优化无阈值显著目标检测的 F 度量

Kai Zhao[1], Shanghua Gao[1], Wenguan Wang[2], Ming-Ming Cheng[1*]
[1]TKLNDST, CS, Nankai University [2]Inception Institute of Artificial Intelligence
{kaizhao.net,shanghuagao,wenguanwang.ai}@gmail.com,cmm@nankai.edu.cn

摘要

当前基于 CNN 的显著目标检测 (SOD) 解决方案主要依赖于交叉熵损失 (CELoss) 的优化。然后，通常根据 F-measure 来评估检测到的显著性图的质量。在本文中，我们研究了一个有趣的问题：我们能否在 SOD 的训练和评估中始终使用 F-measure 公式？通过重新制定标准 F-measure，我们提出了 relaxed F-measure，它与后验是可微的，并且可以很容易地附加到 CNN 的后面作为损失函数。与梯度在饱和区域急剧下降的传统交叉熵损失相比，我们的损失函数 FLoss 即使在激活接近目标时也保持相当大的梯度。因此，FLoss 可以持续迫使网络产生极化激活。对几个流行数据集的综合基准测试表明，FLoss 以相当大的优势优于最先进的技术。更具体地说，由于极化预测，我们的方法能够在不仔细调整最佳阈值的情况下获得高质量的显著图，在实际应用中显示出显著的优势。代码和预训练模型可在 http://kaizhao.net/fmeasure 获得。

1. 简介

我们考虑显著对象检测 (SOD) 的任务，其中给定图像的每个像素都必须归类为显著（优秀）或不显著。人类视觉系统能够以独特的方式感知和处理视觉信号：感兴趣的区域被优先考虑和分析，而其他区域则较少受到关注。这种能力在计算机视觉社区中以“显著物体检测”的名义进行了很长时间的研究，因为它可以简化场景理解的过程 [4]。现代显著目标检测方法的性能通常根据 F-measure 进行评估。源于信息检索 [29]，F-measure 被广泛用作必须检索指定类元素的任务中的评估指标，尤其是当相关类很少时。给定每像素预测 $\hat{Y}(\hat{y}_i \in [0,1], i=1,...,|Y|)$ 和真实显著图 $Y(y_i \in \{0,1\}, i=1,...,|Y|)$，应用阈值 t 来获得二值化预测 $\dot{Y}^t(\dot{y}_i^t \in \{0,1\}, i=1,...,|Y|)$。然后将 F-measure 定义为精度和召回率的调和平均值：

$$F(Y,\dot{Y}^t)=(1+\beta^2)\frac{\text{precision}(Y,\dot{Y}^t)\cdot\text{recall}(Y,\dot{Y}^t)}{\beta^2\text{precision}(Y,\dot{Y}^t)+\text{recall}(Y,\dot{Y}^t)}, \quad (1)$$

其中 $\beta^2>0$ 是准确率和召回率之间的平衡因子。当 $\beta^2>1$ 时，F-measure 偏向于召回，否则偏向于精确度。

大多数基于 CNN 的显著性解决方案 [11, 16, 30, 9, 31, 39, 33] 主要依赖于 FCN [22] 架构中 cross-entropy loss (CELoss) 的优化，显著图的质量通常由 F-measure 评估。优化像素无关的 CELoss 可以看作是最小化平均绝对误差 (MAE=$\frac{1}{N}\sum_i^N |\hat{y}_i - y_i|$)，因为在这两种情况下每个预测/真值对独立工作，对最终得分的贡献相同。如果数据标签有偏差分布，用 CELoss 训练的模型会对多数类做出有偏差的预测。因此，使用 CELoss 训练的 SOD 模型保持先验偏差，并倾向于将未知像素预测为背景，从而导致低召回率检测。F-measure [29] 是一种更复杂、更全面的评估指标，它将精度和召回率合并为一个分数，并自动抵消正/负样本之间的不平衡。

在本文中，我们为 SOD 的训练和评估提供了统一的公式。通过直接采用评估指标，也就是 F-measure，作为优化目标，我们以端到端的方式执行

*M.M. Cheng is the corresponding author.

F-measure 最大化。为了进行端到端的学习，我们提出了 松弛的 F-measure 来克服标准 F-measure 公式中的不可微性。我们所提出的损失函数名为 FLoss，可与后验 $\hat{Y}$ 分解，因此可以毫不费力地将其附加到 CNN 的后面作为监督。我们在几种最先进的 SOD 架构上测试了 FLoss，并见证了明显的性能提升。此外，即使在饱和区域，FLoss 也保持相当大的梯度，从而导致对阈值稳定的极化预测。我们提出的 FLoss 具有三个有利的特性：

- 无阈值显著物体检测。使用 FLoss 训练的模型会生成对比显著图，其中前景和背景清晰地分开。因此，FLoss 可以在很宽的阈值范围内实现高性能。
- 能够处理不平衡的数据。F-measure 定义为精度和召回率的调和平均值，能够在不同类别的样本之间建立平衡。我们通过实验证明我们的方法可以在精度和召回率之间找到更好的折衷。
- 快速收敛。我们的方法在经过数百次迭代后迅速学会专注于显著对象区域，显示出快速的收敛速度。

2. 相关工作

我们回顾了几种基于 CNN 的 SOD 架构以及与 F-measure 优化相关的文献。

显著物体检测 (SOD)。 卷积神经网络 (CNN) 已被证明在计算机视觉的许多子领域中占据主导地位。自从 CNN 在 SOD 中出现以来，已经取得了重大进展。DHS net [19]是使用 CNN 进行 SOD 的先驱之一。DHS 首先生成带有全局线索的粗略显著图，包括对比度、客观性等。然后使用分层循环 CNN 逐步细化粗图。全卷积网络 (FCN) [22] 的出现提供了一种执行端到端像素级推理的优雅方式。DCL [16] 使用双流架构来处理像素和补丁级别的对比度信息。基于 FCN 的子流生成具有像素级精度的显著图，另一个网络流对每个对象段进行推理。最后，一个全连接的 CRF [14] 用于结合像素级和段级语义。

