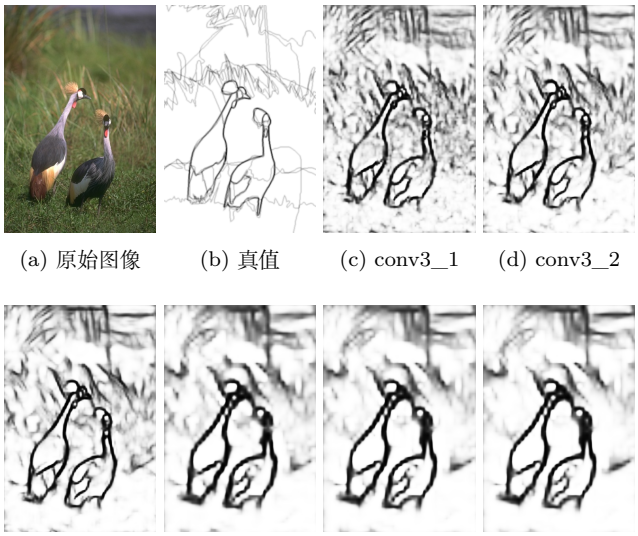
请勿直接用Google Scholar中的默认的会议版BibTex信息，IEEE TPAMI 2019期刊版Bib信息在项目主页上。

基于更丰富的卷积特征的边缘检测

Yun Liu[1] Ming-Ming Cheng[1] * Xiaowei Hu[1] Kai Wang[1] Xiang Bai[2]
[1]Nankai University [2]HUST
http://mmcheng.net/rcfedge/

摘要

在本文中，我们提出了一个基于更丰富的卷积特征（Richer Convolutional Features, RCF）的边缘检测器。由于自然图像中的物体具有各种不同的尺寸和比例，所以学习丰富的层次特征是边缘检测任务的关键。卷积神经网络（Convolutional Neural Network, CNN）已被证明对该任务很有效。并且，随着感受野的变大，CNN 中的卷积特征也会逐渐变得粗糙。根据上述观察，我们尝试使用更丰富的卷积特征来提高边缘检测性能。所提出的网络通过组合所有有意义的卷积特征来充分利用物体的多尺度和多层次信息，以此实现“图像-图像”的预测。当使用 VGG16 网络作为骨干网络时，我们可以在多个公开数据集上达到当前的最好性能。尤其当使用著名的 BSDS500 数据集进行评测时，我们的模型在 ODS F-measure 指标上达到 0.811 的同时，也保持了极高的检测速度（速度可达 8 FPS）。此外，快速版本的 RCF 也可实现在速度为 30 FPS 时，ODS F-measure 达到 0.806。



(a) 原始图像 (b) 真值 (c) conv3_1 (d) conv3_2
(e) conv3_3 (f) conv4_1 (g) conv4_2 (h) conv4_3

图 1. 我们基于 VGG16 [50] 设计了一个简单的神经网络来得到侧输出 conv3_1、conv3_2、conv3_3、conv4_1、conv4_2 和 conv4_3。可以清楚地看到，卷积特征逐渐变粗糙，中间层 conv3_1、conv3_2、conv4_1 和 conv4_2 包含许多其他层没有的细节信息。

1. 引言

边缘检测旨在从自然图像中提取视觉上显著的物体边缘，几十年来一直是计算机视觉领域重要且极具挑战性的任务之一。它通常被认为是一种底层技术，很多高层任务都已经得益于边缘检测的发展，比如物体检测 [17, 55]、拟物性采样 [9, 54, 60–62] 和图像分割 [1, 3, 8, 56] 等。

传统的边缘检测方法先提取亮度、颜色、梯度、纹理或其他手动设计特征（如 Pb [40]、gPb [2] 和 Sketch tokens [36] 等），然后使用复杂的学习方法 [14, 57] 来对边缘和非边缘像素进行分类。尽管近年来使用底层特征进行边缘检测的方法有了很大的进步 [33]，但其局限性也很明显。例如，通常情况下所定义的边缘是拥有语义意义的，而使用底层特征很难表示出物体级别的语义信息。在这种情况下，gPb [2] 和 Structured Edges [14] 都尝试使用复杂的策略来尽可能地捕获全局特征。

在过去的几年中，卷积神经网络（Convolutional Neural Network, CNN）通过大幅推进各种任务的发展，如图像分类 [31, 50, 52]、目标检测 [20, 21, 34, 43]

*M.M. Cheng (cmm@nankai.edu.cn) is the corresponding author.

和语义分割 [7, 38] 等，而在计算机视觉社区变得很流行。由于 CNN 具有强大地自动学习自然图像的高层表征的能力，因此使用 CNN 进行边缘检测已成为最近的趋势。一些著名的基于 CNN 的方法已经显著地推动了该领域的发展，如 DeepEdge [4]、N^4-Fields [19]、CSCNN [26]、DeepContour [47]、和 HED [58] 等。我们的方法也属于此类。

如图1所示，为了观察边缘检测中不同卷积层所学到的信息，我们用具有 5 个卷积阶段的 VGG16 [50] 构建了一个简单的网络来产生中间层的侧输出。我们发现卷积特征会逐渐变得粗糙，且中间层包含许多有用的细节信息。另一方面，由于丰富的卷积特征对于许多视觉任务都非常有效，因此许多研究人员都致力于探索更深的网络结构 [25]。但是，由于梯度消失/爆炸和训练数据不足（例如，用于边缘检测的数据）等原因，使网络架构变得很深时会难以收敛。那么，我们为什么不充分利用现有的卷积特征呢？我们的动机正是基于这些观察。与以前的基于卷积神经网络的方法不同，本文所提出的网络使用所有卷积层的卷积特征来以“图像-图像”的方式进行像素级预测，因此能够获得不同尺度的物体或物体部分的准确表征。具体来说，我们尝试用一个统一的框架来利用所有卷积层的卷积特征，这个框架也具有应用到其他视觉任务的潜力。通过精心设计的一个通用的策略来结合不同层次的卷积特征，我们的系统在边缘检测方面表现得非常出色。

