【论文翻译】Focal Loss for Dense Object Detection

翻译 大数据机器学习实验室 于 2020-08-27 23:14:12 发布 2320 收藏 21 版权

文章标签： 计算机视觉

论文题目：Focal Loss for Dense Object Detection

论文来源：Focal Loss for Dense Object Detection

摘要

迄今为止，精度最高的目标检测器是基于R-CNN推广的两阶段方法，其中 分类器 应用于稀疏的候选对象位置集。相比之下，在可能的物体位置的规则，密集采样上应用的 one-stage 探测器具有更快和更简单的可能性，但迄今为止已经落后于 two-stage 探测器的精度。在本文中，我们将探讨为什么会出现这种情况。我们发现在密集型探测器训练过程中遇到的前景背景类极度不平衡是造成这种情况的主要原因。我们通过重新构造标准交叉熵损失函数来解决这一类不平衡，使其降低分类良好的样例的损失。我们的新的Focal损失在稀疏的困难子集上训练，这样防止了大量负样本的情况。为了评估该损失函数的有效性，我们设计并训练了一个简单的密集检测器，我们称之为RetinaNet。我们的研究结果表明，当使用Focal损失进行训练时，RetinaNet能够与以前的单级探测器的速度相匹配，同时其精度超过了所有现有的两级探测器。

1.引言

目前最先进的目标探测器是基于两阶段的proposal驱动机制。例如R-CNN普及的那样，在第一阶段会产生一系候选框，在第二阶段通过卷积神经网络确其是前景或背景以及候选框位置。通过一系列的改进，这个两阶段的框架连续在COCO挑战赛上达到最高精度。

尽管两级检测器取得了成功，但令人好奇的问题是：一个简单的单级检测器能否达到类似的精度？单级检测器应用于对目标位置、比例和纵横比进行常规、密集采样。最近对单级检测器的研究，如YOLO和SSD，与最先进的两级方法相比，检测速度更快，精度在10-40%以内。

本文进一步推进了这个问题：我们提出了一种单级目标检测器，它在COCO上的实时检测AP首次能与更复杂的两阶段检测器相比拟，例如FPN，Mask RCNN以及Faster R-CNN的变体。为了达到这一结果，我们将训练过程中的类别不平衡视为阻碍单级检测器达到最高精度的主要障碍，并提出了一种新的损失函数来消除这种障碍。

在类R-CNN检测器中，通过两级级联和抽样启发式来解决类不平衡问题。提取proposal阶段（例如，选择性搜索、EdgeBoxes、DeepMask、RPN）迅速将候选对象位置的数量缩小到一个小数目（例如，1-2k），过滤掉大多数背景样本。在第二个分类阶段，执行抽样启发式，例如固定的前景背景比（1:3），或在线难样本挖掘（OHEM），以保持前景和背景之间的平衡。

相比之下，单级检测器必须处理一组更大的候选对象位置，这些候选对象位置在图像中定期采样。实际上，这通常相当于列举~100k个密集覆盖空间位置、比例和纵横比的位置。虽然类似的抽样启发法也可以应用，但由于训练过程仍然由容易分类的背景例子所支配，因此它们效率低下。这种效率低下是目标检测中的一个典型问题，通常通过引导或难样本挖掘等技术来解决。

在这篇文章中，我们提出一个新的损失函数，来处理样本不平衡带来的问题。该损失函数是一个动态缩放的交叉熵损失，当正确类别的置信度增加时，比例因子衰减为零，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示例的权重，并快速将模型集中到难样本 上。实验表明，我们提出的Focal Loss使我们能够训练一个性能明显优于采样式启发或难样本挖掘训练方法的高精度单级检测器，后者是训练单级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，其他实例也能获得类似的结果。