源于边缘检测的 HED [34]，聚合多尺度侧输出被证明在细化密集预测方面是有效的，尤其是当需要保留详细的局部结构时。在类似 HED 的架构中，更深的侧输出捕获丰富的语义，而更浅的侧输出包含高分辨率细节。结合这些不同级别的表示将导致显著的性能改进。DSS [11] 引入了跨不同侧输出的深到浅短连接，以细化具有深层语义特征的浅侧输出。深到浅的短连接使浅侧输出能够从背景中区分真实的显著对象，同时保持高分辨率。Liu et al. [18] 设计了一个基于池化的模块，以从自上而下的路径有效地融合卷积特征。Amulet [38] 也采用了自上而下细化的思想，同时赵 et al. [40] 通过双向细化对其进行了增强。后来，Wang et al. [32] 提出了一种视觉注意力驱动模型，该模型弥合了 SOD 和眼睛注视预测之间的差距。上面提到的这些方法试图通过引入更强大的网络架构来细化 SOD，从循环细化网络到多尺度侧输出融合。我们向读者推荐最近的调查 [3] 以了解更多详细信息。

F 度量优化。 尽管已被用作许多应用领域的通用性能指标，但直到最近，优化 F-measure 才引起较多关注。旨在优化 F-measure 的工作可以分为两个子类别 [6]：(a) 结构化损失最小化方法，例如 [24, 25]，在训练期间以优化 F-measure 作为目标；(b) 在推理阶段优化 F-measure 的插件规则方法 [13, 7, 26, 37]。

大部分注意力都集中在对后一个子类别的研究上：找到一个最佳阈值，在给定预测后验 $\hat{Y}$ 的情况下，该阈值导致最大 F 度量。关于在训练阶段优化 F-measure 的文章很少。Petterson et al. [24] 通过最大化与 F-measure 相关的损失函数来间接优化 F-measure。然后在他们的后续工作中 [25] 他们构造了离散 F-measure 的上限，然后通过优化其上限来最大化 F-measure 。这些先前的研究要么作为后处理，要么是不可微分也即后验，使它们难以应用于深度学习框架。

3. 优化 SOD 的 F-measure

3.1. 松弛的 F-measure

在标准 F-measure 中，真阳性、假阳性和假阴性定义为对应样本的数量：

$$\begin{aligned} TP(\dot{Y}^t, Y) &= \sum\nolimits_i 1(y_i == 1 \text{ and } \dot{y}_i^t == 1), \\ FP(\dot{Y}^t, Y) &= \sum\nolimits_i 1(y_i == 0 \text{ and } \dot{y}_i^t == 1), \\ FN(\dot{Y}^t, Y) &= \sum\nolimits_i 1(y_i == 1 \text{ and } \dot{y}_i^t == 0), \end{aligned} \tag{2}$$

其中 Y 是真值，$\dot{Y}^t$ 是阈值 t 二进制化的二值预测 Y 是真值显著图。$1(\cdot)$ 是一个指示函数，如果其参数为真，则计算为 1，否则为 0。

为了将 F-measure 合并到 CNN 中并以端到端的方式对其进行优化，我们定义了一个可分解的 F-measure，它在后验 $\hat{Y}$ 上是可微的。基于这个动机，我们根据连续后验 $\hat{Y}$ 重新表述真阳性、假阳性和假阴性：

$$\begin{aligned} TP(\hat{Y}, Y) &= \sum\nolimits_i \hat{y}_i \cdot y_i, \\ FP(\hat{Y}, Y) &= \sum\nolimits_i \hat{y}_i \cdot (1 - y_i), \\ FN(\hat{Y}, Y) &= \sum\nolimits_i (1 - \hat{y}_i) \cdot y_i \ . \end{aligned} \tag{3}$$

根据 Eq. 3 中的定义，精度 p 和召回 r 是：

$$p(\hat{Y}, Y) = \frac{TP}{TP + FP}, \quad r(\hat{Y}, Y) = \frac{TP}{TP + FN}. \tag{4}$$

最终，我们的 松弛 F-measure 可以写作：

$$\begin{aligned} F(\hat{Y}, Y) &= \frac{(1+\beta^2)p \cdot r}{\beta^2 p + r}, \\ &= \frac{(1+\beta^2)TP}{\beta^2(TP + FN) + (TP + FP)}, \\ &= \frac{(1+\beta^2)TP}{H}, \end{aligned} \tag{5}$$

其中 $H = \beta^2(TP + FN) + (TP + FP)$。由于 Eq. 3 中的松弛，Eq. 5 可与后验 $\hat{Y}$ 分解，因此可以集成到使用反向传播训练的 CNN 架构中。

3.2. 最大化 CNN 中的 F-measure

为了以端到端的方式最大化 CNN 中的 松弛 F-measure，我们将我们提出的基于 F-measure 的损失（FLoss）函数 $\mathcal{L}_F$ 定义为：

$$\mathcal{L}_F(\hat{Y}, Y) = 1 - F = 1 - \frac{(1+\beta^2)TP}{H}. \tag{6}$$

最小化 $\mathcal{L}_F(\hat{Y}, Y)$ 就相当于最大化 松弛 F-measure。再次注意 $\mathcal{L}_F$ 是直接从原始预测 $\hat{Y}$ 计算出来的，没有进行阈值处理。因此，$\mathcal{L}_F$ 在预测 $\hat{Y}$ 上是可微的，并且可以插入到 CNN 中。损失 $\mathcal{L}_F$ 对网络激活 $\hat{Y}$ 在位置 i 的偏导数是：

$$\begin{aligned} \frac{\partial \mathcal{L}_F}{\partial \hat{y}_i} &= -\frac{\partial F}{\partial \hat{y}_i} \\ &= -\Big(\frac{\partial F}{\partial TP} \cdot \frac{\partial TP}{\partial \hat{y}_i} + \frac{\partial F}{\partial H} \cdot \frac{\partial H}{\partial \hat{y}_i}\Big) \\ &= -\Big(\frac{(1+\beta^2)y_i}{H} - \frac{(1+\beta^2)TP}{H^2}\Big) \\ &= \frac{(1+\beta^2)TP}{H^2} - \frac{(1+\beta^2)y_i}{H}. \end{aligned} \tag{7}$$

