Focal Loss for Dense Object Detection

Abstract

The highest accuracy object detectors to date are based on a two-stage approach popularized by R-CNN, where a classifier is applied to a sparse set of candidate object locations. In contrast, one-stage detectors that are applied over a regular, dense sampling of possible object locations have the potential to be faster and simpler, but have trailed the accuracy of two-stage detectors thus far. In this paper, we investigate why this is the case. We discover that the extreme foreground-background class imbalance encountered during training of dense detectors is the central cause. We propose to address this class imbalance by reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples. Our novel Focal Loss focuses training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training. To evaluate the effectiveness of our loss, we design and train a simple dense detector we call RetinaNet. Our results show that when trained with the focal loss, RetinaNet is able to match the speed of previous one-stage detectors while surpassing the accuracy of all existing state-of-the-art two-stage detectors. Code is at: https://github.com/facebookresearch/Detectron.

迄今为止最高精度的目标检测是基于 在 R-CNN 推广的两阶段方法上，其中分类器应用于一组稀疏的候选对象位置。相比之下，应用于对可能的物体位置进行常规、密集采样的单阶段目标检测有可能更快、更简单，但是到目前为止准确度落后于二阶段目标检测。在本文中， 我们调查为什么会这样。我们发现，在密集检测器的训练过程中遇到的极端前景-背景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，使其降低分配给分类良好的例子的损失，来解决这种类别不平衡。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。我们的结果表明，当用焦点损失进行训练时，RetinaNet能够与以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.*

1. Introduction

Current state-of-the-art object detectors are based on a two-stage, proposal-driven mechanism. As popularized in the R-CNN framework [11], the first stage generates a sparse set of candidate object locations and the second stage classifies each candidate location as one of the foreground classes or as background using a convolutional neural network. Through a sequence of advances [10, 28, 20, 14], this two-stage framework consistently achieves top accuracy on the challenging COCO benchmark [21]. Despite the success of two-stage detectors, a natural question to ask is: could a simple one-stage detector achieve similar accuracy? One stage detectors are applied over a regular, dense sampling of object locations, scales, and aspect ratios. Recent work on one-stage detectors, such as YOLO [26, 27] and SSD [22, 9], demonstrates promising results, yielding faster detectors with accuracy within 10-40% relative to state-of-the-art two-stage methods. This paper pushes the envelop further: we present a one-stage object detector that, for the first time, matches the state-of-the-art COCO AP of more complex two-stage detectors, such as the Feature Pyramid Network (FPN) [20] or Mask R-CNN [14] variants of Faster R-CNN [28]. To achieve this result, we identify class imbalance during training as the main obstacle impeding one-stage detector from achieving state-of-the-art accuracy and propose a new loss function that eliminates this barrier. Class imbalance is addressed in R-CNN-like detectors by a two-stage cascade and sampling heuristics. The proposal stage (e.g., Selective Search [35], EdgeBoxes [39], DeepMask [24, 25], RPN [28]) rapidly narrows down the number of candidate object locations to a small number (e.g., 1-2k), filtering out most background samples. In the second classification stage, sampling heuristics, such as a fixed foreground-to-background ratio (1:3), or online hard example mining (OHEM)

[31], are performed to maintain a manageable balance between foreground and background. In contrast, a one-stage detector must process a much larger set of candidate object locations regularly sampled across an image. In practice this often amounts to enumerating 100k locations that densely cover spatial positions, scales, and aspect ratios. While similar sampling heuristics may also be applied, they are inefficient as the training procedure is still dominated by easily classified background examples. This inefficiency is a classic problem in object detection that is typically addressed via techniques such as bootstrapping [33, 29] or hard example mining [37, 8, 31]. In this paper, we propose a new loss function that acts as a more effective alternative to previous approaches for dealing with class imbalance. The loss function is a dynamically scaled cross entropy loss, where the scaling factor decays to zero as confidence in the correct class increases, see Figure 1. Intuitively, this scaling factor can automatically down-weight the contribution of easy examples during training and rapidly focus the model on hard examples. Experiments show that our proposed Focal Loss enables us to train a high-accuracy, one-stage detector that significantly outperforms the alternatives of training with the sampling heuristics or hard example mining, the previous state-of-the-art techniques for training one-stage detectors. Finally, we note that the exact form of the focal loss is not crucial, and we show other instantiations can achieve similar results. To demonstrate the effectiveness of the proposed focal loss, we design a simple one-stage object detector called RetinaNet, named for its dense sampling of object locations in an input image. Its design features an efficient in-network feature pyramid and use of anchor boxes. It draws on a variety of recent ideas from [22, 6, 28, 20]. RetinaNet is efficient and accurate; our best model, based on a ResNet-101- FPN backbone, achieves a COCO test-dev AP of 39.1 while running at 5 fps, surpassing the previously best published single-model results from both one and two-stage detectors, see Figure 2.

一、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框架中所推广的那样[11]，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]，这个两阶段框架在具有挑战性的COCO基准[21]上持续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：一个简单的单阶段检测器能否达到类似的精度？单阶段检测器被应用于物体位置、比例和长宽比的有规律的密集采样。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，其精确度在10-40%之间。本文进一步推动了这一进程（envelop）：我们提出了一个单阶段物体检测器，它首次与更复杂的两阶段检测器，如特征金字塔网络（FPN）[20]或快速R-CNN[28]的Mask R-CNN[14]变体的最先进的COCO AP相匹配。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阶段目标检测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（eliminates）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（two-stage cascade and sampling heuristics）方法来解决的。建议阶段（例如选择性搜索[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（例如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，一个单阶段检测器必须处理更大的候选物体位置集，并在整个图像中定期采样。在实践中，这往往相当于列举了*10*万个位置，密集地覆盖了空间位置、比例和长宽比。虽然类似的采样启发式方法也可以应用，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，作为以前处理类不平衡的方法的一个更有效的替代。该损失函数是一个动态缩放的交叉熵损失，其中缩放因子随着对正确类的置信度增加而衰减为零，见图1。直观地说，这个比例因子可以在*训练期间自动降低容易的例子的贡献，并迅速将模型集中在困难的例子上。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，其性能明显优于用抽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得名。它的设计特点是一个高效的网络内特征金字塔和使用锚箱。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骨干，在以5帧/秒的速度运行时，COCO test-dev AP达到了39.1，超过了之前公布的单阶段和二阶段目标检测的最佳单模型结果，见图2。

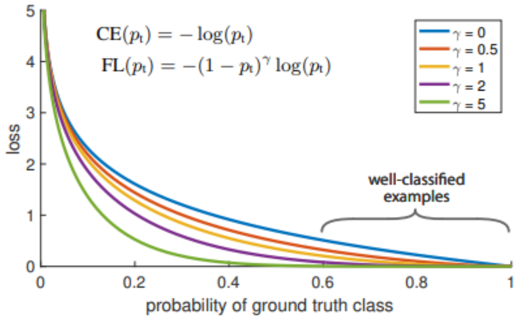
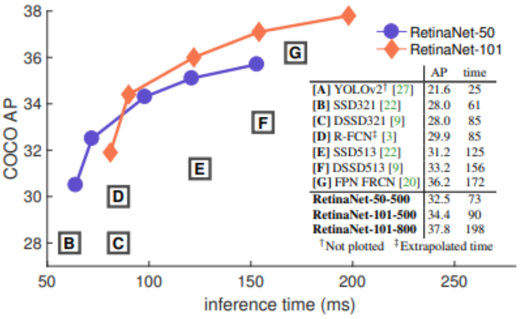


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will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



	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.

2. Related Work

Classic Object Detectors:

The sliding-window paradigm,in which a classifier is applied on a dense image grid, has a long and rich history. One of the earliest successes is the classic work of LeCun et al. who applied convolutional neural networks to handwritten digit recognition [19, 36]. Viola and Jones [37] used boosted object detectors for facedetection, leading to widespread adoption of such models. The introduction of HOG [4] and integral channel features[5] gave rise to effective methods for pedestrian detection.DPMs [8] helped extend dense detectors to more general object categories and had top results on PASCAL [7] for many years. While the sliding window approach was the leading detection paradigm in classic computer vision, with the resurgence of deep learning [18], two-stage detectors, described next, quickly came to dominate object detection.

