

你只看一次： 统一、实时的目标检测

Joseph Redmon*, Santosh Divvala*†, Ross Girshick†, Ali Farhadi*†

University of Washington*, Allen Institute for AI†, Facebook AI Research†

<http://pjreddie.com/yolo/>

Abstract

本文提出`YOLO`，一种新的目标检测方法。之前关于目标检测的工作重新利用分类器来进行检测。相反，我们将目标检测框架为分离的边界框和相关的类概率的回归问题。在一次评估中，单个神经网络直接从完整图像中预测边界框和类概率。由于整个检测流水线是一个单一的网络，因此可以直接对检测性能进行端到端的优化。

我们的统一架构非常快。我们的基本`YOLO`模型以每秒45帧的速度实时处理图像。该网络的一个较小版本`Fast YOLO`，处理速度达到惊人的155帧/秒，同时仍然实现了其他实时检测器的mAP的两倍。与最先进的检测系统相比，`YOLO`有更多的定位误差，但不太可能预测背景的误报。最后，`YOLO`可以学习物体的一般表示。当从自然图像泛化到艺术作品等其他领域时，它优于其他检测方法，包括`DPM`和`R-CNN`。

1. 简介

人类瞥一眼图像，就立刻知道图像中有什么物体，它们在哪里，以及它们如何相互作用。人类的视觉系统是快速和准确的，使我们能够执行复杂的任务，如驾驶时很少有意识的思考。快速、准确的目标检测算法将允许计算机在没有专门传感器的情况下驾驶汽车，使辅助设备能够向人类用户传递实时场景信息，并释放用于通用、响应式机器人系统的潜力。

当前的检测系统利用分类器进行检测。为了检测目标，这些系统采用该目标的分类器，并在测试图像的不同位置和尺度上对其进行评估。像可变形部件模型(DPM)这样的系统使用滑动窗口方法，其中分类器在整个图像的均匀间隔位置运行[10]。

最近的方法，如`R-CNN`，使用区域建议方法首先在图像中生成潜在的边界框，然后在这些建议框上运行分类器。分类后，使用后处理来细化边界框，消除重复检测，并基于场景中的其他物体对框进行重新评分[13]。这些复杂的管道很慢，很难优化，因为每个单独的组件必须单独训练。

将目标检测重构为一个单一的回归问题，直接从图像像素到边界框坐标和类概率。使用我们的系统，你只需要对图像看一次(`YOLO`)就可以预测存在什么物体及其位置。

`YOLO`非常简单：参见图1。一个卷积网络同时预测

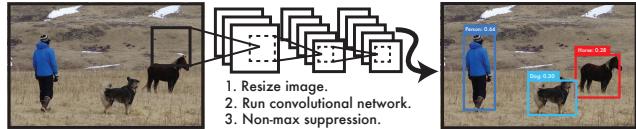


Figure 1: `YOLO`检测系统。使用`YOLO`处理图像简单而直接。我们的系统(1)将输入图像调整为 448×448 ，(2)在图像上运行单个卷积网络，(3)通过模型的置信度对结果检测进行阈值化。

多个边界框和这些框的类别概率。`YOLO`在全图像上进行训练，并直接优化检测性能。这种统一的模型比传统的目标检测方法有几个优点。

首先，`YOLO`非常快。由于我们将帧检测作为回归问题，因此不需要复杂的管道。我们只需在测试时在新图像上运行神经网络以预测检测结果。我们的基础网络运行在Titan X GPU上，没有批处理，每秒45帧，快速版本运行在超过150 fps。这意味着我们可以实时处理视频流，延迟小于25毫秒。此外，`YOLO`达到了其他实时系统平均精度的两倍以上。有关我们的系统在网络摄像头上实时运行的演示，请参阅我们的项目网页：<http://pjreddie.com/yolo/>。

其次，`YOLO`在进行预测时对图像进行全局推理。与滑动窗口和基于区域建议的技术不同，`YOLO`在训练和测试期间查看整个图像，因此它隐式编码了关于类及其外观的上下文信息。`Fast R-CNN`是一种顶级检测方法[14]，它将图像中的背景块错误地识别为物体，因为它无法看到更大的上下文。与`Fast R-CNN`相比，`YOLO`产生的背景误差数量不到`Fast R-CNN`的一半。