Eq. 6 的另一种替代方法是最大化 F-measure 的对数似然：

$$\mathcal{L}_{\log F}(\hat{Y}, Y) = -\log(F), \tag{8}$$

对应梯度时：

$$\frac{\partial \mathcal{L}_{\log F}}{\partial \hat{y}_i} = \frac{1}{F}\left[\frac{(1+\beta^2)TP}{H^2} - \frac{(1+\beta^2)y_i}{H}\right]. \tag{9}$$

我们将从理论上和实验上分析 FLoss 相对于 Log-FLoss 和 CELoss 在生成偏振和高对比度显著图方面的优势。

3.3. FLoss vs 交叉熵 Loss

为了证明我们的 FLoss 优于替代者 Log-FLoss 和 cross-entropy loss (CELoss)，我们比较了这三个损失函数的定义、梯度和曲面图。CELoss 的定义是：

$$\mathcal{L}_{CE}(\hat{Y}, Y) = -\sum\nolimits_i^{|Y|} \left(y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)\right), \tag{10}$$

其中 i 是输入图像的空间位置，$|Y|$ 是输入图像的像素数。$\mathcal{L}_{CE}$ 的预测 $\hat{y}_i$ 的梯度为：

$$\frac{\partial \mathcal{L}_{CE}}{\partial \hat{y}_i} = \frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i}. \tag{11}$$

如公式 7 和公式 11 所示，CELoss 的梯度 $\frac{\partial \mathcal{L}_{CE}}{\partial \hat{y}_i}$ 仅依赖于单个像素 i 的预测/真实情况；而在 FLoss

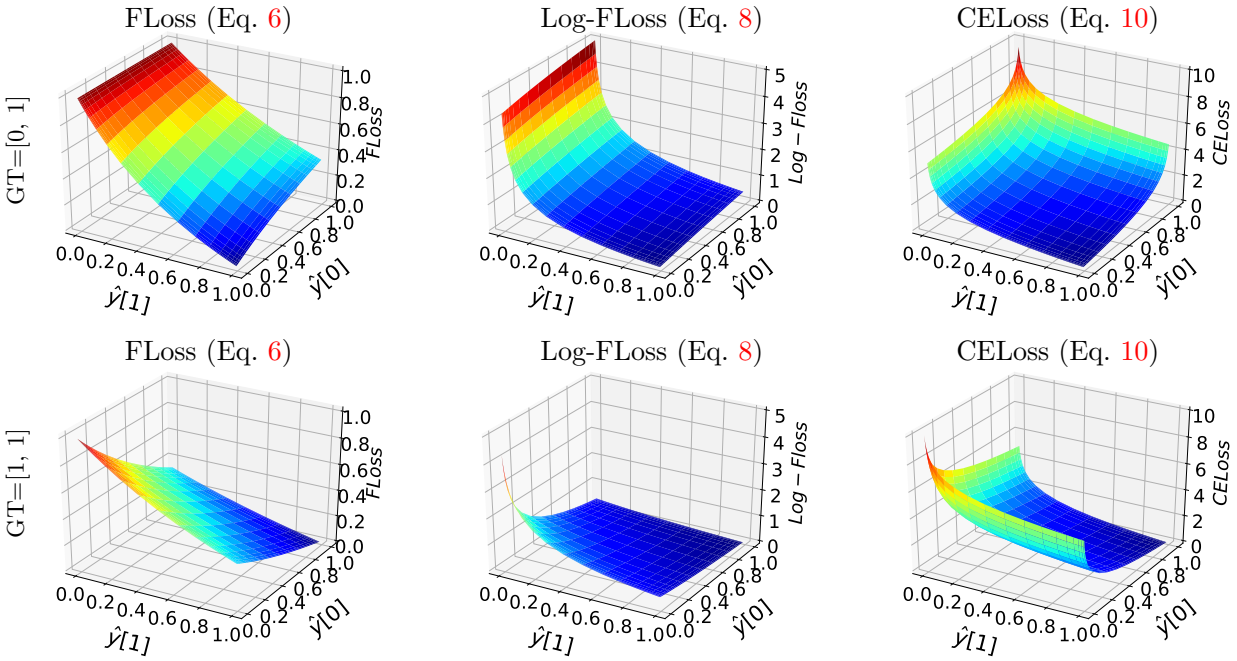

图 1. 2 点 2 类分类情况下不同损失函数的曲面图。列从左至右：F-measure loss defined in 公式 6中定义的 F-measure 损失，log F-measure loss defined in 公式 8中定义的 log F-measure 损失和 cross-entropy loss in 公式 10中定义的交叉熵损失。在顶行中，真值情况是 [0, 1]，在底行中，真值情况是 [1, 1]。与交叉熵损失和 Log-FLoss 相比，FLoss 即使在饱和区域也保持相当大的梯度，这将迫使产生极化预测。

中，$\frac{\partial \mathcal{L}_F}{\partial \hat{y}_i}$ 是由图像中所有像素的预测和真实情况全局决定的。我们进一步比较了两点二元分类问题中 FLoss、Log-FLoss 和 CELoss 的曲面图。结果如图 1。两个空间轴代表预测值 $\hat{y}_0$ 和 $\hat{y}_1$，z 轴表示损失值。

如图 1所示，FLoss 的梯度与 CELoss 和 Log-FLoss 的梯度不同有两个方面：(1) 有限梯度：即使预测与真实情况相距甚远，FLoss 也保持有限的梯度值。这对于 CNN 训练至关重要，因为它可以防止臭名昭著的梯度爆炸问题。因此，正如我们的实验所证明的那样，FLoss 在训练阶段允许更大的学习率。(2) 饱和区有相当大的梯度：在 CELoss 中，当预测接近真实值时梯度会衰减，而 FLoss 即使在饱和区域也保持相当大的梯度。这将迫使网络进行两极分化的预测。图 3 中的显著检测示例说明了“高对比度”和极化预测。