经典物体检测器。
滑动窗口范式（paradigm）。 在这种模式中，分类器被应用于密集的图像网格上，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。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]。Viola和Jones[37]将增强的物体检测器用于人脸检测，从而导致广泛采用。检测，从而导致了此类模型的广泛采用。HOG[4]和积分通道特征的引入 [5]产生了用于行人检测的有效方法。 DPMs[8]有助于将密集型检测器扩展到更普遍的 物体类别，并在PASCAL[7]上取得了多年的最佳结果。多年来在PASCAL[7]上取得了最佳结果。虽然滑动窗口方法是领先的检测范式，但随着深度学习[7]的重新兴起，滑动窗口方法在经典计算机视觉中
随着深度学习[18]的重新兴起，两阶段检测器。接下来描述的两阶段检测器很快就主导了物体检测。

Two-stage Detectors:

The dominant paradigm in modern object detection is based on a two-stage approach. As pioneered in the Selective Search work [35], the first stage generates a sparse set of candidate proposals that should contain all objects while filtering out the majority of negative locations, and the second stage classifies the proposals into foreground classes / background. R-CNN [11] upgraded the second-stage classifier to a convolutional network yielding large gains in accuracy and ushering in the modern era of object detection. R-CNN was improved over the years, both in terms of speed [15, 10] and by using learned object proposals [6, 24, 28]. Region Proposal Networks (RPN) integrated proposal generation with the second-stage classifier into a single convolution network, forming the Faster RCNN framework [28]. Numerous extensions to this framework have been proposed, e.g. [20, 31, 32, 16, 14].

现代物体检测的主流范式 物体检测的主导模式是基于两阶段的方法。正如在选择性搜索工作中所开创的那样[35]，第一阶段产生一个稀疏的候选建议集，该建议集应该包含所有的物体，同时过滤掉大部分的负面位置。第二阶段将这些建议分类为 前景类/背景类。R-CNN[11]将第二阶段的分类器升级为 第二阶段的分类器升级为卷积网络，产生了在准确性方面有了很大的提高，并开创了现代物体检测的时代。多年来，R-CNN得到了改进，包括 在速度方面[15, 10]，以及通过使用学习到的物体提议[6, 24, 28]。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单一的卷积网络中。到一个单一的卷积网络中，形成了Faster RCNN框架[28]。这个框架的许多扩展已经被提出，例如[20, 31, 32, 16, 14]。

One-stage Detectors:

OverFeat [30] was one of the first modern one-stage object detector based on deep networks.More recently SSD [22, 9] and YOLO [26, 27] have renewed interest in one-stage methods. These detectors have been tuned for speed but their accuracy trails that of two stage methods. SSD has a 10-20% lower AP, while YOLO focuses on an even more extreme speed/accuracy trade-off. See Figure 2. Recent work showed that two-stage detectors can be made fast simply by reducing input image resolution and the number of proposals, but one-stage methods trailed in accuracy even with a larger compute budget [17]. In contrast, the aim of this work is to understand if one-stage detectors can match or surpass the accuracy of two-stage detectors while running at similar or faster speeds. The design of our RetinaNet detector shares many similarities with previous dense detectors, in particular the concept of 'anchors' introduced by RPN [28] and use of features pyramids as in SSD [22] and FPN [20]. We emphasize that our simple detector achieves top results not based on innovations in network design but due to our novel loss.

OverFeat[30]是第一个 基于深度网络的现代单阶段物体检测器。 最近，SSD[22, 9]和YOLO[26, 27]重新引起了人们对单阶段方法的兴趣。这些检测器 被调整为速度，但它们的准确性却落后于两阶段的方法。SSD的AP值要低10-20%，而YOLO 则专注于更极端的速度/准确度的权衡。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，但单阶段方法即使有更大的计算预算，其准确性也会落后[17]。相比之下，这项工作的目的是了解单阶段检测器是否能够在运行速度类似或更快的情况下，匹配或超越两阶段检测器的准确性。我们的RetinaNet检测器的设计与以前的密集型检测器有很多相似之处，特别是RPN[28]引入的 "锚 "的概念，以及像SSD[22]和FPN[20]那样使用特征金字塔。我们强调，我们的简单检测器取得的最佳结果不是基于网络设计的创新，而是由于于网络设计的创新，而是由于我们新颖的损失。

Class Imbalance:

Both classic one-stage object detection methods, like boosted detectors [37, 5] and DPMs [8], and more recent methods, like SSD [22], face a large class imbalance during training. These detectors evaluate 104-105 candidate locations per image but only a few locations contain objects. This imbalance causes two problems: (1) training is inefficient as most locations are easy negatives that contribute no useful learning signal; (2) en masse,the easy negatives can overwhelm training and lead to degenerate models. A common solution is to perform some form of hard negative mining [33, 37, 8, 31, 22] that samples hard examples during training or more complex sampling/reweighing schemes [2]. In contrast, we show that our proposed focal loss naturally handles the class imbalance faced by a one-stage detector and allows us to efficiently train on all examples without sampling and without easy negatives overwhelming the loss and computed gradients.

类别不平衡。
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。这些检测器对每幅图像的104-105个候选位置进行评估，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容易被否定的，没有贡献有用的学习信号。
(2)大量的容易被否定的位置会淹没训练，导致退化的模型。一个常见的解决方案是进行某种 一种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，或者采用更复杂的采样/重权方案[2]。与此相反，我们表明，我们的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使我们能够有效地对所有的例子进行训练，而不需要抽样，也不需要轻易负数压倒了损失和计算梯度。

Robust Estimation:

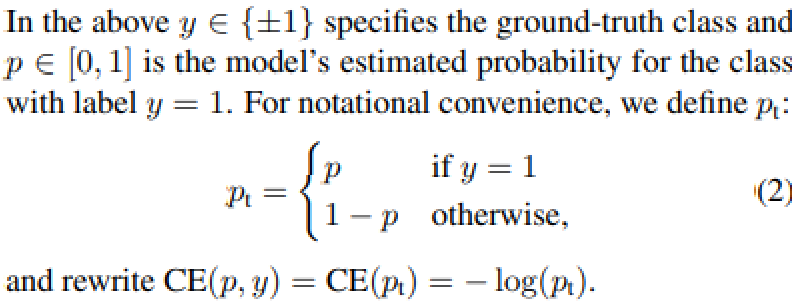
There has been much interest in designing robust loss functions (e.g., Huber loss [13]) that reduce the contribution of outliers by down-weighting the loss of examples with large errors (hard examples). In contrast, rather than addressing outliers, our focal loss is designed to address class imbalance by down-weighting inliers (easy examples) such that their contribution to the total loss is small even if their number is large. In other words, the focal loss performs the opposite role of a robust loss: it focuses training on a sparse set of hard examples.

稳健估计。人们对设计稳健的损失函数（例如Huber损失[13]）很感兴趣，这些函数通过降低误差大的例子（硬例子）的损失权重来减少离群点的贡献。误差大的例子（硬例子）的损失。与此相反。我们的焦点损失不是为了解决离群值，而是为了解决通过降低异常值（容易的例子）的权重来解决类的不平衡问题。例子），从而使它们对总损失的贡献是小，即使它们的数量很大。换句话说，焦点损失损失发挥了与稳健损失相反的作用：它将训练重点放在稀疏的例子上。训练的重点是稀疏的硬例集。

3.Focal Loss

The Focal Loss is designed to address the one-stage object detection scenario in which there is an extreme imbalance between foreground and background classes during training (e.g., 1:1000). We introduce the focal loss starting from the cross entropy (CE) loss for binary classification1:

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



In the above $y \in \{\pm 1\}$ specifies the ground-truth class and $p \in [0,1]$ is the model's estimated probability for the class with label $y=1$. For notational convenience, we define p_t:

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

and rewrite $\text{CE}(p,y) = \text{CE}(p_t) = -\log(p_t)$.

The CE loss can be seen as the blue (top) curve in Figure 1. One notable property of this loss, which can be easily seen in its plot, is that even examples that are easily classified (pt .5) incur a loss with non-trivial magnitude. When summed over a large number of easy examples, these small loss values can overwhelm the rare class.