第三，`YOLO`学习对象的可泛化表示。当在自然图像上进行训练并在艺术品上进行测试时，`YOLO`的性能大大优于`DPM`和`R-CNN`等顶级检测方法。由于`YOLO`具有高度泛化性，因此在应用于新领域或意外输入时不太可能崩溃。

`YOLO`在准确性方面仍然落后于最先进的检测系统。虽然它可以快速识别图像中的物体，但它很难精确定位一些物体，特别是小物体。在实验中进一步研究了这些权衡。

我们所有的训练和测试代码都是开源的。还可以下载各种预训练模型。

2. 统一检测

将目标检测的独立组件统一为一个单一的神经网络。我们的网络使用整个图像的特征来预测每个边界框。它还同时预测图像所有类别的所有边界框。这意味着我们的网络对整个图像和图像中的所有对象进行全局推理。YOLO设计实现了端到端训练和实时速度，同时保持较高的平均精度。

我们的系统将输入图像划分为 $S \times S$ 网格。如果一个物体的中心落在一个网格单元中，该网格单元负责检测该物体。

每个网格单元预测 B 边界框和这些框的置信度分数。这些置信度分数反映了模型对框中包含对象的置信度，以及它认为它所预测的框的准确性。在形式上，我们将信心定义为 $\text{Pr}(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$ 。如果单元格中不存在对象，则置信度得分应为0。否则，我们希望置信度分数等于预测框与真实值之间的交并比(IOU)。

每个边界框由5个预测结果组成: x, y, w, h 和置信度。 (x, y) 坐标表示相对于网格单元格边界的方框中心。宽度和高度相对于整个图像进行预测。最后，置信度预测表示预测框和任何真实框之间的IOU。

每个网格单元格还预测 C 条件类别概率 $\text{Pr}(\text{Class}_i | \text{Object})$ 。这些概率取决于包含对象的网格单元。我们只预测每个网格单元格的一组类别概率，而不管盒子的数量 B 。

在测试时，我们将条件类别概率与单个框置信度预测相乘，

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

它为每个框提供了特定于类别的置信度分数。这些分数编码了该类别出现在框中的概率以及预测框与对象的拟合程度。

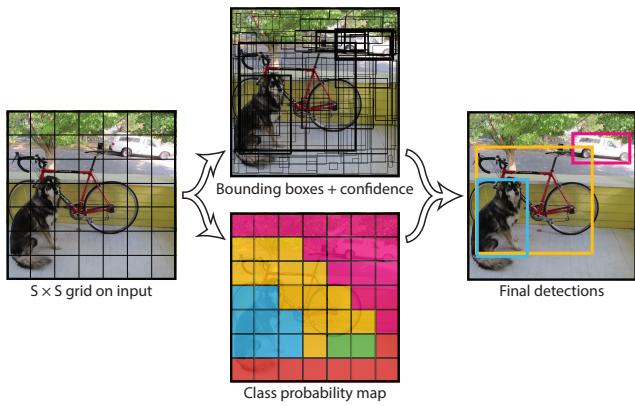


Figure 2: 模型。我们的系统将检测建模为回归问题。它将图像划分为 $S \times S$ 网格，并对每个网格单元预测 B 边界框，这些框的置信度和 C 类概率。这些预测被编码为 $S \times S \times (B * 5 + C)$ 张量。

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，所以 $C = 20$ 。我们的最终预测是一个 $7 \times 7 \times 30$ 张量。

2.1. 网络设计

我们将该模型实现为卷积神经网络，并在PASCAL VOC检测数据集[9]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。

我们的网络架构的灵感来自于用于图像分类的GoogLeNet模型[34]。我们的网络有24个卷积层，其次是2个全连接层。与GoogLeNet使用的inception模块不同，我们简单地使用 1×1 缩减层，然后是 3×3 卷积层，类似于Lin等人[22]。整个网络如图3所示。

本文还训练了一个YOLO的快速版本，旨在推动快速目标检测的边界。Fast YOLO使用的神经网络卷积层更少(9层而不是24层)，这些层中的滤波器也更少。除了网络的大小，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预测的 $7 \times 7 \times 30$ 张量。