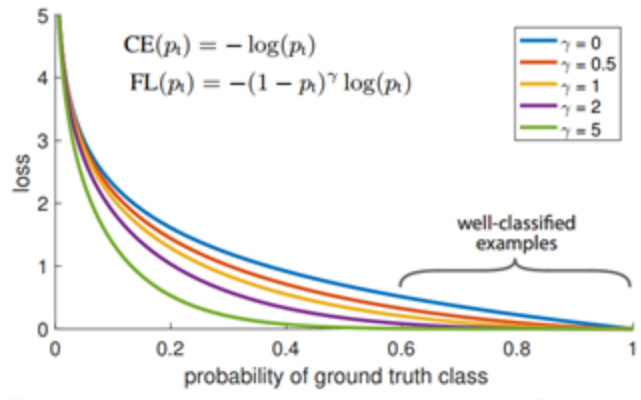


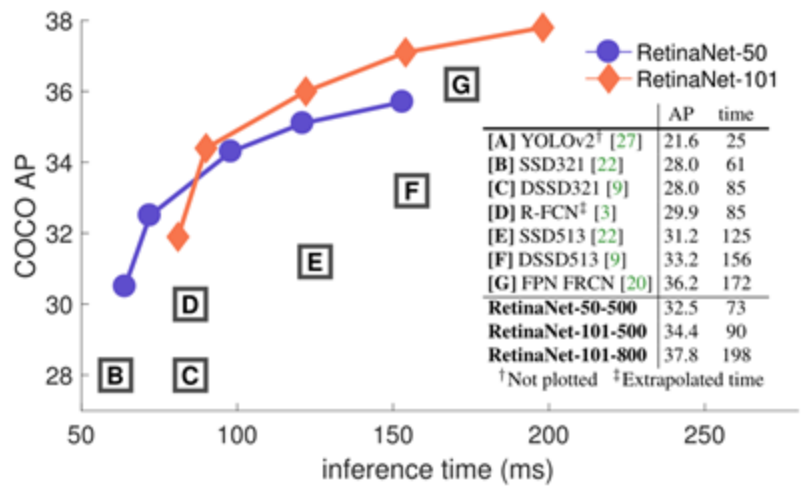
Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^{\gamma}$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

为了证明提出的损失函数的有效性，我们设计了一种简单的单级目标检测器RetinaNet，该检测器以其对输入图像中目标位置的密集采样而命名。它的设计特点是高效的网络特征金字塔和anchor box的使用。RetinaNet是高效和准确的；我们的最佳模型，以ResNet-101和FPN为主干，在以5fps的速度运行时，COCO-test-dev-AP达到39.1，超过了先前发布的单级和两级检测器的最佳结果，见图2。

2.相关工作

经典的目标检测器：拥有着悠久历史的滑动窗口模式，在密集的图像网格上应用分类器。最早的成功之一是LeCun等人的经典著作，他将卷积神经网络应用于手写数字识别。Viola和Jones使用增强型目标检测器进行人脸检测，使这类模型得到了广泛采用。HOG和积分通道特征的引入为行人检测提供了有效的方法。DPMs的帮助将密集型探测器扩展到更一般的物体类别，并在PASCAL上取得了多年的最佳结果。虽然滑动窗口方法是经典计算机视觉中的主要检测范式，但随着深度学习的重新兴起，两级检测器很快开始主导目标检测。

两阶段检测器：现代目标检测的主流方法都是基于两阶段的。正如像在选择性搜索工作中所开创的那样，第一阶段生成一个稀疏的候选方案集，这些方案应该包含所有对象，同时过滤掉大部分的反例背景，第二阶段将这些方案分类为前景类或背景。R-CNN将第二级分类器升级为卷积网络，在精度上有了很大提高，并开创了目标检测的现代时代。多年来，R-CNN在速度和使用学习提取proposal方面都得到了改进。区域建议网络（RPN）将proposal生成与第二阶段分类器集成到单个卷积网络中，形成了Faster RCNN框架。有人提议对该框架进行多次扩展。



	AP	time
[A] YOLOv2† [27]	21.6	25
[B] SSD321 [22]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [22]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [20]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

Figure 2. Speed (ms) versus accuracy (AP) on COCO `test-dev`. Enabled by the focal loss, our simple one-stage *RetinaNet* detector outperforms all previous one-stage and two-stage detectors, including the best reported Faster R-CNN [28] system from [20]. We show variants of RetinaNet with ResNet-50-FPN (blue circles) and ResNet-101-FPN (orange diamonds) at five scales (400-800 pixels). Ignoring the low-accuracy regime (AP<25), RetinaNet forms an upper envelope of all current detectors, and an improved variant (not shown) achieves 40.8 AP. Details are given in §5.

一阶段检测器：OverFeat是第一个基于深度网络的现代单级目标检测器之一。最近SSD和YOLO重新燃起了对单阶段方法研究的兴趣。这些方法速度很快，但是精度还是落后于两阶段方法。SSD的AP降低了10-20%，而YOLO专注于更极端的速度和精度的权衡。见图2。最近的研究表明，只需降低输入图像分辨率和proposal数量，就可以快速生成两级检测器。但即使一阶段方法的算力更大，其精度也落后于两阶段方法。相比之下，我们的工作的目的是研究一阶段检测器在以相似或更快的速度运行时，是否能够匹配或超过两阶段检测器的精度。
我们的RetinaNet检测器的设计与以前的密集型检测器有许多相似之处，特别是RPN引入的"锚"概念，以及在SSD和FPN中使用的特征金字塔。但我们要强调的是，我们的简单探测器达到最高的结果不是由于网络设计的创新，而是由于损失函数的创新。
类不平衡：经典的单级目标检测方法，如增强型检测器和DPMs，以及更新的方法，如SSD，在训练过程中都面临着较大的类不平衡的问题。这些检测器每幅图像要处理104-105个候选位置，但只有少数位置包含物体。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负样本，没有提供有用的学习信号；（2）总的来说，负样本过多会压倒训练，导致退化模型。一个常见的解决方案是执行某种形式的难负样本挖掘，在训练时进行难负样本采样或更复杂的采样/重新称重方案。相比之下，我们的Focla Loss自然地处理了单级检测器所面临的类不平衡，并且允许在所有示例上有效地训练，而不需要采样，也不需要容易的负样本来压倒损失和计算的梯度。
稳健估计：人们对设计具有鲁棒性的损失函数（例如Huber损失）感兴趣，这种函数通过对具有较大误差（难样本）的示例的损失进行加权来减少输出的贡献。相比之下，我们的Focal Loss不是针对异常值，而是通过向下加权（简单示例）来解决类不平衡，这样即使它们的数量很大，它们对总损失的贡献也很小。换言之，Focal Loss的作用与鲁棒损失相反：它将训练集中在一组稀疏的难样本上。

