只看一次：统一、实时的物体检测

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

摘要

我们提出了一种全新的物体检测方法 YOLO。之前的物体检测工作是利用分类器进行检测。而我们将物体检测作为一个回归问题，回归到空间上分离的边界框和相关的类别概率。在一次评估中，单个神经网络可直接从完整图像中预测边界框和类概率。由于整个检测管道是一个单一的网络，因此可以直接根据检测性能进行端到端的优化。

我们的统一架构速度极快。我们的基本 YOLO 模型以每秒 45 帧的速度实时处理图像。该网络的缩小版--快速 YOLO--的处理速度达到惊人的每秒 155 帧，而 mAP 却仍然是其他实时检测器的两倍。与最先进的检测系统相比，YOLO 的定位错误更多，但在背景上预测误报的可能性较小。最后，YOLO 可以学习非常通用的物体表征。从自然图像泛化到艺术品等其他领域时，它的表现优于其他检测方法，包括 DPM 和 R-CNN。

1. 引言

人类瞥一眼图像，就能立即知道图像中有哪些物体，它们在哪里，以及它们是如何相互作用的。人类的视觉系统快速而准确，使我们几乎不需要有意识的思考就能完成驾驶等复杂任务。快速、准确的物体检测算法可以让计算机在没有专门传感器的情况下驾驶汽车，让辅助设备向人类用户传递实时场景信息，并释放通用、反应灵敏的机器人系统的潜力。

当前的检测系统会重新使用分类器来进行检测。为了检测一个物体，这些系统会使用一个针对该物体的分类器，并在测试图像的不同位置和尺度上对其进行评估。可变形部件模型（DPM）等系统使用滑动窗口方法，在整个图像中均匀分布的位置运行分类器[10]。

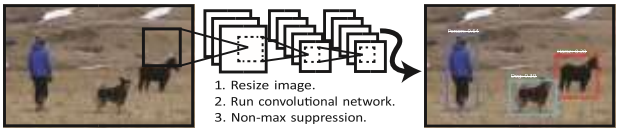


图 1：YOLO 检测系统。使用 YOLO 处理图像简单明了。我们的系统 (1) 将输入图像的大小调整为 448×448，(2) 在图像上运行一个卷积网络，(3) 根据模型的置信度对检测结果进行阈值处理。

较新的方法（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，后处理用于完善边界框，消除重复检测，并根据场景中的其他物体对边界框重新评分 [13]。这些复杂的流水线既缓慢又难以优化，因为每个组件都必须单独训练。

我们将物体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别概率。使用我们的系统，您只需查看一次（YOLO）图像，就能预测出哪些物体存在以及它们在哪里。

YOLO 简单得令人耳目一新：见图 1。一个卷积网络可同时预测多个边界框和这些边界框的类别概率。YOLO 在完整图像上进行训练，并直接优化检测性能。与传统的物体检测方法相比，这种统一模型具有多种优势。

首先，YOLO 速度极快。由于我们将检测视为一个回归问题，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，即可预测检测结果。我们的基础网络在 Titan X GPU 上以每秒 45 帧的速度运行，无需批处理，而快速版本的运行速度则超过了每秒 150 帧。这意味着我们可以实时处理流媒体视频，延迟时间不到 25 毫秒。此外，YOLO 的平均精度是其他实时系统的两倍多。如需观看我们的系统在网络摄像头上实时运行的演示，请访问我们的项目网页：http://pjreddie.com/yolo/。

其次，YOLO 在进行预测时对图像进行全局推理。与基于滑动窗口和区域建议的技术不同，YOLO 在训练和测试时看到的是整个图像，因此它隐含了关于类别及其外观的上下文信息。快速 R-CNN 是一种顶级检测方法 [14]，由于看不到更大的上下文，因此会将图像中的背景斑块误认为是物体。与快速 R-CNN 相比，YOLO 的背景错误率不到一半。