2.2. 培训

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的模型动物园[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，在预训练网络中同时添加卷积层和连接层可以提高性能[29]。按照他们的例子，我们添加了四个卷积层和两个随机初始化权重的全连接层。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 。

我们的最后一层同时预测类别概率和边界框坐标。我们将边界框的宽度和高度归一化为图像的宽度和高度，使它们落在0和1之间。我们将边界框 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泄漏纠正线性激活：

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

我们优化模型输出的平方和误差。我们使用平方和误差，因为它很容易优化，但它并不完全符合我们最大化平均精度的目标。该算法将定位误差与分类误差同等加权，但效果并不理想。此外，在每个图像中，许多网格单元不包含任何对象。这将这些单元格的“信心”分数推向零，通常超过了包含对象的单元格的梯度。这可能会导致模型不稳定，导致训练在早期出现分歧。



Figure 3: 建筑。我们的检测网络有24个卷积层，然后是2个全连接层。交替 1×1 卷积层减少了来自前面层的特征空间。我们在ImageNet分类任务上以一半的分辨率(224×224 输入图像)对卷积层进行预训练，然后将分辨率加倍用于检测。

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

平方和误差对大框和小框中的误差也具有同等的权重。我们的误差指标应该反映出，大盒子中的小偏差比小盒子中的小偏差更重要。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预测每个网格单元有多个边界框。在训练时，我们只需要一个边界框预测器来负责每个对象。我们指定一个预测器“负责”预测一个对象，基于哪个预测对象与真实值有最大的IOU。这导致边界框预测器之间的特化。每个预测器都能更好地预测特定大小、纵横比或对象类别，从而提高整体召回率。

在训练过程中，我们优化以下多部分损失函数：

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

其中 $\mathbb{1}_i^{\text{obj}}$ 表示对象是否出现在单元格*i*中， $\mathbb{1}_{ij}^{\text{obj}}$ 表示单元格*i*中的*j*第3个边界框预测器“负责”该预测。

请注意，损失函数仅在网格单元中存在对象时才惩罚分类错误(因此前面讨论的条件类概率)。它还只惩罚边界框坐标误差，如果预测器“负责”地面事实框(即在该网格单元中具有最高的IOU)。

在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年测试时，我们还包括VOC 2007测试数据用于训练。在整个训练过程中，我们使用的批量大小为64，动量为0.9，衰减为0.0005。

我们的学习率时间表如下：在第一个时期，我们缓慢地将学习率从 10^{-3} 提高到 10^{-2} 。如果我们从高学习率开始，我们的模型经常由于不稳定的梯度而发散。我们继续用 10^{-2} 训练75个纪元，然后用 10^{-3} 训练30个纪元，最后用 10^{-4} 训练30个纪元。

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，速率为0.5的dropout层阻止了层之间的共同适应[18]。对于数据增强，我们引入了高达原始图像大小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，最高可达1.5因子。

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个边界框以及每个框的类别概率。与基于分类器的方法不同，YOLO在测试时非常快，因为它只需要进行一次网络评估。

网格设计加强了边界框预测的空间多样性。通常，对象所属的网格单元很清楚，网络仅为每个对象预测一个框。然而，一些较大的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。非极大值抑制可用于修复这些多重检测。虽然不像RCNN或DPM那样对性能至关重要，但非最大抑制会增加2-3%的mAP。

2.4. YOLO的局限性

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能预测两个框，并且只能有一个类别。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。我们的模型与成群出现的小物体进行斗争，例如鸟群。

由于我们的模型学会了从数据中预测边界框，它很难泛化到具有新的或不寻常的纵横比或配置的对象。我们的模型还使用相对粗糙的特征来预测边界框，因为我们的架构从输入图像中有多个下采样层。

最后，当我们在一个近似于检测性能的损失函数上进行训练时，我们的损失函数在小边界框和大边界框中处理错误是一样的。大盒子里的小错误通常是良性的，但小盒子里的小错误对IOU的影响要大得多。错误的主要来源是不正确的本地化。

