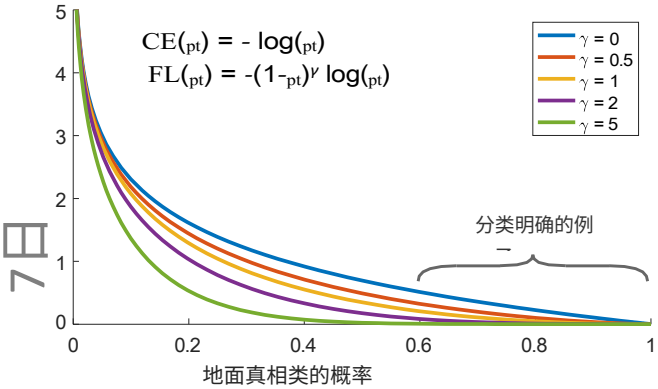
密集物体检测的焦点损失

Tsung-Yi Lin Priya Goyal Ross Girshick Kaiming He Piotr
Dolla´r Facebook AI Research (FAIR)



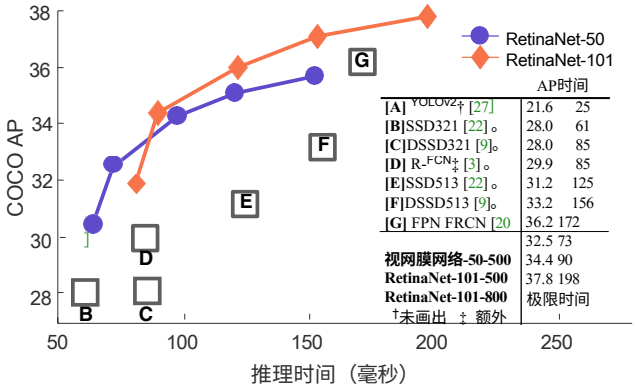


图1.我们提出了一种新的损失，我们称之为*焦点损失*，在标准的交叉熵准则上增加了一个系数（$1-p_t$）$^\gamma$。设置$\gamma>0$可以减少分类良好的例子（$p_t>0.5$）的相对损失，把更多的注意力放在困难的、错误分类的例子上。正如我们的实验所证明的那样，所提出的焦点损失使
在存在大量简单背景实例的情况下，训练高度精确的密集物体检测器。

摘要

迄今为止，准确度最高的物体检测器是基于R-CNN推广的两阶段方法，其中分类器被应用于稀疏的候选物体位置集。相比之下，对可能的物体位置进行有规律的密集采样的单阶段检测器有可能更快、更简单，但到目前为止还落后于两阶段检测器的准确性。在本文中，我们研究了为什么会出现这种情况。我们发现，密集检测器训练过程中遇到的前台-后台类不平衡是核心原因。我们建议通过重塑标准的交叉熵损失来解决这种等级不平衡的问题，从而降低分配给分类良好的例子的损失。我们新颖的焦点损失（Focal Loss）将训练集中在一组稀疏的困难例子上，并防止大量的容易否定的例子在训练期间淹没检测器。为了评估我们损失的有效性，我们设计并训练了一个简单的密集检测器，我们称之为RetinaNet。我们的结果显示，当用焦点损失进行训练时，RetinaNet能够与之前的单阶段检测器的速度相匹配，同时超过了所有现有的最先进的两阶段检测器的准确性。代码见：https://github.com/facebookresearch/Detectron。

图2.速度（毫秒）与准确度（AP）在COCO测试装置上的对比。在焦点损失的支持下，我们简单的单级*RetinaNet*检测器的性能超过了以前所有的单级和双级检测器，包括[20]中报道的最好的Faster R-CNN[28]系统。我们展示了ResNet-50-FPN（蓝色圆圈）和ResNet-101-FPN（橙色钻石）在五个尺度（400-800像素）下的RetinaNet变体。忽略低精度系统（AP<25），RetinaNet形成了目前所有检测器的上限，一个改进的变体（未显示）达到了40.8AP。详细情况见*第5节*。

1. 简介

目前最先进的物体检测器是基于一个两阶段的、建议驱动的机制。正如在R-CNN框架[11]中推广的那样，第一阶段产生一个*稀疏*的候选物体位置集，第二阶段使用卷积神经网络将每个候选位置分类为前景类之一或背景。通过一连串的进展[10, 28, 20, 14]，这个两阶段的框架在具有挑战性的COCO基准[21]上始终取得了最高的准确性。

尽管两阶段检测器取得了成功，一个自然的问题是：简单的单阶段检测器能否达到类似的精度？单阶段检测器适用于对物体位置、比例和As-pect比率的定期*密集*采样。最近关于单阶段检测器的工作，如YOLO[26, 27]和SSD[22, 9]，展示了有希望的结果，产生了更快的检测器，相对于最先进的两阶段方法，其精度在10-40%之间。

本文进一步推动了这一进程：我们提出了一个单级物体检测器，该检测器首次与最先进的COCO AP更复杂的两级去中心化技术相匹配。

诸如特征金字塔网络（FPN）[20]或Faster R-CNN[28]的Mask R-CNN[14]变体。为了实现这一结果，我们将训练过程中的类不平衡确定为阻碍一级检测器实现最先进精度的主要障碍，并提出了一个新的损失函数来消除这一障碍。

类不平衡在类似R-CNN的检测器中是通过两阶段的级联和采样启发式方法来解决的。提出阶段（*如*选择性搜索[35]、EdgeBoxes[39]、DeepMask[24, 25]、RPN[28]）迅速将候选物体位置的数量缩小到一个小数目（*如*1-2k），过滤掉大多数背景样本。在第二个分类阶段，采样启发法，如固定的前景与背景比例（1:3），或在线硬例挖掘（OHEM）[31]，以保持前景和背景之间可控的平衡。

相比之下，单阶段检测器必须处理更大的候选物体位置集，并在图像中定期采样。在实践中，这往往相当于列举了10万个密集覆盖空间位置、比例和长宽比的位置。虽然类似的抽样方法也可以应用，但它们的效率很低，因为训练过程仍然被容易分类的背景例子所支配。这种低效率是物体检测中的一个典型问题，通常通过引导[33, 29]或硬例挖掘[37, 8, 31]等技术解决。

在本文中，我们提出了一个新的损失函数，作为一个更有效的替代以前的方法来处理类不平衡问题。该损失函数是一个按比例计算的交叉熵损失，其中比例因子随着对正确类别的信心增加而衰减为零，见图1。直观地说，这个比例因子可以在训练过程中自动降低简单例子的贡献，并迅速将模型集中在困难例子上。实验表明，我们提出的*Focal Loss*使我们能够训练出一个高准确度的单阶段检测器，该检测器明显优于使用采样启发式训练或硬例挖掘的替代方法，这是以前训练单阶段检测器的最先进技术。最后，我们注意到，焦点损失的确切形式并不重要，我们表明其他实例可以达到类似的结果。