焦点损失焦点损失的设计是为了解决单阶段物体检测的情况，即在训练期间前景和背景类之间存在极端不平衡的情况。训练期间，前景类和背景类之间存在极大的不平衡（例如，1:1000）。我们从交叉熵（CE）开始引入焦点损失从二元分类的交叉熵（CE）损失开始1。CE损失可以看作是图1中的蓝色（顶部）曲线。这种损失的一个显著特点是，即使是在二元分类的情况下，它也能在图中显示出来。这个损失的一个显著特点是，即使是容易分类的例子（pt.5）也会产生一个非微不足道的损失。当把大量容易分类的例子加起来时，这些小的损失值可以压倒稀有类。

3.1. Balanced Cross Entropy

A common method for addressing class imbalance is to introduce a weighting factor α ∈ [0, 1] for class 1 and 1−α for class −1. In practice α may be set by inverse class frequency or treated as a hyperparameter to set by cross validation. For notational convenience, we define αt analogously to how we defined pt We write the α-balanced CE loss as:

$$CE(p_t) = -\alpha_t log(p_t).$$

This loss is a simple extension to CE that we consider as an experimental baseline for our proposed focal loss.

平衡交叉熵

解决类不平衡的一个常用方法是引入一个加权因子α∈[0, 1]来表示类1和1-α代表-1类。在实践中，α可以通过反类别频率来设置，或者作为一个超参数通过交叉验证来设置。为便于记述，我们对αt的定义类似于的定义与我们定义pt. 我们将α-平衡的CE损失写成。
这个损失是CE的一个简单扩展，我们将其视为我们提出的焦点损失的实验基线。

3.2. Focal Loss Definition

As our experiments will show, the large class imbalance encountered during training of dense detectors overwhelms the cross entropy loss. Easily classified negatives comprise the majority of the loss and dominate the gradient. While α balances the importance of positive/negative examples, it does not differentiate between easy/hard examples. Instead, we propose to reshape the loss function to down-weight easy examples and thus focus training on hard negatives. More formally, we propose to add a modulating factor(1 − pt)γto the cross entropy loss, with tunable focusing parameter γ ≥ 0. We define the focal loss as:

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

.2. 焦点损失的定义正如我们的实验所显示的，在密集型检测器的训练过程中遇到的巨大的类不平衡性在密集型检测器的训练过程中遇到的巨大的类不平衡，压倒交叉熵损失。容易分类的负样本包括损失的大部分，并主导着梯度。而α平衡了阳性/阴性样本的重要性，但它没有区分容易/困难的例子。它并不区分容易/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。损失函数，从而将训练的重点放在困难的负面例子上。更正式地说，我们建议增加一个调节因子(1-pt)γ到交叉熵损失中，可调整的聚焦参数γ≥0。

The focal loss is visualized for several values of γ ∈[0, 5] in Figure 1. We note two properties of the focal loss.

(1) When an example is misclassified and pt is small, the modulating factor is near 1 and the loss is unaffected. As pt → 1, the factor goes to 0 and the loss for well-classified examples is down-weighted.

(2) The focusing parameter γ smoothly adjusts the rate at which easy examples are down weighted. When γ = 0, FL is equivalent to CE, and as γ is increased the effect of the modulating factor is likewise increased (we found γ = 2 to work best in our experiments). Intuitively, the modulating factor reduces the loss contribution from easy examples and extends the range in which an example receives low loss. For instance, with γ = 2, an example classified with pt = 0.9 would have 100× lower loss compared with CE and with pt ≈ 0.968 it would have 1000× lower loss. This in turn increases the importance of correcting misclassified examples (whose loss is scaled down by at most 4× for pt ≤ .5 and γ = 2). In practice we use an α-balanced variant of the focal loss:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma} log(p_t)$$

γ∈的几个值的焦距损失是可视化的。[0, 5]的情况下，图1显示了焦点损失。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pt很小，则调制因子接近1，损失不受影响。当pt→1时，该系数变为0，分类良好的例子的损失
例子的损失被降低了权重。

(2) 聚焦参数γ平稳地调整容易的例子被降权的速度。当γ=0时，FL等同于CE，当γ增加时，调节因子的效果就会下降。增加，调节因子的效果也同样增加（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容易的例子的损失贡献，扩大了一个样本获得低损失的范围。例如，在γ=2的情况下，一个归类为pt = 0.9的例子将有100倍的更低损失，而pt≈0.968时，它将有1000倍的更低损失。这反过来又增加了这反过来又增加了纠正错误分类的例子的重要性（这些例子的损失在pt=0.9时最多降低4倍）。在pt≤.5和γ=2的情况下，其损失最多下降4倍）。

In practice we use an α-balanced variant of the focal loss: We adopt this form in our experiments as it yields slightly improved accuracy over the non-α-balanced form. Finally, we note that the implementation of the loss layer combines the sigmoid operation for computing p with the loss computation, resulting in greater numerical stability. While in our main experimental results we use the focal loss definition above, its precise form is not crucial. In the appendix we consider other instantiations of the focal loss and demonstrate that these can be equally effective.

在实践中，我们使用一个α平衡的焦点损失的变体。我们在实验中采用了这种形式，因为它比非α-平衡形式的准确性略有提高。最后。我们注意到，损失层的实现结合了计算p的sigmoid操作与损失计算相结合，从而产生更大的数值稳定性。虽然在我们的主要实验结果中，我们使用了上述的焦距损失定义，但其精确形式并不重要。在在附录中，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. Class Imbalance and Model Initialization

Binary classification models are by default initialized to have equal probability of outputting either y = −1 or 1. Under such an initialization, in the presence of class imbalance, the loss due to the frequent class can dominate total loss and cause instability in early training. To counter this, we introduce the concept of a ‘prior’ for the value of p estimated by the model for the rare class (foreground) at the start of training. We denote the prior by π and set it so that the model’s estimated p for examples of the rare class is low, e.g. 0.01. We note that this is a change in model initialization (see §4.1) and not of the loss function. We found this to improve training stability for both the cross entropy and focal loss in the case of heavy class imbalance.

3.3. 类不平衡和模型初始化

二元分类模型在默认情况下被初始化为有相等的概率输出y=-1或1。在这样的初始化下，**在类不平衡的情况下，频繁出现的类所造成的损失可能会支配总的损失**，造成早期训练的不稳定。为了解决这个问题。我们引入了 "先验 "的概念，即在训练开始时由模型对稀有类（前景）估计的p值。训练的开始。我们用π来表示这个先验，并将其设定为模型对稀有类例子的p估计值很低。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。我们发现这来改善交叉熵和焦点损失的训练稳定性和在严重的类不平衡情况下的焦点损失。

3.4. Class Imbalance and Two-stage Detectors

Two-stage detectors are often trained with the cross entropy loss without use of α-balancing or our proposed loss. Instead, they address class imbalance through two mechanisms:

(1) a two-stage cascade and

(2) biased mini-batch sampling.

The first cascade stage is an object proposal mechanism [35, 24, 28] that reduces the nearly infinite set of possible object locations down to one or two thousand. Importantly, the selected proposals are not random, but are likely to correspond to true object locations, which removes the vast majority of easy negatives. When training the second stage, biased sampling is typically used to construct mini-batches that contain, for instance, a 1:3 ratio of positive to negative examples. This ratio is like an implicit α balancing factor that is implemented via sampling. Our proposed focal loss is designed to address these mechanisms in a one-stage detection system directly via the loss function.

3.4. 类不平衡和两级检测器
两阶段检测器通常使用交叉熵损失进行训练，不使用α-平衡或我们提出的损失。相反，它们通过两种机制解决类不平衡问题。

(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]，该机制将几乎是无限的物体位置集合减少到一个或多个。可能的物体位置减少到一两千个。重要的是，选择的建议不是随机的，而是可能对应于真实的物体位置，这就消除了绝大多数容易被否定的位置。在训练第二阶段时，通常使用有偏见的抽样来构建mini-batch，其中包含，例如，1:3的正反面例子的比例。这个比例就像一个隐含的α平衡因子，通过采样来实现。我们提出的焦点损失旨在解决这些机制。直接通过损失函数在一个单阶段检测系统中解决这些机制。

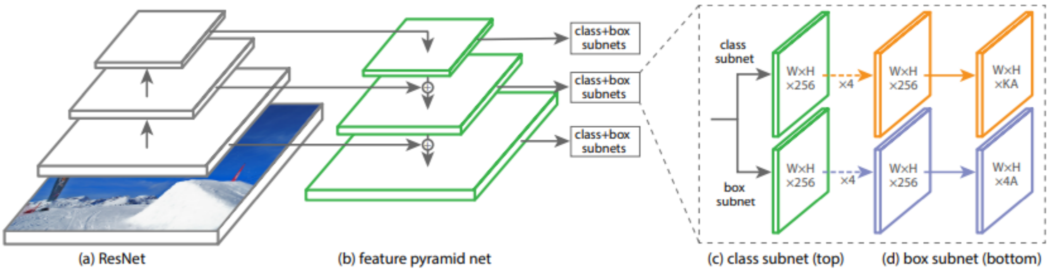


Figure 3. The one-stage **RetinaNet** network architecture uses a Feature Pyramid Network (FPN) [20] backbone on top of a feedforward ResNet architecture [16] (a) to generate a rich, multi-scale convolutional feature pyramid (b). To this backbone RetinaNet attaches two subnetworks, one for classifying 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.