4. 实验及分析

4.1. 实验配置

数据集和数据增强。为了公平比较，我们在 MSRA-B [20] 训练集上统一训练我们的模型和其他对比模型。总共有 5000 张图像的 MSRA-B 数据集被平均分成训练/测试子集。我们在其他 5 个 SOD 数据集上测试训练模型：ECSSD [35], HKU-IS [15], PASCALS [17], SOD [23], 和 DUT-OMRON [23]。表格 1 中显示了这些数据集的更多统计信息。

值得一提的是，数据集的挑战性程度是由许多因素决定的，如图像数量、图像中目标数量、显著目标与背景的对比度、显著目标结构的复杂性，显著目标的中心偏差和图像的大小方差等。分析这些细节超出了本文的范围，我们建议读者参考 [8] 对数据集进行更多分析。

数据增强对于为训练深度 CNN 生成足够的数据至关重要。我们公平地为原始实现及其 FLoss 变体执行数据增强。对于 DSS [11] 和 DHS [19] 架构，我们只对训练图像和显著图执行水平翻转，就

数据集	# 图片	年份	来源	对比度
MSRA-B [20]	5000	2011	TPAMI	High
ECSSD [35]	1000	2013	CVPR	High
HKU-IS [15]	1447	2015	CVPR	Low
PASCALS [17]	850	2014	CVPR	Medium
SOD [23]	300	2010	CVPRW	Low
DUT-OMRON [36]	5168	2013	CVPR	Low

表 1. SOD 数据集的统计。“# 图片” 表示数据集中的图像数量，而 “对比度” 表示前景/背景之间的总体对比度。对比度越低，数据集越具有挑战性。

像 DSS 一样。Amulet [38] 只允许 256×256 输入。我们随机裁剪/填充原始数据以获得方形图像，然后调整它们的大小以满足形状要求。

网络架构和超参数。我们在 3 种基线方法上测试我们提出的 FLoss：Amulet [38], DHS [20] 和 DSS [11]。为了验证 FLoss 的有效性（公式 6），我们将原始实现的损失函数替换为 FLoss，并保持所有其他配置不变。如 Sec. 3.3 中所述，由于梯度有限，FLoss 允许更大的基本学习率。我们使用基本学习率 10^4 乘以原始设置。例如，在 DSS 中，基本学习率为 10^{-8}，而在我们的 F-DSS 中，基本学习率为 10^{-4}。为了公平比较，所有其他超参数都与原始实现一致。

评估指标。我们根据最大 F 测量 (MaxF)、平均 F 测量 (MeanF) 和平均绝对误差 (MAE $=\frac{1}{N}\sum_i^N|\hat{y}_i-y_i|$) 来评估显著图的性能。. 公式 1 中的因子 β^2 设置为 0.3，如 [1, 11, 16, 19, 30]。通过将系列阈值 $t\in\mathcal{T}$ 应用于显著图 $\hat{Y}$，我们获得了具有不同精度、召回率和 F 度量的二值化显著图 $\dot{Y}^t$。

然后通过穷举测试集得到最优阈值 t_o：

$$t_o=\underset{t\in\mathcal{T}}{\operatorname{argmax}}\,F(Y,\dot{Y}^t). \tag{12}$$

最后，我们使用 t_o 对预测进行二值化并评估最佳 F-measure：

$$\text{MaxF}=F(Y,\dot{Y}^{t_o}), \tag{13}$$

其中 $\dot{Y}^{t_o}$ 是用 t_o 二进制化的二值显著图。MeanF 是不同阈值下的平均 F-measure：

$$\text{MeanF}=\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}F(Y,\dot{Y}^t), \tag{14}$$

其中 $\mathcal{T}$ 是可能阈值的集合。

4.2. Log-FLoss vs FLoss

首先，我们将 FLoss 与其替代方案进行比较，即公式 8 中定义的 Log-FLoss，以证明我们的选择是合理的。正如在 Sec. 3.3 中分析的那样，FLoss 享有交叉熵损失和 Log-FLoss 在饱和区域具有大梯度的优势。

为了通过实验验证我们的假设，即 FLoss 将产生高对比度预测，我们分别用 FLoss 和 Log-FLoss 训练 DSS [11] 模型。训练数据是 MSRA-B [20] 并且除了基本学习率，超参数与原始实现保持不变。如 Sec. 3.3 中所述，我们将基本学习率调整为 10^{-4}，因为我们的方法可以接受更大的学习率。定量结果在表 2 中，一些检测到的显著性图示例如图 2 所示。

尽管 Log-FLoss 和 FLoss 都使用 F-measure 作为最大化目标，但 FLoss 导出具有高前景-背景对比度的极化预测，如图 2所示。可以从表 2 中得出相同的结论，其中 FLoss 实现了更高的平均 F-measure。这表明 FLoss 在更大的阈值下实现了更高的 F-measure 分数。

4.3. Evaluation results on open Benchmarks

我们将所提出的方法与 5 个流行数据集的几个基线进行比较。一些示例检测结果如图 3所示，综合定量比较在表 3中。一般来说，与基于交叉熵损失 (CELoss) 的方法相比，基于 FLoss 的方法可以获得相当大的改进，尤其是在平均 F-measure 和 MAE 方面。这主要是因为我们的方法对阈值是稳定的，从

