COD 的代替指标。一键评价代码可在项目主页上找到。

5.1.2 基线模型

本文根据以下准则选择 12 个深度学习基线模型 [8], [13], [26], [64], [72], [73], [74], [75], [76], [77], [78], [79]：*a)* 经典结构，*b)* 近期发表和 *c)* 在特定领域中达到前沿的性能。

5.1.3 训练/测试协议

为了公平地与之前的版本 [1] 进行比较，本文所对比的基线模型采用了与 [1] 相同的训练设定[4]。本文以整个 CHAMELEON [25] 数据集和 CAMO 及 COD10K 的测试集来评估模型。

5.2 结果和数据分析

本节提供了 CHAMELEON、CAMO 和 COD10K 数据集上的量化评估结果。

CHAMELEON 数据集上的性能： 在表. 2 中，*SINet* 与 12 个前沿的目标检测基线模型及 ANet-SRM 进行比较，*SINet* 模型在每个指标上均达到新的前沿性能。注意，本文所提出的模型并没有使用边缘辅助特征（例如：EGNet [13]、PFANet [78]）和预处理技术 [87] 或后处理策略 [88], [89]。

4. 为了验证 *SINet* 模型的泛化能力，本文只使用 CAMO [26] 和 COD10K [1] 作为训练数据集而没有使用其他（额外的）数据。

表 3

在 COD10K 的四个父类中用四个常用评价指标的量化结果。所有的方法都以 [1] 中的数据集训练。↑ 代表越高分越好，而 ↓ 代表越低分越好。

基线模型	Amphibian (124 images)				Aquatic (474 images)				Flying (714 images)				Terrestrial (699 images)			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$M \downarrow$
FPN [72]	0.745	0.776	0.497	0.065	0.684	0.732	0.432	0.103	0.726	0.766	0.440	0.061	0.601	0.656	0.353	0.109
MaskRCNN [73]	0.665	0.785	0.487	0.081	0.560	0.721	0.344	0.123	0.644	0.767	0.449	0.063	0.611	0.630	0.380	0.075
PSPNet [74]	0.736	0.774	0.463	0.072	0.659	0.712	0.396	0.111	0.700	0.743	0.394	0.067	0.669	0.718	0.332	0.071
UNet++ [75]	0.677	0.745	0.434	0.079	0.599	0.673	0.347	0.121	0.659	0.727	0.397	0.068	0.608	0.749	0.288	0.070
PiCANet [76]	0.686	0.702	0.405	0.079	0.616	0.631	0.335	0.115	0.663	0.676	0.347	0.069	0.658	0.708	0.273	0.074
MSRCNN [77]	0.722	0.786	0.555	0.055	0.614	0.686	0.398	0.107	0.675	0.744	0.466	0.058	0.594	0.661	0.361	0.081
PFANet [78]	0.693	0.677	0.358	0.110	0.629	0.626	0.319	0.155	0.658	0.648	0.299	0.102	0.611	0.603	0.237	0.111
CPD [64]	0.794	0.839	0.587	0.051	0.739	0.792	0.529	0.082	0.777	0.827	0.544	0.046	0.714	0.771	0.445	0.058
HTC [79]	0.606	0.598	0.331	0.088	0.507	0.495	0.183	0.129	0.582	0.559	0.274	0.070	0.530	0.485	0.170	0.078
EGNet [13]	0.785	0.854	0.606	0.047	0.725	0.793	0.528	0.080	0.766	0.826	0.543	0.044	0.700	0.775	0.445	0.053
PraNet [8]	0.842	0.905	0.717	0.035	0.781	0.883	0.696	0.065	0.819	0.888	0.669	0.033	0.756	0.835	0.565	0.046
SINet_cvpr [1]	0.827	0.866	0.654	0.042	0.758	0.803	0.570	0.073	0.798	0.828	0.580	0.040	0.743	0.778	0.491	0.050
***SINet* (OUR)**	**0.858**	**0.916**	**0.756**	**0.030**	**0.811**	**0.883**	**0.696**	**0.051**	**0.839**	**0.908**	**0.713**	**0.027**	**0.787**	**0.866**	**0.623**	**0.039**

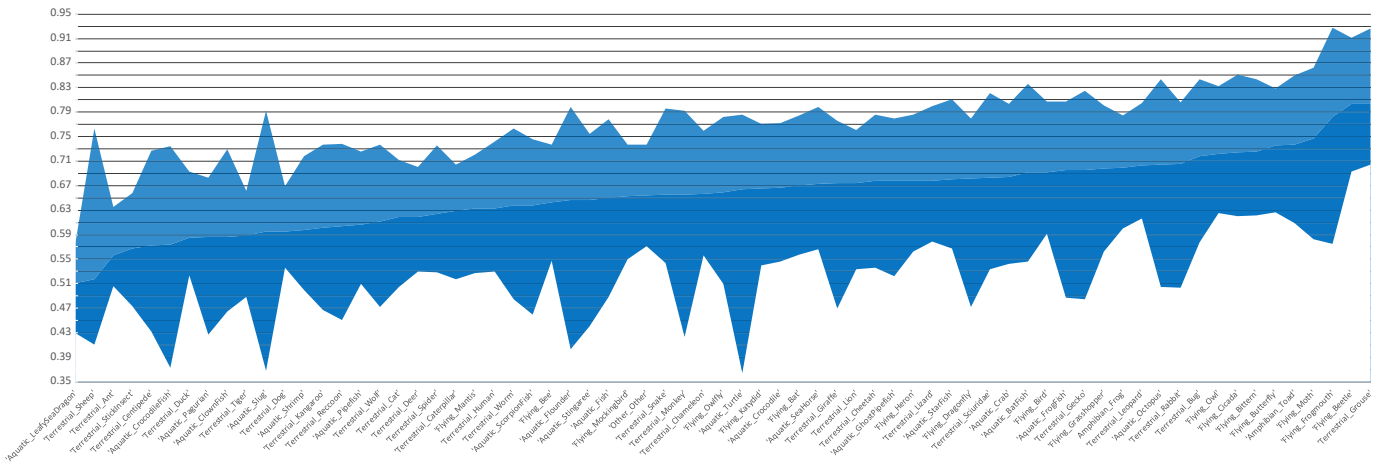


图 15. **每个子类的性能。本文以 12 个基线模型的 S_α [84] 得分来排序子类的难度。同时提供每个子类 S_α 的最小值（底线）和最大值（顶线）。**

CAMO 数据集上的性能： 我们也在 CAMO [26] 数据集上测试本文的模型，该数据集包含了各式隐蔽的目标。基于表. 2 上整体的性能表现，可观察到 CAMO 数据集比 CHAMELEON 更具挑战性。*SINet* 再次达到最佳的性能，进一步展现了其鲁棒性。

COD10K 数据集上的性能： 在 COD10K 测试集上 (2,026 张图像)，观察到 *SINet* 模型再次优于其他的对比模型。这是因为它精心设计的搜索和识别模块能自动地从粗糙至精细地学习到丰富且多元的特征，这对于发现具有挑战性的模糊目标的边界极为重要。结果分别展示于表. 2 和 表. 3 。