为了证明所提出的焦点损失的有效性，我们设计了一个简单的单阶段物体检测器，称为*RetinaNet*，因其对输入图像中物体位置的密集采样而命名。它的设计特点是一个有效的网络内特征金字塔和使用锚定盒。它借鉴了[22, 6, 28, 20]中的各种最新想法。RetinaNet是高效和准确的；我们的最佳模型，基于ResNet-101-FPN主干，在运行速度为5帧/秒时，COCO测试开发AP为39.1，超过了之前公布的单级和双级检测器的最佳模型结果，见图2。

2. 相关工作

经典的物体检测器：滑动窗口范式，即在密集的图像网格上应用分类器，有着悠久而丰富的历史。最早的成功案例之一是LeCun等人的经典工作，他们将卷积神经网络应用于手写数字识别[19, 36]。Vi- ola和Jones[37]将提升的物体检测器用于人脸检测，导致了这种模型的广泛采用。HOG[4]和积分通道特征的引入
[5]产生了有效的行人检测方法。DPMs[8]有助于将密集型检测器扩展到更多的一般物体类别，并在PASCAL[7]上取得了多年的顶级结果。虽然滑动窗口方法是经典计算机视觉中领先的检测范式，但随着深度学习[18]的重新兴起，接下来描述的两阶段检测器迅速成为物体检测的主导。

两阶段检测器：现代物体检测的主流范式是基于两阶段的方法。正如选择性搜索工作[35]中所展示的那样，第一阶段产生一个稀疏的候选方案集，该方案集应该包含所有的物体，同时过滤掉大部分的负面位置，第二阶段将这些方案分类为前景类/背景类。R-CNN[11]将第二阶段的分类器升级为卷积网络，在准确度上有了很大的提高，并开创了物体检测的现代时代。多年来，R-CNN在速度方面[15, 10]以及通过使用学习到的物体提议[6, 24, 28]都得到了改进。区域提议网络（RPN）将提议生成与第二阶段分类器整合到一个卷积网络中，形成了快速R-CNN框架[28]。这个框架的许多扩展已经被提出，*例如*[20, 31, 32, 16, 14]。

单阶段检测器：OverFeat[30]是第一个基于深度网络的现代单阶段物体检测器之一。最近，SSD[22, 9]和YOLO[26, 27]重新激发了对单阶段方法的兴趣。这些检测器已被调整为速度，但它们的准确度落后于两阶段的方法。SSD的AP值要低10-20%，而YOLO专注于更极端的速度/精度权衡。见图2。最近的工作表明，仅仅通过降低输入图像的分辨率和建议的数量，就可以使两阶段检测器变得快速，但是单阶段方法即使有更大的计算预算，其准确性也会落后[17]。与此相反，这项工作的目的是了解单级检测器是否能够在运行速度相似或更快的情况下，匹配或超过两级检测器的精度。

我们的RetinaNet检测器的设计与之前的密集检测器有许多相似之处，特别是RPN[28]引入的 "锚 "概念，以及SSD[22]和FPN[20]中使用的金字塔结构。我们强调，我们的简单检测器取得了最好的结果，不是基于网络设计的创新，而是由于我们新的损失。

类不平衡：经典的单阶段物体检测方法，如提升检测器[37, 5]和DPMs[8]，以及最近的方法，如SSD[22]，在训练期间都面临着巨大的类不平衡。这些检测器评估10^{-4}
每幅图像有10^5个候选地点，但只有少数地点...
含有对象。这种不平衡导致了两个问题：
(1) 训练效率低下，因为大多数位置都是容易否定的，没有贡献任何有用的学习信号；(2) 大量的容易否定的例子会淹没训练，导致模型的生成失败。一个常见的解决方案是进行某种形式的硬阴性挖掘[33, 37, 8, 31, 22]，在训练过程中对硬例子进行采样，或者进行更复杂的采样/重新权衡方案[2]。相比之下，我们表明，我们提出的焦点损失自然地处理了单阶段检测器所面临的类不平衡问题，并允许我们在不抽样的情况下对所有的例子进行有效的训练，而且没有容易的负面因素来压倒损失和计算梯度的情况。

稳健估计：人们对去签署稳健的损失函数（如Huber损失[13]）很感兴趣，该函数通过降低误差大的例子（硬例子）的损失权重来重新控制*离群值*的贡献。相比之下，我们的焦点损失不是为了解决离群值，而是通过降低*离群值*（容易的例子）的权重来解决类的不平衡，这样即使它们的数量很大，它们对总损失的贡献也很小。换句话说，焦点损失发挥了与稳健损失*相反的*作用：它把训练重点放在稀少的硬例子上。

3. 焦点损失

*焦点损失*的设计是为了解决单阶段目标检测的情况，在训练过程中，前景和背景类之间存在极端的平衡（*例如*，1：1000）。我们从二元分类的交叉熵（CE）损失开始引入焦点损失。[1]：

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{如果} y=1 \\ -\text{对数}(1-p)，\text{否则。} \end{cases} \quad (1)$$

在上面的例子中，$y \in \{\pm 1\}$指定了基础真理类，$p \in [0, 1]$是模型对标签为$y=1$的类的估计概率。为了符号上的方便，我们定义p_t：

$$p_t = \begin{cases} p & \text{如果} y=1 \\ 1-p，\text{否则、} \end{cases} \quad (2)$$

并重写$\mathrm{CE}(p, y) = \mathrm{CE}(p_t) = -\log(p_t)$。
CE的损失可以看作是图中的蓝色（顶部）曲线。

3.1. 平衡交叉熵

解决类不平衡的一个常用方法是为第1类引入一个加权因子$\alpha \in [0, 1]$，为第1类引入$1-\alpha$。在实践中，α可以通过反类自由度来设置，或者作为一个超参数，通过交叉验证来设置。
为方便记述，我们将α定义为：，类似于我们对p的定义。为便于记述，我们定义α_t，类似于我们定义p_t。我们把α-平衡*的*CE损失写成：

$$\mathrm{CE}(p_t) = -\alpha_t \log(p_t)。 \quad (3)$$

这种损失是对CE的简单扩展，我们认为这是我们提出的焦点损失的实验基线。

3.2. 焦点损失的定义

正如我们的实验所显示的，在密集检测器的训练过程中遇到的巨大的类别不平衡压倒了交叉熵损失。容易分类的负数占了损失的大部分，并主导了梯度。虽然α平衡了正面/负面例子的重要性，但它并没有区分容易/困难的例子。相反，我们建议重塑损失函数，降低容易的例子的权重，从而将训练重点放在困难的负面因素上。