3.Focal Loss

焦点损失旨在解决单级目标检测场景在训练期间前景类和背景类之间存在极端不平衡（例如，1:1000）的问题。我们引入了从二元分类的交叉熵（CE）损失开始的焦点损失

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases} \tag{1}$$

其中y∈{±1}指定了基本真值类，p∈[0,1]是该类的模型估计概率。为了便于标注，我们将pt定义为：

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases} \tag{2}$$

并重新写成CE(p, y) = CE(pt) = −log(pt).
CE损耗可以看作图1中的蓝色（顶部）曲线。这一损失的一个显著特征是，即使是那些容易分类的例子（pt>0.5），也可以很容易地从图中看出并损失较大。在大量简单的例子中进行总结时，这些小的损失值就会覆盖稀有类。

3.1平衡交叉熵损失

解决类不平衡的一种常见方法是为类1引入权重因子α∈[0,1]，为类−1引入1−α。在实践中，α可以通过逆类频率来设置，也可以作为一个超参数通过交叉验证来设置。为了便于注释，我们将α定义为pt的定义。我们将α-平衡CE损耗写为：

$$\mathrm{CE}(p_t) = -\alpha_t \log(p_t). \tag{3}$$

这个损耗是CE的一个简单扩展，我们将其作为我们提出的焦点损耗的实验基线。

3.2Focla Loss定义

我们的实验表明，在密集检测器训练过程中遇到的类不平衡压倒了交叉熵损失。容易分类的负样本构成了大部分的损失，并主导了梯度。虽然α平衡了正反两个例子的重要性，但它并不区分简单和困难的例子。相反，我们建议重构损失函数，以减少简单的样例，从而集中训练难负样本
更正式地说，我们建议在交叉熵损失中增加一个调制因子（1−pt）γ，可调聚焦参数γ≥0。我们将焦点损失定义为：

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

。
图1显示了几个γ∈[0,5]值的焦点损失。我们注意到焦点损失的两个性质。（1） 当一个例子被错误分类并且pt很小时，调制因子接近1，损失不受影响。当pt→1时，调制因子为0，分类良好的示例的损失权重减小。（2） 聚焦参数γ平滑地调整简单例子的加权速率。当γ=0时，FL相当于CE，随着γ的增加，调制因子的作用也同样增加（我们发现γ=2在我们的实验中效果最好）。
直观地说，调制因子减少了来自简单示例的损耗贡献，并且扩展了示例接收低损失的范围。例如，当γ=2时，与CE相比，分类为pt=0.9的示例的损耗将降低100倍，而当pt≈0.968时，其损耗将降低1000倍。这反过来又增加了纠正错误分类示例的重要性（对于pt≤.5和γ=2，其损失最多减少4倍）。
在实践中，我们使用α-平衡的焦距损失：

$$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma} \log(p_t). \tag{5}$$

我们在实验中采用这种形式，因为它比非α-平衡形式的精确度稍有提高。最后，我们注意到损失层的实现结合了计算p的sigmoid运算和损失计算，从而使数值稳定性更高。
虽然在我们的主要实验结果中，我们使用了上面的Focal Loss定义，但它的精确形式并不重要。在附录中，我们考虑了焦点损失的其他实例，并证明这些实例同样有效。

3.3类不平衡和模型初始化

默认情况下，二进制分类模型初始化输出y=-1或1的概率相等。在这样的初始化条件下，在存在类不平衡的情况下，由于更多那一类会占据损失的主要部分，会导致在早期训练中的不稳定。为了解决这一问题，我们引入了"先验"的概念，即在训练开始时由稀有类（前景）模型估计p值。我们用π表示先验值，并对其进行设置，使模型对稀有类示例的估计值较低，例如0.01。我们注意到，这是模型初始化（见§4.1）的变化，而不是损失函数的变化。我们发现在类严重不平衡的情况下，对于交叉熵和焦点损失，这可以改善训练的稳定。