第三，YOLO 可学习对象的通用表征。在自然图像上进行训练并在艺术品上进行测试时，YOLO 的表现远远超过 DPM 和 R-CNN 等顶级检测方法。由于 YOLO 具有很强的通用性，因此在应用于新领域或意外输入时，不太可能出现问题。

在精确度方面，YOLO 仍然落后于最先进的检测系统。虽然它能快速识别图像中的物体，但在精确定位某些物体，尤其是小物体方面却很吃力。我们将在实验中进一步研究这些权衡问题。

我们所有的训练和测试代码都是开源的。各种预训练模型也可供下载。

2. 统一检测

我们将物体检测的各个组成部分统一到一个神经网络中。我们的网络使用整个图像的特征来预测每个边界框。它还能同时预测图像中所有类别的所有边界框。这意味着我们的网络可以对整个图像和图像中的所有物体进行全局推理。YOLO 设计可实现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 × S 网格。如果一个物体的中心落在一个网格单元内，则该网格单元负责检测该物体。

每个网格单元预测 B 边框和这些边框的置信度分数。这些置信度分数反映了模型对该方框包含一个物体的置信度，以及模型认为其预测的方框的准确度。形式上，我们将置信度定义为 Pr(Object) ∗ IOUtruthpred。如果该单元格中不存在任何对象，那么置信度分数应为零。否则，我们希望置信度分数等于预测方框与地面实况之间的交集大于联合（IOU）。

每个边界框由 5 个预测值组成：x、y、w、h 和置信度。(x,y) 坐标表示相对于网格单元边界的边界框中心。宽度和高度是相对于整个图像的预测值。最后，置信度预测表示预测方框与任何地面实况方框之间的 IOU。

每个网格单元还预测 C 个条件类概率，即 Pr(Classi|Object)。这些概率以包含对象的网格单元为条件。我们只预测每个网格单元的一组类别概率，与方格 B 的数量无关。

测试时，我们将条件类概率和单个方框置信度预测值相乘

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

从而得出每个方框中特定类别的置信度分数。这些分数既表示该类出现在方框中的概率，也表示预测方框与对象的匹配程度。

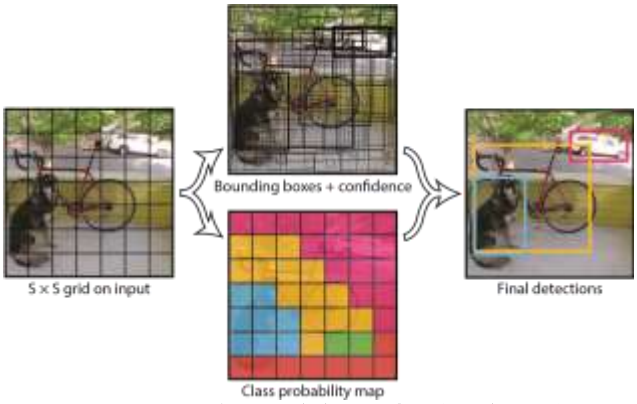


图 2：模型。我们的系统将检测建模为一个回归问题。它将图像划分为 S×S 网格，并为每个网格单元预测 B 边框、这些边框的置信度和 C 类概率。这些预测结果被编码为 S×S×(B*5+C) 张量。

在 PASCAL VOC 上评估 YOLO 时，我们使用 S = 7，B = 2。PASCAL VOC 有 20 个标签类，因此 C = 20。我们的最终预测结果是一个 7 × 7 × 30 的张量。

2.1. Network Design

我们以卷积神经网络的形式实现了这一模型，并在 PASCAL VOC 检测数据集上对其进行了评估[9]。网络的初始卷积层从图像中提取特征，而全连接层则预测输出概率和坐标。

我们的网络架构受 GoogLeNet 图像分类模型的启发 [34]。我们的网络有 24 个卷积层，然后是 2 个全连接层。我们没有使用 GoogLeNet 使用的入门模块，而是简单地使用了 1×1 还原层，然后是 3×3 卷积层，这与 Lin 等人的做法类似[22]。整个网络如图 3 所示。