更正式地说，我们建议在交叉熵损失中加入一个调制因子$(1-p_t)^{\gamma}$，可调控的*聚焦*参数$\gamma \geq 0$。我们将焦点损失定义为：：

$$\mathrm{FL}(p_t) = -(1-p_t)^{\gamma} \log(p_t)。 \quad (4)$$

$\gamma \in$的几个值的焦点损失是可视化的
图1中的$[0, 5]$。我们注意到焦点损失的两个特性。
(1) 当一个例子被错误分类，并且p_t小的时候，调节系数接近1，损失不受影响。当$p_t \to 1$时，系数变为0，分类良好的例子的损失被降权。(2) 聚焦参数γ
平稳地调整容易的例子被降权的速度。当$\gamma=0$时，FL等同于CE，而当γ是
在我们的实验中，我们发现$\gamma=2$的效果最好。直观地说，
调制因子减少了损失的影响。
但从简单的例子中，扩展了一个例子获得低损失的范围。例如，在$\gamma=2$的情况下，与CE相比，一个以$p_t=0.9$分类的例子将有100倍的损失，而在$p_t \approx 0.968$的情况下，将有
损失降低1000倍。这反过来又增加了重要性
纠正错误分类的例子（其损失是按比例计算的）。
图1。这个损失的一个值得注意的特性是，即使是容易

分类的例子(p_t .5)也会产生非微不足道的损失，这一点可以从它的图中轻易看出。当在大量容易分类的例子上求和时，这些小的损失值可以压倒稀有类。

[1] 将焦点损失扩展到多类情况是直截了当的，而且效果很好；为了简单起见，我们在这项工作中侧重于二进制损失。

对于$p_t \leq .5$和$\gamma = 2$，最多下降4倍）。

在实践中，我们使用一个α-平衡的焦点损失的变体：

$$FL(p_t) = -\alpha_t (1 - p)_t^{\gamma} \log(p_t)。 \quad (5)$$

我们在实验中采用了这种形式，因为它比非α-平衡形式产生的精度略有提高。最后，我们注意到损失层的实现结合了计算p的sigmoid操作和损失计算，从而产生了更大的数值稳定性。

虽然在我们的主要实验结果中，我们使用了上述的焦点损失定义，但它的精确形式并不关键。在附录中，我们考虑了焦点损失的其他实例，并证明这些实例可以同样有效。

3.3. 类不平衡和模型初始化

二元分类模型默认初始化为输出y=-1或1的概率相同。在这样的初始化下，在存在类不平衡的情况下，频繁出现的类所造成的损失会支配总的损失，导致早期训练的不稳定。为了解决这个问题，我们引入了"先验 "的概念，即p es-的值。

*在训练开始时，*由模型对稀有类（前景）进行计时。我们用π表示先验，并设置它使模型对稀有类的例子的估计p很低、

例如：0.01。我们注意到这是模型初始化的一个变化（见第4.1节），而*不是*损失函数的变化。我们发现这样做可以提高交叉熵和损失函数的训练稳定性。

在班级严重失衡的情况下，焦点损失。

3.4. 等级不平衡和两级检测器

两阶段检测器通常用交叉熵损失进行训练，而不使用α-平衡或我们提出的损失。相反，它们通过两种机制来解决类的不平衡问题：(1)两级级联和(2)有偏见的小批量采样。第一个级联阶段是一个对象建议机制[35, 24, 28]，它将几乎无限的可能对象位置集减少到一两千个。重要的是，所选择的建议不是随机的，而是有可能对应于真实的物体位置，这就消除了绝大多数的容易被否定的情况。在训练第二阶段时，通常使用有偏见的抽样来构建迷你批，其中包含例如1:3的正面和负面例子的比例。这个比例就像一个隐含的α-平衡因子，通过抽样来实现。我们提出的焦点损失旨在通过损失函数直接解决单阶段检测系统中的这些机制。

4. 视网膜网络检测器

RetinaNet是一个单一的、统一的网络，由一个*骨干*网络和两个特定任务的*子网络*组成。主干网络负责计算整个输入图像的卷积图，是一个非自体卷积网络。第一个子网络在主干网的输出上进行对话式物体分类；第二个子网络进行卷积边界盒的回归。这两个子网络的特点是设计简单，我们专门为单阶段密集检测提出了设计，见图3。虽然这些组件的细节有许多可能的选择，但大多数设计参数对精确的值并不特别敏感，正如实验中所显示的那样。接下来我们将描述RetinaNet的每个组件。

特征金字塔网络骨干网：我们采用[20]中的特征金字塔网络（FPN）作为RetinaNet的骨干网络。简而言之，FPN通过自上而下的路径和横向连接增强了标准的卷积网络，因此该网络可以从单一分辨率的输入图像中有效地构建一个丰富的、多尺度的特征金字塔，见图3(a)-(b)。金字塔的每一层都可用于检测不同尺度的物体。FPN改善了完全卷积网络（FCN）[23]的多尺度预测，正如其对RPN[28]和DeepMask式的提议[24]，以及在两阶段检测器如Fast R-CNN[10]或Mask R-CNN[14]的收益所示。

按照[20]，我们在ResNet架构[16]的基础上构建FPN。我们构建了一个金字塔，级别为P_3到P_7，其中l表示金字塔级别（P_l的Reso- lution 2^l比输入低）。如同[20]中的所有金字塔级别有$C = 256$个通道。金字塔的细节一般遵循[20]，但有一些适度的差异。[2]虽然许多设计的选择并不关键，我们强调使用FPN主干是关键；初步实验只使用最后的ResNet层的特征，产生了低AP。

锚点：我们使用与[20]中的RPN变体类似的翻译不变的锚点盒。锚点的面积为32^2到512^2，分别位于金字塔层P_3到P_7。与[20]一样，在每个金字塔层，我们使用三种长宽比的锚点{1:2, 1:1, 2:1}。对于比[20]中更密集的规模覆盖，在每一级我们增加了原3个长宽比锚的大小$\{2^0, 2^{1/3}, 2^{2/3}\}$的锚。这在我们的环境中，改善AP。总共有$A$=9个锚每级和各级都涵盖了32-30的规模范围。相对于网络的输入图像，813个像素。

每个锚被分配一个长度为K的分类目标单热向量，其中K是物体类别的数量，以及一个4向量的箱体回归目标。我们使用RPN[28]的分配规则，但为多类检测进行了修改，并调整了阈值。具体来说，锚被分配到地面真实的物体盒子上，使用0.5的交叉联合（IoU）阈值；如果它们的IoU在[0, 0.4]，则分配到背面的地面。由于每个锚点最多分配给一个物体箱，我们将其长度为K的标签向量中的相应条目设置为1，其他条目为0。如果一个锚没有被分配，这可能发生在重叠的时候在[0.4, 0.5]中，它在训练中被忽略了。箱体回归目标被计算为每个锚和其分配的对象箱体之间的偏移，如果没有分配，则省略不计。

