# FCOS: Fully Convolutional One-Stage Object Detection

Zhi Tian, Chunhua Shen, Hao Chen, Tong He

**摘要**

我们提出了一种全卷积的单阶段目标检测器 (FCOS)，以类似于语义分割的每像素预测方式解决目标检测问题。几乎所有最先进的物体检测器，如 RetinaNet、SSD、YOLOv3 和 Faster R-CNN 都依赖于预定义的锚框。相比之下，我们提出的检测器 FCOS 没有锚框，也没有候选。通过消除预定义的锚框集合，FCOS 完全避免了与锚框相关的复杂计算，例如在训练过程中计算重叠。更重要的是，我们还避免了与锚框相关的所有超参数，最终检测性能通常对这些超参数非常敏感。凭借唯一的后处理——非极大值抑制（NMS），FCOS with ResNeXt-64x4d-101 在单模型单尺度测试的 AP 中达到了 44.7%，超越了之前的单阶段检测器，同时有着更简单的优势。我们第一次展示了一个更简单和灵活的检测框架，可以提高检测精度。我们希望提出的 FCOS 框架可以作为许多其他实例级任务的简单而强大的替代方案。代码位于：tinyurl.com/FCOSv1。

## 1 Introduction

目标检测是计算机视觉中一项基本但具有挑战性的任务，它需要算法为图像中的每个感兴趣的实例预测带有类别标签的边界框。目前所有主流检测器如 Faster R-CNN[**fasterrcnn**]、SSD[**ssd**] 和 YOLOv2、v3[**yolov3**] 都依赖于一组预定义的锚框，长期以来一直认为使用锚框是检测器成功的关键。尽管取得了巨大的成功，但重要的是要注意基于锚的检测器有一些缺点：1) 如 [15, 24] 所示，检测性能对锚框的大小、长宽比和数量很敏感。例如，在 RetinaNet[**retinanet**] 中，改变这些超参数最多可以在 COCO 基准测试上影响 4% 的 AP [16]。因此，需要在基于锚的检测器中仔细调整这些超参数。2）即使经过精心设计，由于锚框的尺度和纵横比保持固定，检测器在处理具有较大形状变化的候选对象时遇到困难，尤其是对于小对象。预定

义的锚框也阻碍了检测器的泛化能力，因为它们需要在具有不同对象大小或长宽比的新检测任务上重新设计。3）为了实现高召回率，一个基于锚的检测器需要在输入图像上密集放置锚框（例如，对于短边为 800 的图像，特征金字塔网络（FPN）[**fpn**] 中存在超过 180K 个锚框）。大多数这些锚框在训练期间被标记为负样本。过多的负样本加剧了训练中正负样本的不平衡。4) 锚框还涉及复杂的计算，例如使用 ground-truth 边界框计算 IoU 分数。

最近，全卷积网络 (FCN) [**fcn**] 在语义分割 [20、28、9、19]、深度估计 [17、31]、关键点检测 [3] 和计数 [2] 等密集预测任务中取得了巨大成功。作为高级视觉任务之一，目标检测可能是唯一一个偏离整洁的全卷积每像素预测框架的任务，主要是由于使用了锚框。提出一个问题是很自然的：*我们能否以简洁的每像素预测方式解决对象检测，例如，类似于语义分割的 FCN*？因此，这些基本的视觉任务可以（几乎）统一在一个框架中。我们证明答案是肯定的。此外，我们首次证明，更简单的基于 FCN 的检测器比基于锚的检测器实现了更好的性能。

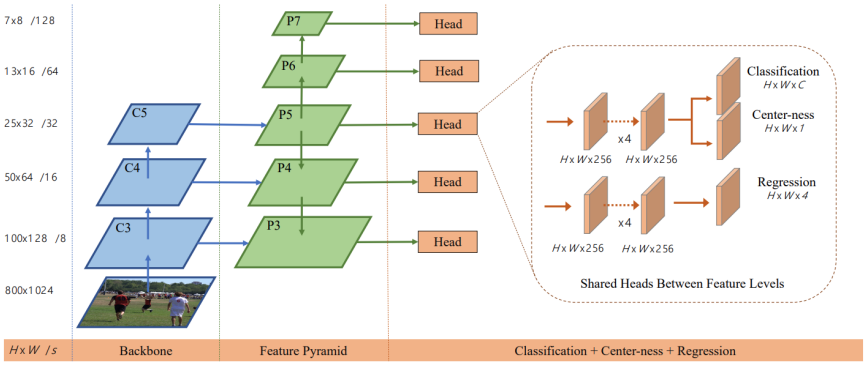


图 1: **模型**。我们的系统将检测建模为一个回归问题。它将图片划分为 $S \times S$ 个网格，并为每个网格预测 $B$ 个边界框、这些边界框的置信度和 $C$ 个类别概率。这些预测被编码为一个 $S \times S \times (B * 5 + c)$ 的张量。