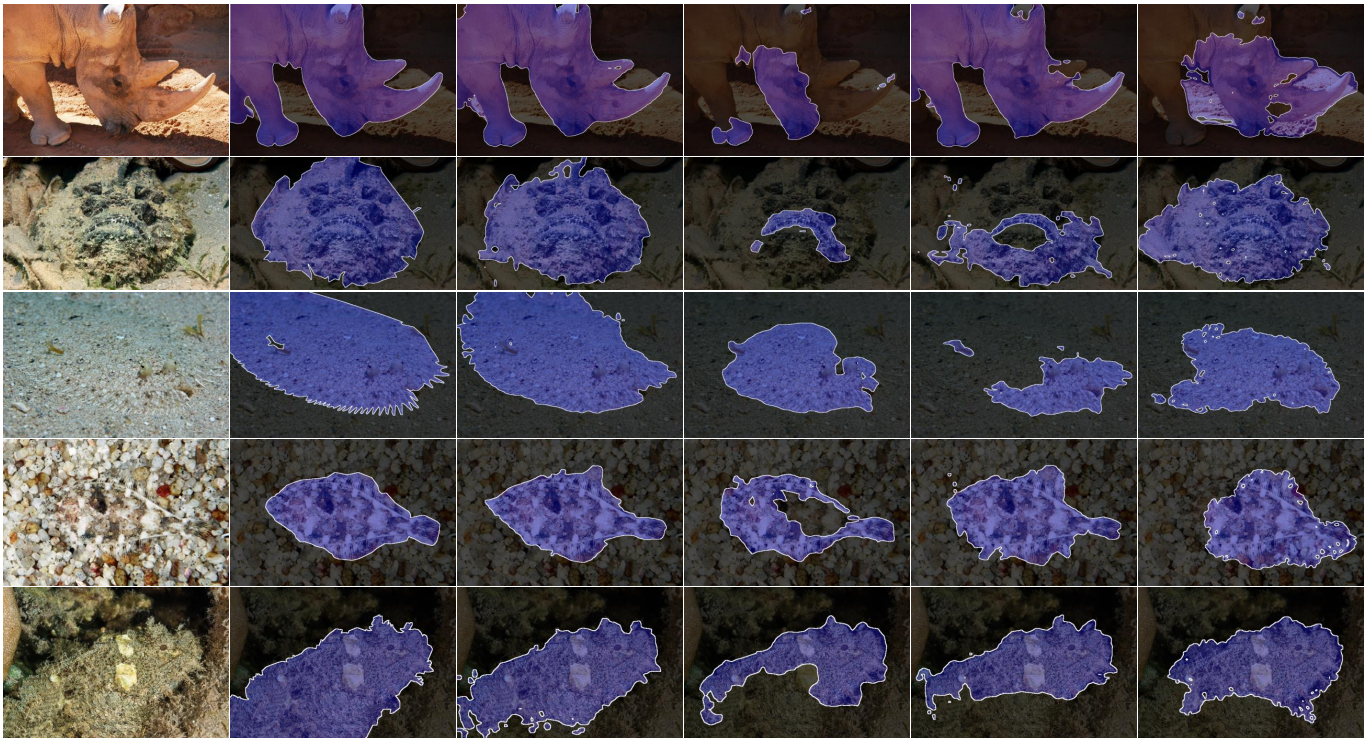
各子类上的性能： 除了在 COD10K 数据集上整体的量化指标外，本文也在表. 4 中呈现了每个子类的评估结果，为了让未来的研究人员更了解当前模型的优劣。在图. 15 中，本文额外展示了所有基线模型在各个子类 S 指标的最小值、平均值和最大值。在陆生动物和水生动物中最简单和最难的子类分别是 “Grouse” 和 “LeafySeaDragon”。

定性结果： 在补充材料中还为会议版本的模型 (SINet_cvpr) 呈现了更多针对各种挑战性的隐蔽目标检测的结果，例如：蜘蛛、蚕、海马和蟾蜍。如图. 16 所示，*SINet* 在不同光照情况 (1^{st} 行)、外观改变 (2^{nd} 行)、模糊边界 (3^{rd} 到 5^{th} 行) 下相较 SINet_cvpr 在视觉上，进一步提升了结果。PFANet [78] 能够定位隐蔽的目标，但是预测结果总是不精确。通过进一步地利用反向注意力模块，PraNet [8] 在第一个案例中达到相对于 PFANet 更准确的定位。然而，它仍然缺失了目标的细节，尤其是在 2^{nd} 行和 3^{rd} 行中鱼的结果图。针对这些具有挑战性的案例，*SINet* 能够预测出正确的隐蔽目标与其细节，展现出了本文架构的鲁棒性。

GOS 与 SOD 基线模型对比： 值得注意是，在最好的三个模型之中，GOS 模型 (即：FPN [72]) 比 SOD 模型 (即：CPD [64] 和 EGNet [13]) 表现得差，体现出 SOD 架构或许更适合扩展至 COD 任务。相较于 GOS [72], [73], [74], [75], [77], [79] 和 SOD [13], [64], [76], [78] 模型，*SINet* 明显地减少了训练时间 (例如：*SINet*: 4 小时 *vs.* EGNet: 48 小时) 并在所有数据集上达到了前沿性能，体现出其为 COD 任务中最优的解决方案。由于篇幅限制，全面地与现有前沿的 SOD 模型进行比较超出了本文的讨论范围。本文的核心目标是为未来的研究提供更全面的观察。更多的 SOD 模型可参见项目主页。