[2] RetinaNet使用特征金字塔级别P_3到P_7，其中P_3到P_5是由相应的ResNet残差阶段（C_3到C_5）的输出计算出来的，与[20]一样，P_6是通过对C_5的3×3 stride-2 conv得到的，P_7是通过应用ReLU然后对P_6进行3×3 stride-2 conv计算的。这与[20]略有不同：(1)由于商业原因，我们不使用高分辨率的金字塔级别P_2，(2)P_6是通过串联卷积而不是下采样计算的，(3)我们包括P_7以提高大型物体检测。这些细微的修改在保持精度的同时提高了速度。

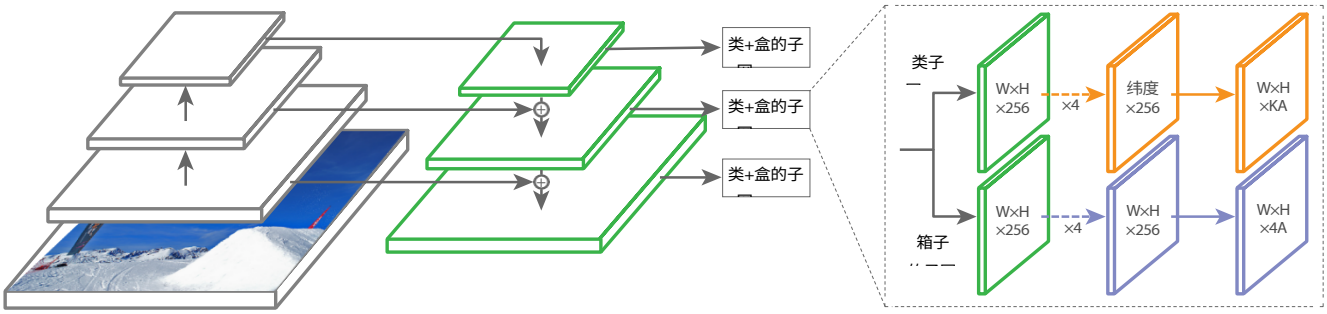


图3.一阶段的**RetinaNet**网络架构在前馈ResNet架构[16]的基础上使用了一个特征金字塔网络（FPN）[20]（a）来生成一个丰富的、多尺度卷积特征金字塔（b）。在这个骨干网络上，RetinaNet附加了两个子网络，一个用于分类锚定盒（c），一个用于从锚定盒回归到地面真实物体盒（d）。网络的设计是故意简单的，这使得这项工作能够专注于一个新的焦点损失函数，它消除了我们的单阶段检测器与最先进的两阶段检测器（如带FPN的Faster R-CN[20]）之间的准确性差距，同时以更快的速度运行。

分类子网：分类子网预测每个空间位置的物体存在的概率，每个A锚和K物体类别。这个子网是一个附属于每个FPN级别的小型FCN；这个子网的参数在所有金字塔级别中都是共享的。它的设计很简单。以一个具有C通道的输入特征图
从一个给定的金字塔层开始，子网应用了四个3×3的信念层，每个都有C滤波器，每个都有ReLU激活，然后是一个有KA滤波器的3×3信念层。最后附加sigmoid激活来输出每个空间位置的KA二元预测，见图3（c）。我们在大多数实验中使用C=256和A=9。

与RPN[28]相比，我们的对象分类子网更深，只使用3×3的convs，并且不与盒式回归子网（接下来描述）共享参数。我们发现这些更高层次的设计决定更有意义。
比超参数的具体数值更重要。

盒子回归子网：在对象分类子网的同时，我们在每个pyra-mid水平上附加一个小的FCN，目的是对每个锚定箱的偏移量进行回归，如果存在一个地面真实的对象的话。盒式回归子网的设计与 "地物分类子网 "相同。分类子网，除了它在每个空间位置终止于$4A$线性输出，见图3（d）。对于每个空间位置的A个锚点，这4个输出预先决定了锚点和地面真实箱之间的相对偏移（我们使用R-CNN[11]的标准箱体参数化）。我们注意到，与最近的工作不同的是，我们使用了一个与类别无关的界线盒调节器，它使用了较少的参数。
我们发现这些参数同样有效。物体分类子网和盒式回归子网虽然共享一个共同的结构，但使用的是单独的参数。

4.1. 推理和训练

推论：RetinaNet形成了一个单一的FCN，由ResNet-FPN主干网、一个分类子网和一个盒子组成。

回归子网，见图3。因此，推理包括简单地通过网络转发图像。为了提高速度，我们只对每个FPN级别中最多1k个最高分的预测进行解码，在检测器置信度达到0.05的阈值后。所有级别的最高预测被合并，并采用0.5的阈值进行非最大抑制，以获得最终的检测结果。

焦点损失：我们使用这项工作中引入的焦点损失作为分类子网输出的损失。由于我们
我们将在第5节中展示，我们发现γ=2在实践中效果很好，而且RetinaNet对$\gamma \in [0.5, 5]$相对稳健。我们强调，在训练RetinaNet时，焦点损失是
适用于每个采样图像中的*所有*~10万个锚点。这与使用启发式抽样（RPN）或硬例挖掘（OHEM，SSD）为每个minibatch选择一小部分锚点（*如*256个）的常见做法形成对比。一幅图像的总焦点损失被计算为所有□10万个锚点的焦点损失之和，并*以分配给地面实况箱的锚点数量为标准*。我们通过指定的锚的数量，而不是锚的总数来进行归一化，因为绝大多数的锚是容易被否定的，在焦点损失下得到的损失值可以忽略不计。最后，我们注意到α，分配给稀有类的权重，也有一个稳定的范围，但它与γ相互作用，使得有必要将两者结合起来选择（见表1a和1b）。一般来说，α应该随着γ的增加而略微减少。
增加（对于$\gamma = 2$，$\alpha = 0.25$效果最好）。

初始化：我们用ResNet-50-FPN和ResNet-101-FPN骨干进行实验[20]。基础的ResNet-50和ResNet-101模型是在ImageNet1k上预训练过的；我们使用的是[16]发布的模型。为FPN添加的新层按[20]的方法进行初始化。除了RetinaNet子网中的最后一个，所有新的conv层都被初始化为偏置$b = 0$
和一个高斯权重填充，σ=0.01。对于最终的定罪
在分类子网的层中，我们将偏见的初始化设置为$b = -\log((1 - \pi)/\pi)$，其中π指定在

α	美联社	美联社$_{50}$	美联社$_{75}$
.10	0.0	0.0	0.0
.25	10.8	16.0	11.7
.50	30.2	46.7	32.8
.75	31.1	49.4	33.0
.90	30.8	49.7	32.3
.99	28.7	47.4	29.9
.999	25.1	41.7	26.1

(a) **变化α的CE损失**（$\gamma = 0$）。

γ	α	美联社	美联社$_{50}$	美联社$_{75}$
0	.75	31.1	49.4	33.0
0.1	.75	31.4	49.9	33.1
0.2	.75	31.9	50.7	33.4
0.5	.50	32.9	51.7	35.2
1.0	.25	33.7	52.0	36.2
2.0	.25	**34.0**	**52.5**	**36.5**
5.0	.25	32.2	49.6	34.8