在文献中，一些工作尝试利用基于 FCN 的框架进行对象检测，例如 DenseBox[**densebox**]。具体来说，这些基于 FCN 的框架直接在每个特征图层级上的每个空间位置预测 4D 向量和类别。如图1（左）所示，4D 向量描绘了从边界框的四个边到该位置的相对偏移量。这些框架类似于用于语义分割的 FCN，不同之处在于每个位置都需要回归 4D 连续向量。然而，为

了处理不同大小的边界框，DenseBox[**densebox**] 将训练图像裁剪和调整到固定尺寸。因此 DenseBox 必须对图像金字塔进行检测，这违背了 FCN 一次计算所有卷积的理念。此外，更重要的是，这些方法主要用于特殊领域的物体检测，例如场景文本检测 [33, 10] 或人脸检测 [32, 12]，因为人们认为这些方法在应用于可能高度重叠的通用物体检测时效果不佳。如图 1（右）所示，高度重叠的边界框导致难以处理的歧义：不清楚 w.r.t. 重叠区域中的像素应该回归哪个边界框。

在续集中，我们仔细研究了这个问题，并表明使用 FPN 可以在很大程度上消除这种歧义。因此，我们的方法已经可以获得与那些传统的基于锚的检测器相当的检测精度。此外，我们观察到我们的方法可能会在远离目标对象中心的位置产生许多低质量的预测边界框。为了抑制这些低质量的检测，我们引入了一个新的“中心度”分支（只有一层）来预测像素与其相应边界框中心的偏差，如方程**??**中所定义。然后使用该分数降低检测到的低质量边界框的权重，并在 NMS 中合并检测结果。简单而有效的中心分支使得基于 FCN 的检测器在完全相同的训练和测试设置下优于基于锚的检测器。

这个新检测框架有如下优势。

- 检测现在与许多其他可以使用 FCN 解决的任务（例如语义分割）统一起来，从而更容易重用这些任务中的想法。
- 检测不再需要候选和锚框，这显着减少了设计参数的数量。设计参数通常需要启发式调整，并涉及许多技巧以实现良好的性能。因此，我们的新检测框架使检测器，尤其是其训练变得更加简单。
- 通过消除锚框，我们的新检测器完全避免了与锚框相关的复杂计算，例如训练期间锚框与真实框之间的 IOU 计算和匹配。从而相对于基于锚的对应检测器，实现更快的训练和测试以及更少的训练内存占用。
- 没有花里胡哨，我们在单阶段检测器中实现了最好的结果。我们还表明，所提出的 FCOS 可以用作两阶段检测器中的区域候选网络 (RPN)，并且可以实现比基于锚的 RPN 对应物明显更好的性能。鉴于更简单的无锚检测器的性能甚至更好，我们鼓励社区重新考虑锚框在目标检测中的必要性，它目前被认为是事实上的检测标准。
- 所提出的检测器可以立即以最少的修改扩展到其他视觉任务，包括实例分割和关键点检测。我们相信这种新方法可以成为许多实例预测问题的新基线。

# 2 Our Approach

我们将物体检测的独立组件统一到一个神经网络中。我们的网络使用来自整个图像的特征来预测每个边界框。它还同时预测图像的所有类别的所有边界框。这意味着我们的网络对完整图像和图像中的所有物体进行全局推理。YOLO 设计支持端到端训练和实时速度，同时保持高平均精度。

我们的系统将输入图像划分为 $S \times S$ 网格。如果物体的中心落入网格单元中，则该网格单元负责检测该物体。

每个网格单元预测 $B$ 个边界框和这些框的置信度。这些置信度反映了模型对框包含物体的信心程度，以及它认为盒子预测的准确度。形式上，我们将置信度定义为 $\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$。如果该单元格中不存在物体，则置信度分数应为零。否则，我们希望置信度得分等于预测框和 ground truth 之间的交集（IOU）。