3.4两阶段和类不平衡

两阶段检测器通常使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。它们通过两种机制来解决类不平衡：（1）两级级联和（2）有偏小批量采样。第一个阶段是目标框的建立，它将几乎无限的可能对象位置集减少到一千或两千个。重要的是，所选目标框不是随机的，而是可能与真实的目标位置相对应的，这样可以消除绝大多数容易产生的负样本。当训练第二阶段时，有偏抽样通常用于构造小批量，例如，含有1:3的阳性和阴性样本。这个比率就像是一个隐式的α平衡因子，通过抽样实现。我们提出的焦点损失是直接通过损失函数来解决一阶段系统的类不平衡问题。

4.RetianNet

RetinaNet是一个由一个主干网和两个子网组成的统一网络。主干网负责计算整个输入图像的卷积特征映射，是一个非自卷积网络。第一个子网对主干网的输出进行分类；第二个子网进行BoundingBox回归。这两个子网的设计简单，是我们专门为一阶段密集检测而提出，见图3。这些组成部分有很多的选择，但大多数设计参数对实验中显示的精确值并不特别敏感。接下来我们将描述视RetinaNet的每个组成部分。

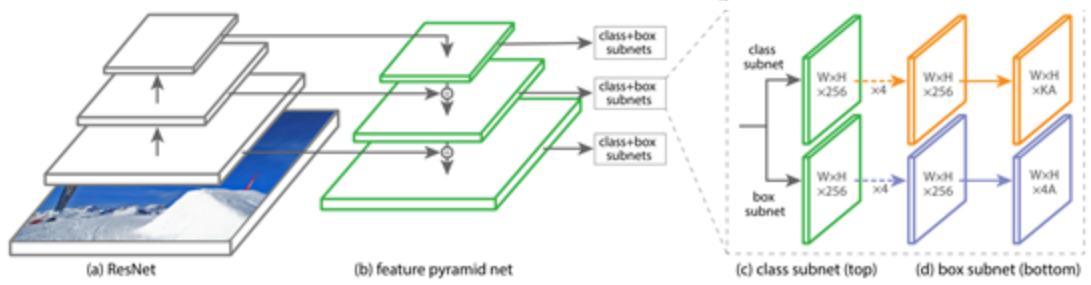


Figure 3. The one-stage **RetinaNet** network architecture uses a Feature Pyramid Network (FPN) [20] backbone on top of a feedforward ResNet architecture [16] (a) to generate a rich, multi-scale convolutional feature pyramid (b). To this backbone RetinaNet attaches two subnetworks, one for classifying anchor boxes (c) and one for regressing from anchor boxes to ground-truth object boxes (d). The network design is intentionally simple, which enables this work to focus on a novel focal loss function that eliminates the accuracy gap between our one-stage detector and state-of-the-art two-stage detectors like Faster R-CNN with FPN [20] while running at faster speeds.

特征金字塔网络主干：我们采用特征金字塔网络（FPN）作为RetinaNet的骨干网络。简言之，FPN通过自上而下的路径和横向连接来增强标准卷积网络，因此网络可以有效地从单个分辨率的输入图像构建一个丰富的、多尺度的特征金字塔，见图3（a）-（b）。金字塔的每一层都可以用来检测不同尺度的物体。FPN改进了全卷积网络（FCN）的多尺度预测，如RPN和DeepMask风格方案的增益所示，以及在Fast R-CNN或Mask R-CNN等两阶段检测器上的增益。

之后，我们在ResNet架构的基础上构建了FPN。构建了一个P3到P7的金字塔，其中l表示金字塔级别（Pl的分辨率比输入低2l）。所有级别的金字塔都有C=256个通道。金字塔的细节通常遵循FPN网络，但有一些细微的差异。我们强调使用FPN主干网络，其他的设计选择并不重要；我们初始的实验使用的特征仅使用ResNet的最后几层来产生，产生了较低的AP。

Anchor：我们使用与RPN变体类似的平移不变anchor。P3到P7的anchor的面积分别在322到5122之间。在FPN中，我们在每个金字塔级别使用三个纵横比{1:2，1:1，2:1}的anchor。对于比FPN中更密集的标度覆盖，我们在每一层添加尺寸为{20，21/3，22/3}的anchor，其尺寸为原始的3个纵横比anchor。这提高了AP。总的来说，每个级别会生成9个anchor，在各个级别上，它们覆盖了相对于网络输入图像的32-813像素的缩放范围。

每个anchor会以一个长度为K的one-hot向量表示。其中K是目标类的数目，和一个4向量的box回归目标。我们使用RPN中的赋值规则，但对多类检测进行了修改，并调整了阈值。具体地说，计算anchor和ground-truth的IOU阈值；如果anchor的IoU在[0，0.4]中，则anchor为背景。由于每个anchor最多分配给一个对象框，因此我们将其长度K标签向量中的对应项设置为1，而将所有其他项设置为0。如果某个anchor未指定，可能会在[0.4，0.5]中重叠，则在训练期间忽略该anchor。目标框回归的目标是计算每个anchor与其指定目标框之间的偏移，如果没有指定，则忽略。