当在 BSDS500 数据集 [2] 上评测所提出的方法时，我们在 ODS F-measure 为 0.811、速度为 8 FPS 时取得了检测性能和效率之间的最佳平衡，它甚至超过了人类对边缘感知的结果（ODS F-measure 为 0.803）。此外，我们还介绍了一个快速版本的 RCF，该版本可实现在速度为 30 FPS 时，ODS F-measure 达到 0.806。

2. 相关工作

边缘检测是计算机视觉中最基本的问题之一 [15, 18, 46]，国内外研究人员已经进行了近 50 年的相关研究，并取得了很多成就。从广义上讲，我们可以将这些方法大致分为三类：早期的开拓性方法、提取手工特征并进行学习的方法和基于深度学习的方法。在这里，我们简要回顾一下过去几十年来的一些代表性方法。

早期的开拓性方法主要利用了图像的强度和颜色梯度。Robinson [46] 提出了通过选择颜色坐标来提取视觉上显著的边缘的定量度量方法。[39, 53] 等提出了基于零交叉理论的算法。Sobel [51] 提出了著名的 Sobel 算子来计算图像的梯度图，然后通过对梯度图进行阈值化来得到边缘。Canny [6] 是 Sobel 的扩展，它添加了高斯平滑作为预处理步骤，并使用双阈值来得到边缘。通过这种方式，Canny 对噪声更加鲁棒。由于它较高的效率，它现在在各种任务中仍然很受欢迎。但是，早期的这些方法准确性较低，很难适用于当今的任务。

后来，研究人员开始使用一些底层信息（例如强度、梯度和纹理等）来手动设计特征，然后采用复杂的学习方法对边缘和非边缘像素进行分类 [13, 44]。Konishi 等人 [30] 通过学习两组边缘滤波器的响应的概率分布，提出了第一种数据驱动的方法。Martin 等人 [40] 将亮度、颜色和纹理的变化规范化为 Pb 特征，并训练一个分类器以结合这些特征中的信息。Arbeláez 等人 [2] 将 Pb 改进为 gPb，他们通过使用标准的标准化切割（Normalized Cuts）[48] 将上述局部信息整合进一个全局框架。Lim 等人 [36] 提出了新的可表示中间层信息的特征 Sketch tokens。Dollár 等人 [14] 使用随机决策森林来表示局部图像小块的结构。只要输入颜色和梯度特征，结构化森林就能输出高质量的边缘。然而，这些方法都是基于手工设计的特征的，对具有语义意义的边缘进行检测时，这些手工特征在表征高层信息时能力有限。

随着最近深度学习的快速发展，很多基于深度学习的方法被提出。Ganin 等人 [19] 通过结合 CNN 和最近邻搜索提出了 N^4-Fields 方法。Shen 等人 [47] 将边缘数据划分为多个子类，并通过学习模型参数来拟合每个子类。Hwang 等人 [26] 将边缘检测看成逐像素的分类问题。他们使用 DenseNet [27] 为每个像素提取一个特征向量，然后使用 SVM 分类器将每个像素分为边缘或非边缘类别。Xie 等人 [58] 提出了一个高效且准确的边缘检测器，HED，它可以实现“图像-图像”的训练和预测。HED 在 VGG16 [50] 每一个阶段的最后一个卷积层后连接一个侧输出层，这个侧输出层由一个卷积核为 1 的卷积层、一个反卷积层和一个 sigmoid 层组成。最近，Liu 等人 [35] 使用了由下而上的边缘生成的松弛标签来指导 HED 的训练过程，并在性能上取得了一些提升。Li 等人 [35] 提出了一个复杂的无监督学习模型，但其性能比使用有限的 BSDS500 数据集训练的模型差。

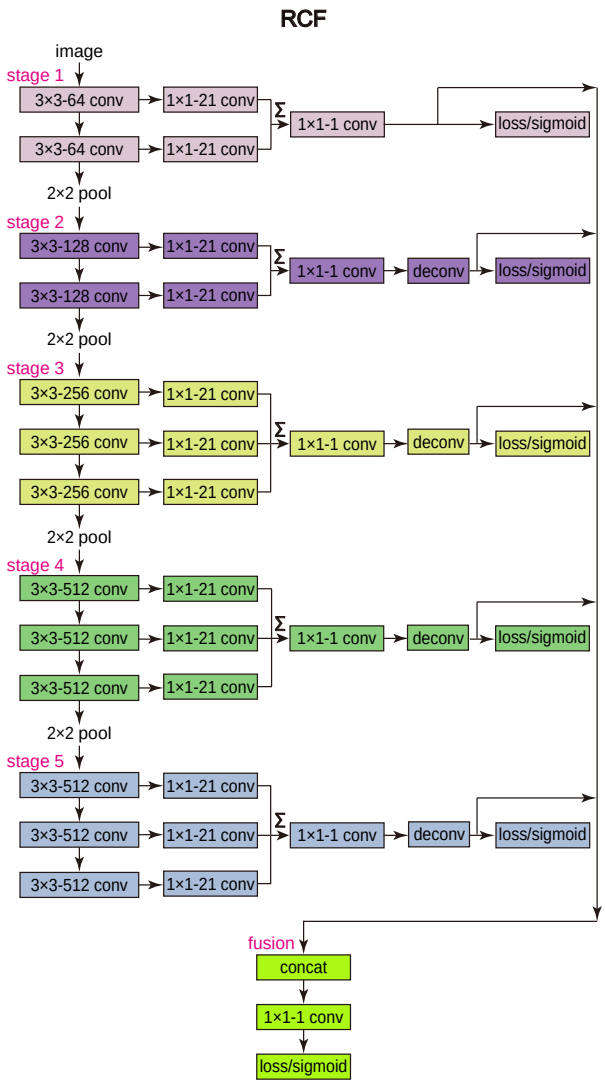


图 2. 所提出的 RCF 网络架构。输入为任意大小的图像，输出相同大小的边缘概率图。

前面所提到的基于 CNN 的模型已经显著提高了边缘检测的性能，但当对像素点进行分类时，这些方法通常仅使用每个卷积阶段的最后一层的特征，因此他们都损失了部分有用的卷积特征。为了解决这个问题，我们提出了一个可以有效地组合每个卷积层特征的全卷积网络。我们将在下面详细介绍我们的方法。