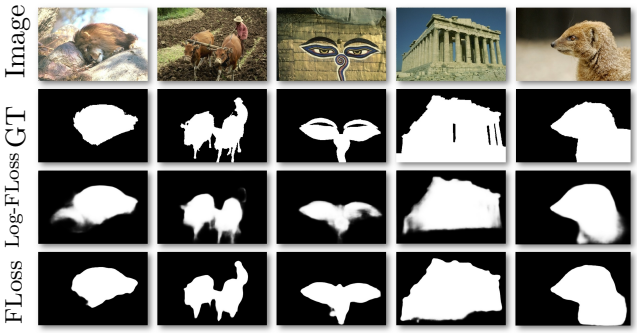

图 2. FLoss（底部）和 Log-FLoss（中间）的显著性图示例。我们提出的 FLoss 方法可以产生高对比度的显著图。

	Training data		ECSSD [35]			HKU-IS [15]			PASCALS [17]			SOD [23]			DUT-OMRON [23]		
Model	Train	#Images	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE
Log-FLoss	MB [20]	2.5K	.909	.891	.057	.903	.881	.043	.823	.808	.101	.838	.817	.122	.770	.741	.062
FLoss	MB [20]	2.5K	.914	.903	.050	.908	.896	.038	.829	.818	.091	.843	.838	.111	.777	.755	.067

表 2. Log-FLoss (Eq. 8) 和 FLoss (Eq. 6) 的性能比较。在 MaxF、MeanF 和 MAE 方面，FLoss 在大多数数据集上的表现优于 Log-FLoss。特别是 FLoss 由于其高对比度预测，在 MeanF 方面享有很大的改进。

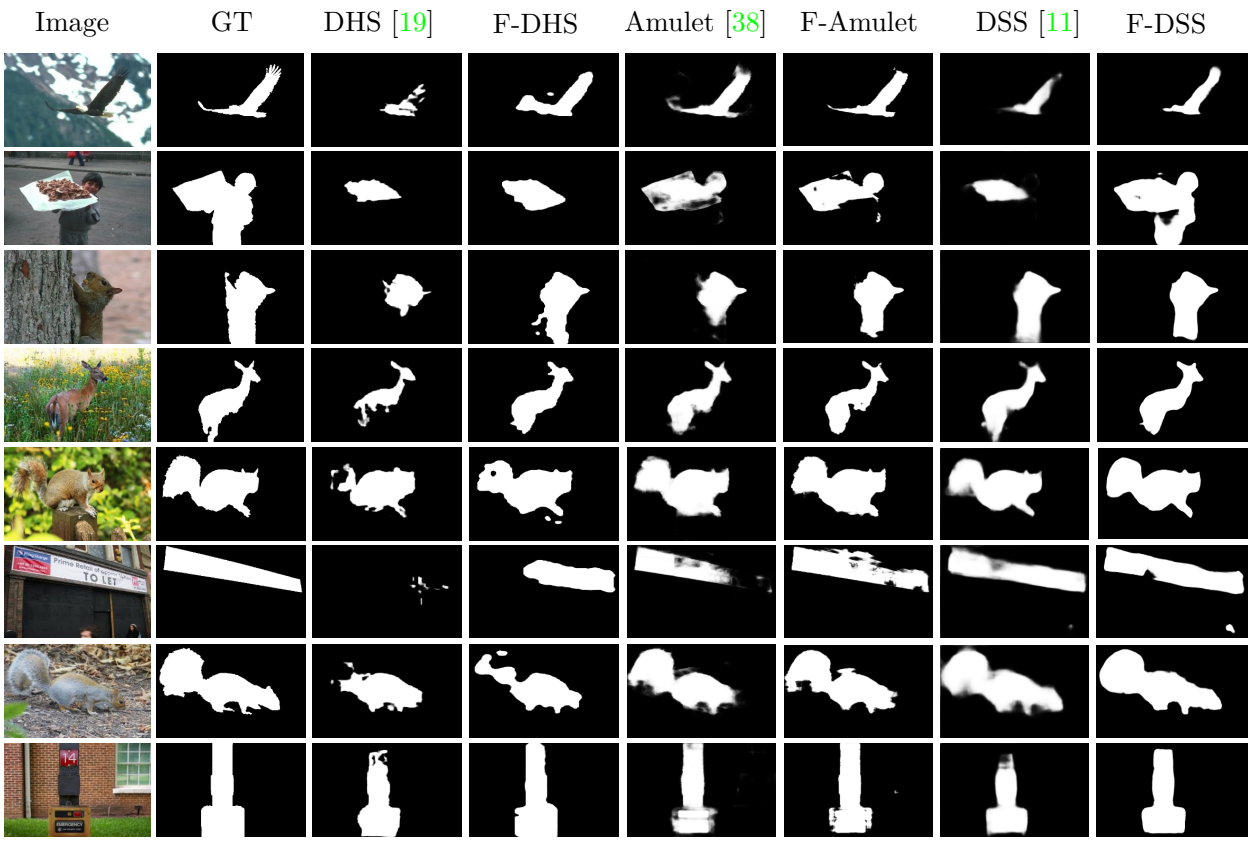

图 3. 几个流行数据集上的显著对象检测示例。F-DHS、F-Amulet 和 F-DSS 表示使用我们提出的 FLoss 训练的原始架构。FLoss 导致强烈的显著置信度，尤其是在对象边界上。

而导致在宽阈值范围下的高性能显著图。如图 3 所示，及 Sec. 3.3 所述，在我们检测到的显著图中，前景（显著对象）和背景很好地分离。

4.4. 无阈值显著对象检测

最先进的 SOD 方法 [11, 16, 19, 38] 通常按照如下方法评估最大 F-measure：(a) 使用预训练模型获得显著图 $\hat{Y}_i$；(b) 通过对测试集（公式 12）的详尽搜索来调整最佳阈值 t_o，并使用 t_o 对预测进行二值化；(c) 根据公式 13 评估最大 F-measure。

上述程序有一个明显的缺陷：通过对测试集的详尽搜索获得最佳阈值。这样的过程对于现实世界的应用程序是不切实际的，因为我们不会对测试数据进行标注。即使我们在一个数据集上调整了最佳阈值，也不能广泛应用于其他数据集。