(b) **变化的γ用于FL**（最佳α）。

#sc	#ar	美联社	美联社$_{50}$	美联社$_{75}$
1	1	30.3	49.0	31.8
2	1	31.9	50.0	34.0
3	1	31.8	49.4	33.7
1	3	32.4	52.3	33.9
2	3	**34.2**	**53.1**	**36.5**
3	3	34.0	52.5	**36.5**
4	3	33.8	52.1	36.2

(c) **不同的锚定尺度和方面**

方法	批量大小	nms thr	美联社	美联社$_{50}$	美联社$_{75}$
羟基甲烷(OHEM)	128	.7	31.1	47.2	33.2
羟基甲烷(OHEM)	256	.7	31.8	48.8	33.9
羟基甲烷(OHEM)	512	.7	30.6	47.0	32.6
羟基甲烷(OHEM)	128	.5	32.8	50.3	35.1

(d) **FL*与*OHEM基线的*对比***（使用ResNet-101-FPN）。

深度	规模	美联社	美联社$_{50}$	美联社$_{75}$	美联社$_S$	美联社$_M$	美联社$_L$	时间
50	400	30.5	47.8	32.7	11.2	33.8	46.1	64
50	500	32.5	50.9	34.8	13.9	35.8	46.7	72
50	600	34.3	53.2	36.9	16.2	37.4	47.4	98
50	700	35.1	54.2	37.7	18.0	39.3	46.4	121
50	800	35.7	55.0	38.5	18.9	38.9	46.3	153
101	400	31.9	49.5	34.1	11.6	35.8	48.5	81
101	500	34.4	53.1	36.8	14.7	38.5	49.1	90
101	600	36.0	55.2	38.7	17.4	39.6	49.7	122
101	700	37.1	56.6	39.8	19.1	40.6	49.4	154

(e) **准确度/速度权衡** RetinaNet (在测试装置上)

表1.**RetinaNet和Focal Loss（FL）的消融实验。**所有模型都在`trainval35k`上训练，并在`minival`上测试，除非注明。如果没有指定，默认值是：γ=2；3种比例和3种长宽比的锚；ResNet-50-FPN主干；以及600像素的训练和测试图像比例。(a) 使用α-*平衡*CE的RetinaNet最多能达到31.1AP。(b) 相比之下，使用FL与相同的确切网络，可获得2.9的AP增益，并且对确切的γ/α设置相当稳健。(c) 使用2-3个比例和3个长宽比的锚点会产生良好的结果，之后性能就会饱和。(d) FL比在线硬例挖掘(OHEM)的最佳变体[31, 22]的AP超过3点。(e) RetinaNet在不同的网络深度和图像比例下的准确度/速度权衡（也见图2）。

在训练开始时，每一个锚都应该被标记为前地，其置信度为π。我们在所有的实验中都使用π=0.01，尽管结果对准确的数值很稳健。正如第3.3节所解释的，这种初始化可以防止大量的背景锚在训练的第一次迭代中产生一个大的、破坏性的损失值。

优化：RetinaNet是用随机梯度下降法（SGD）训练的。我们在8个GPU上使用同步SGD，每个minibatch共有16幅图像（每个GPU2幅）。除非另有说明，所有模型都训练了9万次，初始学习率为0.01，然后在6万次时除以10，在8万次时再次除以10。除非另有说明，我们使用水平图像翻转作为唯一的数据增加形式。使用0.0001的权重衰减和0.9的动量。训练损失是焦点损失和用于盒式回归的标准平滑L_1损失之和[10]。表1e中的模型的训练时间在10到35小时之间。

我们在具有挑战性的COCO基准[21]的边界盒检测轨道上展示了实验结果。对于训练，我们遵循通常的做法[1, 20]，并使用COCO `trainval35k`分割（由80k图像组成的联盟）。

5. 实验

训练和一个随机的35K图像子集从40K图像中分割出来的年龄）。我们通过评估`minival`分割（来自`val`的剩余5k图像）来报告病变和敏感性研究。对于我们的主要结果，我们在测试-开发部分报告COCO AP，它没有公共标签，需要使用评估服务器。

5.1. 训练密集型检测

我们进行了许多实验来分析密集检测的损失函数的行为以及各种优化策略。在所有的实验中，我们使用深度为50或101的ResNets[16]，并在上面构建一个特征金字塔网络（FPN）[20]。对于所有的消融研究，我们使用600像素的图像比例进行训练和测试。

网络初始化：我们第一次尝试训练Reti- naNet时使用了标准的交叉熵（CE）损失，没有对初始化或学习策略进行任何修改。这很快就失败了，网络在训练过程中出现了分歧。然而，简单地初始化我们模型的最后一层，使检测到物体的先验概率为$\pi=0.01$(见第4.1节)能够有效地学习。训练RetinaNet与ResNet-50，这个初始化已经产生了一个再在COCO上可观察到的AP为30.2。结果对π的精确值不敏感，所以我们在所有的实验中使用$\pi=0.01$。

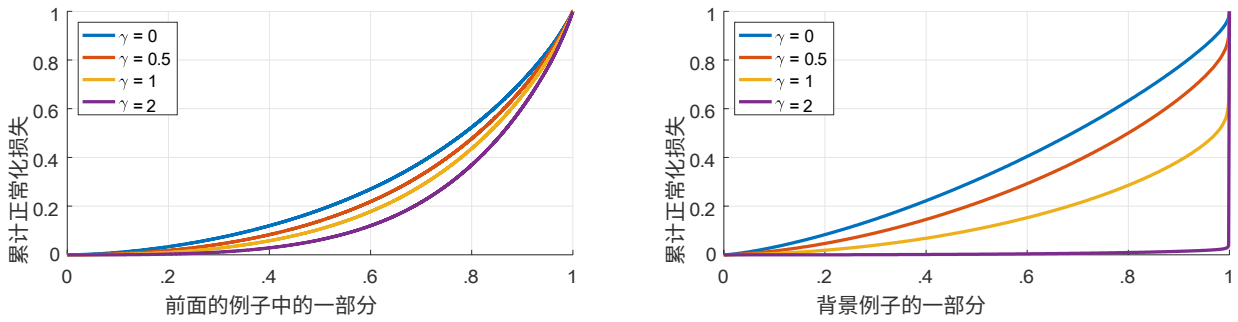


图4.对于一个*收敛的*模型，在不同的γ值下，正、负样本的归一化损失的累积分布函数。改变γ对正向样本的损失分布的影响很小。然而，对于负面的例子，增加γ会使损失严重集中在困难的例子上，几乎把所有的注意力从容易的负面例子上集中起来。