3. 与其他检测系统的比较

目标检测是计算机视觉中的一个核心问题。检测管道通常首先从输入图像中提取一组鲁棒的特征(Haar [25], SIFT [23], HOG [4], 卷积特征[6])。然后，使用分类器[36, 21, 13, 10]或定位器[1, 32]来识别特征空间中的对象。这些分类器或定位器要么以滑动窗口的方式在整个图像上运行，要么在图像中的某些区域子集[35, 15, 39]上运行。将YOLO检测系统与几个顶级检测框架进行了比较，强调了关键的相似性和差异性。

可变形部件模型。可变形部件模型(DPM)使用滑动窗口方法进行目标检测[10]。DPM使用不相交的管道来提取静态特征，对区域进行分类，预测高分区域的边界框等。我们的系统用一个卷积神经网络替换了所有这些不同的部分。该网络同时执行特征提取、边界框预测、非极大值抑制和上下文推理。网络不是静态特征，而是在线训练特征并针对检测任务进行优化。我们的统一架构导致了比DPM更快，更准确的模型。

r-cnn。R-CNN及其变体使用建议区域而不是滑动窗口在图像中查找目标。选择性搜索[35]生成潜在的边界框，卷积网络提取特征，SVM对边界框进行评分，线性模型调整边界框，非最大抑制消除重复检测。这个复杂管道的每个阶段都必须独立进行精确调整，结果系统非常慢，在测试时每张图像需要超过40秒[14]。

YOLO与R-CNN有一些相似之处。每个网格单元提出潜在的边界框，并使用卷积特征对这些框进行评分。然而，该系统对网格单元建议框施加了空间约束，有助于减轻对同一目标的多次检测。该系统还提出了更少的边界框，每个图像只有98个边界框，而选择搜索的约2000个。最后，系统将这些单独的组件组合成一个单独的、联合优化的模型。

其他快速检测器Fast and Faster R-CNN专注于通过共享计算和使用神经网络提出区域而不是选择性搜索来加速R-CNN框架[14][28]。虽然它们比R-CNN提供了速度和精度的改进，但仍然低于实时性能。

许多研究工作都集中在加速DPM管道[31][38][5]。它们可以加速HOG计算，使用级联并将计算推到gpu上。然而，实际上只有30Hz DPM [31]实时运行。

YOLO不是试图优化大型检测管道的单个组件，而是完全抛弃管道，从设计上看速度很快。

单个类别(如人脸或人)的检测器可以被高度优化，因为它们必须处理更少的变化[37]。YOLO是一个通用的检测器，它学会同时检测各种物体。

多盒。与R-CNN不同，Szegedy等人训练卷积神经网络来预测感兴趣的区域[8]而不是使用选择性搜索。MultiBox还可以通过将置信度预测替换为单类预测来执行单目标检测。然而，MultiBox不能执行一般的目标检测，仍然只是更大的检测管道中的一部分，需要进一步的图像块分类。YOLO和MultiBox都使用卷积网络来预测图像中的边界框，但YOLO是一个完整的检测系统。

过度。Sermanet等人训练一个卷积神经网络来执行定位，并使该定位器进行检测[32]。OverFeat可以有效地执行滑动窗口检测，但它仍然是一个不相交的系统。OverFeat优化的是定位性能，而不是检测性能。与DPM一样，本地化者在进行预测时只能看到本地信息。OverFeat无法推理全局上下文，因此需要大量的后处理来产生一致的检测。

多抓。我们的工作在设计上与Redmon等人的抓取检测工作相似[27]。所提出的网格包围盒预测方法基于MultiGrasp系统，用于回归抓取。然而，抓取检测比目标检测简单得多。对于包含一个物体的图像，MultiGrasp只需要预测一个可抓取区域。它不需要估计物体的大小、位置或边界或预测它的类别，只需要找到适合抓取的区域。YOLO同时预测图像中多个类别的多个对象的边界框和类别概率。

4. 实验

首先在PASCAL VOC 2007上将YOLO与其他实时检测系统进行比较。为了了解YOLO和R-CNN变体之间的差异，我们探索了YOLO和Fast R-CNN在VOC 2007上的错误，Fast R-CNN是性能最高的R-CNN版本之一[14]。基于不同的错误轮廓，本文表明YOLO可以用来对快速R-CNN检测进行重新评分，并减少背景假阳性的错误，从而带来显著的性能提升。展示了VOC 2012的结果，并将mAP与当前最先进的方法进行了比较。在两个艺术品数据集上表明，YOLO比其他检测器更好地泛化到新领域。