表 4
针对 *COD10K* 中每一个子类的 S_α 指标结果。每个模型的最佳结果用加粗高亮。

Sub-class	HTC [79]	PFANet [78]	MRCNN [73]	BASNet [60]	UNet++ [75]	PiCANet [76]	MSRCNN [77]	PoolNet [90]	PSPNet [74]	FPN [72]	EGNet [13]	CPD [64]	PraNet [8]	***SINet* OUR**
Amphibian-Frog	0.600	0.678	0.664	0.692	0.656	0.687	0.692	0.732	0.697	0.731	0.745	0.752	*0.823*	**0.837**
Amphibian-Toad	0.609	0.697	0.666	0.717	0.689	0.714	0.739	0.786	0.757	0.752	0.812	0.817	*0.853*	**0.870**
Aquatic-BatFish	0.546	0.746	0.634	0.749	0.626	0.624	0.637	0.741	0.724	0.764	0.707	0.761	**0.879**	*0.873*
Aquatic-ClownFish	0.547	0.519	0.509	0.464	0.548	0.636	0.571	0.626	0.531	0.730	0.632	0.646	*0.707*	**0.787**
Aquatic-Crab	0.543	0.661	0.691	0.643	0.630	0.675	0.634	0.727	0.680	0.724	0.760	0.753	*0.792*	**0.815**
Aquatic-Crocodile	0.546	0.631	0.660	0.599	0.602	0.669	0.646	0.743	0.636	0.687	0.772	0.761	*0.806*	**0.825**
Aquatic-CrocodileFish	0.436	0.572	0.558	0.475	0.559	0.373	0.479	0.693	0.624	0.515	*0.709*	0.690	0.669	**0.746**
Aquatic-Fish	0.488	0.622	0.597	0.625	0.574	0.619	0.680	0.703	0.650	0.699	0.717	0.778	*0.784*	**0.834**
Aquatic-Flounder	0.403	0.663	0.539	0.633	0.569	0.704	0.570	0.782	0.646	0.695	0.798	0.774	*0.835*	**0.889**
Aquatic-FrogFish	0.595	0.768	0.650	0.736	0.671	0.670	0.653	0.719	0.695	*0.807*	0.806	0.730	0.781	**0.894**
Aquatic-GhostPipefish	0.522	0.690	0.556	0.679	0.651	0.675	0.636	0.717	0.709	0.744	0.759	0.763	*0.784*	**0.817**
Aquatic-LeafySeaDragon	0.460	0.576	0.442	0.481	0.523	0.499	0.500	0.547	0.563	0.507	0.522	0.534	*0.587*	**0.670**
Aquatic-Octopus	0.505	0.708	0.598	0.644	0.663	0.673	0.720	0.779	0.723	0.760	0.810	0.812	**0.896**	*0.887*
Aquatic-Pagurian	0.427	0.578	0.477	0.607	0.553	0.624	0.583	0.657	0.608	0.638	*0.683*	0.611	0.615	**0.698**
Aquatic-Pipefish	0.510	0.553	0.531	0.557	0.550	0.612	0.566	0.625	0.642	0.632	0.681	0.704	*0.769*	**0.781**
Aquatic-ScorpionFish	0.459	0.697	0.482	0.686	0.630	0.605	0.600	0.729	0.649	0.668	0.730	0.746	*0.766*	**0.808**
Aquatic-SeaHorse	0.566	0.656	0.581	0.664	0.663	0.623	0.657	0.698	0.687	0.715	0.750	0.765	*0.810*	**0.823**
Aquatic-Shrimp	0.500	0.574	0.520	0.631	0.586	0.574	0.546	0.605	0.591	0.667	0.647	0.669	*0.727*	**0.735**
Aquatic-Slug	0.493	0.581	0.368	0.492	0.533	0.460	0.661	0.732	0.547	0.664	**0.777**	*0.774*	0.701	0.729
Aquatic-StarFish	0.568	0.641	0.617	0.611	0.657	0.638	0.580	0.733	0.722	0.756	*0.811*	0.787	0.779	**0.890**
Aquatic-Stingaree	0.519	0.721	0.670	0.618	0.571	0.569	0.709	0.733	0.616	0.670	0.741	*0.754*	0.704	**0.815**
Aquatic-Turtle	0.364	0.686	0.594	0.658	0.565	0.734	0.762	0.757	0.664	0.745	0.752	**0.786**	*0.773*	0.760
Flying-Bat	0.589	0.652	0.611	0.623	0.557	0.638	0.679	0.725	0.657	0.714	0.765	0.784	*0.817*	**0.847**
Flying-Bee	0.578	0.579	0.628	0.547	0.588	0.616	0.679	0.670	0.655	0.665	0.737	0.709	*0.763*	**0.777**
Flying-Beetle	0.699	0.741	0.693	0.810	0.829	0.780	0.796	0.860	0.808	0.848	0.830	0.887	*0.890*	**0.903**
Flying-Bird	0.591	0.628	0.680	0.627	0.643	0.674	0.681	0.735	0.696	0.708	0.763	0.785	*0.822*	**0.835**
Flying-Bittern	0.639	0.621	0.703	0.650	0.673	0.741	0.704	0.785	0.701	0.751	0.802	*0.838*	0.827	**0.849**
Flying-Butterfly	0.653	0.692	0.697	0.700	0.725	0.714	0.762	0.777	0.736	0.758	0.816	0.818	*0.871*	**0.883**
Flying-Cicada	0.640	0.682	0.620	0.729	0.675	0.691	0.708	0.781	0.744	0.733	0.820	0.812	*0.845*	**0.883**
Flying-Dragonfly	0.472	0.679	0.624	0.712	0.670	0.694	0.682	0.695	0.681	0.707	0.761	*0.779*	*0.779*	**0.837**
Flying-Frogmouth	0.684	0.766	0.648	0.828	0.813	0.722	0.773	0.883	0.741	0.795	0.901	*0.928*	0.927	**0.941**
Flying-Grasshopper	0.563	0.671	0.651	0.689	0.656	0.692	0.666	0.734	0.710	0.740	0.773	0.779	*0.821*	**0.833**
Flying-Heron	0.563	0.579	0.629	0.598	0.670	0.647	0.699	0.718	0.654	0.743	0.783	0.786	*0.810*	**0.823**
Flying-Katydid	0.540	0.661	0.593	0.657	0.653	0.659	0.615	0.696	0.687	0.709	0.730	0.739	*0.802*	**0.809**
Flying-Mantis	0.527	0.622	0.569	0.618	0.614	0.629	0.603	0.661	0.658	0.670	0.696	0.690	*0.749*	**0.775**
Flying-Mockingbird	0.641	0.550	0.622	0.593	0.636	0.596	0.664	0.670	0.674	0.683	0.721	0.737	*0.788*	**0.838**
Flying-Moth	0.583	0.720	0.726	0.737	0.707	0.685	0.747	0.783	0.753	0.798	0.833	0.854	*0.878*	**0.917**
Flying-Owl	0.625	0.671	0.705	0.656	0.657	0.718	0.710	0.781	0.712	0.750	0.793	0.809	*0.837*	**0.868**
Flying-Owlfly	0.614	0.690	0.524	0.669	0.633	0.580	0.599	0.778	0.583	0.743	*0.782*	0.756	0.758	**0.863**
Other-Other	0.571	0.613	0.603	0.593	0.638	0.653	0.675	0.687	0.671	0.665	0.725	0.700	*0.777*	**0.779**
Terrestrial-Ant	0.506	0.516	0.508	0.519	0.523	0.585	0.538	0.552	0.572	0.564	0.627	0.605	**0.676**	*0.669*
Terrestrial-Bug	0.578	0.681	0.682	0.687	0.686	0.701	0.691	0.743	0.710	0.799	0.799	0.803	*0.828*	**0.856**
Terrestrial-Cat	0.505	0.585	0.591	0.557	0.562	0.608	0.613	0.669	0.624	0.634	0.682	0.678	*0.745*	**0.772**
Terrestrial-Caterpillar	0.517	0.643	0.569	0.691	0.636	0.581	0.575	0.638	0.640	0.685	0.684	0.704	*0.729*	**0.776**
Terrestrial-Centipede	0.432	0.573	0.476	0.485	0.496	0.554	0.629	0.703	0.561	0.536	0.727	0.643	*0.704*	**0.762**
Terrestrial-Chameleon	0.556	0.651	0.627	0.653	0.619	0.619	0.632	0.695	0.659	0.673	0.713	0.732	*0.789*	**0.804**
Terrestrial-Cheetah	0.536	0.649	0.699	0.624	0.603	0.662	0.598	0.717	0.720	0.667	0.732	0.769	*0.800*	**0.826**
Terrestrial-Deer	0.530	0.581	0.610	0.564	0.558	0.600	0.623	0.650	0.644	0.660	0.667	0.670	*0.719*	**0.757**
Terrestrial-Dog	0.572	0.560	0.596	0.536	0.559	0.574	0.614	0.608	0.588	0.613	0.607	0.648	*0.666*	**0.707**
Terrestrial-Duck	0.530	0.535	0.557	0.539	0.524	0.558	0.619	0.582	0.602	0.548	0.598	0.682	*0.742*	**0.746**
Terrestrial-Gecko	0.485	0.674	0.662	0.725	0.683	0.705	0.606	0.733	0.724	0.747	0.789	0.771	*0.833*	**0.848**
Terrestrial-Giraffe	0.469	0.628	0.697	0.620	0.611	0.701	0.635	0.681	0.718	0.722	0.747	0.776	**0.809**	*0.784*
Terrestrial-Grouse	0.704	0.760	0.726	0.721	0.774	0.805	0.780	0.879	0.803	0.806	0.904	*0.919*	0.888	**0.921**
Terrestrial-Human	0.530	0.629	0.608	0.613	0.549	0.577	0.658	0.697	0.636	0.665	0.708	0.700	*0.765*	**0.817**
Terrestrial-Kangaroo	0.482	0.586	0.599	0.467	0.548	0.588	0.571	0.644	0.630	0.623	0.650	0.620	*0.798*	**0.816**
Terrestrial-Leopard	0.617	0.647	0.742	0.616	0.640	0.652	0.673	0.736	0.720	0.704	0.744	*0.791*	*0.791*	**0.823**
Terrestrial-Lion	0.534	0.634	0.695	0.599	0.660	0.656	0.658	0.720	0.714	0.663	0.754	0.751	*0.805*	**0.813**
Terrestrial-Lizard	0.579	0.629	0.634	0.635	0.633	0.656	0.627	0.710	0.702	0.716	0.744	0.777	*0.804*	**0.830**
Terrestrial-Monkey	0.423	0.693	0.724	0.593	0.611	0.730	0.663	0.792	0.678	0.614	0.709	0.699	*0.851*	**0.888**
Terrestrial-Rabbit	0.504	0.657	0.685	0.634	0.635	0.721	0.731	0.794	0.722	0.758	0.789	0.806	*0.829*	**0.843**
Terrestrial-Reccoon	0.451	0.525	0.536	0.461	0.482	0.702	0.723	0.643	0.532	0.592	0.691	0.659	**0.781**	*0.766*
Terrestrial-Sciuridae	0.533	0.612	0.638	0.573	0.608	0.693	0.661	0.745	0.725	0.721	0.775	0.757	*0.810*	**0.842**
Terrestrial-Sheep	0.434	0.451	0.721	0.410	0.482	0.467	**0.763**	*0.660*	0.466	0.430	0.489	0.487	0.481	0.500
Terrestrial-Snake	0.544	0.590	0.586	0.603	0.567	0.614	0.597	0.714	0.695	0.652	0.738	*0.788*	0.771	**0.831**
Terrestrial-Spider	0.528	0.594	0.593	0.594	0.580	0.621	0.572	0.650	0.649	0.651	0.685	0.687	*0.740*	**0.771**
Terrestrial-StickInsect	0.473	0.548	0.486	0.526	0.535	0.600	0.491	0.578	0.607	0.629	0.616	0.647	*0.660*	**0.696**
Terrestrial-Tiger	0.489	0.583	0.576	0.555	0.573	0.563	0.565	0.638	0.602	0.599	0.647	0.621	*0.690*	**0.703**
Terrestrial-Wolf	0.472	0.574	0.602	0.535	0.534	0.568	0.621	0.650	0.656	0.651	0.704	0.662	*0.737*	**0.749**
Terrestrial-Worm	0.485	0.652	0.596	0.642	0.628	0.558	0.651	0.692	0.629	0.684	0.763	0.670	*0.724*	**0.806**