平衡交叉熵：我们的下一个尝试是使用§3.1中描述的α-*平衡*CE损失来改善学习。表1a中显示了不同α的结果。设置-
α=*0.*75时，AP的收益为0.9分。

焦点损失：使用我们提出的焦点损失的结果显示在表1b。焦点损失引入了一个新的透明参数，即聚焦参数γ，控制调制项的强度。当γ=0时，我们的损失等同于CE损失。随着γ的增加，焦点损失的形状
损失的变化使低损失的 "简单 "例子得到更多的折扣，见图1。随着γ的增加，FL显示出比CE更大的收益。在γ=2的情况下，FL*比*α*平衡的CE损失产生了2.9个AP的提升*。

对于表1b中的实验，为了公平比较，我们找到了每个γ的最佳α。我们观察到，较低的α被选择为较高的γ（因为容易否定的东西被降低权重，需要放在正面的东西上较少强调）。然而，总的来说，改变γ的好处要大得多，事实上，最好的α的范围仅在[.25,.75]（我们测试了α∈[.01,.999]）。我们使用γ=2.0，α=*0.*25，对所有的
实验，但α=*0.*5的效果几乎一样好（0.4AP低）。

焦点损失的分析：为了更好地理解焦点损失，我们分析了一个*收敛*模型的损失的经验分布。为此，我们采取我们默认的ResNet- 101 600像素模型，用γ=2进行训练（它有36.0个AP）。我们将这个模型应用于大量的随机图像。
年龄，并对$\approx 10^7$ 阴性窗口和$\approx 10^5$ 阳性窗口的预测概率进行采样。接下来，分别针对阳性和阴性，我们计算这些样本的FL、
并将损失归一，使其总和为1。考虑到归一化的损失，我们可以将损失从低到高进行排序，并绘制其在正负样本和不同的γ设置下的累积分布函数（CDF）（尽管模型是在γ=2的情况下训练的）。
正数和负数的累积分布函数。
图4中显示了正向样本。如果我们观察阳性样本，我们会发现，对于不同的γ值，CDF看起来相当相似。例如，大约20%的最难的阳性样本占了大约一半的比例。

正的损失，随着γ的增加，更多的损失集中在前20%的例子中，但影响不大。

γ对阴性样本的影响是显著不同的。对于γ=0，正面和负面的CDFs非常相似。然而，随着γ的增加，更多的权重集中在困难的负面例子上。事实上，在γ = 2的情况下（我们的默认设置），绝大部分的损失来自一小部分的样本。可以看出，FL可以有效地抵消容易否定的影响，将所有的注意力集中在困难的否定例子上。

在线硬例挖掘（OHEM）：[31]提出通过使用高损失的例子构建迷你批来改善两阶段检测器的训练。具体来说，在OHEM中，每个例子都根据其损失进行评分，然后应用非最大抑制（nms），用损失最大的例子组成一个迷你批。nms阈值和批量大小是可调整的参数。与焦点损失一样，OHEM更强调错误分类的例子，但与FL不同的是，OHEM完全抛弃了容易的例子。我们还实现了SSD[22]中使用的OHEM的一个变体：在对所有的例子应用nms后，minibatch的构造是强制执行阳性和阴性之间的1:3比例，以帮助确保每个minibatch有足够的阳性。

我们在具有较大类不平衡性的单阶段检测环境中测试了OHEM的两个变体。原始OHEM策略和 "OHEM 1:3 "策略在选定批次大小和nms阈值下的结果见图1d。这些结果使用了ResNet-101，我们用FL训练的基线在这种设置下达到了36.0的AP。相比之下，OHEM的最佳设置（没有1:3的比例，批次大小为128，nms为.5）达到了32.8AP。这是一个3.2AP的差距，表明FL比OHEM在训练密集型检测器方面更有效。我们注意到，我们尝试了OHEM的其他参数设置和变体，但并没有取得更好的结果。

铰链损失：最后，在早期的实验中，我们尝试用铰链损失[13]对p_t进行训练，该损失在p_t的某一数值之上设置为0。然而，这是不稳定的，我们没有设法获得有意义的结果。探索其他损失函数的结果在附录中。

	骨干力量	美联社	美联社$_{50}$	美联社$_{75}$	美联社$_S$	美联社$_M$	美联社$_L$
两阶段方法							
更快的R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
更快的R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
通过G-RMI实现更快的R-CNN [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7	13.5	38.1	52.0
更快的R-CNN w TDM [32]	初始-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
一个阶段的方法							
YOLOv2 [27]	暗网-19 [27]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [22, 9]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513[9]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
视网膜网络（我们的）	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
视网膜网络（我们的）	镍钴锰锌合金-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2

表2.**物体检测***单一模型的*结果（边界框AP），*与*COCO测试开发中的最先进技术*相比*。我们展示了我们的RetinaNet-101-800模型的结果，该模型的训练时间比表1e的同一模型长1.5*倍*。我们的模型取得了顶级的结果，超过了一阶段和两阶段的模型。关于速度与准确率的详细分类，见表1e和图2。

5.2. 模型架构设计

锚点密度：一级检测系统中最重要的设计因素之一是它如何密集地覆盖可能的图像框空间。两阶段检测器可以使用区域池化操作对任何位置、比例和长宽比的盒子进行分类[10]。相比之下，由于单阶段检测器使用固定的采样网格，在这些方法中实现盒子的高覆盖率的流行方法是在每个空间位置使用多个 "锚点"[28]，以覆盖不同尺度和长宽比的盒子。

我们对FPN中每个空间位置和每个金字塔级别所使用的尺度和长宽比锚的数量进行扫瞄。我们考虑了从每个位置的单个方形锚到每个位置的12个锚的情况，跨越4个亚八度的尺度（$2^{k/4}$，对于$k\leq 3$）和3个长宽比[0.5, 1, 2]。使用ResNet-50的结果显示在表1c。令人惊讶的是
好的AP（30.3）是只用一个方形锚实现的。然而，当每个位置使用3个尺度和3个长宽比时，AP可以提高近4点（达到34.0）。我们在这项工作的所有其他实验中都使用了这种设置。

最后，我们注意到，增加到6-9个锚点以上并没有显示出进一步的收益。因此，虽然两级系统可以对图像中的任意方框进行分类，但密度的饱和度意味着两级系统更高的潜在密度可能不会带来优势。

速度与准确性的关系：较大的骨干网络产生较高的准确性，但推断速度也较慢。同样，对于输入图像的规模（由较短的图像边定义）也是如此。我们在表1e中显示了这两个因素的影响。在图2中，我们绘制了RetinaNet的速度/准确率权衡曲线，并与最近使用COCO `test-dev`上的公共数字的方法进行了比较。该图显示，RetinaNet在我们的焦点损失的支持下，形成了一个超过所有现有方法的上限，不包括低精确度的制度。带有ResNet-101-FPN和600像素图像比例的RetinaNet（为了简单起见，我们用RetinaNet-101-600表示）与最近发表的ResNet-101-FPN Faster R-CN[20]的精度相匹配，同时每分钟运行时间为122ms。