4.1. 与其他实时系统的比较

目标检测的许多研究工作都集中在快速实现标准检测管道上。[5][38][31][14][17][28]然而，只有Sadeghi等人实际上产生了一个实时运行的检测系统(每秒30帧或更好)[31]。我们将YOLO与他们在30Hz或100Hz下运行的DPM的GPU实现进行了比较。虽然其他努力没有达到实时里程碑，但我们还比较了它们的相对mAP和速度，以检查目标检测系统中可用的精度-性能权衡。

Fast YOLO是PASCAL上最快的目标检测方法；据我们所知，它是现存最快的物体探测器。通过52.7% mAP，它的实时检测精度是之前工作的两倍以上。YOLO将mAP推送到63.4%，同时仍然保持实时性能。

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
<hr/>			
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Table 1: 实时系统帕斯卡 **Voc 2007**。比较快速检测器的性能和速度。Fast YOLO是有记录以来最快的帕斯卡 VOC 检测检测器，其精度仍然是任何其他实时检测器的两倍。YOLO比快速版本更准确，但在速度上仍远高于实时。

我们还使用VGG-16训练YOLO。该模型更准确，但也明显慢于YOLO。它有助于与其他依赖VGG-16的检测系统进行比较，但由于它比实时慢，本文的其余部分专注于我们更快的模型。

最快的DPM有效地加速了DPM，而不牺牲太多的mAP，但它仍然错过了2倍的实时性能[38]。与神经网络方法相比，DPM的检测精度相对较低，这也限制了它。

R- cnn minus R将选择性搜索替换为静态边界框建议[20]。虽然它比R-CNN快得多，但它仍然低于实时性，并且由于没有好的建议，准确性受到了很大的影响。

Fast R-CNN加快了R-CNN的分类阶段，但它仍然依赖于选择性搜索，每个图像大约需要2秒来生成建议边界框。因此，它有很高的地图，但在0.5 fps上，它仍然离实时性很远。

最近的Faster R-CNN用神经网络取代选择性搜索来提出边界框，类似于Szegedy等人。[8]在我们的测试中，他们最准确的模型达到了7 fps，而较小的、不太准确的模型运行在18 fps。Faster R-CNN的VGG-16版本比YOLO高10 mAP，但也慢6倍。Zeiler-Fergus Faster R-CNN仅比YOLO慢2.5倍，但准确性也较低。

4.2. VOC 2007误差分析

为了进一步研究YOLO和最先进的检测器之间的差异，我们将详细分析VOC 2007上的结果。我们将YOLO与Fast R-CNN进行了比较，因为Fast R-CNN是PASCAL上性能最高的检测器之一，它的检测结果是公开的。

我们对每个类别使用Hoiem等人的方法和工具[19]。在测试时，我们查看该类别的前N个预测。每个预测要么是正确的，要么是根据错误类型进行分类的：

- 正确:正确的类和IOU > .5

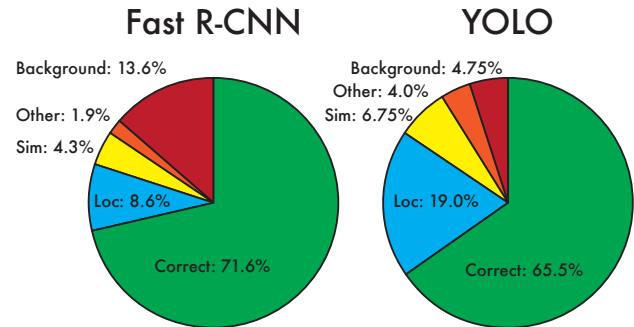


Figure 4: 错误分析:Fast R-CNN与YOLO这些图表显示了各种类别(N = 该类别中的#对象)的前N个检测中的定位和背景误差的百分比。

- 本地化:正确的类， $.1 < \text{IOU} < .5$
- 相似:类相似， $\text{IOU} > .1$
- 其他:class是错误的， $\text{IOU} > .1$
- 背景: $\text{IOU} < .1$ 为任何对象

图4显示了所有20个类中每个错误类型的平均分解。

YOLO很难正确定位对象。定位错误占YOLO错误的比例超过所有其他来源的总和。Fast R-CNN的定位误差要少得多，但背景误差要大得多。13.6%的顶级检测是不包含任何对象的误报。Fast R-CNN预测背景检测的可能性几乎是YOLO的3倍。