4.RetinaNet Detector

RetinaNet is a single, unified network composed of a backbone network and two task-specific subnetworks. The backbone is responsible for computing a convolutional feature map over an entire input image and is an off-the-self convolutional network. The first subnet performs convolutional object classification on the backbone's output; the second subnet performs convolutional bounding box regression. The two subnetworks feature a simple design that we propose specifically for one-stage, dense detection, see Figure 3. While there are many possible choices for the details of these components, most design parameters are not particularly sensitive to exact values as shown in the experiments. We describe each component of RetinaNet next.

4.视网膜网络检测器
RetinaNet是一个单一的、统一的网络，由一个主干网络和两个特定的子网络组成。主干网络负责计算整个输入图像的卷积特征图，是一个离自卷积网络。第一个子网络在主干网络的输出上进行卷积物体分类；第二个子网执行卷积边界盒回归。这两个子网络的特点是设计简单。专门为单阶段密集检测提出的简单设计，见图3。虽然这些组件的细节有许多可能的选择这些组件的细节有许多可能的选择，但大多数设计参数对确切的值并不特别敏感，正如实验中所显示的那样。接下来我们描述一下RetinaNet的每个组件。

Feature Pyramid Network Backbone:

We adopt the Feature Pyramid Network (FPN) from [20] as the backbone network for RetinaNet. In brief, FPN augments a standard convolutional network with a top-down pathway and lateral connections so the network efficiently constructs a
rich, multi-scale feature pyramid from a single resolution input image, see Figure 3(a)-(b). Each level of the pyramid can be used for detecting objects at a different scale. FPN improves multi-scale predictions

from fully convolutional networks (FCN) [23], as shown by its gains for RPN [28] and DeepMask-style proposals [24], as well at two-stage detectors such as Fast R-CNN [10] or Mask R-CNN [14]. Following [20], we build FPN on top of the ResNet architecture [16]. We construct a pyramid with levels P3 through P7, where l indicates pyramid level (Pl has resolution 2 l lower than the input). As in [20] all pyramid levels have C = 256 channels. Details of the pyramid generally follow [20] with a few modest differences.2 While many design choices are not crucial, we emphasize the use of the FPN backbone is; preliminary experiments using features from only the final ResNet layer yielded low AP.

特征金字塔网络骨架。

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用一个自上而下的路径和横向连接增强了标准的卷积网络，因此该网络可以有效地构建一个丰富的、多尺度的特征金字塔，从单一分辨率的输入图像，见图3（a）-（b）。金字塔的每一层可用于检测不同尺度的物体。FPN改善了来自全卷积网络（FCN）的多尺度预测。网络(FCN)[23]的多尺度预测，如其对RPN[28]的增益所示和DeepMask式的建议[24]，以及两阶段的检测器，如快速R-CNN[10]或Mask R-CNN[14]。按照[20]，我们在ResNet架构[16]的基础上建立FPN。我们构建了一个金字塔，级别为P3到P7，其中l表示金字塔级别（Pl的分辨率为2 l 低于输入）。如同在[20]中，所有的金字塔级别都有C = 256个通道。金字塔的细节大体上遵循[20]，但有一些小的差别。虽然许多设计选择并不关键，但我们强调使用FPN骨干是关键。虽然许多设计选择并不关键，但我们强调使用FPN主干是关键的；初步实验中只使用最后的ResNet层的特征，产生的AP很低。

Anchors:

We use translation-invariant anchor boxes similar to those in the RPN variant in [20]. The anchors have areas of 32^2 to 512^2 on pyramid levels P3 to P7, respectively. As in [20], at each pyramid level we use anchors at three aspect ratios {1:2, 1:1, 2:1}. For denser scale coverage than in [20], at each level we add anchors of sizes {2^0,2^1/3, 2^2/3} of the original set of 3 aspect ratio anchors. This improve AP in our setting. In total there are A = 9 anchors per level and across levels they cover the scale range 32 -
813 pixels with respect to the network’s input image. Each anchor is assigned a length K one-hot vector of classification targets, where K is the number of object classes, and a 4-vector of box regression targets. We use the assignment rule from RPN [28] but modified for multiclass detection and with adjusted thresholds. Specifically, anchors are assigned to ground-truth object boxes using an intersection-over-union (IoU) threshold of 0.5; and to background if their IoU is in [0, 0.4). As each anchor is assigned to at most one object box, we set the corresponding entry in its length K label vector to 1 and all other entries to 0. If an anchor is unassigned, which may happen with overlap in [0.4, 0.5), it is ignored during training. Box regression targets are computed as the offset between each anchor and its assigned object box, or omitted if there is no assignment.

锚点。

我们使用了类似于[20]中RPN变体的翻译不变的锚点盒。锚点有面积为322到5122，分别位于金字塔的P3到P7层。和[20]一样，在每个金字塔层，我们使用的锚点是三种长宽比{1：2，1：1，2：1}。为了实现比[20]更密集的尺度覆盖，我们在每一级增加了尺寸为{2^0,2^1/3, 2^2/3}的3个长宽比锚的原始集合。这在我们的设置中改善了AP。每个级别总共有A = 9个锚点每个级别有A = 9个锚点，在不同的级别中，它们覆盖的比例范围为32 -813个像素，相对于网络的输入图像。每个锚被分配一个长度为K的分类目标的单次向量。分类目标的长度为K的单次向量，其中K是物体类别的数量类的数量，以及一个4个盒式回归目标的向量。我们使用我们使用RPN[28]中的分配规则，但为多类检测和调整阈值进行了修改。具体来说。锚被分配到地面真实的物体箱中，使用交叉联合（IoU）阈值为0.5；如果它们的IoU在[0, 0.4]，则分配给背景。由于每个锚被分配到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在其长度为K的标签向量中的相应条目为1，所有其他条目为0。如果一个锚没有被分配，这可能发生在重叠的在[0.4, 0.5]中可能会出现这种情况，那么在训练中就会忽略它。箱体回归目标被计算为每个锚点和其指定的对象盒之间的偏移量。的偏移量，如果没有分配，则省略不计。

Classification Subnet:

The classification subnet predicts the probability of object presence at each spatial position for each of the A anchors and K object classes. This subnet is a small FCN attached to each FPN level; parameters of this subnet are shared across all pyramid levels. Its design is simple. Taking an input feature map with C channels from a given pyramid level, the subnet applies four 3×3 conv layers, each with C filters and each followed by ReLU activations, followed by a 3×3 conv layer with KA filters. Finally sigmoid activations are attached to output the KA binary predictions per spatial location, see Figure 3 (c). We use C = 256 and A = 9 in most experiments. In contrast to RPN [28], our object classification subnet is deeper, uses only 3×3 convs, and does not share parameters with the box regression subnet (described next). We found these higher-level design decisions to be more important than specific values of hyperparameters.

分类子网。

分类子网预测了物体在每个空间位置出现的概率锚点和K个物体类别中的每一个。这个子网是一个附属于每个FPN层的小型FCN；该子网的参数在所有金字塔层中共享。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。从一个给定的金字塔层中获取一个具有C通道的输入特征图的输入特征图，该子网应用四个3×3的卷积层，每个卷积层有C个滤波器，每个滤波器后有ReLU激活，然后是一个带有KA滤波器的3×3卷积层。最后附加sigmoid激活来输出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。RPN[28]相比，我们的物体分类子网更加深入，只使用3×3 convs，并且不与盒式回归子网共享参数（接下来描述）。我们我们发现这些更高层次的设计决定比超参数的具体数值更重要。

Box Regression Subnet:

In parallel with the object classification subnet, we attach another small FCN to each pyramid level for the purpose of regressing the offset from each anchor box to a nearby ground-truth object, if one exists. The design of the box regression subnet is identical to the classification subnet except that it terminates in 4A linear outputs per spatial location, see Figure 3 (d). For each of the A anchors per spatial location, these 4 outputs predict the relative offset between the anchor and the groundtruth box (we use the standard box parameterization from RCNN [11]). We note that unlike most recent work, we use a class-agnostic bounding box regressor which uses fewer parameters and we found to be equally effective. The object classification subnet and the box regression subnet, though sharing a common structure, use separate parameters.