与172毫秒相比（均在Nvidia M40 GPU上测量）。使用更大的尺度使RetinaNet超越了所有两阶段方法的准确性，同时速度仍然更快。对于更快的运行时间，只有一个操作点（500像素的输入），使用ResNet-50-FPN比ResNet-101-FPN更有优势。解决高帧率制度可能需要特殊的网络设计，如[27]，这超出了本工作的范围。我们注意到，在文章发表后，现在可以通过[12]中的Faster R-CNN的一个变体获得更快更准确的结果。

5.3. 与技术现状的比较

我们在具有挑战性的COCO数据集上评估了RetinaNet，并将测试结果与最近的最先进的方法（包括单阶段和双阶段模型）进行比较。表2列出了我们的RetinaNet-101-800模型的结果，该模型使用比例抖动训练，时间比表1e中的模型长1.5倍（给予1.3个AP增益）。与前
与单阶段方法相比，我们的方法实现了一个健康的
与最接近的竞争对手DSSD[9]*相比，*有5.9分的AP差距（39.1*对*33.2），同时也更快，见图2。与最近的两阶段方法相比，RetinaNet比基于Inception-ResNet-v2-TDM[32]的表现最好的Faster R-CNN模型高出2.3分。插入ResNeXt- 32x8d-101-FPN[38]作为RetinaNet的主干，又进一步提高了1.7个AP，超过了COCO的40个AP。

6. 总结

在这项工作中，我们发现类的不平衡是阻碍一级物体检测器超越顶级两级方法的主要障碍。为了解决这个问题，我们提出了*焦点损失*，它在交叉熵损失中应用了一个调制项，以便将学习重点放在困难的负面例子上。我们的方法很简单，而且非常有效。我们通过设计一个完全卷积的单级检测器来证明它的功效，并报告了广泛的实验分析，表明它达到了最先进的精度和速度。源代码可在 https://github.com/facebookresearch/Detectron [12]。

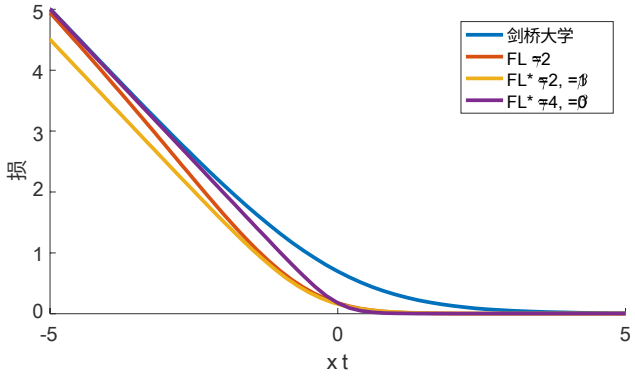


图5.焦点损失变体与交叉熵的比较，作为$_{xt}$=γx的函数。原始的FL和替代的FL变体* ，都减少了分类良好的例子（$_{xt}$>0）的相对损失。

损失	γ	β	AP	AP_{50}	AP_{75}
消费品	-	-	31.1	49.4	
FL	33.0 2.0	-	34.0	52.5	36.5
FL□ 2	.0 1 . 0		33.8	52.7	36.3

表3.FL和FL* 与CE的选择设置的结果。

附录A：焦点损失 *

焦点损失的确切形式并不重要。我们现在展示的是焦点损失的另一种实例，它具有类似的属性，并能产生类似的结果。下文还给出了对焦点损失属性的更多见解。

我们首先考虑交叉熵（CE）和焦点损失（FL），其形式与正文中略有不同。具体来说，我们定义一个数量x_t 如下：

$$x_t = yx、 \tag{6}$$

其中y⍰{±1}如前所述指定了基础真理类。然后我们可以写成$p_t = \sigma(x_t)$（这与方程2中p_t 的定义是一致的）。当$x_t > 0$时，一个例子被正确分类，在这种情况下$p_t > .5$。

我们现在可以用x_t 来定义焦点损失的另一种形式。我们定义p^* 和FL* 如下：

$$p^*_t = \sigma\ (\gamma x_t + \beta)\ 、 \tag{7}$$

$$FL^* = - \log(p^*_t)/\gamma。 \tag{8}$$

FL* 有两个参数，γ和β，控制损失曲线的陡度和移动。我们在图5中把两个选定的γ和β设置的FL* 与CE和FL一起绘制。可以看出，与FL一样，FL* 在选定的参数下会降低
分配给分类良好的例子的损失。

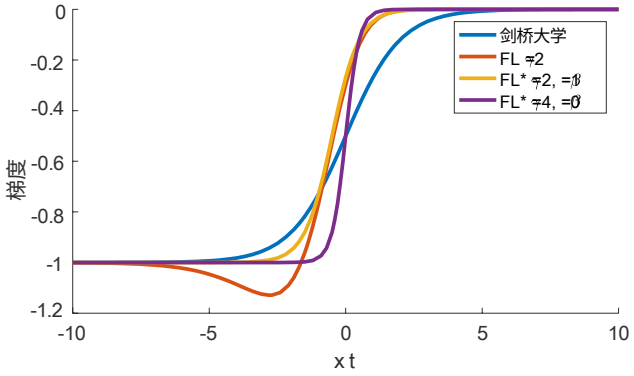


图6.图5中损失函数的派生，与x有关。

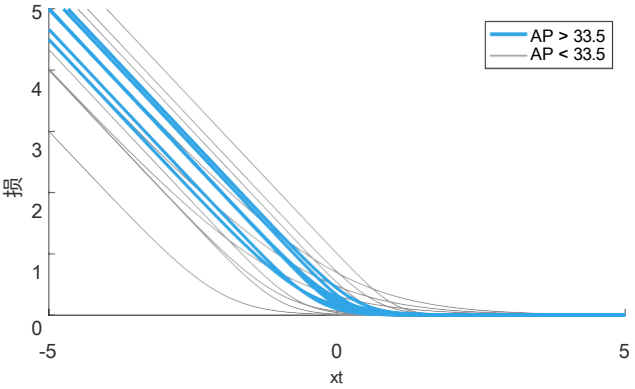


图7.不同设置γ和β的FL* 的效果。图中用颜色编码，有效设置显示为蓝色。

我们发现，各种γ和β的设置都有很好的效果。在图7中，我们展示了RetinaNet-50-600与FL* 的广泛参数设置的结果。损失图是用颜色编码的，有效的设置（模型收敛且AP超过33.5）显示为蓝色。为了简单起见，我们在所有实验中使用了α=0.25。可以看出，减少分类良好的例子的权重的损失（$x_t > 0$）是有效的。