分类子网：分类子网预测每个Aanchor和K个目标在每个空间位置存在的概率。此子网是一个连接到每个FPN级别的小型FCN；此子网的参数在所有金字塔级别上共享。它的设计很简单。从一个给定的金字塔级别获取一个带有C通道的输入特征映射，子网应用四个3×3 conv层，每个层都有C个卷积核，每个层都有ReLU激活，然后是一个3×3conv层和K*A个卷积核。最后附加sigmoid激活来输出每个空间位置的KA二进制预测。见图3（c）。我们在大多数实验中使用C=256和A=9。

与RPN相比，我们的对象分类子网更深，只使用3×3的卷积，并且不与box回归子网共享参数（如下所述）。我们发现这些更高层次的设计决策比特定的超参数值更重要。

盒回归子网：与目标分类子网并行，我们在每个金字塔层上附加另一个小的FCN，以便将每个anchor的偏移量回归到附近的ground-truth（如果存在的话）。盒回归子网的设计与分类子网相同，只是它以每个空间位置的4A线性输出终止，见图3（d）。对于每个空间位置的每个anchor，这4个输出预测anchor和ground-truth之间的相对偏移（我们使用RCNN中的标准框参数化）。我们注意到，与最近的工作不同，我们使用了一个与类无关的边界盒回归器，它使用的参数较少，但同样有效。目标分类子网和盒回归子网虽然共享一个公共结构，但使用不同的参数。

4.1推理与训练

推论：RetinaNet形成了一个由ResNet-FPN为主干网、分类子网和盒回归子网组成的单一FCN，见图3。因此，推理只涉及通过网络简单的前馈图像。为了提高速度，在将检测器置信度设为0.05后，我们只解码每个FPN级别最多1k个最高得分预测的框预测。将所有级别的顶层预测合并，并应用阈值为0.5的非极大值抑制来获得最终的检测结果。

Focal损失：我们使用本文中引入的焦点损失作为分类子网输出上的损失。如我们在§5中所示，我们发现γ=2在实际中工作良好，并且当γ∈[0.5,5]时RetinaNet是相对稳健的。我们强调在训练RetinaNet时，焦点损失将应用于每个采样图像中所有~100k anchor。这与使用启发式抽样（RPN）或难样本挖掘（OHEM，SSD）来为每个小批量选择一组anchor（例如256个）的常见做法形成鲜明对比。图像的总焦距损失计算为所有~100k anchor的焦距损失之和，按分配给ground-truth的anchor 数量进行归一化。我们通过指定的anchor的数量来进行标准化，而不是总anchor，因为绝大多数anchor

都是负例，并且在Focal Loss的损失值可以忽略不计。最后，我们注意到，分配给稀有类的权重α也有一个稳定的范围，但它与γ相互作用，因此有必要同时选择两者（见表1a和1b）。一般来说，随着γ的增加，α应略微减小（对于γ=2，α=0.25效果最好）。

初始化：我们用ResNet-50-FPN和ResNet-101-FPN作为主干网络进行训练。其中基本ResNet-50和ResNet-101模型是在ImageNet1k上预先训练的。为FPN添加的新层初始化为。除了RetinaNet子网中的最后一层外，所有新的conv层都初始化为偏置b=0，高斯权重填充为σ=0.01。对于分类子网的最后一个conv层，我们将偏差初始化设置为b=−log（（1−π）/π），其中π指定训练开始时，每个anchor都应标记为前景，置信度为~π。我们在所有实验中都使用π=.01，尽管结果对精确值是可靠的。如§3.3所述，该初始化可防止大量背景anchor在训练的第一次迭代中产生较大的不稳定损失值。

优化：采用随机梯度下降（SGD）训练RetinaNet。我们在8个GPU上同步进行SGD，每个小批量总共有16张图像（每个GPU 2个图像）。除非另有规定，否则所有模型都被训练为90k次迭代，初始学习率为0.01，然后在60k时除以10，在80k迭代时再次除以10。我们使用水平图像翻转作为数据增强的唯一形式，除非另有说明。重量衰减为0.0001，动量为0.9。训练损失是焦点损失和用于盒回归的标准平滑l1损失的总和。表1e中模型的训练时间在10到35小时之间。

α	AP	AP_{50}	AP_{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) **Varying α for CE loss** ($\gamma = 0$)

γ	α	AP	AP_{50}	AP_{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) **Varying γ for FL** (w. optimal α)

#sc	#ar	AP	AP_{50}	AP_{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) **Varying anchor scales and aspects**

method	batch size	nms thr	AP	AP_{50}	AP_{75}
OHEM	128	.7	31.1	47.2	33.2
OHEM	256	.7	31.8	48.8	33.9
OHEM	512	.7	30.6	47.0	32.6
OHEM	128	.5	32.8	50.3	35.1
OHEM	256	.5	31.0	47.4	33.0
OHEM	512	.5	27.6	42.0	29.2
OHEM 1:3	128	.5	31.1	47.2	33.2
OHEM 1:3	256	.5	28.3	42.4	30.3
OHEM 1:3	512	.5	24.0	35.5	25.8
FL	n/a	n/a	**36.0**	**54.9**	**38.7**