3. 更丰富的卷积特征

3.1. 网络架构

受之前深度学习相关研究 [20, 38, 43, 58] 的启发，我们通过修改 VGG16 网络 [50] 来设计我们的网络。由

表 1. 标准 VGG16 网络 [50] 的感受野和步幅的具体大小。

layer	conv1_1	conv1_2	pool1	conv2_1	conv2_2	pool2
rf size	3	5	6	10	14	16
stride	1	1	2	2	2	4
layer	conv3_1	conv3_2	conv3_3	pool3	conv4_1	conv4_2
rf size	24	32	40	44	60	76
stride	4	4	4	8	8	8
layer	conv4_3	pool4	conv5_1	conv5_2	conv5_3	pool5
rf size	92	100	132	164	196	212
stride	8	16	16	16	16	32

13 个卷积层和 3 个全连接层组成的 VGG16 网络已经在包括图像分类 [50] 和目标检测 [20, 21, 43] 在内的许多任务上实现了最佳性能。它的卷积层可以分为五个阶段，其中每个阶段之后都连接一个池化层。每个卷积层所学到的有用信息会随着其感受野的增加而越来越粗糙。在表1中可以看到不同层的感受野的具体大小。而我们网络设计的初衷就在于假设使用这种丰富的多层信息会对边缘检测有所帮助。

我们所提出的网络架构如图2所示。与标准的 VGG16 相比，我们做出如下修改：

- 由于全连接层不符合我们的全卷积神经网络的设计思想，而添加 pool5 层将使步幅增加两倍，这对定位边缘位置是有害的。因此，我们移除了所有的全连接层和 pool5 层。
- VGG16 中的每个卷积层后都再连接一个卷积核为 1×1、输出通道数为 21 的卷积层，然后将各个阶段内的卷积结果使用逐元素相加来生成混合特征。
- 在每个逐元素相加操作之后都再连接一个卷积核为 1×1、输出通道数为 1 卷积层，然后使用反卷积层对该特征图进行上采样。
- 在每个阶段的反卷积层之后连接一个交叉熵损失或者 sigmoid 层。
- 将所有反卷积层的输出结果都拼接到一起，然后使用一个 1×1 卷积层融合每个阶段的特征图。最后，再连接一个交叉熵损失或者 sigmoid 层来获取融合后的损失或者输出。

因此，我们将来自所有卷积层的分层的特征整合进一个整体框架内，且该框架的所有参数都是自动学习的。

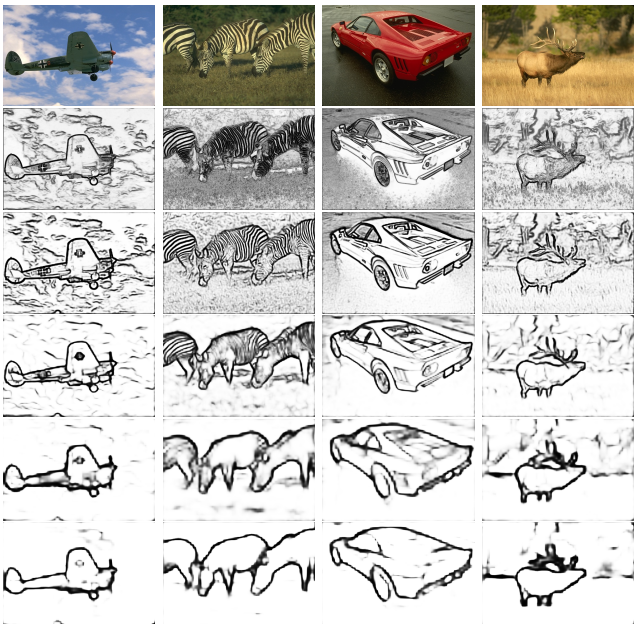


图 3. RCF 每个阶段的输出的示例。第一行是 BSDS500 [2] 中的原始图像，从第二行到第六行分别是第 1、2、3、4 和 5 阶段的输出。

由于 VGG16 中不同卷积层的感受野大小互不相同，因此我们的网络可以学习多尺度信息，包括底层信息和高层信息，而这些信息均有助于边缘检测。图3展示了每个阶段的中间结果。可以看出，从上到下，边缘响应变得越来越粗糙，同时对较大的物体或物体部分的边缘响应会更强烈，这与我们预期的卷积层会随着感受野的变大来检测较大的物体是一致的。由于 RCF 模型结合了所有可用的卷积层来获取更丰富的特征，因此有望提高边缘检测的准确性。

3.2. 标注鲁棒的损失函数

边缘检测的数据集通常是由多个标注者根据他们自身对物体或物体部分的认知来标记的。尽管每个人的认知略有不同，但是他们对同一张图像所标注的边缘却基本一致。对于每张图像，我们对所有真值图求平均来生成范围为 $[0,1]$ 的边缘概率图。其中，0 表示没有标注者将此像素标注为边缘，而 1 表示所有标注者都将此像素标注为边缘。我们将边缘概率高于 η 的像素视为正样本，将边缘概率等于 0 的像素视为负样本。如果一个像素的边缘概率大于 0 且小于 η，那么该像素点将是一个有争议的边缘点，无论将其视为正样本还是负样本都将干扰神经网络的训练，因此，我们忽略这类像素点。

我们按下式计算每个像素相对于标签的损失

$$l(X_i;W)=\begin{cases}\alpha\cdot log\ (1-P(X_i;W)) & if\ y_i=0,\\ 0 & if\ 0<y_i\le\eta,\\ \beta\cdot log\ P(X_i;W) & otherwise,\end{cases} \tag{1}$$

其中，

$$\begin{aligned}\alpha&=\lambda\cdot\frac{|Y^+|}{|Y^+|+|Y^-|},\\ \beta&=\frac{|Y^-|}{|Y^+|+|Y^-|}.\end{aligned} \tag{2}$$

Y^+ 和 Y^- 分别表示正样本集和负样本集。超参数 λ 用于平衡正负样本。在像素 i 处的激活值（CNN 特征向量）和真值边缘概率分别用 X_i 和 y_i 表示。$P(X)$ 是标准的 sigmoid 函数，W 表示神经网络中所有需要学习的参数。因此，我们所改进的损失函数可以表示为

$$L(W)=\sum_{i=1}^{|I|}\Big(\sum_{k=1}^{K}l(X_i^{(k)};W)+l(X_i^{fuse};W)\Big), \tag{3}$$