(a) Image (b) GT (c) SINet (OUR) (d) SINet_cvpr (e) PraNet (f) PFANet

图 16. ***SINet* 与三个最优基线模型的性能比较**，包含（d）SINet_cvpr [1]、（e）PraNet [8] 和（f）PFANet [78]。

表 5
跨数据集泛化性能的结构指标 ($S_\alpha \uparrow$ [84]) 分数。 SINet_cvpr 模型在一个数据集上训练（行）并在所有的数据集上进行测试（列）。"Self"：代表在同一个数据集上进行训练和测试（对角线）。"Mean others"：代表除自身以外的所有数据集上的平均分数。

Trained on: \ Tested on:	CAMO [26]	COD10K (OUR)	Self	Mean others	Drop↓
CAMO [26]	0.803	0.702	0.803	0.702	12.6%
COD10K (OUR)	0.742	0.700	0.700	0.742	-6.0%
Mean others	0.742	0.702			

泛化性： 数据集的泛化性和难度是训练和评估不同算法的关键因素 [36]。因此，针对这些方面研究现有的 COD 数据集，利用跨数据集分析方法 [91]，即分别在不同数据集上训练和测试模型。本文选择 CAMO [26] 和 *COD10K*。根据 [36]，针对每个数据集，随机选择 800 张图像作为训练集和 200 张图像作为测试集。为了公平的比较，在每个数据集上训练 SINet_cvpr 直到损失趋于稳定。

表. 5 提供跨数据集上泛化性的 S 指标结果。每行列出分别在不同数据集训练和测试的模型，代表训练数据集的泛化性。每列代表在其他数据集训练和在特定数据集测试的模型，代表测试数据集的难度。请注意训练和测试设定与表. 2 中的不同，因此性能是不可比较的。不出所料，本文发现 *COD10K* 比 CAMO 数据集有更好的泛化能力（例如：最后一列中 '*Drop*↓: -6.0%'）。这是因为本文的数据集包含各式具有挑战性的隐蔽目标 (参见第3节)。因此可以看出 COD10K 数据集包含更多挑战性的场景。

5.3 消融实验

本节提供了 *SINet* 在 CHAMELEON、CAMO 和 COD10K 上细节的分析。通过解耦各个子模块（包括 NCD、TEM 和 GRA）来证明其有效性，其结果总结于 表. 6 。注意，在重新训练每个消融变体模型时使用第4.4节中相同的超参数。

NCD 的有效性： 本节探索 *SINet* 模型在搜索阶段中解码器的影响从而证实其有效性，移除 NCD（No.#1）并重新训练网络后，发现相较于 #OUR 变体（表. 6 中的最后一行），NCD 在 CAMO 数据集上有贡献，平均 E_ϕ 性能从 0.869 提升到了 0.882。进一步地，以部分解码器 [64] 替换 NCD（即：No.#2 的 PD）来证明近邻连接（No.#OUR）的优势所在。比较 No.#2 和 #OUR，本文的设计能略微提升性能，在 CHAMELEON 数据集上的 F_β^w 指标增加了 1.7%。

如图. 17 所示，在改良的类 UNet 解码器结构的基础上提出了一个新颖的特征聚合策略（移除了具高分辨率的最底两层），名为 NCD，相邻的层间带有近邻连接。该设计是受到高层特征更适合提升语义和准确定位，但引进了噪声及模糊边缘事实的启发。

不在密集连接层以短连接 [33] 或带有跳层连接的部分解码器 [64] 来传播特征，本文的 NCD 以近邻连接挖掘语义信息，提供了一个简单并有效的方法来减少不同特征之间的不一致性。使用短路连接 [33] 来聚合所有的特征会增加网络的总体参数。这个是 DSS（图. 17 a）和 NCD 之间主要的差别。相较于 CPD [64]（图. 17 b），其忽略了 f_5' 和 f_4' 之间的特征

表 6
在三个测试集上针对各模块的消融实验。细节请参见 第5.3节.

No.	Decoder		TEM		GRA		CHAMELEON [25]				CAMO-Test [26]				COD10K-Test (OUR)			
	PD	NCD	Sy. Conv.	Asy. Conv,	Reverse	Group Size	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$
#1				✓	{1, 0, 0}	{32; 8; 1}	0.884	0.940	0.811	0.033	0.812	0.869	0.730	0.073	0.812	0.884	0.679	0.039
#2	✓			✓	{1, 0, 0}	{32; 8; 1}	0.881	0.934	0.799	0.034	**0.820**	0.877	0.740	0.071	0.813	0.884	0.673	0.038
#3		✓			{1, 0, 0}	{32; 8; 1}	0.887	0.934	0.813	0.033	0.811	0.867	0.731	0.074	**0.815**	**0.888**	0.680	**0.036**
#4		✓	✓		{1, 0, 0}	{32; 8; 1}	**0.888**	**0.944**	**0.818**	**0.030**	0.810	0.866	0.730	0.073	0.814	0.883	0.678	0.037
#5		✓		✓	{0, 0, 0}	{32; 8; 1}	0.886	0.942	0.814	0.031	0.814	0.873	0.739	0.073	0.814	0.887	**0.682**	0.037
#6		✓		✓	{1, 1, 0}	{32; 8; 1}	0.879	0.928	0.794	0.035	**0.820**	0.877	0.738	0.071	0.807	0.878	0.661	0.040
#7		✓		✓	{1, 1, 1}	{32; 8; 1}	0.886	0.939	0.812	0.031	0.817	0.875	0.736	0.073	0.810	0.884	0.670	0.037
#8		✓		✓	{1, 0, 0}	{1; 1; 1}	**0.888**	0.940	0.812	0.031	0.819	0.877	0.741	0.072	0.814	0.887	0.681	0.037
#9		✓		✓	{1, 0, 0}	{8; 8; 8}	0.886	0.943	0.814	0.032	0.816	0.872	0.738	0.074	**0.815**	0.886	**0.682**	0.037
#10		✓		✓	{1, 0, 0}	{32; 32; 32}	0.884	**0.944**	0.810	0.033	0.819	0.876	0.738	0.071	0.813	0.884	0.675	0.037
#11		✓		✓	{1, 0, 0}	{1; 8; 32}	0.883	0.940	0.812	0.032	0.811	0.869	0.734	0.073	**0.815**	0.887	0.679	**0.036**
#OUR		✓		✓	{1, 0, 0}	{32; 8; 1}	**0.888**	0.942	0.816	**0.030**	**0.820**	**0.882**	**0.743**	**0.070**	**0.815**	0.887	0.680	0.037

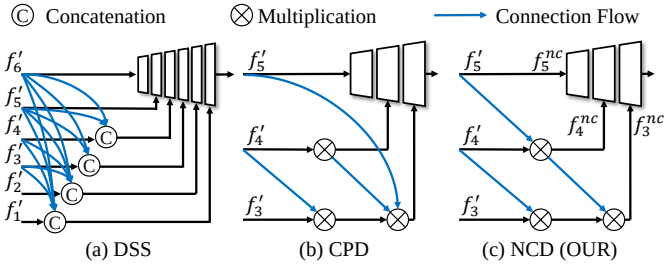


图 17. 各种层间特征聚合与短路连接的比较。(a) DSS [33] 提出了自顶向下的密集短路连接。(b) CPD [64] 丢弃浅层的大分辨率特征构建了一个部分解码器，以提升计算资源利用和速度。(c) 本文的近邻连接解码器只在近邻层传播特征。