每个边界框由 5 个预测组成：$x, y, w, h$ 和置信度。$(x, y)$ 坐标表示相对于网格单元边界的框中心。宽度和高度是相对于整个图像预测的。最后，置信度预测表示预测框和任何 ground truth 框之间的 IOU。

每个网格单元还预测 $C$ 个条件类概率，$\Pr(\text{Class}_i|\text{Object})$。这些概率以网格单元包含物体的为条件。我们只为每个网格单元预测一组类概率，而不管框 $B$ 的数量。

在测试时，我们将条件类概率和单个框置信度预测相乘，

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}} \quad (1)$$

这为我们提供了每个框特定类的置信度分数。这些分数编码了该类出现在框中的概率以及预测的框与物体的匹配程度。

为了在 PASCAL VOC 上评估 YOLO，我们使用 $S = 7$，$B = 2$。PASCAL VOC 有 20 个标记类别，因此 $C = 20$。我们的最终预测是一个 $7 \times 7 \times 30$ 张量。

## 2.1 网络设计

我们将此模型实现为卷积神经网络，并在 PASCAL VOC 检测数据集 [4] 上对其进行评估。网络的初始卷积层从图像中提取特征，而全连接层预测输出概率和坐标。

我们的网络架构受到用于图像分类的 GoogLeNet 模型的启发 [6]。我们的网络有 24 个卷积层，后跟 2 个全连接层。我们不使用 GoogLeNet 使用

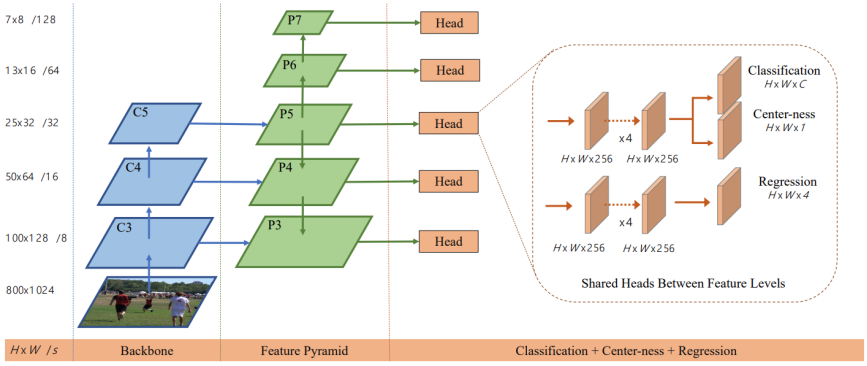


图 2: **模型**。我们的系统将检测建模为一个回归问题。它将图片划分为 $S \times S$ 个网格，并为每个网格预测 $B$ 个边界框、这些边界框的置信度和 $C$ 个类别概率。这些预测被编码为一个 $S \times S \times (B * 5 + c)$ 的张量。

的 inception 模块，而是简单地使用 $1 \times 1$ 缩减层和 $3 \times 3$ 卷积层，类似于 Lin 等人 [2]。完整的网络如图3所示。

我们还训练了一个快速版本的 YOLO，旨在突破快速目标检测的边界。Fast YOLO 使用较少卷积层（9 个而不是 24 个）并在这些层中使用较少的卷积核。除了网络的大小之外，YOLO 和 Fast YOLO 的所有训练和测试参数都是相同的。

我们网络的最终输出是 $7 \times 7 \times 30$ 的预测张量。

## 2.2 训练

我们在 ImageNet 1000 类竞赛数据集 [30] 上预训练我们的卷积层。对于预训练，我们使用图3中的前 20 个卷积层，然后是平均池化层和全连接层。我们对该网络进行了大约一周的训练，并在 ImageNet 2012 验证集上实现了 88% 的单次裁剪 top-5 准确率，与 Caffe 的 Model Zoo [24] 中的 GoogLeNet 模型相当。我们使用 darknet 框架进行所有训练和推理 [26]。

然后我们转换模型以执行检测。Ren 等人表明将卷积层和连接层添加到预训练网络可以提高性能 [29]。按照他们的例子，我们添加了具有随机初始化权重的四个卷积层和两个全连接层。检测通常需要细粒度的视觉信息，因此我们将网络的输入分辨率从 224×224 增加到 448×448。

我们的最后一层预测类别概率和边界框坐标。我们通过图像的宽度和

高度对边界框的宽度和高度进行归一化，使它们落在 0 和 1 之间。我们将边界框的 $x$ 和 $y$ 坐标参数化为特定网格单元位置的偏移量，因此它们也被限制在 0 和 1 之间.