(d) **FL *vs.* OHEM** baselines (with ResNet-101-FPN)

depth	scale	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	time
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
101	800	37.8	57.5	40.8	20.2	41.1	49.2	198

(e) **Accuracy/speed trade-off** RetinaNet (on `test-dev`)

Table 1. **Ablation experiments for RetinaNet and Focal Loss (FL).** All models are trained on `trainval35k` and tested on `minival` unless noted. If not specified, default values are: $\gamma = 2$; anchors for 3 scales and 3 aspect ratios; ResNet-50-FPN backbone; and a 600 pixel train and test image scale. (a) RetinaNet with α-balanced CE achieves at most 31.1 AP. (b) In contrast, using FL with the same exact network gives a 2.9 AP gain and is fairly robust to exact γ/α settings. (c) Using 2-3 scale and 3 aspect ratio anchors yields good results after which point performance saturates. (d) FL outperforms the best variants of online hard example mining (OHEM) [31, 22] by over 3 points AP. (e) Accuracy/Speed trade-off of RetinaNet on `test-dev` for various network depths and image scales (see also Figure 2).

5.实验

我们给出了COCO比赛基准的Bounding Box检测轨迹的实验结果。训练时，我们遵循常见做法，并使用COCO trainval35k进行分割（将来自训练集的80k张图像与从40k 测试集中随机的35k图像子集合并）。我们报告了病变和敏感性研究，通过评估minival分裂（剩下的5k图像来自测试集）。对于我们的主要结果，我们报告了测试开发部分的COCO AP，它没有公共标签，需要使用评估服务器。

5.1训练密集检测

我们进行了大量的实验来分析密集检测中损失函数的行为以及各种优化策略。对于所有的实验，我们使用深度为50或101的resnet，并在顶部构建一个特征金字塔网络（FPN）。对于所有的消融实验，我们使用600像素的图像尺度进行训练和测试。

网络初始化：我们第一次尝试训练RetinaNet时使用了标准的交叉熵（CE）损失，而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发散。但是，简单地初始化模型的最后一层，使得检测到一个对象的先验概率为π=.01（见§4.1），可以有效地学习。用ResNet-50训练RetinaNet，这个初始化已经在COCO上产生了一个值得比较好的AP值30.2。结果对π的精确值不敏感，因此我们在所有实验中使用π=.01。

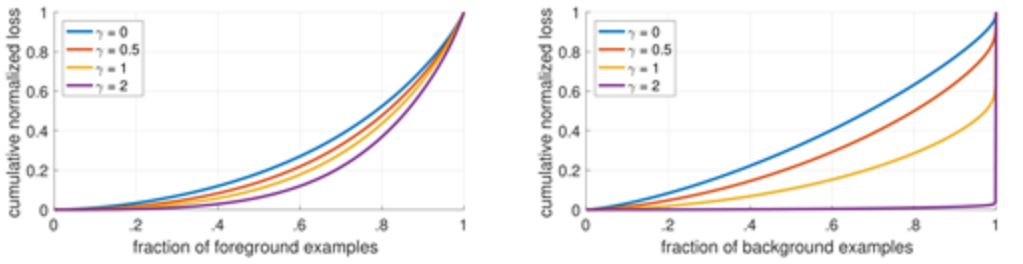


Figure 4. Cumulative distribution functions of the normalized loss for positive and negative samples for different values of γ for a *converged* model. The effect of changing γ on the distribution of the loss for positive examples is minor. For negatives, however, increasing γ heavily concentrates the loss on hard examples, focusing nearly all attention away from easy negatives.

平衡交叉熵：我们下一个改进学习的方法是使用§3.1中描述的α-平衡CE损失。各种α的结果如表1a所示。设置α=.75，可获得0.9点的增益AP。

焦点损失：表1b显示了使用我们提出的焦损失的结果。焦点损失引入了一个新的超参数，聚焦参数γ，它控制调制项的强度。当γ=0时，我们的损失等于CE损失。随着γ的增加，损失的形式会发生变化，因此低损失的"简单"示例将会很小，见图1。随着γ的增加，FL显示出比CE大的增益。当γ=2时，FL比α平衡CE损失提高了2.9AP。

对于表1b中的实验，为了公平比较，我们找到每个γ的最佳α。我们观察到，较低的α值被选为较高的γ值（因为负样本是向下加权的，所以不需要把重点放在正值上）。然而，总的来说，改变γ的好处要大得多，实际上最好的α值仅为[.25，.75]（我们测试了α∈[.01，.999]）。我们在所有实验中使用γ=2.0α=0.25，但α=0.5几乎同样有效（低0.4ap）。