透明性，NCD 能更有效地逐步传播特征。

TEM 的有效性： 本节提供了两个不同的变量：(a) 去除 TEM (No.#3) 和 (b) 使用带有对称的卷积层 [65] (No.#4)。相较于 No.#3，发现在 CAMO 数据集上使用带有非对称卷积层的 TEM 对于性能的提升是必要的。除此之外，以非对称卷积层 (No.#OUR) 替换对称卷积层 (No.#4) 会小幅度影响模型的学习能力，其在 CAMO 数据集进一步提升 E_ϕ 的平均值从 0.866 至 0.882。

GRA 的有效性： *反向引导*。如同 表. 6 的 'Reverse' 列，{*,*,*} 代表每个 GRA 模块 G_i^k 前的引导是否为反向 (详见图. 14 (b))。例如，{1,0,0} 代表只有第一个模块 (即 r_1^k) 的引导是反向的而另外两个模块 (即：r_2^k 和 r_3^k) 没有使用反向操作。

本节研究 GRA 内反向引导的贡献，包含三个方案：(a) 去除所有的反向操作，即 No.#5 中的 {0,0,0}、 (b) 反转前两个引导 $r_i^k, i \in \{1,2\}$, 即 No.#6 中的 {1,1,0} 和 (c) 反转全部的引导 $r_i^k, i \in \{1,2,3\}$, 即 No.#7 中的 {1,1,1}。相较于 *SINet* 的默认设定 (即：No.#OUR 的 {1,0,0})，发现只反转第一个引导或许能从两个方面 (即注意力和反向注意力) 帮助模型去提取多样的表征，然而在中间过程采用多次反向引导可能会导致学习过程的混淆，特别是在设定 #6 中的 CHAMELEON 和 COD10K 数据集。

GGO 的分组大小。如 表. 6 中 'Group Size' 列所示，{∗; ∗; ∗} 表示 GGO 中第一个模块 G_1^k 至最后一个模块 G_3^k 中特征切片的数量 (即：分组大小 g_i)。举例来说，{32; 8; 1} 代表在每个 GRA 模块分别切分候选特征 $p_i^k, i \in \{1,2,3\}$ 至 32、8 和 1 分组大小。在此讨论两种选择分组大小的方法，即：统一的策略 (#8 中的 {1; 1; 1} ，#9 中的 {8; 8; 8}，#10 中的 {32; 32; 32}) 和渐进的策略 (#11 中的 {1; 8; 32} 和 #OUR 中的 {32; 8; 1})。发现基于渐进策略的设计能有效地维持模型的泛化性，相较其他变体模型能提供更优的性能。

6 下游应用

隐蔽目标检测系统在医学、艺术和农业等领域中有丰富的下游应用。本节设想了一些具有共性特征的潜在应用，因为这类应用中目标与背景具有类似的表观特征。更多的细节请参见项目主页。在这种情况下，COD 任务的一些模型非常适合作为这类应用中的核心组件来挖掘伪装的目标。请注意，这些应用只是激发未来研究而提出的一些有趣想法的案例。

6.1 应用 I: 医学

6.1.1 息肉分割

众所周知，通过医学成像进行早期诊断在疾病治疗中扮演着关键的作用。然而，早期病变/病灶区域通常与周围的组织器官有着高度的同质性。因此，医师难以从早期阶段的医学影像中识别病变区域。一个典型的案例就是早期结肠镜检查中息肉的分割，这将会降低结直肠癌约 30% 的发病率 [8]。与隐蔽目标检测相似的是，息肉分割 (参见图. 18) 同样面临着诸多挑战，例如：表观变化和模糊边缘。最先进的息肉分割模型 PraNet [8] 在息肉分割 (Top-1) 和隐蔽目标分割 (Top-2) 任务中取得了不错的表现。从这个角度来看，将所提出的 SINet 模型嵌入到这个应用中可能会获得更鲁棒的预测结果。

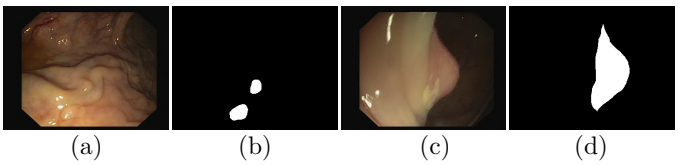


图 18. **息肉分割**。(a) & (c) 是输入的息肉图像。(b) & (d) 是所对应的真值图。

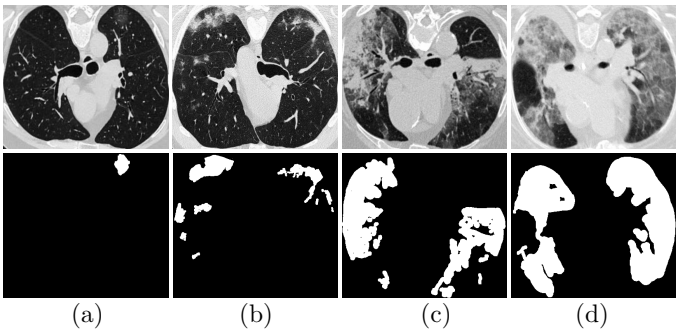


图 19. **肺部感染区域分割**。第一行表示 COVID-19 肺部感染区域 CT 扫描切片，而第二行对应的表示经过医师标注的真值图。从 (a) 到 (d)，COVID-19 患者的感染程度从中期到晚期。

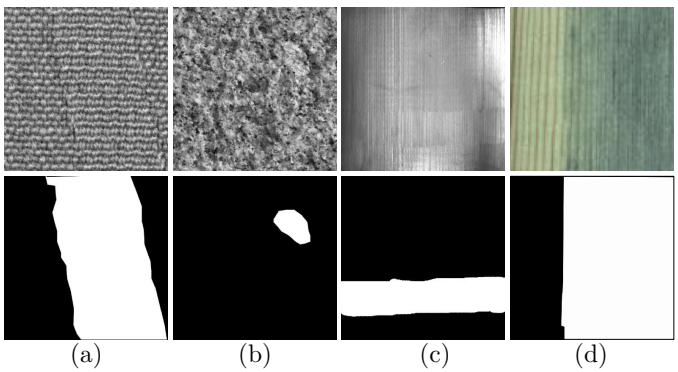


图 20. **表观缺陷检测**。缺陷类型分别为：(a) 纺织品、(b) 石头、(c) 磁瓦和 (d) 木材。第二行代表其所对应的真值图。图片来源于 [92]。

6.1.2 肺部感染区域分割

另一个隐蔽检测案例是在医疗领域中的肺部感染分割任务。近来，COVID-19 疾病尤其令人关注并导致全球疾病大流行。配备 COVID-19 肺部感染分割模型的人工智能系统将有助于早期 COVID-19 疾病的筛查。

更多有关该应用的细节可参见近来的分割模型 [9] 和综述文献 [93]。使用 COVID-19 肺部感染分割数据集重新训练 SINet 模型将会成为另外一个有趣的潜在应用。

6.2 应用 II: 制造业

6.2.1 表观缺陷检测

在工业制造中，产品 (如：木材、纺织品和磁砖) 质量不佳必然会对经济造成不利影响。如图. 20 所示，表观缺陷因其低对比度、难以定义的边界等因素而具有挑战性。由于传统的表观缺陷检测系统主要依赖于人，最大的问题便是高度主观性和耗时的鉴别过程。因此，设计一个基于人工智能

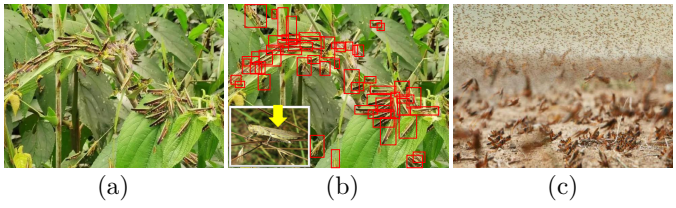


图 21. **病虫检测**。对于病虫检测应用，系统能够针对每一张局部筛选的图像 (a) 生成边界框 (b)，或者针对提供统计数据（病虫数量）以用于在整个环境 (c) 中对蝗灾密度进行监测。