我们从两个方面进一步分析了模型对阈值的敏感性：(1) 不同阈值下的模型性能，反映了一种方法对阈值变化的稳定性，(2) 不同数据集上最优阈值 t_o

	Training data		ECSSD [35]			HKU-IS [15]			PASCALS [17]			SOD [23]			DUT-OMRON [23]		
Model	Train	#Images	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE
RFCN [30]	MK [5]	10K	.898	.842	.095	.895	.830	.078	.829	.784	.118	.807	.748	.161	-	-	-
DCL [16]	MB [20]	2.5K	.897	.847	.077	.893	.837	.063	.807	.761	.115	.833	.780	.131	.733	.690	.095
DHS [19]	MK [5]+D [23]	9.5K	.905	.876	.066	.891	.860	.059	.820	.794	.101	.819	.793	.136	-	-	-
Amulet [38]	MK [5]	10K	.912	.898	.059	.889	.873	.052	.828	.813	.092	.801	.780	.146	.737	.719	.083
DHS [19]	MB	2.5K	.874	.867	.074	.835	.829	.071	.782	.777	.114	.800	.789	.140	.704	.696	.078
DHS+FLoss [19]	MB	2.5K	.884	.879	.067	.859	.854	.061	.792	.786	.107	.801	.795	.138	.707	.701	.079
Amulet [38]	MB	2.5K	.881	.857	.076	.868	.837	.061	.775	.753	.125	.791	.776	.149	.704	.663	.098
Amulet-FLoss	MB	2.5K	.894	.883	.063	.880	.866	.051	.791	.776	.115	.805	.800	.138	.729	.696	.097
DSS [11]	MB	2.5K	.908	.889	.060	.899	.877	.048	.824	.806	.099	.835	.815	.125	.761	.738	.071
DSS+FLoss	MB	2.5K	.914	.903	.050	.908	.896	.038	.829	.818	.091	.843	.838	.111	.777	.755	.067

表 3. 6 个流行数据集上不同方法的定量比较。我们提出的 FLoss 在 MAE（越小越好）和 F-measure（越大越好）方面都提高了性能。特别是在 Mean F-measure 方面，我们以非常明显的优势超越了最先进的技术，因为我们的方法能够产生高对比度预测，可以在广泛的阈值范围内实现高 F-measure。

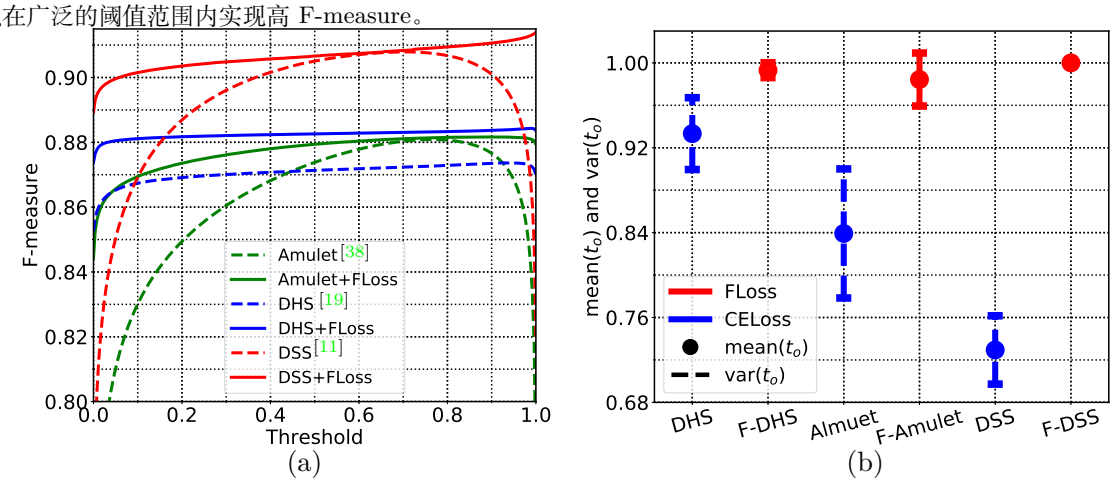

图 4. (a) ECSSD 数据集上不同阈值下的 F-measure。(b) 最优阈值 t_o 的均值和方差。基于 FLoss 的方法在不同的数据集（较低的 t_o 方差）和不同的主干架构（F-DHS、F-Amulet 和 F-DSS 保持非常接近的平均值 t_o）中保持稳定的 t_o。

的均值和方差，代表 t_o 在一个数据集上调整到其他数据集的泛化能力。

Fig. 4 (a) 说明了对应不同的阈值下的 F-measure。对于大多数没有 FLoss 的方法，F-measure 随阈值变化剧烈，最大 F-measure（MaxF）只出现在狭窄的阈值跨度内。而基于 FLoss 的方法几乎不受阈值变化的影响。

Fig. 4 (b) 反映了 t_o 在不同数据集上的均值和方差。传统方法（DHS、DSS、Amulet）在不同的数据集上表现出不稳定的 t_o，其巨大的差异证明了这一点。而基于 FLoss 的方法（F-DHS、F-Amulet、F-DSS）的 t_o 在不同数据集和不同骨干网络架构中保持不变。

总而言之，我们提出的的 FLoss 在三个方面对阈值 t 是稳定的：(1) 在较宽的阈值范围内实现高性能；(2) 在一个数据集上调整的最佳阈值 t_o 可以转移到其他数据集，因为 t_o 在不同的数据集上略有不同；(3) 从一种骨干架构中获得的 t_o 可以应用于其他架构。

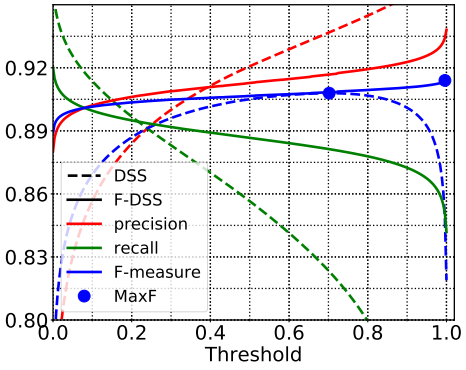

图 5. DSS (- - -) 和 F-DSS (—) 在不同阈值下的Precision, Recall, F-measure 和最大 F-measure (•)。DSS 倾向于将未知像素预测为多数类——背景，导致精度高但召回率低。FLoss 能够在精度和召回率之间找到更好的折衷方案。