更普遍的是，我们希望任何具有与FL或FL类似属性的损失函数* ，同样有效。

附录B：衍生品

*

作为参考，CE、FL和FL相对于x的导数为：：

$$\frac{dCE}{dx} = y(p_t - 1) \tag{9}$$

$$\frac{dFL}{dx} = y(1-p\ )_t^{\gamma}\ (\gamma p_t \log(p_t\ ) + p_t - 1) \tag{10}$$

$$\frac{dFL\ \square^*}{}$$

我们使用与之前相同的设置来训练RetinaNet-50-600

，但我们将FL换成了FL^{*}，并选择了参数等。这些模型达到的AP几乎与用FL训练的模型相同，见表3。换句话说，FL^{*} 是一个可以替代FL的、在实践中运行良好的选择。

$$\frac{dx}{} = y(_{pt} - 1) \tag{11}$$

图6显示了选定设置的图。对于所有的损失函数，高置信度预测的导数都趋向于-1或0。然而，与CE不同，对于FL和FL^{*} 的有效设置，只要$x_t > 0$，导数就很小。

参考文献

[1] S.Bell, C. L. Zitnick, K. Bala, and R. Girshick.内-外网：用跳过集合 和递归神经网络检测上下文中的物体。In *CVPR*, 2016.6

[2] S.R. Bulo, G. Neuhold, and P. Kontschieder.用于语义图像分割的损失最大值集合。In *CVPR*, 2017.3

[3] J.Dai, Y. Li, K. He, and J. Sun.R-FCN：通过 基于区域的全卷积网络进行物体检测。在*NIPS*, 2016.1

[4] N.Dalal and B. Triggs.面向梯度的直方图用于 人类检测。In *CVPR*, 2005.2

[5] P.Dolla´r, Z. Tu, P. Perona, and S. Belongie.综合频道 特征。在*BMVC*，2009年。2, 3

[6] D.Erhan, C. Szegedy, A. Toshev, and D. Anguelov.使用深度神经网络进行可扩展的物体检测。In *CVPR*, 2014.2

[7] M.Everingham, L. Van Gool, C. K. Williams, J. Winn, and A.Zisserman.PASCAL可视化对象类(VOC)挑战。*IJCV*, 2010.2

[8] P.F. Felzenszwalb, R. B. Girshick, and D. McAllester.用可变形部件模型进行案例对象检测。In *CVPR*, 2010.2, 3

[9] C.-Y.Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg.DSSD：*arXiv:1701.06659*, 2016.1, 2, 8

[10] R.Girshick.快速R-CNN.In *ICCV*, 2015.1, 2, 4, 6, 8

[11] R.Girshick, J. Donahue, T. Darrell, and J. Malik.用于精确物体检测和语义 分割的丰富图像层次结构。In *CVPR*, 2014.1, 2, 5

[12] R.Girshick, I. Radosavovic, G. Gkioxari, P. Dolla´r, and K. He 。 Detectron 。 https://github.com/facebookresearch/detectron, 2018.8

[13] T.Hastie, R. Tibshirani, and J. Friedman.*统计学习的要素*.斯普林格统计学系列 斯普林格、柏林，2008年。3, 7

[14] K.He, G. Gkioxari, P. Dolla´r, and R. Girshick.面罩R-CNN。In *ICCV*, 2017.1, 2, 4

[15] K.He, X. Zhang, S. Ren, and J. Sun.用于视觉识别的深度卷积网络中的空间金字塔集合。在 *ECCV*。2014.2

[16] K.He, X. Zhang, S. Ren, and J. Sun.深度残差学习 ，用于图像识别。In *CVPR*, 2016.2, 4, 5, 6, 8

[17] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara、A.Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K.Murphy.现代 convolu- tional object detectors 的速度/准确度权衡。In *CVPR*, 2017.2, 8

[18] A.Krizhevsky, I. Sutskever, and G. Hinton.用深度卷积神经网络进行ImageNet分类。在*NIPS*, 2012.2

[19] Y.LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel.应用于手写邮政编码识别的反向传播法。*Neural compu- tation*, 1989.2

[20] T.-Y.Lin, P. Dolla´r, R. Girshick, K. He, B. Hariharan, and S.Belongie.用于物体检测的特征金字塔网络。In *CVPR*, 2017.1, 2, 4, 5, 6, 8

[21] T.-Y.Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ra-manan, P. Dolla´r, and C. L. Zitnick.Microsoft COCO: Com- mon objects in context.在*ECCV*，2014年。1, 6

[22] W.Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed.SSD：单枪多盒检测器。In *ECCV*, 2016.1, 2, 3, 6, 7, 8

[23] J.Long, E. Shelhamer, and T. Darrell.用于语义分割的完全卷积 网络。In *CVPR*, 2015.4

[24] P.O. Pinheiro, R. Collobert, and P. Dollar.学习分离 ment 对象候选人。In *NIPS*, 2015.2, 4

[25] P.O. Pinheiro, T.-Y.Lin, R. Collobert, and P. Dolla´r.学习-完善物体段。在*ECCV*，2016年。2

[26] J.Redmon, S. Divvala, R. Girshick, and A. Farhadi.你只看一次：统一、实时的物体检测。在 *CVPR*, 2016。1, 2

[27] J.雷德蒙和A.法哈迪。YOLO9000：更好，更快， 强 。在*CVPR*, 2017.1, 2, 8

[28] S.Ren, K. He, R. Girshick, and J. Sun.更快的R-CNN：用区域建议网（ works）进行实时物体检测。In *NIPS*, 2015.1, 2, 4, 5, 8

[29] H.Rowley, S. Baluja, and T. Kanade.视觉场景中的人脸识别。技术报告CMU-CS-95-158R、
卡内基梅隆大学，1995年。2

[30] P.Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun.Overfeat：使用卷积网络进行综合识别、定位和检测。In *ICLR*, 2014.2

[31] A.Shrivastava, A. Gupta, and R. Girshick.用在线硬例挖掘训练基于区域的物体检测器。在 *CVPR*, 2016。2, 3, 6, 7

[32] A.Shrivastava, R. Sukthankar, J. Malik, and A. Gupta.Be-yond skip connections：自上而下调制的对象去 tection. *arXiv:1612.06851*, 2016.2, 8

[33] K.-K.Sung and T. Poggio.对象和模式检测的学习和实例选择。在*麻省理工学院A.I.备忘录No. 1521*，1994。2, 3

[34] C.Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi.Inception-v4，inception-resnet和剩余连接对学习的影响。在*AAAI会议上，人工 智能*，2017。8

[35] J.R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders.对象识别的选择性搜索。*IJCV*, 2013.2, 4

[36] R.Vaillant, C. Monrocq, and Y. LeCun.图像中物体定位的原始方法。*IEE Proc. on Vision*、
图像和信号处理，1994。2

[37] P.Viola and M. Jones.使用简单特征的提升 级联的快速物体检测。In *CVPR*, 2001.2, 3

[38] S.Xie, R. Girshick, P. Dolla´r, Z. Tu, and K. He。深度神经网络的聚合残差转换。在*CVPR*， 2017。8

[39] C.L. Zitnick和P. Dolla´r。边缘盒：从边缘定位物体建议 。在*ECCV*，2014年。2