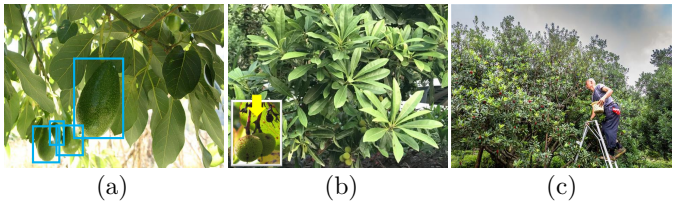


图 22. **水果成熟度检测**。与传统人工监测水果 (c) 相比，基于 AI 的成熟度检测系统（例如：牛油果 (a) 和杨梅 (b)）将会极大地提升产能效率。

的自动识别系统是提高生产效率的必要条件。我们正在积极构建这样的数据集，以推进相关研究进展。相关的文献可参见：https://github.com/Charmve/Surface-Defect-Detection/tree/master/Papers。

6.3 应用 III：农业

6.3.1 病虫检测

自 2020 年初以来，沙漠蝗虫的祸害已侵入非洲到南亚各国。大量的蝗虫在田野上啃食并彻底摧毁农产品，因粮食短缺造成严重经济损失和饥荒。如图. 21 所示，引入基于人工智能的技术来实施科学的监测，对各国政府实现可持续的管制/遏制病虫害是可行的。然而，收集昆虫相关的数据用于训练 COD 模型需要丰富的生物学知识，这也是该应用所面临的难题。

6.3.2 水果成熟度检测

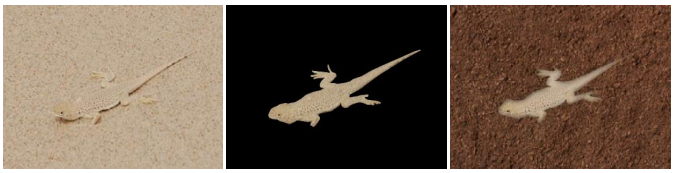
在水果成熟的早期阶段，许多果实看起来像绿色的叶子，图. 22 展示了两种水果：牛油果和杨梅。这类水果具有相似的特征从而被隐蔽起来，因此可利用 COD 算法来识别这些水果以提高监测的效率。

6.4 应用 IV：艺术

6.4.1 文娱艺术

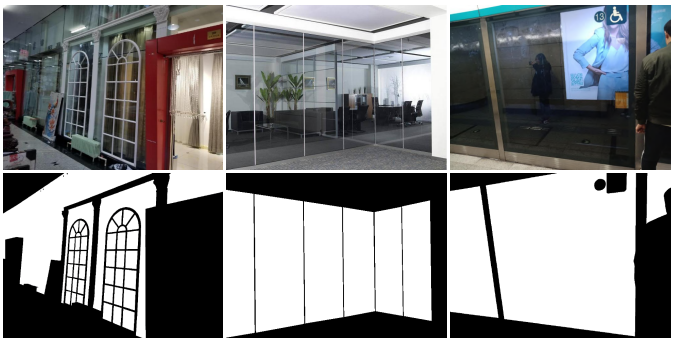
在 SIGGRAPH 社区里，通过将隐蔽的显著物体嵌入到背景中是一类有趣的技术。图. 23 展示了由 Chu 等人 [10] 生成的一些案例。本文技术将为现有的、数据缺乏的深度学习模型提供更多的训练数据。因此，如 Treisman 和 Wolfe [94], [95] 所述，探讨其背后特征搜索和连接搜索理论之间的内在机制是很有价值的。

图 23. **文娱艺术**。通过算法，一些动物被嵌入到背景中。图片来源于 Chu 等人 [10]，版权由 2010 年 John Van Straalen 所保留。



(a) (b) (c)

图 24. **从隐蔽目标转变为显著目标**。图片来源于 [26]。一个有趣的应用就是识别出 (a) 图中的一个特别的隐蔽目标 (b)，然后将其转变成为显著目标 (c)。



(a) (b) (c)

图 25. **透明东西/目标检测**。在日常生活中，人类看到、触摸或与各种透明材料产生交互，例如：窗户 (a)、玻璃门 (b) 和玻璃墙 (c)。第二行是其所对应的真值图。教会 AI 机器人识别透明的东西/目标以避开隐蔽障碍是必要的。

6.4.2 从隐蔽目标走向显著目标

隐蔽目标检测和显著目标检测是两种对立的视觉任务，可以便捷地设计一个多任务学习框架，同时提高网络的鲁棒性。如图. 24 所示，子图 (a) 和 (c) 中存在两种反向的目标。一个有趣的应用是提供一个滚动条来允许用户自定义从伪装目标转换为显著目标的程度值。

6.5 应用 V：日常生活

6.5.1 透明东西/目标检测

玻璃制品等透明物品在日常生活中很常见。如图. 25 所示，这类物体/东西（包括门和墙）所固有的背景外观使其不引人注意。作为隐蔽目标检测的一个子任务，透明物体检测 [48] 和透明对象跟踪 [96] 已经展现出巨大的潜力。

6.5.2 搜索引擎

图. 26 展示了来自谷歌的搜索结果示例。从结果 (图. 26 a) 中可注意到搜索引擎无法检测隐藏的蝴蝶，因此只提供类似背景的图像。有趣的是，当搜索引擎配备了一个隐蔽检测系统 (这里仅简单地改变了关键字)，该算法可以识别隐藏目标，然后反馈出多幅蝴蝶图像（见图. 26 b）。

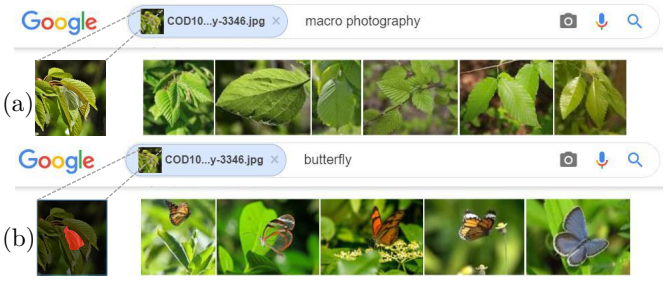


图 26. **搜索引擎**。互联网搜索引擎应用不配备 (a) 与配备 (b) 隐蔽检测系统对比。

7 潜在的研究方向

近十年来，尽管隐蔽目标检测领域取得了不错的进展，与一般目标检测 [97] 相比，深度学习时代的前沿算法仍然有限，还不能有效地解决现实世界中的挑战，比如在 *COD10K* 基准测试中最好的模型性能 $F_\beta^w < 0.7$。在此，本文强调下列长期存在的一些挑战:

- 受限条件下的隐蔽目标检测：少次/零次学习、弱监督学习、无监督学习、自监督学习、有限训练数据、未知目标类别等。
- 与其他模态融合的隐蔽目标检测：文本、语音、视频、RGB 和深度图像、RGB 和红外热图像、三维数据等。
- 基于 *COD10K* 提供的丰富标注的新方向，如：隐蔽实例分割、隐蔽边缘检测、隐蔽目标提议、隐蔽目标排序等。

基于上述挑战，一些未来可预见的研究方向如下:

(1) 弱监督/半监督检测： 现有的基于深度的方法以完全监督的方式从带有对象级标签的图像中提取特征。但是，像素级标签通常是使用 LabelMe 或带有大量专业交互功能的 Adobe Photoshop 软件进行人工标注。因此，为了降低标注费用，采用弱/半 (部分) 标注的数据进行训练显得尤为重要。

(2) 自监督检测： 近来，使用自监督学习来获得视觉 (例如：图像、音频和视频) 表征的研究工作 [98], [99] 取得了举世闻名的成就而备受关注。因此，为隐蔽目标检测任务建立自监督的基准模型也是显而易见的方向。