其中，$X_i^{(k)}$ 是来自阶段 k 的激活值，而 X_i^{fuse} 表示来自融合层的激活值。$|I|$ 是图像 I 中的像素数，K 是神经网络的阶段数（这里为 5）。

3.3. 多尺度的分层边缘检测

在单尺度边缘检测中，我们将原图像输入到经过微调的 RCF 网络中，然后输出边缘概率图。为了进一步提高边缘的质量，我们在测试时使用了图像金字塔。具体来说，我们通过缩放图像来构建图像金字塔，再分别将金字塔中的每一张图像输入到我们的单尺度检测器中。然后，使用双线性插值将得到的所有边缘概率图缩放到原图像大小。最后，通过求这些边缘图的平均值来获得最终的预测图。图4展示了我们所提出的多尺度算法的流程。此外，我们还尝试了使用加权和的方式来获得最终的预测图，但最后发现使用简单的平均操作的效果要更好。考虑到精度和速度之间的权衡，我们在本文中使用 0.5、1.0 和 1.5 三个尺度。在 BSDS500 数据集 [2] 上进行评测时，尽管使用这个简单的多尺度策略会让检测速度从 30 FPS 下降到 8 FPS，但却将 ODS F-measure 从 0.806 提高到 0.811。具体实验细节请参见第4节。

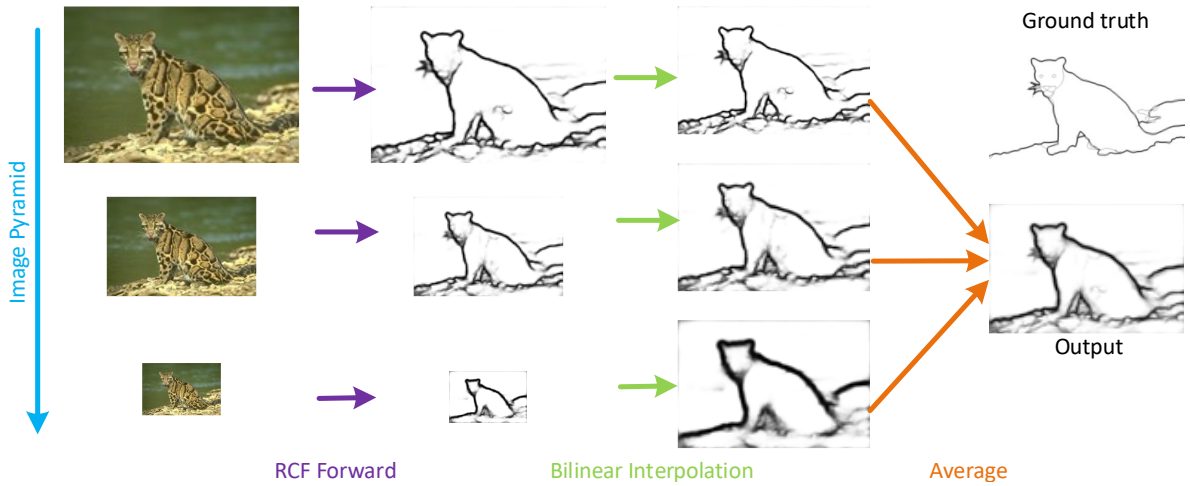


图 4. 所提出的多尺度算法的流程图。首先，缩放原图像的大小来构建图像金字塔，并将这些多尺度图像输入到 RCF 网络进行正向传播。然后，我们使用双线性插值法将生成的边缘响应图还原为原始大小。最后，将这些边缘图进行简单的平均就可以得到高质量的边缘。

3.4. 与 HED 的比较

我们所提出的 RCF 与 HED [58] 相比有三个最明显的区别。首先，HED 仅考虑 VGG16 各个阶段中的最后一个卷积层，因此忽略了许多对边缘检测有用的信息。与之对比，RCF 使用来自所有卷积层的更丰富的特征，使得它可以准确地捕获更多的各种尺度下的物体或物体部分的边缘。其次，本文提出了一个新的损失函数来合理地处理训练样本。我们只考虑被大多数标注者标记为正样本的边缘像素，由于这些边缘像素是高度一致的，因此易于训练。我们忽略了一些由少量标注者标注的边缘像素，因为他们的边缘属性是模棱两可的。最后，我们使用多尺度的层次结构来增强边缘。我们的评测结果证明了这些选择的优势（与 HED 相比，ODS F-measure 提高了 2.3%）。更多的详细信息请参见第4节。

4. 实验

我们使用社区内著名的公开框架 Caffe [28] 来实现所提出的网络，并使用在 ImageNet [11] 上预训练的 VGG16 模型来初始化该网络。我们将 pool4 的步幅改为 1，并使用 atrous 算法填充空洞。在 RCF 的训练中，第 1-5 阶段的 1×1 卷积层的权重由标准偏差为 0.01 的零均值高斯分布初始化，卷积层的偏差初始化为 0。融合阶段的 1×1 卷积层的权重初始化为 0.2，而偏差同样初始化为 0。整个网络使用小批量随机梯度下降（Stochastic Gradient Descent, SGD）进行训练，在每次迭代中为小批量随机采样 10 张图像。对于其他 SGD 的超参数，全局学习率设置为 1e-6，并且每 10k 次迭代后就将其除以 10。动量（Momentum）和权重衰减（Weight Decay）分别设置为 0.9 和 0.0002。训练总共进行 40k 次迭代。此外，损失函数中的参数 η 和 λ 是根据训练数据设置的。本文中所有实验均使用一块 NVIDIA TITAN X GPU 完成。

给定一张边缘概率图，需要设置一个阈值来得到相应的边缘图像。设置此阈值的方法有两种。第一种为最佳数据集尺度（Optimal Dataset Scale, ODS），它对数据集中的所有图像采用固定的阈值。第二种称为最佳图像尺度（Optimal Image Scale, OIS），它为每张图像选择相应的最佳阈值。我们在实验中同时使用了 ODS 和 OIS 两种方式下的 F-measure （$\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$）来进行评测。