箱体回归子网。

在物体分类子网的同时，我们在每个金字塔层面上附加一个小的FCN，目的是对每个锚箱的偏移量进行回归。锚箱到附近地面真实物体的偏移量（如果存在的话）。盒子回归子网的设计与分类子网相同的设计与分类子网相同，只是它在每个空间位置终止于4A的线性输出。每个空间位置的输出，见图3（d）。对于每一个每个空间位置的A个锚点，这4个输出预测锚点和真实箱体之间的相对偏移（我们使RCN[11]的标准箱体参数化）。我们注意到，与最近的大多数工作不同，我们使用了一个类的边界盒回归器，它使用的参数较少，我们发现它同样有效。对象分类子网和盒式回归子网，虽然共享一个共同的结构，但使用不同的参数。共享一个共同的结构，但使用单独的参数。

4.1. Inference and Training

Inference: RetinaNet forms a single FCN comprised of a ResNet-FPN backbone, a classification subnet, and a box regression subnet, see Figure 3. As such, inference involves simply forwarding an image through the network. To improve speed, we only decode box predictions from at most 1k top-scoring predictions per FPN level, after thresholding detector confidence at 0.05. The top predictions from all levels are merged and non-maximum suppression with a threshold of 0.5 is applied to yield the final detections.

4.1. 推理和训练

推理。RetinaNet形成了一个单一的FCN，由以下部分组成ResNet-FPN主干网、一个分类子网和一个盒式回归子网，见图3。因此，推理包括简单地通过网络转发一幅图像。为了提高速度，我们最多只从以下几个方面解码盒子预测每个FPN级别有1k个最高分预测，在检测器置信度达到0.05的阈值后。所有级别的最高预测值被合并所有级别的顶级预测被合并，并以0.5的阈值进行非最大限度的压制。阈值为0.5的非最大抑制，以产生最终的检测结果。

Focal Loss:

We use the focal loss introduced in this work as the loss on the output of the classification subnet. As we will show in §5, we find that γ = 2 works well in practice and the RetinaNet is relatively robust to γ ∈ [0.5, 5]. We emphasize that when training RetinaNet, the focal loss is applied to all ~100k anchors in each sampled image. This stands in contrast to common practice of using heuristic sampling (RPN) or hard example mining (OHEM, SSD) to select a small set of anchors (e.g., 256) for each minibatch. The total focal loss of an image is computed as the sum of the focal loss over all ~100k anchors, normalized by the number of anchors assigned to a ground-truth box. We perform the normalization by the number of assigned anchors, not total anchors, since the vast majority of anchors are easy negatives and receive negligible loss values under the focal loss. Finally we note that α, the weight assigned to the rare class, also has a stable range, but it interacts with γ making it necessary to select the two together (see Tables 1a and 1b). In general α should be decreased slightly as γ is increased (for γ = 2, α = 0.25 works best).

焦点损失。

我们使用本工作中引入的焦点损失作为分类子网输出的损失。正如我们我们将在第5节中展示，我们发现γ=2在实践中效果很好而且RetinaNet对γ∈[0.5, 5]相对稳健。我们我们强调，在训练RetinaNet时，焦点损失是应用于每个采样图像中的所有~10万个锚点。这这与使用启发式取样（RPN）或硬例的常见做法形成鲜明对比。启发式采样（RPN）或硬例挖掘（OHEM，SSD）来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）的常见做法不同。一幅图像的总焦点损失被计算为
所有10万个锚的焦点损失之和，以分配给一个地面的锚的数量为标准。归一化为分配给一个地面真实箱的锚的数量。我们通过指定的锚的数量进行归一化。锚的数量，而不是锚的总数，因为绝大多数的锚都是简单的因为绝大多数的锚都是容易被否定的，在焦点损失下收到的损失值可以忽略不计。损失。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，即类的权重，也有一个稳定的范围，但它与γ相互作用，使得有必要同时选择这两者（见表1a和1b）。一般来说，α应该随着γ的增加而略微下降。增加（对于γ = 2，α = 0.25效果最好）。

Initialization:

We experiment with ResNet-50-FPN and ResNet-101-FPN backbones [20]. The base ResNet-50 and ResNet-101 models are pre-trained on ImageNet1k; we use the models released by [16]. New layers added for FPN are initialized as in [20]. All new conv layers except the final one in the RetinaNet subnets are initialized with bias b = 0 and a Gaussian weight fill with σ = 0.01. For the final conv layer of the classification subnet, we set the bias initialization to b = − log((1 − π)/π), where π specifies that at the start of training every anchor should be labeled as foreground with confidence of ~π. We use π = .01 in all experiments, although results are robust to the exact value. As explained in §3.3, this initialization prevents the large number of background anchors from generating a large, destabilizing loss value in the first iteration of training.

初始化。

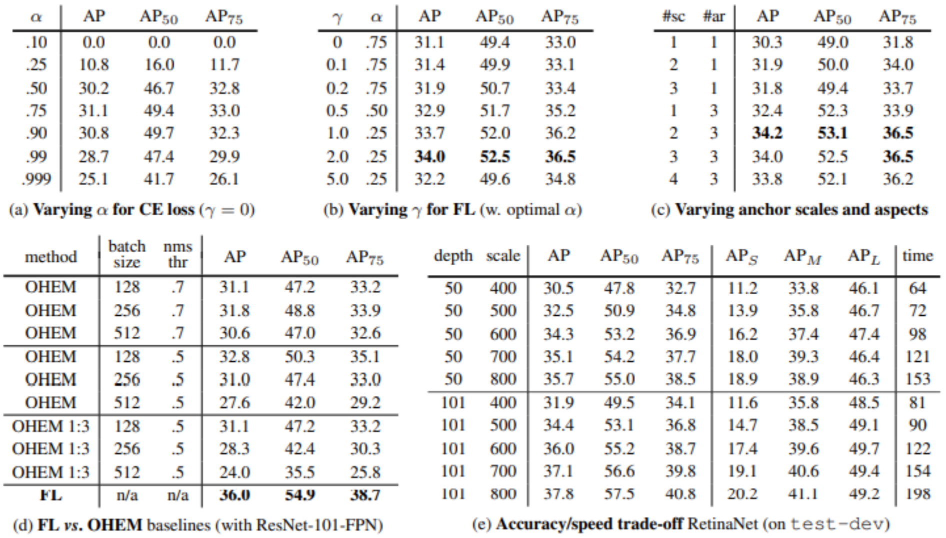
我们用ResNet-50-FPN和ResNet-101-FPN骨干网[20]。基础的ResNet-50和ResNet-50和ResNet-101模型是在ImageNet1k上预训练的；我们使用我们使用的是[16]发布的模型。为FPN添加的新层是初始化，如[20]。所有新的说服层，除了最后的在视网膜子网中的所有新的卷积层都被初始化为偏置b = 0和高斯权重填充 σ = 0.01。对于分类子网的最后一个卷积层的最后一个卷积层，我们将偏置初始化设置为b = -log((1 - π)/π)，其中π指定在训练开始时，每个锚点都应该是训练开始时，每个锚点都应该被标记为前景，置信度为~π。我们在所有的实验中都使用π=0.01，尽管结果对准确的数值很稳健。正如如第3.3节所述，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、不稳定的损失值。

Optimization:

RetinaNet is trained with stochastic gradient descent (SGD). We use synchronized SGD over 8 GPUs with a total of 16 images per minibatch (2 images per GPU). Unless otherwise specified, all models are trained for 90k iterations with an initial learning rate of 0.01, which is then divided by 10 at 60k and again at 80k iterations. We use horizontal image flipping as the only form of data augmentation unless otherwise noted. Weight decay of 0.0001 and momentum of 0.9 are used. The training loss is the sum the focal loss and the standard smooth L1 loss used for box regression [10]. Training time ranges between 10 and 35 hours for the models in Table 1e.