(3) 其他模态下的隐蔽目标检测： 现有的隐蔽数据仅基于静态图像或动态视频 [100]。然而，暗光下的害虫监测、机器人技术和艺术家设计等领域中的伪装目标检测任务可以是多个紧密相联的模态。与 RGB-D SOD [54]、RGB-T SOD [101]、CoSOD [102], [103] 和 VSOD [104] 任务相类似，这些模态可以是音频、红外热图或深度图，从而在这些特定场景下提出了新的挑战。

(4) 隐蔽目标分类： 一般目标分类是计算机视觉中的一项基本任务。因此，隐蔽目标分类也可能在未来得到关注。利用

COD10K 中提供的类和子类标签，学者可以构建一个大规模的细粒度分类任务。

(5) 隐蔽目标检测和追踪：在本文中，隐蔽目标的检测实际上是一个分割任务。这与传统的生成一个提议（proposal）或边界框作为预测的目标检测任务不同。因此，隐蔽目标的检测加跟踪在未来将是一个全新且有趣的研究方向 [105]。

(6) 隐蔽目标排序：目前，隐蔽目标检测算法是建立在二值化真值图的基础上的用真值图来构建隐藏物体的掩膜，仅有限的工作分析了隐蔽目标的排序 [40]。然而，了解隐蔽程度有助于更好地探索模型背后的机理，提供更深入的见解。建议读者参考 [40], [106] 来获得灵感。

(7) 隐蔽实例分割：如 [21] 所述，实例分割比目标分割在实际应用中更重要。例如，可以将伪装目标分割向伪装实例分割转换，从而推进了相关的研究进展。

(8) 针对多种任务的统一框架： Zamir 等人提出的 Taskonomy [22] 研究表明，不同的视觉任务有很强的关联。因此，可以在不增加模型复杂度的情况下将监督信号进行复用。自然而然地，可以想到设计一种通用网络使其能同时针对隐蔽目标进行定位、分割和排序。

(9) 神经架构搜索：用于隐蔽目标检测的传统算法和深度学习模型往往需要较强的先验知识或熟练的专家参与设计。通常，经算法工程师人工设计的特征和架构可能不是最优的。因此，利用神经架构搜索技术是一个潜在的研究方向，如最近流行的自动机器学习 [107]。

(10) 将显著目标转变为隐蔽目标：由于篇幅限制，本文只在评测章节部分评估了典型的显著目标检测模型。然而，这还有许多值得进一步研究的科学问题，例如从显著目标到隐蔽目标的转换，从而增加训练样本数量，并在 SOD 和 COD 任务之间引入生成对抗机制，从而提高网络的特征提取能力。

上述与隐蔽目标相关的十个全新研究问题依旧还有很长的一段路要走。值得庆幸的是，依然有很多经典的文献可供参考，从隐蔽的角度为研究人员看待目标检测任务夯实了基础。

8 结语

本文从隐蔽的角度呈现了针对目标检测的首个完备研究。具体而言，本文提供了极具挑战性的、密集标注的全新 *COD10K* 数据集，并进行了大规模的基准测评，同时设计了一个简单而高效的端到端搜索识别框架（*SINet*），最后强调了若干潜在应用。与现有的前沿基线模型相比，本文的 *SINet* 模型更具竞争力，并生成了更契合人类视觉的预测结果。上述贡献为研究社区提供了一个为 COD 任务设计新模型的机会。未来，我们计划将 *COD10K* 数据集拓展为多种输入形式，例如：多视角图像（如：RGB-D SOD [108], [109]）、文本描述、视频（如：视频显著目标检测 [104]）等。我们也计划将自动化搜索最优的感受野 [110] 应用于增强特征表达 [111]，以获得更好的性能。

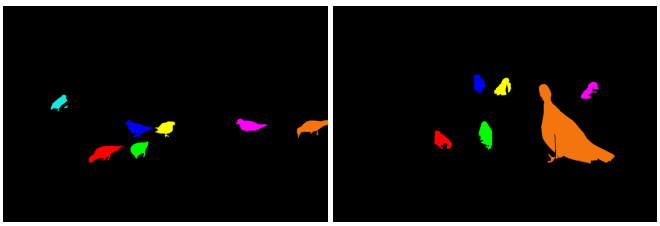


图 27. 图. 1 所展示的图像样本的**真值图**。

参考文献

[1] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 2777–2787.

[2] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, "Concealed object detection," *arXiv*, 2021.

[3] I. C. Cuthill, M. Stevens, J. Sheppard, T. Maddocks, C. A. Párraga, and T. S. Troscianko, "Disruptive coloration and background pattern matching," *Nature*, vol. 434, no. 7029, p. 72, 2005.

[4] A. Owens, C. Barnes, A. Flint, H. Singh, and W. Freeman, "Camouflaging an object from many viewpoints," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2782–2789.

[5] M. Stevens and S. Merilaita, "Animal camouflage: current issues and new perspectives," *Phil. Trans. R. Soc. B: Biological Sciences*, vol. 364, no. 1516, pp. 423–427, 2008.

[6] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 9404–9413.

[7] T. Troscianko, C. P. Benton, P. G. Lovell, D. J. Tolhurst, and Z. Pizlo, "Camouflage and visual perception," *Phil. Trans. R. Soc. B: Biological Sciences*, vol. 364, no. 1516, pp. 449–461, 2008.

[8] D.-P. Fan, G.-P. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, "Pranet: Parallel reverse attention network for polyp segmentation," in *Med. Image. Comput. Comput. Assist. Interv.*, 2020.

[9] D.-P. Fan, T. Zhou, G.-P. Ji, Y. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, "Inf-Net: Automatic COVID-19 Lung Infection Segmentation from CT Images," *IEEE Trans. Med. Imaging*, 2020.

[10] H.-K. Chu, W.-H. Hsu, N. J. Mitra, D. Cohen-Or, T.-T. Wong, and T.-Y. Lee, "Camouflage images." *ACM Trans. Graph.*, vol. 29, no. 4, pp. 51–1, 2010.

[11] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE T. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019.

[12] D.-P. Fan, J. Zhang, G. Xu, M.-M. Cheng, and L. Shao, "Salient objects in clutter," *arXiv preprint arXiv:*, 2021.

[13] 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.*, 2019.

[14] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[15] 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.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[17] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 633–641.

[18] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 724–732.

[19] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, 2019.

[20] G. Medioni, "Generic object recognition by inference of 3-d volumetric," *Object Categorization: Comput. Hum. Vis. Perspect.*, vol. 87, 2009.

[21] G. Li, Y. Xie, L. Lin, and Y. Yu, "Instance-level salient object segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 247–256.

[22] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3712–3722.

[23] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.

[24] Y. Zhang, L. Gong, L. Fan, P. Ren, Q. Huang, H. Bao, and W. Xu, "A late fusion cnn for digital matting," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 7469–7478.

[25] P. Skurowski, H. Abdulameer, J. Błaszczyk, T. Depta, A. Kornacki, and P. Kozieł, "Animal camouflage analysis: Chameleon database," 2018, unpublished Manuscript.

[26] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto, "Anabranch network for camouflaged object segmentation," *Comput. Vis. Image Underst.*, vol. 184, pp. 45–56, 2019.

[27] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Eur. Conf. Comput. Vis.*, 2006, pp. 1–15.

[28] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2010.

[29] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015.

[30] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.

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

[32] 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.

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

[34] X. Qin, D.-P. Fan, C. Huang, C. Diagne, Z. Zhang, A. C. Sant' Anna, A. Su'arez, M. Jagersand, and L. Shao, "Boundary-aware segmentation network for mobile and web applications," *arXiv preprint arXiv:2101.04704*, 2021.

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

[36] W. Wang, Q. Lai, H. Fu, J. Shen, and H. Ling, "Salient object detection in the deep learning era: An in-depth survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[37] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, "Salient object detection: A survey," *Computational Visual Media*, vol. 5, no. 2, pp. 117–150, 2019.

[38] G. H. Thayer and A. H. Thayer, *Concealing-coloration in the Animal Kingdom: An Exposition of the Laws of Disguise Through Color and Pattern: Being a Summary of Abbott H. Thayer's Discoveries.* Macmillan Company, 1909.