4.5. SOD 中的标签不平衡问题

前景和背景在 SOD 中存在偏差，其中大多数像素属于非显著区域。不平衡的训练数据将导致模型趋向于将未知像素预测为背景的局部极小值。因此，召回将成为评估期间性能的瓶颈，如图 5所示。

尽管为正/负样本分配损失权重是抵消不平衡问题的一种简单方法，但表 4 中的额外实验表明，我们的方法比简单地分配损失权重表现更好。我们用正/负样本之间的权重因子定义平衡交叉熵损失：

$$\mathcal{L}_{balance} = \sum_{i}^{|Y|} w_1 \cdot y_i \log \hat{y}_i + w_0 \cdot (1 - y_i) \log (1 - \hat{y}_i). \tag{15}$$

正如在 [34] 和 [28] 中所建议的，正/负样本的损失权重由小批量中的正/负比例决定：$w_1 = \frac{1}{|Y|}\sum_i^{|Y|} 1(y_i == 0)$ and $w_0 = \frac{1}{|Y|}\sum_i^{|Y|} 1(y_i == 1)$

4.6. 精确率和召回率之间的折衷

召回率和精确率是两个互相冲突的指标。在某些应用中，我们更关心召回率而不是精确率，而在其他任务中，精确率可能比召回率更重要。Eq. 1 中的 β^2 在评估特定任务的性能时平衡了精确率和召回率之间的偏差。比如最近边缘检测的研究使用 $\beta^2 = 1$ [2, 34, 28]，说明它对精度和召回率的考虑是平等的。而显著性检测 [1, 11, 16, 19, 30] 通常使用 $\beta^2 = 0.3$ 来强调召回的精度。

作为优化目标，FLoss 还应该能够平衡精度和召回率之间的优势。我们用不同的 β^2 训练模型，并在精确率、召回率和 F-measure 方面综合评估它们的性能。Fig. 6 中的结果表明 β^2 是精确率和召回率之间的偏差调整器：较大的 β^2 导致更高的召回率，而较低的 β^2 导致更高的精确率。

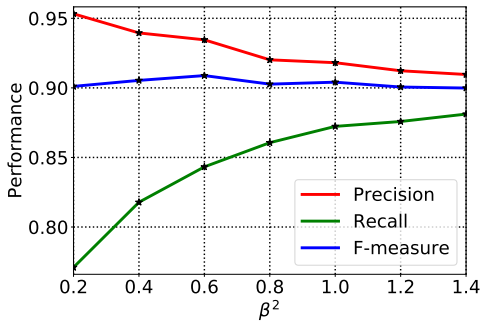

图 6. Precision, Recall, F-measure of model trained under different β^2 (Eq. 1). 精确率随着 β^2 的增加而减少，而召回率却随之增加。这个特性给了我们很大的灵活性来调整召回率和精确率之间的平衡：在召回优先应用程序中使用较大的 β^2 ，否则使用较低的 β^2 。

4.7. 更快的收敛速度和更好的性能

在这个实验中，我们训练了三个最先进的显著性检测器 (Amulet [38]，DHS [20]和 DSS [?]) 以及它们的 FLoss 对应项。然后我们绘制了所有方法在每个检查点的性能，以确定各自模型的收敛速度和收敛性能。所有模型在 MB [20]数据集上训练，并在 ECSSD [35]数据集上测试。结果如图7。

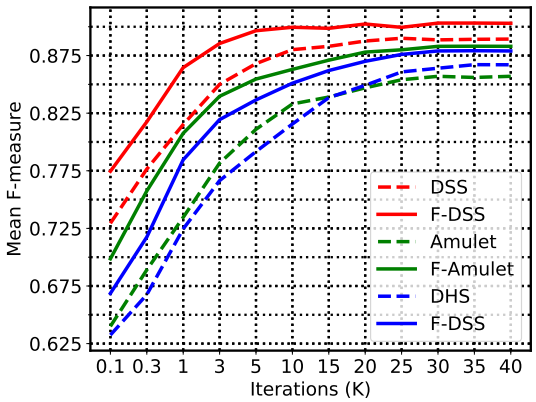

图 7. 性能 vs 训练迭代。我们的方法提供了更快的收敛性和较高的聚合性能。

我们观察到，我们的 FLoss 为所有这三个显著模型提供了每个迭代的性能提升。我们还发现，基

	Training data		ECSSD [35]			HKU-IS [15]			PASCALS [17]			SOD [23]			DUT-OMRON [23]		
Model	Train	#Images	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE
DSS [11]	MB [20]	2.5K	.908	.889	.060	.899	.877	.048	.824	.806	.099	.835	.815	.125	.761	.738	.071
DSS+Balance	MB [20]	2.5K	.910	.890	.059	.900	.877	.048	.827	.807	.097	.837	.816	.124	.765	.741	.069
DSS+FLoss	MB [20]	2.5K	.914	.903	.050	.908	.896	.038	.829	.818	.091	.843	.838	.111	.777	.755	.067

表 4. 原始交叉熵损失（Eq. 10）、平衡交叉熵损失（Eq. 15）和我们提出的 FLoss（Eq. 15）的性能比较。 6）。原始交叉熵学习对主要类（背景）有偏见的先验。召回率低证明了这一点：由于先验偏差，许多正点被错误预测为负。通过在前景/背景样本上分配损失权重，平衡交叉熵损失可以缓解不平衡问题。我们提出的方法比 平衡交叉熵损失表现更好，因为 F-measure 标准可以自动调整数据不平衡。

于 floss 的方法可以快速学会关注突出的目标区域，并在数百次迭代后获得较高的 F-measure 评分。而基于交叉熵的方法产生模糊的输出，不能很好地定位显著区域。如图 ??所示，基于 FLoss 的方法比其交叉熵竞争对手收敛更快，收敛性能更高。