4.1. BSDS500 数据集

BSDS500 [2] 是边缘检测中广泛使用的一个数据集。它由 200 张训练图像、100 张验证图像和 200 张测试图像组成，且每张图像都被 4 到 9 个标注者所标注。我们使用训练集和验证集对网络进行微调，然后利用测试集进行评测。数据增强与论文 [58] 中相同。受之前研究 [29,37,59] 的启发，我们将进行数据增强后的 BSDS500 和经过翻转的 PASCAL VOC Context 数据

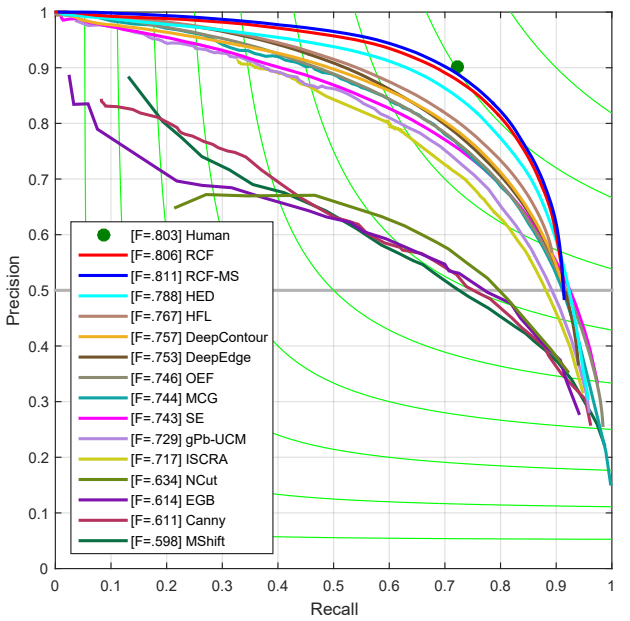


图 5. 在标准的 BSDS500 数据集 [2] 上的评测结果。单尺度和多尺度版本的 RCF 都取得了比人眼更好的性能。

集 [42] 混合在一起作为训练数据。在训练时，损失函数中的参数 η 和 λ 分别设置为 0.5 和 1.1。在评测时，使用标准的非极大值抑制（Non-Maximum Suppression, NMS [14]）来使检测到的边缘变细。我们将所提出的方法与一些非深度学习算法，如 Canny [6]、EGB [16]、gPb-UCM [2]、ISCRA [45]、MCG [3]、MShift [10]、NCut [48]、SE [14] 和 OEF [24]，以及一些基于深度学习的方法，如 DeepContour [47]、DeepEdge [4]、HED [58]、HFL [5] 和 MIL+G-DSN+MS+NCuts [29] 等进行了比较。

评测结果如图5所示。人眼在边缘检测中的性能被认为是 0.803 ODS F-measure，而单尺度和多尺度（MS）版本的 RCF 都实现了比人类更好的性能。在 ODS F-measures 指标上，所提出的 RCF-MS 和 RCF 分别比 HED [58] 高 2.3% 和 1.8%，而且我们的方法的准确率-召回率曲线也高于 HED。这些显著的提升证明了我们所提出的更丰富的卷积特征的有效性，也就是说，每个卷积阶段中，不只最后一层而是所有的卷积层都包含有用的层次信息。

定量比较结果如表2所示。从 RCF 到 RCF-MS，虽然检测速度从 30 FPS 下降到 8 FPS，但 ODS F-measure 从 0.806 增加到 0.811，这证明了我们多尺度策略的有效性。此外，当在 BSDS500 基准上使用默认

表 2. 在 BSDS500 数据集 [2] 上与其他方法的比较。† 表示 GPU 时间。性能最好的三个结果分别以红色、绿色和蓝色突出显示。

Method	ODS	OIS	FPS
Canny [6]	.611	.676	28
EGB [16]	.614	.658	10
MShift [10]	.598	.645	1/5
gPb-UCM [2]	.729	.755	1/240
Sketch Tokens [36]	.727	.746	1
MCG [3]	.744	.777	1/18
SE [14]	.743	.763	2.5
OEF [24]	.746	.770	2/3
DeepContour [47]	.757	.776	1/30†
DeepEdge [4]	.753	.772	1/1000†
HFL [5]	.767	.788	5/6†
N^4-Fields [19]	.753	.769	1/6†
HED [58]	.788	.808	30†
RDS [37]	.792	.810	30†
CEDN [59]	.788	.804	10†
MIL+G-DSN+MS+NCuts [29]	.813	.831	1
RCF	.806	.823	30†
RCF-MS	.811	.830	8†

参数进行评测时，RCF 曲线的长度不如其他方法长，这一现象表明 RCF 更倾向于检测置信度更高的边缘。与最近的边缘检测器（例如 RDS [37] 和 CEDN [59]）相比，我们的方法也取得了更好的检测结果。RDS 使用松弛标签和额外的训练数据来重新训练 HED 网络，与 HED 相比，在 ODS F-measure 上提高了 0.4%。而我们所提出的 RCF 方法在 ODS F-measure 上比 RDS 还要高 1.4%，这表明我们的改进不是微不足道的。

我们可以看出，RCF 实现了检测性能和检测效率之间的最佳平衡。尽管 MIL+G-DSN+MS+NCuts [29] 的精度比我们的方法要好一些，但是 RCF 和 RCF-MS 的速度要比它快很多。单尺度的 RCF 的速度能达到 30 FPS，RCF-MS 也可以达到 8 FPS。值得注意的是，我们的 RCF 网络仅向 HED 添加了一些 1×1 卷积层，因此时间消耗与 HED 几乎相同。此外，Iasonas 等人 [29] 往 HED 中添加了一些有用的组件，例如多实例学习（Multiple Instance Learning, MIL）[12]、G-DSN [32]、多尺度、使用 PASCAL Context 数据集 [42] 作为额外