[39] H. B. Cott, *Adaptive coloratcottion in animals.* Methuen & Co., Ltd., 1940.

[40] Q. Zhai, X. Li, F. Yang, C. Chen, H. Cheng, and D.-P. Fan, "Mutual graph learning for camouflaged object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[41] Y. Lyu, J. Zhang, Y. Dai, L. Aixuan, B. Liu, N. Barnes, and D.-P. Fan, "Simultaneously localize, segment and rank the camouflaged objects," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[42] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, and D.-P. Fan, "Camouflaged object segmentation with distraction mining," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[43] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *J. Intell. Manuf.*, vol. 31, no. 3, pp. 759–776, 2020.

[44] Y. He, K. Song, Q. Meng, and Y. Yan, "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1493–1504, 2020.

[45] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "Pga-net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Trans. Industr. Inform.*, 2020.

[46] A. Kalra, V. Taamazyan, S. K. Rao, K. Venkataraman, R. Raskar, and A. Kadambi, "Deep polarization cues for transparent object segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 8602–8611.

[47] Y. Xu, H. Nagahara, A. Shimada, and R.-i. Taniguchi, "Transcut: Transparent object segmentation from a light-field image," in *Int. Conf. Comput. Vis.*, 2015, pp. 3442–3450.

[48] E. Xie, W. Wang, W. Wang, M. Ding, C. Shen, and P. Luo, "Segmenting transparent objects in the wild," in *Eur. Conf. Comput. Vis.*, 2020.

[49] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3213–3223.

[50] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 4990–4999.

[51] 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.*, vol. 115, no. 3, pp. 211–252, 2015.

[52] W. Wang, J. Shen, F. Guo, M.-M. Cheng, and A. Borji, "Revisiting video saliency: A large-scale benchmark and a new model," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 4894–4903.

[53] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, 2017.

[54] D.-P. Fan, Z. Lin, Z. Zhang, M. Zhu, and M.-M. Cheng, "Rethinking rgb-d salient object detection: Models, data sets, and large-scale benchmarks," *IEEE T. Neural Netw. Learn. Syst.*, 2021.

[55] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *Eur. Conf. Comput. Vis.*, 2018, pp. 720–736.

[56] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 909–918.

[57] Y. Zeng, P. Zhang, J. Zhang, Z. Lin, and H. Lu, "Towards high-resolution salient object detection," in *Int. Conf. Comput. Vis.*, 2019.

[58] 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.*, 2014, pp. 280–287.

[59] J. R. Hall, I. C. Cuthill, R. Baddeley, A. J. Shohet, and N. E. Scott-Samuel, "Camouflage, detection and identification of moving targets," *Proc. Royal Soc. B: Biological Sciences*, vol. 280, no. 1758, p. 20130064, 2013.

[60] 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.

[61] N. Xu, B. Price, S. Cohen, and T. Huang, "Deep image matting," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2970–2979.

[62] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, 2021.

[63] S. Liu, D. Huang *et al.*, "Receptive field block net for accurate and fast object detection," in *Eur. Conf. Comput. Vis.*, 2018, pp. 385–400.

[64] 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.

[65] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2818–2826.

[66] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," in *Brit. Mach. Vis. Conf.*, 2014.

[67] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," *Adv. Neural Inform. Process. Syst.*, vol. 27, pp. 1269–1277, 2014.

[68] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1568–1576.

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

[70] S. Chen and Y. Fu, "Progressively guided alternate refinement network for rgb-d salient object detection," in *Eur. Conf. Comput. Vis.*, 2020, pp. 520–538.

[71] D.-P. Fan, G.-P. Ji, X. Qin, and M.-M. Cheng, "Cognitive vision inspired object segmentation metric and loss function (in chinese)," *SCIENTIA SINICA Informationis*, 2021.

[72] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 936–944.

[73] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[74] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 6230–6239.

[75] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *D. Learn. Med. Image Anal.*, 2018, pp. 3–11.

[76] 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.

[77] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 6409–6418.

[78] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3085–3094.

[79] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 4974–4983.

[80] J. Wei, S. Wang, and Q. Huang, "F3Net: Fusion, Feedback and Focus for Salient Object Detection," in *AAAI Conf. Art. Intell.*, 2020.

[81] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learn. Represent.*, 2015.

[82] 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.

[83] S.-M. Hu, D. Liang, G.-Y. Yang, G.-W. Yang, and W.-Y. Zhou, "Jittor: a novel deep learning framework with meta-operators and unified graph execution," *Science China Information Sciences*, vol. 63, no. 12, pp. 1–21, 2020.

[84] 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.

[85] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 733–740.

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

[87] G. Mori, "Guiding model search using segmentation," in *Int. Conf. Comput. Vis.*, 2005, pp. 1417–1423.

[88] P. Krahenbuhl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Adv. Neural Inform. Process. Syst.*, 2011, pp. 109–117.

[89] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 1999, pp. 377–384.

[90] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[91] A. Torralba, A. A. Efros *et al.*, "Unbiased look at dataset bias," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1521–1528.

[92] T. He, Y. Liu, C. Xu, X. Zhou, Z. Hu, and J. Fan, "A fully convolutional neural network for wood defect location and identification," *IEEE Access*, vol. 7, pp. 123 453–123 462, 2019.

[93] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen, "Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19," *IEEE Rev. Biomed. Eng.*, 2020.

[94] A. Treisman, "Features and objects: The fourteenth bartlett memorial lecture," *Q. J. Exp. Psychol. (Hove)*, vol. 40, no. 2, pp. 201–237, 1988.

[95] J. M. Wolfe, "Guided search 2.0 a revised model of visual search," *Psychon. Bull. Rev.*, vol. 1, no. 2, pp. 202–238, 1994.

[96] H. Fan, H. A. Miththanthaya, S. R. Rajan, X. Liu, Z. Zou, Y. Lin, H. Ling *et al.*, "Transparent object tracking benchmark," *arXiv preprint arXiv:2011.10875*, 2020.

[97] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, 2020.

[98] T. Afouras, A. Owens, J. S. Chung, and A. Zisserman, "Self-supervised learning of audio-visual objects from video," *arXiv preprint arXiv:2008.04237*, 2020.

[99] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 9729–9738.

[100] H. Lamdouar, C. Yang, W. Xie, and A. Zisserman, "Betrayed by motion: Camouflaged object discovery via motion segmentation," in *Asian Conf. Comput. Vis.*, 2020.

[101] Q. Zhang, N. Huang, L. Yao, D. Zhang, C. Shan, and J. Han, "Rgb-t salient object detection via fusing multi-level cnn features," *IEEE Trans. Image Process.*, vol. 29, pp. 3321–3335, 2019.

[102] D.-P. Fan, T. Li, Z. Lin, G.-P. Ji, D. Zhang, M.-M. Cheng, H. Fu, and J. Shen, "Re-thinking co-salient object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[103] Q. Fan, D.-P. Fan, H. Fu, C.-K. Tang, L. Shao, and Y.-W. Tai, "Group collaborative learning for co-salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[104] D.-P. Fan, W. Wang, M.-M. Cheng, and J. Shen, "Shifting more attention to video salient object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 8554–8564.

[105] A. Mondal, "Camouflaged object detection and tracking: A survey," *Int. J. Image Graph.*, vol. 20, no. 04, p. 2050028, 2020.

[106] M. Kalash, M. A. Islam, and N. Bruce, "Relative saliency and ranking: Models, metrics, data and benchmarks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[107] Q. Yao, M. Wang, Y. Chen, W. Dai, H. Yi-Qi, L. Yu-Feng, T. Wei-Wei, Y. Qiang, and Y. Yang, "Taking human out of learning applications: A survey on automated machine learning," *arXiv preprint arXiv:1810.13306*, 2018.

[108] K. Fu, D.-P. Fan, G.-P. Ji, Q. Zhao, J. Shen, and C. Zhu, "Siamese network for rgb-d salient object detection and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[109] J. Zhang, D.-P. Fan, Y. Dai, S. Anwar, F. Saleh, S. Aliakbarian, and N. Barnes, "Uncertainty inspired rgb-d saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[110] 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.

[111] 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.