4.3. 结合Fast R-CNN和YOLO

YOLO比Fast R-CNN犯的背景错误要少得多。通过使用YOLO消除Fast R-CNN中的背景检测，我们获得了性能的显著提升。对于R-CNN预测的每个边界框，我们检查YOLO是否预测相似的框。如果是，我们根据YOLO预测的概率和两个框之间的重叠程度对预测进行提升。

最好的Fast R-CNN模型在VOC 2007测试集上取得了71.8%的mAP。当与YOLO结合时，其mAP增加了3.2%到75.0%。我们还尝试将顶级的Fast R-CNN模型与其他几个版本的Fast R-CNN相结合。这些组合产生的mAP在0.3%到0.6%之间的小幅增加，详见表2。

	mAP	Combined	Gain
Fast R-CNN	71.8	-	-
Fast R-CNN (2007 data)	66.9	72.4	.6
Fast R-CNN (VGG-M)	59.2	72.4	.6
Fast R-CNN (CaffeNet)	57.1	72.1	.3
YOLO	63.4	75.0	3.2

Table 2: VOC 2007上的模型组合实验。本文研究了将各种模型与最佳版本的Fast R-CNN相结合的效果。其他版本的Fast R-CNN只提供了很小的好处，而YOLO提供了显著的性能提升。

VOC 2012 test	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
MR_CNN_MORE_DATA [1]	73.9	85.5	82.9	76.6	57.8	62.7	79.4	77.2	86.6	55.0	79.1	62.2	87.0	83.4	84.7	78.9	45.3	73.4	65.8	80.3	74.0
HyperNet_VGG	71.4	84.2	78.5	73.6	55.6	53.7	78.7	79.8	87.7	49.6	74.9	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	79.9	65.7
HyperNet_SP	71.3	84.1	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	81.6	83.2	81.6	48.4	73.2	59.3	79.7	65.6
Fast R-CNN + YOLO	70.7	83.4	78.5	73.5	55.8	43.4	79.1	73.1	89.4	49.4	75.5	57.0	87.5	80.9	81.0	74.7	41.8	71.5	68.5	82.1	67.2
MR_CNN_S_CNN [11]	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Faster R-CNN [28]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
DEEP_ENS_COCO	70.1	84.0	79.4	71.6	51.9	51.1	74.1	72.1	88.6	48.3	73.4	57.8	86.1	80.0	80.7	70.4	46.6	69.6	68.8	75.9	71.4
NoC [29]	68.8	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1
Fast R-CNN [14]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
UMICH_FGS_STRUCT	66.4	82.9	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	55.8	82.7	77.1	79.9	68.7	41.4	69.0	60.0	72.0	66.2
NUS_NIN_C2000 [7]	63.8	80.2	73.8	61.9	43.7	43.0	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	38.6	68.3	58.0	68.7	63.3
BabyLearning [7]	63.2	78.0	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	55.7	68.7	62.6
NUS_NIN	62.4	77.9	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	38.4	66.9	56.2	66.9	62.7
R-CNN VGG BB [13]	62.4	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3
R-CNN VGG [13]	59.2	76.8	70.9	56.6	37.5	36.9	62.9	63.6	81.1	35.7	64.3	43.9	80.4	71.6	74.0	60.0	30.8	63.4	52.0	63.5	58.7
YOLO	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
Feature Edit [33]	56.3	74.6	69.1	54.4	39.1	33.1	65.2	62.7	69.7	30.8	56.0	44.6	70.0	64.4	71.1	60.2	33.3	61.3	46.4	61.7	57.8
R-CNN BB [13]	53.3	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1
SDS [16]	50.7	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7
R-CNN [13]	49.6	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6

Table 3: PASCAL VOC 2012排行榜。 2015年11月6日，YOLO与完整的comp4(允许外部数据)公开排行榜的对比。给出了各种检测方法的平均精度均值和类平均精度均值。YOLO是唯一的实时检测器。Fast R-CNN + YOLO是第四种得分最高的方法，比Fast R-CNN提高了2.3%。