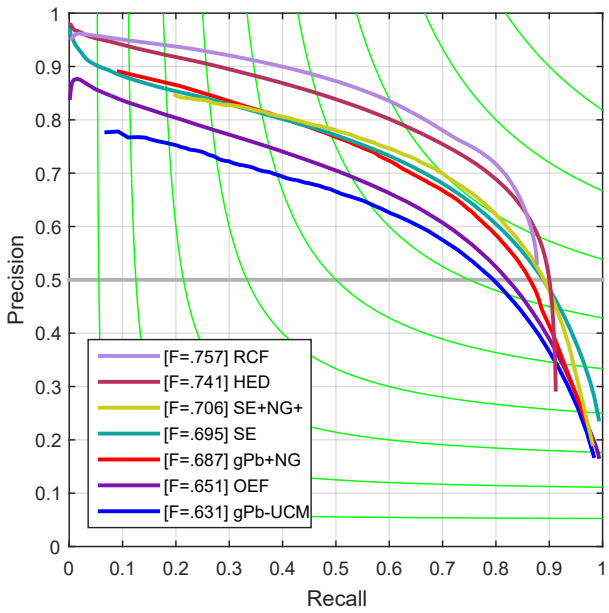


图 6. 在 NYUD 数据集 [49] 上的评测结果。这里的 RCF 为单尺度版本。

训练数据、以及标准化切割（Normalized Cuts）[2] 等。我们提出的方法比 [29] 要简单得多。由于我们的边缘检测器简单而高效，因此可以很容易地将它们应用到其他高层计算机视觉任务中去。

4.2. NYUD 数据集

NYUD 数据集 [49] 由 1449 个密集标记的成对的 RGB 图和深度图组成。最近，许多研究 [14, 57] 都在此数据集上对边缘检测进行评测。Gupta 等人 [22] 将 NYUD 数据集划分为 381 张训练图像、414 张验证图像和 654 张测试图像。我们保持和他们相同的设置，并像 HED [58] 中一样，使用全分辨率的训练集和验证集来训练 RCF 网络。

我们通过使用 HHA [23] 来利用深度信息，其中，深度信息被编码为三个通道：水平视差、地上高度和重力角。因此，HHA 特征可以看作一张彩色图像。然后，分别使用 RGB 图像和 HHA 特征图像来训练两个模型。我们将图像和相应的标注旋转到四个不同的角度（0、90、180 和 270 度），并在每个角度都对它们进行翻转。在训练过程中，将 RGB 和 HHA 的 λ 都设置为 1.2。由于 NYUD 的每张图像都只有一个相应的真值图，因此 η 在这里是无效的。其他的网络设置与在 BSDS500 上的设置相同。在测试时，通过将 RGB 模型和 HHA 模型的输出取平均来得到最终的边缘预测。在评测时，由于 NYUD 数据集中的图像大于 BSDS500 数据集中的图像，因此我们将定位公差（该值控制了预测的边缘与真值图之间的匹配中允许的最大距离）从 0.0075 提高到 0.011。

表 3. 在 NYUD 数据集 [49] 上与一些方法的比较。† 表示 GPU 时间。

Method	ODS	OIS	FPS
OEF [24]	.651	.667	1/2
gPb-UCM [2]	.631	.661	1/360
gPb+NG [22]	.687	.716	1/375
SE [14]	.695	.708	5
SE+NG+ [23]	.706	.734	1/15
HED-HHA [58]	.681	.695	20†
HED-RGB [58]	.717	.732	20†
HED-RGB-HHA [58]	.741	.757	10†
RCF-HHA	.705	.715	20†
RCF-RGB	.729	.742	20†
RCF-HHA-RGB	.757	.771	10†

我们仅将单尺度版本的 RCF 与一些著名的边缘检测方法进行比较。其中，OEF [24] 和 gPb-UCM [24] 仅使用了 RGB 图像，而其他方法则同时使用了深度和 RGB 信息。图6展示了准确率-召回率曲线。从图中可知，RCF 在 NYUD 数据集上获得了最佳性能，其次是 HED [58]。表3展示了定量比较的结果。可以看出，RCF 不仅在单独的 HHA 或 RGB 数据上，在合并的 HHA-RGB 数据上也取得了比 HED 更好的结果。对于 HHA 和 RGB 数据，RCF 在 ODS F-measure 指标上分别比 HED 高 2.4% 和 1.2%。对于合并的 HHA-RGB 数据，RCF 比 HED 要高 1.6%。此外，只用 HHA 的边缘的性能要比只用 RGB 的差，但将 HHA 和 RGB 边缘进行平均后就可取得更高的结果。这说明组合不同类型的信息对于边缘检测非常有用，这可能也是 OEF 和 gPb-UCM 比其他方法表现更差的原因。

4.3. Multicue 数据集

最近，Mély 等人 [41] 提出了 Multicue 数据集来研究边缘检测的心理物理学理论。它由立体摄像机捕获的 100 个具有挑战性的自然场景的双目短视频序列组成。每个场景都包含一个左视角和一个右视角的彩色

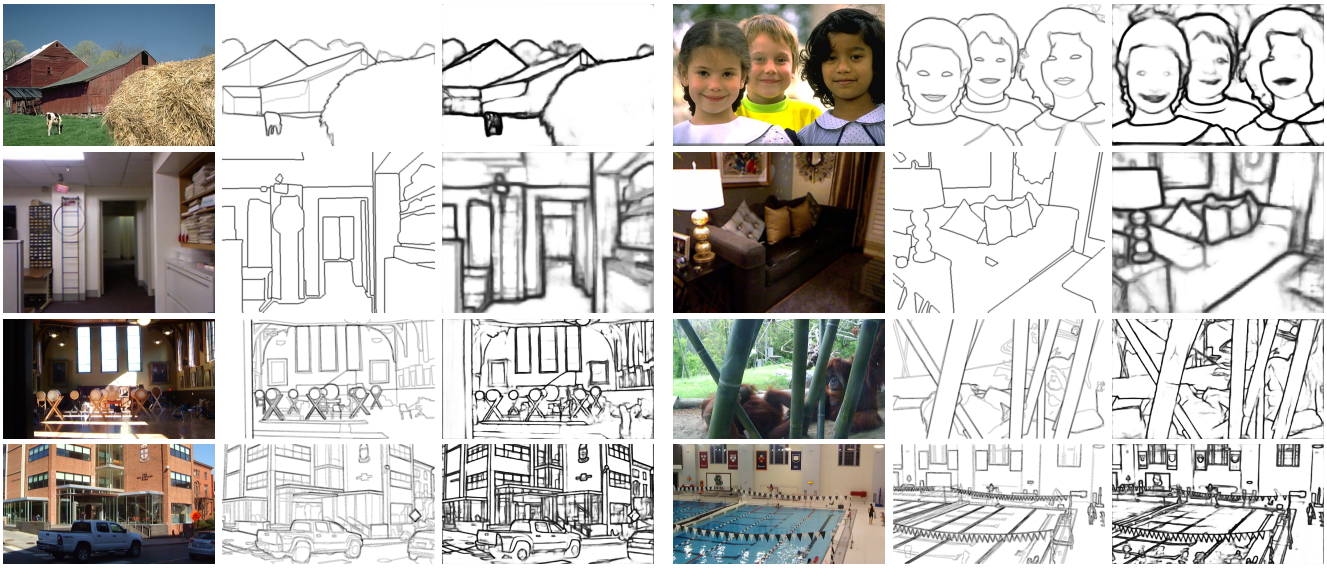


图 7. RCF 的一些示例图。从上到下依次是：BSDS500 [2]、NYUD [49]、Multicue-Boundary [41] 和 Multicue-Edge [41]。从左到右依次是：原图、真值图、RCF 边缘图、原图、真值图、RCF 边缘图。