优化。

RetinaNet是用随机梯度下降法（SGD）训练的。我们在8个GPU上使用同步的SGD每个minibatch有16幅图像（每个GPU有2幅图像）。除非另有说明，所有的模型都训练了9万次迭代，初始学习率为0.01，然后在6万次迭代时除以10，再除以0.01。在6万次迭代时除以10，在8万次迭代时再次除以10。我们使用水平图像翻转作为唯一的数据增强形式，除非另有说明。权重衰减为0.0001，动量为使用了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L1损失之和。回归[10]。训练时间在10-35小时。



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`)

5. Experiments

We present experimental results on the bounding box detection track of the challenging COCO benchmark [21]. For training, we follow common practice [1, 20] and use the COCO trainval35k split (union of 80k images from train and a random 35k subset of images from the 40k image val split). We report lesion and sensitivity studies by evaluating on the minival split (the remaining 5k images from val). For our main results, we report COCO AP on the test-dev split, which has no public labels and requires use of the evaluation server.

实验

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。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]，使用COCO trainval35k（来自训练的8万张图片和随机的3万5千张图片的组合）。训练中的80k图像和40k图像估值中的35k随机图像子集的联合）。我们通过以下方式报告病变和敏感性研究我们报告了病变和敏感性的研究，通过评估minival分割（剩余的5k图像价值）。对于我们的主要结果，我们报告了COCO AP在我们的主要结果是在测试-开发部分报告COCO AP，该部分没有公共标签，需要使用评估服务器。使用评估服务器。

5.1. Training Dense Detection

We run numerous experiments to analyze the behavior of the loss function for dense detection along with various optimization strategies. For all experiments we use depth 50 or 101 ResNets [16] with a Feature Pyramid Network (FPN) [20] constructed on top. For all ablation studies we use an image scale of 600 pixels for training and testing.

Network Initialization:

Our first attempt to train RetinaNet uses standard cross entropy (CE) loss without any modifications to the initialization or learning strategy. This fails quickly, with the network diverging during training. However, simply initializing the last layer of our model such that the prior probability of detecting an object is π = .01 (see §4.1) enables effective learning. Training RetinaNet with ResNet-50 and this initialization already yields a respectable AP of 30.2 on COCO. Results are insensitive to the exact value of π so we use π = .01 for all experiments

5.1. 训练密集型检测
我们进行了大量的实验来分析密集检测的损失函数的行为。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50或101的ResNets[16]，并在上面构建一个特征金字塔网络(FPN)[20]构建在上面。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RetinaNet时使用了标准的交叉熵（CE）损失，没有对初始化或学习进行任何修改。对初始化或学习策略进行修改。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，使之成为探测到物体的先验概率为π=0.01。(见第4.1节）就可以有效地学习。训练RetinaNet用ResNet-50和这种初始化已经在COCO上产生了令人尊敬的30.2的AP。结果对π的精确值不敏感所以我们在所有的实验中都使用π=0.01。

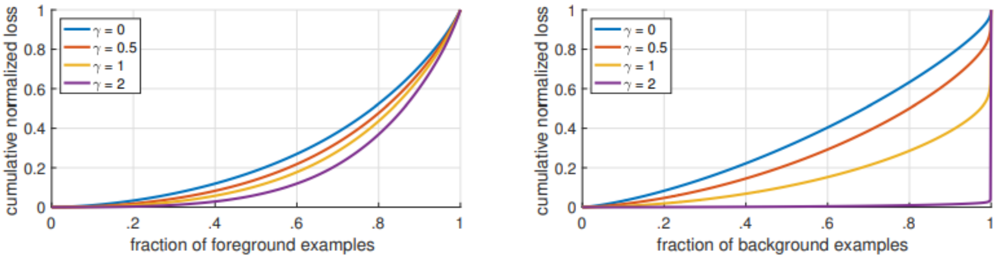


Figure 4. Cumulative distribution functions of the normalized loss for positive and negative samples for different values of γ for a *converged* model. The effect of changing γ on the distribution of the loss for positive examples is minor. For negatives, however, increasing γ heavily concentrates the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

Balanced Cross Entropy:

Our next attempt to improve learning involved using the α-balanced CE loss described in §3.1. Results for various α are shown in Table 1a. Setting α = .75 gives a gain of 0.9 points AP.

平衡的交叉熵。

我们的下一个尝试是改善我们的下一个尝试是使用§3.1中描述的α-平衡CE损失来改进学习。在第3.1节中描述。表1a中显示了各种α的结果。设置α=0.75，可以得到0.9分的AP增益。

Focal Loss:

Results using our proposed focal loss are shown in Table 1b. The focal loss introduces one new hyperparameter, the focusing parameter γ, that controls the strength of the modulating term. When γ = 0, our loss is equivalent to the CE loss. As γ increases, the shape of the loss changes so that “easy” examples with low loss get further discounted, see Figure 1. FL shows large gains over CE as γ is increased. With γ = 2, FL yields a 2.9 AP improvement over the α-balanced CE loss. For the experiments in Table 1b, for a fair comparison we find the best α for each γ. We observe that lower α’s are selected for higher γ’s (as easy negatives are down weighted, less emphasis needs to be placed on the positives). Overall, however, the benefit of changing γ is much larger, and indeed the best α’s ranged in just [.25,.75] (we tested α ∈ [.01, .999]). We use γ = 2.0 with α = .25 for all experiments but α = .5 works nearly as well (.4 AP lower).

焦点损失。

使用我们提出的焦点损失的结果是表1b所示。焦点损失引入了一个新的超参数，即聚焦参数γ，用来控制调制项的强度。当γ=0时，我们的损失相当于CE损失。相当于CE损失。随着γ的增加，损失的形状损失的形状发生了变化，因此低损失的 "简单 "例子会被进一步打折扣，见图1。随着γ的增加，FL显示出对CE的巨大收益。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损失有2.9个AP的改进。对于表1b中的实验，为了进行公平的比较我们观察到，较低的α在较高的γ中被选为最佳α。被选为较高的γ（因为容易的负面因素被降低了权重，所以不需要强调正面因素）。然而，总体而言，改变γ的好处要大得多。较大，事实上，最佳α的范围仅为[.25,.75]（我们测试了α∈[.01,]）。测试了α∈[.01, .999]）。我们在所有的实验中使用γ=2.0，α=0.25。实验，但α=.5的效果几乎一样好（0.4AP更低）。

Analysis of the Focal Loss:

To understand the focal loss better, we analyze the empirical distribution of the loss of a converged model. For this, we take take our default ResNet101 600-pixel model trained with γ = 2 (which has 36.0 AP). We apply this model to a large number of random images and sample the predicted probability for ~107 negative windows and ~105 positive windows. Next, separately for positives and negatives, we compute FL for these samples, and normalize the loss such that it sums to one. Given the normalized loss, we can sort the loss from lowest to highest and plot its cumulative distribution function (CDF) for both positive and negative samples and for different settings for γ (even though model was trained with γ = 2). Cumulative distribution functions for positive and negative samples are shown in Figure 4. If we observe the positive samples, we see that the CDF looks fairly similar for different values of γ. For example, approximately 20% of the hardest positive samples account for roughly half of the positive loss, as γ increases more of the loss gets concentrated in the top 20% of examples, but the effect is minor. The effect of γ on negative samples is dramatically different. For γ = 0, the positive and negative CDFs are quite similar. However, as γ increases, substantially more weight becomes concentrated on the hard negative examples. In fact, with γ = 2 (our default setting), the vast majority of the loss comes from a small fraction of samples. As can be seen, FL can effectively discount the effect of easy negatives, focusing all attention on the hard negative examples.