5. 总结

在本文中，我们提出直接最大化 F-measure 用于 SOD 任务。我们引入了对预测后端可微的 FLoss 作为 cnn 的优化目标。所提出的方法在更好地处理有偏差的数据分布方面取得了更好的性能。此外，我们的方法对阈值是稳定的，并且能够在很宽的阈值范围内生成高质量的显著图，在实际应用中显示出巨大的潜力。通过调整 β^2 因子，可以很容易地调整精确率和召回率之间的折衷，从而灵活地处理各种应用。几个流行数据集的综合基准证明了所提出方法的优势。

Future work. 104/5000 我们计划使用最新的骨干模型例如 [10, 27]来提高所提方法的性能和效率。此外，FLoss 对其他二值密集预测任务，如边缘检测 [21]、阴影检测 [12]和骨架检测 [40]也有潜在的帮助。

致谢。 该研究得到了国家自然科学基金（61572264、61620106008）、国家青年人才支持计划和天津市自然科学基金（17JCJQJC43700、18ZXZNGX00110）的支持。

参考文献

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In CVPR, pages 1597–1604, 2009. 5, 8

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. IEEE PAMI, 33(5):898–916, 2011. 8

[3] 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. 2

[4] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. TIP, 24(12):5706–5722, 2015. 1

[5] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. IEEE PAMI, 37(3):569–582, 2015. 7

[6] K. Dembczynski, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Hüllermeier. Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In ICML, pages 1130–1138, 2013. 2

[7] K. J. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier. An exact algorithm for f-measure maximization. In NeurIPS, pages 1404–1412, 2011. 2

[8] D. Fan, J. Liu, S. Gao, Q. Hou, A. Borji, and M. Cheng. Salient objects in clutter: Bringing salient object detection to the foreground. ECCV, 2018. 4

[9] D.-P. Fan, W. Wang, M.-M. Cheng, and J. Shen. Shifting more attention to video salient object detection. In CVPR, pages 8554–8564, 2019. 1

[10] 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 TPAMI, 2019. 9

[11] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. IEEE TPAMI, 41(4):815–828, 2019. 1, 2, 4, 5, 6, 7, 8, 9

[12] X. Hu, L. Zhu, C.-W. Fu, J. Qin, and P.-A. Heng. Direction-aware spatial context features for shadow detection. In CVPR, June 2018. 9

[13] M. Jansche. A maximum expected utility framework for binary sequence labeling. In Proceedings of the Annual Meeting of the Association of Computational Linguistics, pages 736–743, 2007. 2

[14] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In NeurIPS, pages 109–117, 2011. 2

[15] G. Li and Y. Yu. Visual saliency based on multiscale deep features. CVPR, 2015. 4, 5, 6, 7, 9

[16] G. Li and Y. Yu. Deep contrast learning for salient object detection. In CVPR, pages 478–487, 2016. 1, 2, 5, 6, 7, 8

[17] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In CVPR, pages 280–287, 2014. 4, 5, 6, 7, 9

[18] 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 CVPR, 2019. 2

[19] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In CVPR, pages 678–686, 2016. 2, 4, 5, 6, 7, 8

[20] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. IEEE PAMI, 33(2):353–367, 2011. 4, 5, 6, 7, 8, 9

[21] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, and J. Tang. Richer convolutional features for edge detection. IEEE TPAMI, 41(8):1939 – 1946, 2019. 9

[22] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, pages 3431–3440, 2015. 1, 2

[23] V. Movahedi and J. H. Elder. Design and perceptual validation of performance measures for salient object segmentation. In CVPR-workshop, pages 49–56, 2010. 4, 5, 6, 7, 9

[24] J. Petterson and T. S. Caetano. Reverse multi-label learning. In NeurIPS, pages 1912–1920, 2010. 2

[25] J. Petterson and T. S. Caetano. Submodular multi-label learning. In NeurIPS, pages 1512–1520, 2011. 2

[26] J. R. Quevedo, O. Luaces, and A. Bahamonde. Multi-label classifiers with a probabilistic thresholding strategy. Pattern Recognition, 45(2):876–883, 2012. 2

[27] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In CVPR, June 2018. 9

[28] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In CVPR, pages 3982–3991, 2015. 8

[29] C. J. Van Rijsbergen. Foundation of evaluation. Journal of Documentation, 30(4):365–373, 1974. 1

[30] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In ECCV, pages 825–841, 2016. 1, 5, 7, 8

[31] W. Wang, J. Shen, M.-M. Cheng, and L. Shao. An iterative and cooperative top-down and bottom-up inference network for salient object detection. In CVPR, pages 5968–5977, 2019. 1

[32] W. Wang, J. Shen, X. Dong, and A. Borji. Salient object detection driven by fixation prediction. In CVPR, pages 1711–1720, 2018. 2

[33] W. Wang, S. Zhao, J. Shen, S. C. Hoi, and A. Borji. Salient object detection with pyramid attention and salient edges. In CVPR, pages 1448–1457, 2019. 1

[34] S. Xie and Z. Tu. Holistically-nested edge detection. In ICCV, pages 1395–1403, 2015. 2, 8

[35] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In CVPR, pages 1155–1162, 2013. 4, 5, 6, 7, 8, 9

[36] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In CVPR, pages 3166–3173, 2013. 5

[37] N. Ye, K. M. A. Chai, W. S. Lee, and H. L. Chieu. Optimizing f-measures: a tale of two approaches. In ICML, pages 1555–1562, 2012. 2

[38] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. ICCV, 2017. 2, 5, 6, 7, 8

[39] J. Zhao, J. liu, D. Fan, Y. Cao, J. Yang, and M.-M. Cheng. Egnet:edge guidance network for salient object detection. In ICCV, Oct 2019. 1

[40] K. Zhao, W. Shen, S. Gao, D. Li, and M.-M. Cheng. Hi-Fi: Hierarchical feature integration for skeleton detection. In IJCAI, pages 1191–1197, 7 2018. 2, 9