我们对最后一层使用线性激活函数，所有其他层使用以下 leaky 校正线性激活：

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases} \tag{2}$$

我们针对模型输出中的平方和误差进行了优化。我们使用平方和误差是因为它很容易优化，但是它并不完全符合我们最大化平均精度的目标。它将定位误差与分类误差同等加权，这可能并不理想。此外，在每个图像中，许多网格单元不包含任何物体。这会将这些单元格的“置信度”分数推向零，通常会压倒包含物体的单元格的梯度。这可能会导致模型不稳定，从而导致训练早期出现发散。

为了解决这个问题，我们增加了边界框坐标预测的损失，并减少了不包含物体的框的置信度预测的损失。我们使用两个参数 $\lambda_{\text{coord}}$ 和 $\lambda_{\text{noobj}}$ 来实现这一点。我们设置 $\lambda_{\text{coord}} = 5$ 和 $\lambda_{\text{noobj}} = 0.5$。

平方和误差也同样加权大框和小框的错误。我们的误差度量应该反映大盒子中的小偏差比小盒子中的小偏差重要性小。为了部分解决这个问题，我们预测边界框宽度和高度的平方根，而不是直接预测宽度和高度。

YOLO 为每个网格单元预测多个边界框。在训练时，对于每个物体，我们只希望一个边界框预测器对它负责。我们根据哪个预测与 ground truth 具有最高的 IOU 来指定哪一个预测器为此 ground truth “负责”。这导致边界框预测器之间的专业化。每个预测器在预测特定大小、长宽比或物体类别方面都变得更好，从而提高了整体召回率。

在训练期间，我们优化了以下多部分损失函数：

$$
\begin{aligned}
& \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[ \left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 \right] \\
+ \quad & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[ \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right] \\
+ \quad & \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 \\
+ \quad & \lambda_{\text{noobj}} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2 \\
+ \quad & \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \left(p_i\left(c\right) - \hat{p}_i\left(c\right)\right)^2
\end{aligned}
\tag{3}
$$

其中 $\mathbb{1}_i^{\text{obj}}$ 表示物体是否在单元 $i$ 出现，$\mathbb{1}_{ij}^{\text{obj}}$ 表示单元 $i$ 的第 $j$ 个边界框预测器对这个检测“负责”。

请注意，仅当该网格单元中存在物体时（因此是前面讨论的条件类概率），损失函数才会惩罚分类错误。仅当该预测器对 ground truth 框“负责”（即具有该网格单元中任何预测器的最高 IOU）时，它才会惩罚边界框坐标错误。

我们在来自 PASCAL VOC 2007 和 2012 的训练和验证数据集上训练网络约 135 个 epoch。在 2012 测试时，我们在训练时还使用了 VOC 2007 的测试数据。在整个训练过程中，我们使用 64 的批量大小、0.9 的动量和 0.0005 的衰减。

我们的学习率计划如下：对于第一个 epoch，我们慢慢地将学习率从 $10^{-3}$ 提高到 $10^{-2}$。如果我们以高学习率开始，我们的模型通常会因梯度不稳定而发散。我们继续用 $10^{-2}$ 训练 75 个 epoch，然后用 $10^{-3}$ 训练 30 个 epoch，最后用 $10^{-4}$ 训练 30 个 epoch。

为了避免过拟合，我们使用 dropout 和广泛的数据增强。在第一个连接层之后具有 rate = .5 的 dropout 层来防止层之间的协同适应 [18]。对于数据增强，我们引入了最多原始图像大小 20% 的随机缩放和平移。我们还在 HSV 色彩空间中随机调整图像的曝光和饱和度，最高达 1.5。

### 2.3 推理

就像在训练中一样，预测测试图像的检测只需要一次网络评估。在 PASCAL VOC 上，网络预测每个图像的 98 个边界框和每个框的类别概率。YOLO 在测试时非常快，因为它只需要一次网络评估，这与基于分类器的方法不同。

网格设计在边界框预测中强制执行空间多样性。通常很清楚一个物体属于哪个网格单元，并且网络只为每个物体预测一个框。但是，一些大型物体或靠近多个单元格边界的物体可以被多个单元格很好地定位。非最大抑制可用于修复这些重复检测。虽然对 R-CNN 或 DPM 的性能并不重要，但非最大抑制在 mAP 中增加了 2-3%。