短序列（10 帧）。每个场景的左视角图像的最后一帧有两个标注，即物体边界和底层边缘。与通常情况下不区分边界和边缘有所不同，他们根据不同阶段的视觉感知严格定义了边界和边缘：边界是指有意义的物体的边界像素，而边缘是指亮度、颜色或立体感发生急剧变化的突变像素。在本小节中，我们使用 Mély 等人 [41] 所定义的边界和边缘，而前面几节中的边界和边缘则具有相同的含义。

正如 Mély 等人 [41] 和 HED [58] 中所做的那样，我们将这些人工标注的图像随机划分为 80 张训练样本和 20 张测试样本，并将三次独立实验的平均值作为最终结果。在 Multicue 上训练时，λ 设为 1.1；而 η 在边界任务下为 0.4，在边缘任务下为 0.3。对于边界检测任务，学习率设为 1e-6 并使用 SGD 训练 2k 次迭代；而对于边缘检测任务，学习率设为 1e-7 并使用 SGD 训练 4k 次迭代。训练数据的增强方式和在 NYUD 数据集上的相同。由于 Multicue 的图像分辨率非常高，我们从原图像中随机裁剪出 500×500 的图像小块。

评测结果如表4所示，我们所提出的 RCF 取得了远高于 HED 的结果。对于边界任务，在 ODS F-measure 指标上，RCF-MS 比 HED 高 1.1%；在 OIS F-measure 指标上，RCF-MS 比 HED 高 1.4%。对于边缘任务，在 ODS F-measure 指标上，RCF-MS 比 HED 高 0.9%。另外，RCF 的波动也比 HED 小，这说明 RCF 对于不同类型的图像更加鲁棒。图7中给出了一些定性的比较结果。

表 4. 在 Multicue 数据集 [41] 上与其他方法的比较。

Method	ODS	OIS
Human-Boundary [41]	.760 (.017)	–
Multicue-Boundary [41]	.720 (.014)	–
HED-Boundary [58]	.814 (.011)	.822 (.008)
RCF-Boundary	.817 (.004)	.825 (.005)
RCF-MS-Boundary	.825 (.008)	.836 (.007)
Human-Edge [41]	.750 (.024)	–
Multicue-Edge [41]	.830 (.002)	–
HED-Edge [58]	.851 (.014)	.864 (.011)
RCF-Edge	.857 (.004)	.862 (.004)
RCF-MS-Edge	.860 (.005)	.864 (.004)

4.4. 网络讨论

为了进一步探究我们所提出的网络结构的有效性，我们使用 VGG16 [50] 实现了一些混合网络，方法是将我们的基于更丰富的卷积特征的侧输出连接到一些卷积阶段，而将 HED 的侧输出连接到其他卷积阶段。当仅在 BSDS500 数据集 [2] 上进行训练并使用单尺度进行测试时，这些混合网络的评测结果如表5所示。该表的最后两行分别对应 HED 和 RCF。我们可以观察到，

表 5. 一些混合网络的结果。

RCF Stage	HED Stage	ODS	OIS
1, 2	3, 4, 5	.792	.810
2, 4	1, 3, 5	.795	.812
4, 5	1, 2, 3	.790	.810
1, 3, 5	2, 4	.794	.810
3, 4, 5	1, 2	.796	.812
–	1, 2, 3, 4, 5	.788	.808
1, 2, 3, 4, 5	–	.798	.815

所有混合网络的性能都比 HED 好，但比完全连接了 RCF 侧输出的 RCF 要差。它清楚地证明了我们的更丰富的卷积特征的重要性。

为了研究包括更多的非线性激活是否有帮助，我们在每个阶段的 $1 \times 1 - 21$ 或 $1 \times 1 - 1$ 卷积层后连接 ReLU 层，却发现网络性能变得更差。特别是当我们将非线性层添加到 $1 \times 1 - 1$ 卷积层后时，发现网络无法正常收敛。

5. 总结

在本文中，我们提出了一个新的 CNN 架构 RCF，该架构充分利用语义信息和细节特征来进行边缘检测。我们将其精心设计为一个可扩展的体系结构。所提出的 RCF 方法可以非常高效地生成高质量的边缘，这使其有望被应用于其他计算机视觉任务中。RCF 架构可以看作是全卷积神经网络（例如 FCN [38] 和 HED [58]）的未来的发展方向。将所提出的网络架构应用于其他领域（例如显著性物体检测和语义分割）也非常值得探索。源代码已公开于https://github.com/yun-liu/rcf。

致谢 我们要感谢匿名审稿人的有用的反馈。这项研究得到了国家自然科学基金委员会（项目编号 61572264、61620106008）、华为创新研究计划（HIRP）和 CAST 青年人才计划的支持。

参考文献

[1] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In IEEE CVPR, pages 2294–2301. IEEE, 2009. 1

[2] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. IEEE TPAMI, 33(5):898–916, 2011. 1, 2, 4, 5, 6, 7, 8

[3] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In IEEE CVPR, pages 328–335, 2014. 1, 6

[4] G. Bertasius, J. Shi, and L. Torresani. DeepEdge: A multi-scale bifurcated deep network for top-down contour detection. In IEEE CVPR, pages 4380–4389, 2015. 2, 6

[5] G. Bertasius, J. Shi, and L. Torresani. High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. In IEEE ICCV, pages 504–512, 2015. 6