YOLO的提升不仅仅是模型集成的副产品，因为组合不同版本的Fast R-CNN几乎没有好处。相反，正是因为YOLO在测试时犯了不同类型的错误，所以它在提高Fast R-CNN的性能方面非常有效。

不幸的是，这种组合不能从YOLO的速度中受益，因为我们分别运行每个模型，然后组合结果。然而，由于YOLO非常快，与fast R-CNN相比，它没有增加任何显著的计算时间。

4.4. VOC 2012测试结果

在VOC 2012测试集上，YOLO得分57.9%。这比目前的技术水平要低，更接近使用VGG-16的原始R-CNN，参见表3。与最接近的竞争对手相比，我们的系统在处理小对象时很吃力。在瓶子，绵羊和电视/显示器等类别上，YOLO的得分比R-CNN或Feature Edit低8-10%。然而，在猫和火车等其他类别上，YOLO取得了更高的性能。

我们结合的Fast R-CNN + YOLO模型是性能最高的检测方法之一。通过与YOLO的结合，Fast R-CNN的性能提升了2.3%，使其在公共排行榜上上升了5个名次。

4.5. 泛化性:艺术品中的人物检测

用于目标检测的学术数据集从相同的分布中提取训练和测试数据。在实际应用程序中，很难预测所有可能的用例，测试数据可能与系统之前看到的不同[3]。我们将YOLO与其他检测系统在Picasso数据集[12]和People-Art数据集[3]上进行了比较，这两个数据集用于测试艺术品上的人物检测。

图5显示了YOLO与其他检测方法的对比性能。作为参考，我们给出了VOC 2007对人的检测AP，其中所有模型仅在VOC 2007数据上训练。毕加索模型是在VOC 2012上训练的，而人艺术模型是在VOC 2010上训练的。

R-CNN在VOC 2007上有较高的AP。然而，R-CNN在应用于艺术品时下降很大。R-CNN对针对自然图像调整的边界框建议框使用选择性搜索。R-CNN中的分类器步骤只能看到小区域，需要好的建议。

DPM很好地维护了它的AP应用于美术作品。之前的工作理论认为，DPM表现良好是因为它具有强大的物体形状和布局的空间模型。虽然DPM不像R-CNN那样降级，但它从较低的AP开始。

YOLO在VOC 2007上表现良好，应用于艺术品时AP下降较小。像DPM一样，YOLO对大小和形状进行建模对象、对象之间的关系以及对象经常出现的位置。艺术品和自然图像在像素级别上非常不同，但它们在物体的大小和形状方面是相似的，因此YOLO仍然可以预测良好的边界框和检测。

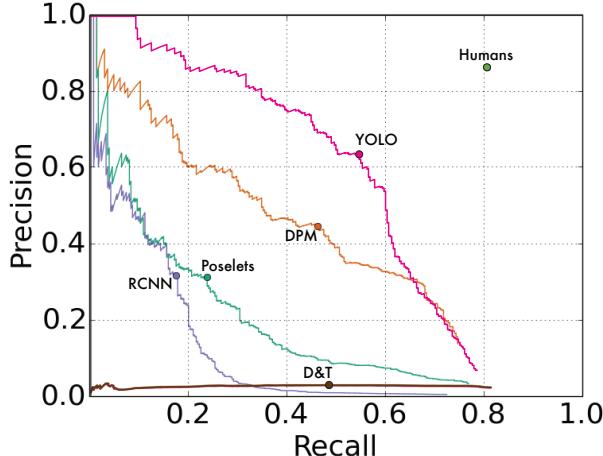
5. 野外实时检测

YOLO是一种快速、准确的目标检测器，非常适合计算机视觉应用。将YOLO连接到网络摄像头，并验证了它保持了实时性能，包括从摄像头获取图像和显示检测结果的时间。

由此产生的系统具有交互性和参与性。虽然YOLO单独处理图像，但当连接到网络摄像头时，它的功能就像一个跟踪系统，在物体移动和外观变化时检测它们。该系统的演示和源代码可以在我们的项目网站上找到:<http://pjreddie.com/yolo/>。

6. 结论

本文提出YOLO，一种用于目标检测的统一模型。该模型构建简单，可以直接在完整图像上进行训练。与基于分类器的方法不同，YOLO是在一个直接对应于检测性能的损失函数上训练的，整个模型是联合训练的。