焦点损失分析：为了更好地理解焦点损失，我们采用一个收敛模型分析了损失的经验分布。为此，我们采用默认的resnet101 600像素模型，γ=2（有36.0ap）。我们将此模型应用于大量随机图像，并对107个负样本和~105个正样本的预测概率进行抽样。接下来，分别计算这些正负样本的FL，并规范化损失，使之和为1。给定归一化损失，我们可以将损失从最低到最高排序，并绘制正样本和负样本以及γ不同设置下的累积分布函数（CDF）（即使模型是用γ=2训练的）。

正样本和负样本的累积分布函数如图4所示。如果我们观察正样本，我们会发现不同γ值的CDF看起来非常相似。例如，大约20%难样本的正样本约占正损失的一半，因为γ增加更多的损失集中在前20%的例子中，但影响很小。

γ对负样本的影响有显著差异。对于γ=0，正和γ非常相似。然而，随着γ的增加，更多的权重集中在难负样本上。事实上，在γ=2（我们的默认设置）下，绝大多数损失来自小部分样本。可以看出，FL可以有效地消除容易负样本带来的影响，把所有的注意力集中在难负样本上。

线上难样本挖掘（OHEM）：有文章建议通过使用高损失样例来构造小批量来改进两阶段检测器的训练。具体地说，在OHEM中，每个例子都是根据其损失来评分的，然后应用非极大值抑制（nms），并用损失最大的例子构造一个小批量。nms阈值和批处理大小是可调参数。和焦点损失一样，OHEM更强调错误分类的例子，但与FL不同，OHEM完全抛弃了简单的例子。我们还实现了SSD中使用的OHEM的一个变体：在将nms应用于所有示例之后，构造minibatch来强制执行正负1:3的比率，以帮助确保每个小批量都有足够的正样本。

我们在我们的一阶段检测设置中测试了两个OHEM变体，它具有很大的类不平衡性。原始OHEM策略和所选批次大小和nms阈值的"OHEM 1:3"策略的结果如表1d所示。这些结果使用FL在ResNet-101上训练的基线在该设置下达到36.0AP。相比之下，OHEM的最佳设置（无1:3比率，批次大小128，nms为0.5）达到32.8 AP。相差了3.2的AP，表明FL比OHEM更有效地训练密集型探测器。我们注意到，我们尝试了OHEM的其他参数设置和变体，但没有获得更好的结果。

Hinge损失：最后，在早期的实验中，我们尝试在pt上使用Hinge损失进行训练，将高于pt的某个确定值设为0。然而，这是不稳定的，我们没有设法取得有意义的结果。研究交替损失函数的结果见附录。

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [32]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [27]	DarkNet-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
RetinaNet (ours)	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet (ours)	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2

Table 2. **Object detection** *single-model* results (bounding box AP), *vs.* state-of-the-art on COCO `test-dev`. We show results for our RetinaNet-101-800 model, trained with scale jitter and for 1.5× longer than the same model from Table 1e. Our model achieves top results, outperforming both one-stage and two-stage models. For a detailed breakdown of speed versus accuracy see Table 1e and Figure 2.

5.2模型架构设计

anchor密度：在一阶段检测系统中，最重要的设计因素之一是anchor的覆盖密度。两阶段检测器可以使用区域池操作在任何位置、比例和纵横比下对box进行分类。相比之下，由于一阶段检测器使用固定的采样网格，我们通常在每个空间位置使用多个anchor，以覆盖不同比例和宽高比的box。

我们扫描了在FPN中每个空间位置和每个金字塔级别上使用的比例和纵横比anchor的数量。我们考虑从每个位置的单个方形anchor到每个位置的12个4中不同大小比例的anchor。使用ResNet-50的结果如表1c所示。仅使用一个方形anchor，就可以获得很好的AP（30.3）。然而，当每个位置使用3个尺度和3个纵横比时，AP可以提高近4个点（达到34.0）。在这项工作中，我们将此设置用于所有其他实验。最后，我们注意到，增加超过6-9个anchor没有更大的作用。因此，当两阶段系统可以对图像中的任意box进行分类时，更高的w.r.t.效果并不一定会更好。

速度与精度：较大的主干网会产生更高的准确性，但推理速度也更慢。同样，对于输入图像比例（由较短的图像侧定义）。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/精度折中曲线，并将其与最近在COCO test-dev公开数据的方法进行了比较。该图显示，加入了FL后，RetinaNet在所有现有方法中形成了一个上包线，低精度区域不起作用。带有ResNet-101-FPN和600像素图像比例尺（为简单起见，我们用RetinaNet-101-600表示）的RetinaNet与最近发布的ResNet101 FPN Faster R-CNN的精度相匹配，而每幅图像的运行时间为122毫秒，而在Nvidia M40 GPU上都测量到了这一点。使用更大的比例尺可以使RetinaNet超越所有两阶段方法的精确度，同时仍然更快。对于更快的运行时间，只有一个操作点（500像素输入），使用ResNet-50-FPN比ResNet-101-FPN更好。解决高帧速率机制可能需要特殊的网络设计，这超出了本工作的范围。我们注意到，在发表之后，现在可以通过中Faster R-CNN的变体来获得更快和更准确的结果。