对病灶损失的分析。

为了更好地理解焦点损失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。收敛的模型。为此，我们以默认的ResNet101 600像素模型为例，用γ=2进行训练（它有36.0AP）。我们将这个模型应用于大量的随机图像，并对~107个负面窗口和~105个正面窗口的预测概率进行抽样。窗口和105个正面窗口的预测概率。接下来，分别对我们计算这些样本的FL。并将损失归一，使其总和为1。鉴于归一化的损失，我们可以将损失从低到高排序并绘制其累积分布函数（CDF）。正负样本和不同设置的γ（尽管模型是以γ=2训练的）。正、负样本的累积分布函数如图4所示。如果我们观察阳性样本，我们会发现，在不同的γ值下，CDF看起来相当相似。不同的γ值相当相似。例如，大约20%的最难的阳性样本大约占了一半的正面损失，随着γ的增加，更多的损失集中在前20%的例子中，但这种影响是很小

的。γ对负面样本的影响则截然不同。对于γ=0，正面和负面的CDFs非常相似。相似。然而，随着γ的增加，大量的权重变得集中在硬性的负面例子上。事实上事实上，在γ = 2的情况下（我们的默认设置），绝大部分的损失来自于一小部分的样本。正如我们所看到的
可以看出，FL可以有效地消除简单的负面因素的影响，将所有的注意力集中在困难的负面例子上。

Online Hard Example Mining (OHEM):

[31] proposed to improve training of two-stage detectors by constructing minibatches using high-loss examples. Specifically, in
OHEM each example is scored by its loss, non-maximum suppression (nms) is then applied, and a minibatch is constructed with the highest-loss examples. The nms threshold and batch size are tunable parameters. Like the focal loss, OHEM puts more emphasis on misclassified examples, but unlike FL, OHEM completely discards easy examples. We also implement a variant of OHEM used in SSD [22]: after applying nms to all examples, the minibatch is constructed to enforce a 1:3 ratio between positives and negatives to help ensure each minibatch has enough positives. We test both OHEM variants in our setting of one-stage detection which has large class imbalance. Results for the original OHEM strategy and the 'OHEM 1:3' strategy for selected batch sizes and nms thresholds are shown in Table 1d. These results use ResNet-101, our baseline trained with FL achieves 36.0 AP for this setting. In contrast, the best setting for OHEM (no 1:3 ratio, batch size 128, nms of .5) achieves 32.8 AP. This is a gap of 3.2 AP, showing FL is more effective than OHEM for training dense detectors. We note that we tried other parameter setting and variants
for OHEM but did not achieve better results.

在线硬例挖掘（OHEM）。

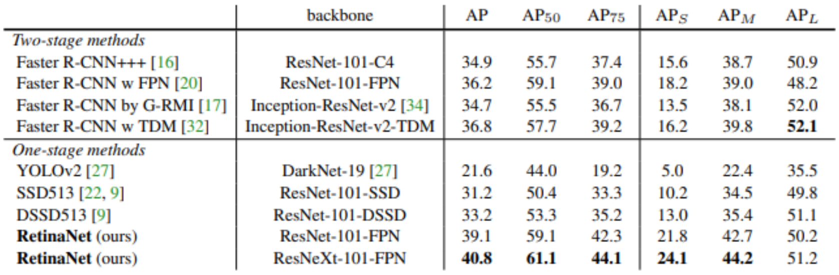
[31]提出通过使用高损失的例子构建小批，来改善两阶段检测器的训练。具体来说，在OHEM中，每个例子都根据其损失进行评分，然后应用非最大限度的抑制（nms），然后用最高损失的例子构建一个小批。nms阈值和批量大小是可调整的参数。像焦点损失一样。OHEM更强调被错误分类的例子，但是与FL不同的是，OHEM完全抛弃了简单的例子。我们我们还实现了SSD[22]中使用的OHEM的一个变体：在对所有的例子应用nms之后对所有的例子应用nms后，构建minibatch强制执行阳性和阴性之间的1:3比例，以帮助确保每个minibatch有足够的阳性。我们在第一阶段的检测中测试了OHEM的两个变体。检测，该检测具有较大的类不平衡性。原有的OHEM策略和 "OHEM "策略的结果是原始OHEM策略和 "OHEM 1:3 "策略的结果。表1d显示了原始OHEM策略和 "OHEM 1:3 "策略在选定的批次大小和nms阈值上的结果。这些结果使用ResNet-101，我们的基线是用FL训练的。在这种情况下，用FL训练出来的AP达到36.0。相比之下，OHEM的OHEM的最佳设置（没有1:3的比例，批次大小为128，nms为.5）实现了32.8个AP。这是一个3.2AP的差距，表明FL在训练密集型检测器方面比OHEM更有效。我们注意到，我们尝试了其他参数设置和变体但没有取得更好的结果。

Hinge Loss:

Finally, in early experiments, we attempted to train with the hinge loss [13] on pt , which sets loss to 0 above a certain value of pt
. However, this was unstable and we did not manage to obtain meaningful results. Results exploring alternate loss functions are in the appendix.

铰链损失。

最后，在早期的实验中，我们试图用铰链损失[13]来训练pt训练，它将损失设置为0在pt的某一数值以上. 然而，这是不稳定的，而且我们并没有设法获得有意义的结果。结果探索替代损失函数的结果在附录中。



	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. Model Architecture Design

Anchor Density:

One of the most important design factors in a one-stage detection system is how densely it covers the space of possible image boxes. Two-stage detectors can classify boxes at any position, scale, and aspect ratio using a region pooling operation [10]. In contrast, as one-stage detectors use a fixed sampling grid, a popular approach for achieving high coverage of boxes in these approaches is to use multiple 'anchors' [28] at each spatial position to cover boxes of various scales and aspect ratios. We sweep over the number of scale and aspect ratio anchors used at each spatial position and each pyramid level in FPN. We consider cases from a single square anchor at each location to 12 anchors per location spanning 4 sub-octave scales (2 k/4 , for k ≤ 3) and 3 aspect ratios [0.5, 1, 2]. Results using ResNet-50 are shown in Table 1c. A surprisingly good AP (30.3) is achieved using just one square anchor. However, the AP can be improved by nearly 4 points (to 34.0) when using 3 scales and 3 aspect ratios per location. We used this setting for all other experiments in this work. Finally, we note that increasing beyond 6-9 anchors did not shown further gains. Thus while two-stage systems can classify arbitrary boxes in an image, the saturation of performance w.r.t. density implies the higher potential density of two-stage systems may not offer an advantage.

锚点密度。

一级检测系统中最重要的设计因素之一是它对可能的图像盒空间的覆盖密度如何。可能的图像盒空间的密度。两阶段检测器可以在任何位置、比例和长宽比上对盒子进行分类。区域池化操作[10]。相比之下，由于单阶段检测器使用一个固定的采样网格，在这些方法中实现盒子的高覆盖率的一个流行方法是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上使用多个 "锚"[28]来覆盖不同尺度和长宽比的盒子。我们在每个空间位置和每个金字塔级别上使用的尺度和长宽比锚的数量上进行了扫瞄。FPN。我们考虑的情况是，从每个位置上的单个方形锚点到每个位置上的12个锚点。到每个位置有12个锚，横跨4个亚八度尺度(2k/4，对于k≤3）和3个纵横比[0.5，1，2]。使用ResNet-50的结果显示在表1c。一个令人惊讶的好的AP（30.3）是用一个方形锚实现的。然而，当使用3种比例时，AP可以提高近4点（达到34.0），当每个位置使用3个尺度和3个长宽比时。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个锚点以上并没有没有显示出进一步的收益。因此，虽然两阶段系统可以对图像中的任意方块进行分类，但在密度方面的性能饱和意味着两级系统的更高的潜在密度的两阶段系统可能不会提供优势。

Speed versus Accuracy:

Larger backbone networks yield higher accuracy, but also slower inference speeds. Likewise for input image scale (defined by the shorter image side). We show the impact of these two factors in Table 1e. In Figure 2 we plot the speed/accuracy trade-off curve for RetinaNet and compare it to recent methods using public numbers on COCO test-dev. The plot reveals that RetinaNet, enabled by our focal loss, forms an upper envelope over all existing methods, discounting the low-accuracy regime. RetinaNet with ResNet-101-FPN and a 600 pixel image scale (which we denote by RetinaNet-101-600 for simplicity) matches the accuracy of the recently published ResNet101-FPN Faster R-CNN [20], while running in 122 ms per image compared to 172 ms (both measured on an Nvidia M40 GPU). Using larger scales allows RetinaNet to surpass the accuracy of all two-stage approaches, while still being faster. For faster runtimes, there is only one operating point (500 pixel input) at which using ResNet-50-FPN improves over ResNet-101-FPN. Addressing the high frame rate regime will likely require special network design, as in [27], and is beyond the scope of this work. We note that after publication, faster and more accurate results can now be obtained by a variant of Faster R-CNN from [12].