[6] J. Canny. A computational approach to edge detection. IEEE TPAMI, (6):679–698, 1986. 2, 6

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062, 2014. 2

[8] M.-M. Cheng, Y. Liu, Q. Hou, J. Bian, P. Torr, S.-M. Hu, and Z. Tu. HFS: Hierarchical feature selection for efficient image segmentation. In ECCV, pages 867–882. Springer, 2016. 1

[9] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In IEEE CVPR, 2014. 1

[10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE TPAMI, 24(5):603–619, 2002. 6

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In IEEE CVPR, pages 248–255. IEEE, 2009. 5

[12] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence, 89(1):31–71, 1997. 6

[13] P. Dollár, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In IEEE CVPR, volume 2, pages 1964–1971. IEEE, 2006. 2

[14] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. IEEE TPAMI, 37(8):1558–1570, 2015. 1, 2, 6, 7

[15] R. O. Duda, P. E. Hart, et al. Pattern classification and scene analysis, volume 3. Wiley New York, 1973. 2

[16] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. IJCV, 59(2):167–181, 2004. 6

[17] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. IEEE TPAMI, 30(1):36–51, 2008. 1

[18] J. R. Fram and E. S. Deutsch. On the quantitative evaluation of edge detection schemes and their comparison with human performance. IEEE TOC, 100(6):616–628, 1975. 2

[19] Y. Ganin and V. Lempitsky. N^4-Fields: Neural network nearest neighbor fields for image transforms. In ACCV, pages 536–551. Springer, 2014. 2, 6

[20] R. Girshick. Fast R-CNN. In IEEE ICCV, pages 1440–1448, 2015. 1, 3

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In IEEE CVPR, pages 580–587, 2014. 1, 3

[22] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In IEEE CVPR, pages 564–571, 2013. 7

[23] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In ECCV, pages 345–360. Springer, 2014. 7

[24] S. Hallman and C. C. Fowlkes. Oriented edge forests for boundary detection. In IEEE CVPR, pages 1732–1740, 2015. 6, 7

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In IEEE CVPR, pages 770–778, 2016. 2

[26] J.-J. Hwang and T.-L. Liu. Pixel-wise deep learning for contour detection. arXiv preprint arXiv:1504.01989, 2015. 2

[27] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869, 2014. 2

[28] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In ACM MM, pages 675–678. ACM, 2014. 5

[29] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. arXiv preprint arXiv:1511.07386, 2015. 5, 6, 7

[30] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu. Statistical edge detection: Learning and evaluating edge cues. IEEE TPAMI, 25(1):57–74, 2003. 2

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, pages 1097–1105, 2012. 1

[32] C.-Y. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In AISTATS, volume 2, page 5, 2015. 6

[33] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Generalized boundaries from multiple image interpretations. IEEE TPAMI, 36(7):1312–1324, 2014. 1

[34] Y. Li, K. He, J. Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In NIPS, pages 379–387, 2016. 1

[35] Y. Li, M. Paluri, J. M. Rehg, and P. Dollár. Unsupervised learning of edges. In IEEE CVPR, pages 1619–1627, 2016. 2

[36] J. J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In IEEE CVPR, pages 3158–3165, 2013. 1, 2, 6

[37] Y. Liu and M. S. Lew. Learning relaxed deep supervision for better edge detection. In IEEE CVPR, pages 231–240, 2016. 5, 6

[38] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In IEEE CVPR, pages 3431–3440, 2015. 2, 3, 9

[39] D. Marr and E. Hildreth. Theory of edge detection. Proceedings of the Royal Society of London B: Biological Sciences, 207(1167):187–217, 1980. 2

[40] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE TPAMI, 26(5):530–549, 2004. 1, 2

[41] D. A. Mély, J. Kim, M. McGill, Y. Guo, and T. Serre. A systematic comparison between visual cues for boundary detection. Vision research, 120:93–107, 2016. 7, 8

[42] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In IEEE CVPR, pages 891–898, 2014. 6

[43] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, pages 91–99, 2015. 1, 3

[44] X. Ren. Multi-scale improves boundary detection in natural images. In ECCV, pages 533–545. Springer, 2008. 2

[45] Z. Ren and G. Shakhnarovich. Image segmentation by cascaded region agglomeration. In IEEE CVPR, pages 2011–2018, 2013. 6

[46] G. S. Robinson. Color edge detection. Optical Engineering, 16(5):165479–165479, 1977. 2

[47] 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 IEEE CVPR, pages 3982–3991, 2015. 2, 6

[48] J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE TPAMI, 22(8):888–905, 2000. 2, 6

[49] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In European Conference on Computer Vision, pages 746–760. Springer, 2012. 7, 8

[50] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. 1, 2, 3, 8

[51] I. Sobel. Camera models and machine perception. Technical report, DTIC Document, 1970. 2

[52] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In IEEE CVPR, pages 1–9, 2015. 1

[53] V. Torre and T. A. Poggio. On edge detection. IEEE TPAMI, (2):147–163, 1986. 2

[54] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. IJCV, 104(2):154–171, 2013. 1

[55] S. Ullman and R. Basri. Recognition by linear combinations of models. IEEE TPAMI, 13(10):992–1006, 1991. 1

[56] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. IEEE TPAMI, 2016. 1

[57] R. Xiaofeng and L. Bo. Discriminatively trained sparse code gradients for contour detection. In NIPS, pages 584–592, 2012. 1, 7

[58] S. Xie and Z. Tu. Holistically-nested edge detection. In IJCV. Springer, 2017. 2, 3, 5, 6, 7, 8, 9

[59] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang. Object contour detection with a fully convolutional encoder-decoder network. arXiv preprint arXiv:1603.04530, 2016. 5, 6

[60] Z. Zhang, Y. Liu, T. Bolukbasi, M.-M. Cheng, and V. Saligrama. Bing++: A fast high quality object proposal generator at 100fps. arXiv preprint arXiv:1511.04511, 2015. 1

[61] S. Zheng, V. A. Prisacariu, M. Averkiou, M.-M. Cheng, N. J. Mitra, J. Shotton, P. H. Torr, and C. Rother. Object proposals estimation in depth image using compact 3d shape manifolds. In German Conference on Pattern Recognition, pages 196–208, 2015. 1

[62] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In ECCV, pages 391–405. Springer, 2014. 1