(a) Picasso数据集的准确率-召回率曲线。

	VOC 2007	Picasso		People-Art
	AP	AP	Best F_1	AP
YOLO	59.2	53.3	0.590	45
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	

(b) 在VOC 2007、Picasso和People-Art数据集上的定量结果。Picasso数据集在AP和最佳 F_1 分数上进行评估。

Figure 5: Picasso和People-Art数据集上的泛化结果。



Figure 6: 定性结果。YOLO在来自互联网的艺术品样本和自然图像上运行。虽然它认为一个人是一架飞机，但它基本上是准确的。

Fast YOLO是文献中最快的通用目标检测器，YOLO在实时目标检测方面推动了最先进的技术。YOLO还可以很好地推广到新领域，使其成为依赖于快速、鲁棒的目标检测的应用程序的理想选择。

致谢:这项工作得到ONR N00014-13-1-0720、NSF IIS-1338054和Allen杰出研究员奖的部分支持。

References

- [1] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *Computer Vision-ECCV 2008*, pages 2–15. Springer, 2008. 4
- [2] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision (ICCV)*, 2009. 7
- [3] H. Cai, Q. Wu, T. Corradi, and P. Hall. The cross-depiction problem: Computer vision algorithms for recognising objects in artwork and in photographs. *arXiv preprint arXiv:1505.00110*, 2015. 6
- [4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*

- tion, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 4, 7
- [5] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, J. Yagnik, et al. Fast, accurate detection of 100,000 object classes on a single machine. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1814–1821. IEEE, 2013. 4
- [6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 4
- [7] J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and semantic segmentation. In *Computer Vision-ECCV 2014*, pages 299–314. Springer, 2014. 6
- [8] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2155–2162. IEEE, 2014. 4, 5
- [9] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 2
- [10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 1, 4
- [11] S. Gidaris and N. Komodakis. Object detection via a multi-region & semantic segmentation-aware CNN model. *CoRR*, abs/1505.01749, 2015. 6
- [12] S. Ginosar, D. Haas, T. Brown, and J. Malik. Detecting people in cubist art. In *Computer Vision-ECCV 2014 Workshops*, pages 101–116. Springer, 2014. 6
- [13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014. 1, 4, 6
- [14] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. 1, 4, 5, 6
- [15] S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In *Advances in neural information processing systems*, pages 655–663, 2009. 4
- [16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Computer Vision-ECCV 2014*, pages 297–312. Springer, 2014. 6
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *arXiv preprint arXiv:1406.4729*, 2014. 4
- [18] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 3
- [19] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *Computer Vision-ECCV 2012*, pages 340–353. Springer, 2012. 5
- [20] K. Lenc and A. Vedaldi. R-cnn minus r. *arXiv preprint arXiv:1506.06981*, 2015. 5
- [21] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900. IEEE, 2002. 4
- [22] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 2
- [23] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 4
- [24] D. Mishkin. Models accuracy on imagenet 2012 val. <https://github.com/BVLC/caffe/wiki/Models-accuracy-on-ImageNet-2012-val>. Accessed: 2015-10-2. 2
- [25] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Computer vision, 1998. sixth international conference on*, pages 555–562. IEEE, 1998. 4
- [26] J. Redmon. Darknet: Open source neural networks in c. <http://pjreddie.com/darknet/>, 2013–2016. 2
- [27] J. Redmon and A. Angelova. Real-time grasp detection using convolutional neural networks. *CoRR*, abs/1412.3128, 2014. 4
- [28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 4, 5, 6
- [29] S. Ren, K. He, R. B. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. *CoRR*, abs/1504.06066, 2015. 2, 6
- [30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015. 2
- [31] M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In *Computer Vision-ECCV 2014*, pages 65–79. Springer, 2014. 4, 5
- [32] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013. 4
- [33] Z. Shen and X. Xue. Do more dropouts in pool5 feature maps for better object detection. *arXiv preprint arXiv:1409.6911*, 2014. 6
- [34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 2
- [35] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 4
- [36] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 4:34–47, 2001. 4
- [37] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 4

- [38] J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2497–2504. IEEE, 2014. [4](#), [5](#)
- [39] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014. [4](#)