速度与准确度。

较大的骨干网络产生更高的准确性，但推理速度也更慢。同样地对于输入图像的比例（由较短的图像边定义）。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/准确率权衡曲线，并与最近使用COCO test-dev上的公开数据的方法进行了比较。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。我们的焦点损失使RetinaNet形成了一个超过所有现有方法的上限，不包括低精度的方法。RetinaNet与ResNet-101-FPN和600像素的图像规模的RetinaNet（为简单起见，我们用RetinaNet-101-600表示）与最近发表的ResNet101-FPN Faster R-CNN[20]的准确度相当，而每幅图像的运行时间为122毫秒，而我们的RetinaNet-101-FPN Faster R而每幅图像的运行时间为122毫秒，相比之下为172毫秒（均在NvidiaM40 GPU上测量）。使用更大的尺度使RetinaNet超过了所有两阶段方法的准确性，同时仍然更快。为了加快运行时间，只有一个操作点（500像素的输入），使用ResNet-50-FPN比ResNet-101-FPN更快。要解决高帧率的问题解决高帧率制度可能需要特殊的网络设计，如[27]，而且超出了本工作的范围。我们注意到我们注意到，在文章发表后，更快、更准确的结果现在可以通过一个变种的FNet-101-FPN获得。可以通过[12]中的Faster R-CNN的一个变种获得更快、更准确的结果。

5.3. Comparison to State of the Art

We evaluate RetinaNet on the challenging COCO dataset and compare test-dev results to recent state-of-the-art methods including both one-stage and two-stage models. Results are presented in Table 2 for our RetinaNet-101-800 model trained using scale jitter and for 1.5× longer than the models in Table 1e (giving a 1.3 AP gain). Compared to existing one-stage methods, our approach achieves a healthy 5.9 point AP gap (39.1 vs. 33.2) with the closest competitor, DSSD [9], while also being faster, see Figure 2. Compared to recent two-stage methods, RetinaNet achieves a 2.3 point gap above the top-performing Faster R-CNN model based on Inception-ResNet-v2-TDM [32]. Plugging in ResNeXt32x8d-101-FPN [38] as the RetinaNet backbone further improves results another 1.7 AP, surpassing 40 AP on COCO.

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包括单阶段和双阶段模型。表2显示了我们的RetinaNet-101-800模型的结果。使用比例抖动训练的模型，比表1e中的模型长1.5倍。表1e中的模型（给予1.3个AP增益）。与现有的单阶段方法相比，我们的方法实现了一个健康的与最接近的竞争者相比，我们的方法实现了5.9个点的AP差距（39.1对33.2）。DSSD [9]，同时速度也更快，见图2。与最近的两阶段方法相比与最近的两阶段方法相比，RetinaNet实现了2.3分的差距。与最近的两阶段方法相比，RetinaNet比表现最好的基于Inception-ResNet-v的Faster R-CNN模型高出2.3分。基于Inception-ResNet-v2-TDM[32]。插入ResNeXt32x8d-101-FPN[38]作为RetinaNet的主干，又进一步提高了1.7个AP的结果，超过了COCO的40个AP。

6. Conclusion

In this work, we identify class imbalance as the primary obstacle preventing one-stage object detectors from surpassing top-performing, two-stage methods. To address this, we propose the focal loss which applies a modulating term to the cross entropy loss in order to focus learning on hard negative examples. Our approach is simple and highly effective. We demonstrate its efficacy by designing a fully convolutional one-stage detector and report extensive experimental analysis showing that it achieves stateof-the-art accuracy and speed. Source code is available at https://github.com/facebookresearch/Detectron [12].

在这项工作中，我们发现类的不平衡是阻碍一阶段物体检测器超越表现最好的两阶段方法的主要障碍。为了解决这个问题，我们提出了焦点损失，它将一个调节项应用于交叉熵损失，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，并报告了大量的实验分析，显示它达到了最先进的精度和速度。源代码可在https://github.com/facebookresearch/Detectron [12]。

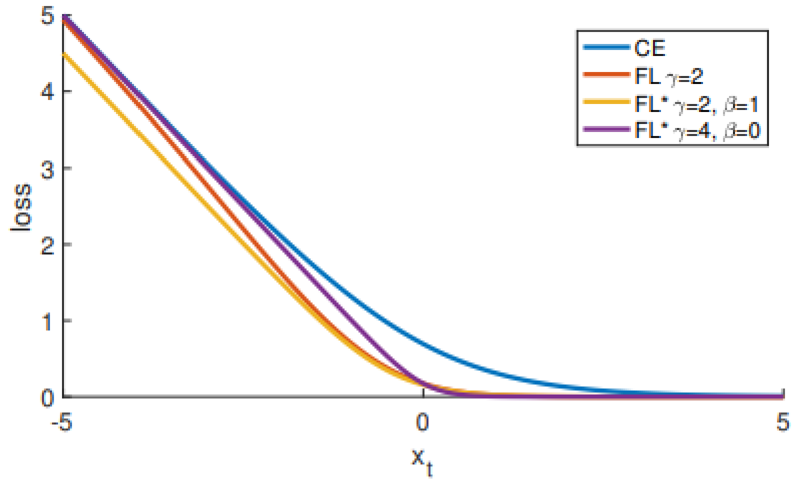
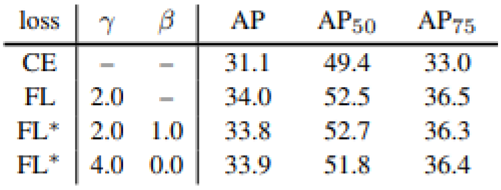


Figure 5. Focal loss variants compared to the cross entropy as a function of $x_t = yx$. Both the original FL and alternate variant FL* reduce the relative loss for well-classified examples ($x_t > 0$).



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.

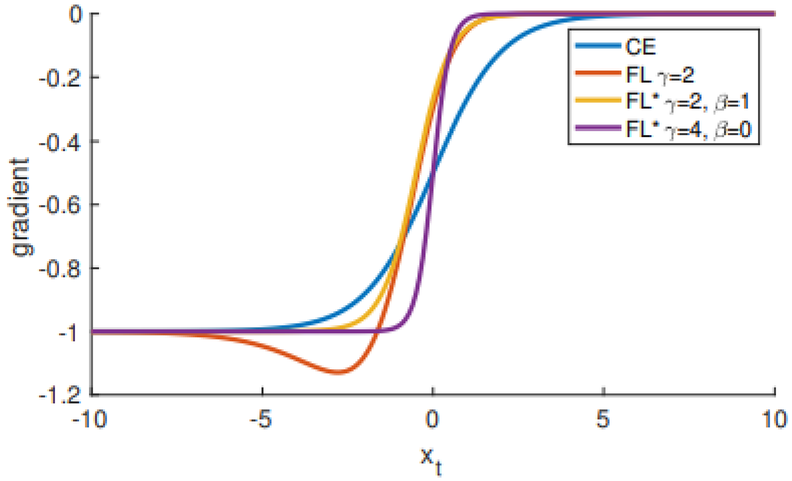


Figure 6. Derivates of the loss functions from Figure 5 w.r.t. x.

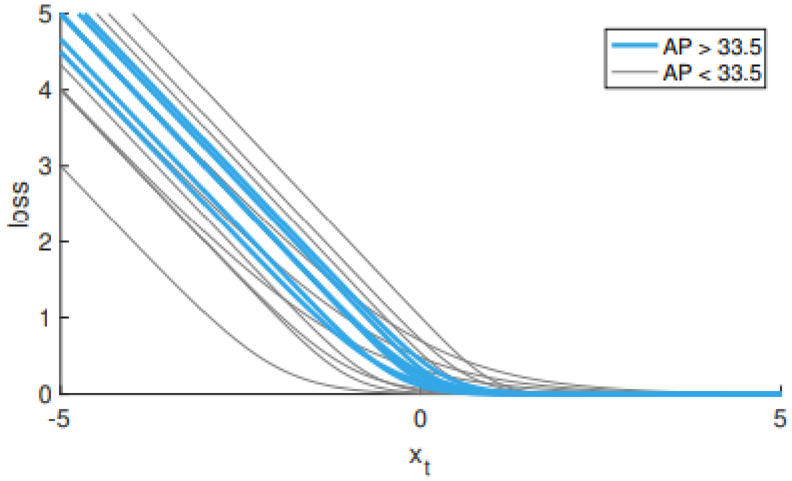


Figure 7. Effectiveness of FL* with various settings γ and β. The plots are color coded such that effective settings are shown in blue.