### 2.4 YOLO 的局限

YOLO 对边界框预测施加了很强的空间约束，因为每个网格单元只预测两个框并且只能有一个类。这种空间约束限制了我们的模型可以预测的附近物体的数量。我们的模型在处理成群出现的小物体时遇到了困难，例如成群的鸟。

由于我们的模型学习从数据中预测边界框，因此它很难泛化到具有新的或不寻常的长宽比或配置的物体。我们的模型还使用相对粗略的特征来预测边界框，因为我们的架构包含对输入图像的多个下采样层。

最后，当我们训练近似检测性能的损失函数时，我们的损失函数将小边界框与大边界框的错误处理相同。大框的小错误通常是良性的，但小框的小错误对 IOU 的影响要大得多。我们的主要错误来源是不正确的定位。

## References

[1] Pedro F Felzenszwalb et al. “Object detection with discriminatively trained part-based models”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2009), pp. 1627–1645.

[2] Min Lin, Qiang Chen, and Shuicheng Yan. “Network in network”. In: *arXiv preprint arXiv:1312.4400* (2013).

[3] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[4] Mark Everingham et al. “The pascal visual object classes challenge: A retrospective”. In: *International journal of computer vision* 111.1 (2015), pp. 98–136.

[5] Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[6] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

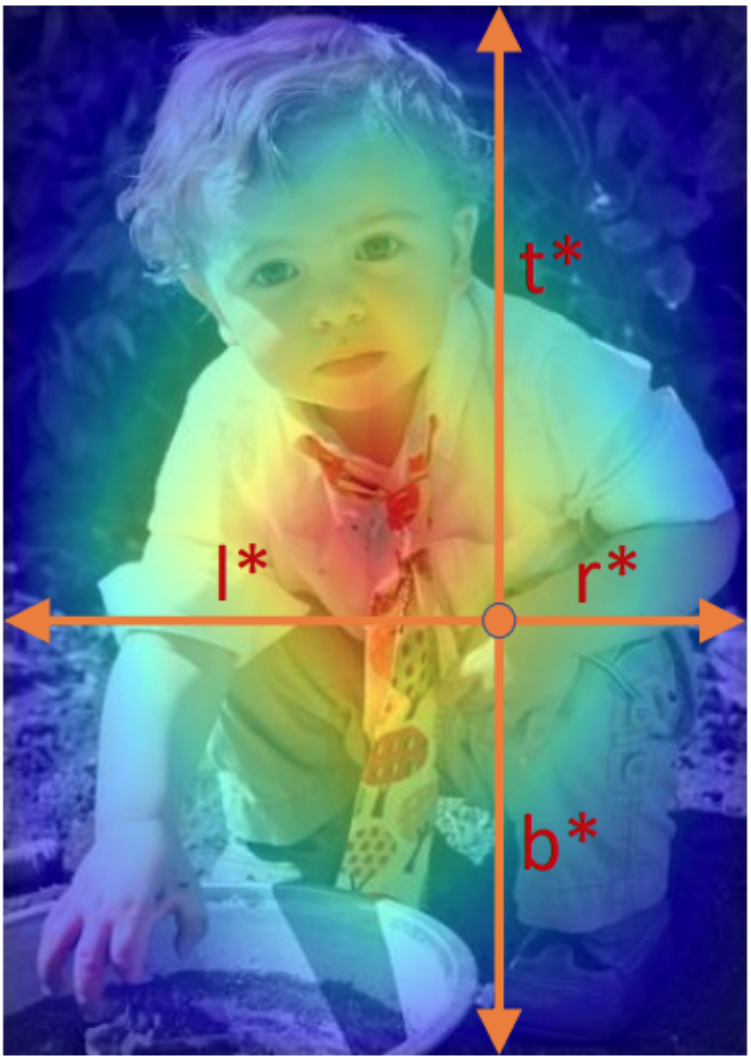


图 3: **网络结构**。我们的检测网络有 24 个卷积层，后跟 2 个全连接层。交替的 $1 \times 1$ 卷积层减少了前一层的特征空间。我们在 ImageNet 分类任务上以一半的分辨率（$224 \times 224$ 输入图像）预训练卷积层，然后将分辨率提高一倍以进行检测。