5.3与最新技术相比

我们在COCO比赛数据集上评估RetinaNet，并将测试开发结果与最新最先进的方法（包括单阶段和两阶段模型）进行比较。表2中给出了我们使用比例抖动训练的RetinaNet-101-800模型的结果，该模型比表1e中的模型长1.5倍（给出了1.3ap增益）。与现有的单阶段方法相比，我们的方法与最接近的竞争对手DSSD实现了5.9点的AP差距（39.1比33.2），同时速度更快，见图2。与最近的两阶段方法相比，RetinaNet比基于Inception-ResNet-v2-TDM的性能最好的R-CNN模型高出2.3个点的差距。以ResNeXt32x8d-101-FPN作为RetinaNet主干网络进一步提高了结果1.7AP，超过了COCO上的40AP。

6. 结论

在这项工作中，我们确定了类不平衡是阻碍一阶段目标检测器性能没发超过两阶段方法的主要障碍。为了解决这一问题，我们提出了焦点损失，它将调制项应用于交叉熵损失，以便集中学习难负样本。我们的方法简单有效。我们通过设计一个全卷积的单级检测器来证明它的有效性，并报告了大量的实验分析，表明它达到了最先进的精度和速度。

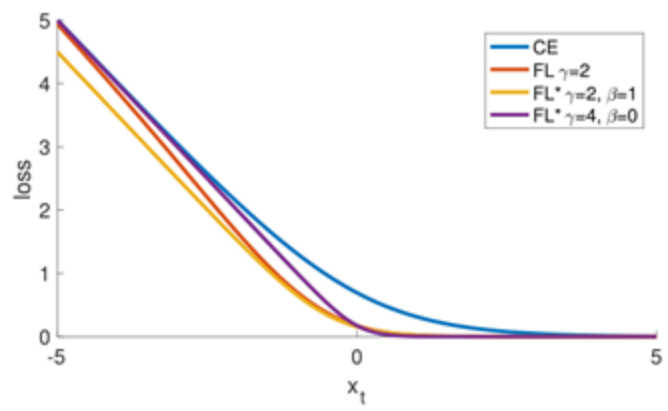
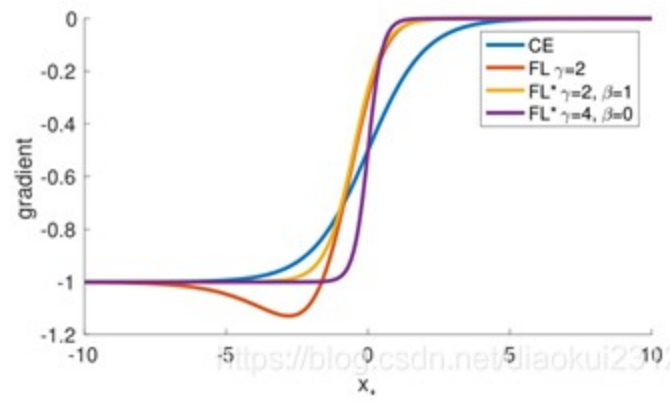


Figure 5. Focal loss variants compared to the cross entropy as a function of $x_t = yx$. Both the original FL and alternate variant FL* reduce the relative loss for well-classified examples ($x_t > 0$).



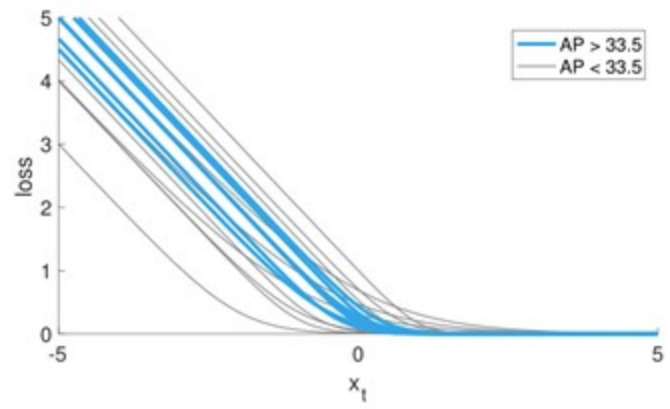


Figure 7. Effectiveness of FL* with various settings γ and β. The plots are color coded such that effective settings are shown in blue.

loss	γ	β	AP	AP_{50}	AP_{75}
CE	–	–	31.1	49.4	33.0
FL	2.0	–	34.0	52.5	36.5
FL*	2.0	1.0	33.8	52.7	36.3
FL*	4.0	0.0	33.9	51.8	36.4

Table 3. Results of FL and FL* versus CE for select settings.