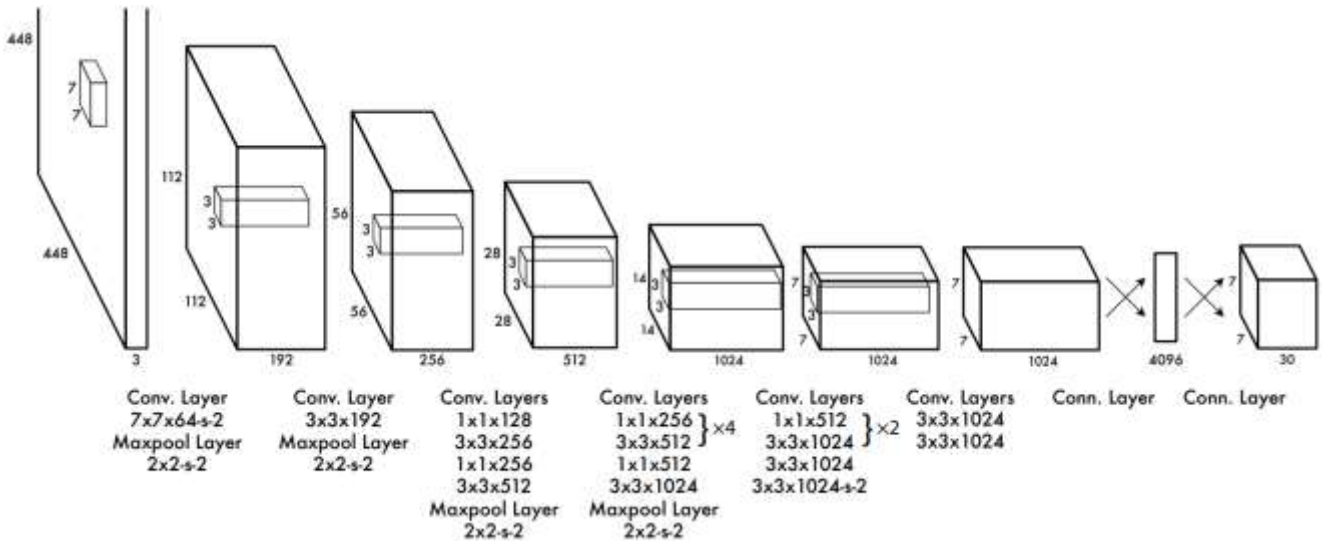


图 3：架构。我们的检测网络有 24 个卷积层，之后是 2 个全连接层。交替使用的 1 × 1 卷积层缩小了前几层的特征空间。我们在 ImageNet 分类任务中以一半的分辨率（224 × 224 输入图像）对卷积层进行预训练，然后以两倍的分辨率进行检测。

我们还训练了一个快速版本的 YOLO，旨在突破快速物体检测的极限。快速 YOLO 使用的神经网络的卷积层数较少（9 层而不是 24 层），卷积层中的过滤器也较少。除网络规模外，YOLO 和快速 YOLO 的所有训练和测试参数均相同。

我们网络的最终输出是 7 × 7 × 30 的预测张量。

2.2. Training

我们在 ImageNet 1000 级竞赛数据集 [30] 上对卷积层进行预训练。在预训练中，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，在 ImageNet 2012 验证集上的单作物前五名准确率达到 88%，与 Caffe 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，我们将模型转换为执行检测。Ren 等人的研究表明，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。检测通常需要精细的视觉信息，因此我们将网络的输入分辨率从 224 × 224 提高到 448 × 448。

最后一层预测类别概率和边界框坐标。我们根据图像的宽度和高度对边界框的宽度和高度进行归一化处理，使其介于 0 和 1 之间。我们将边界框的 x 坐标和 y 坐标参数化为特定网格单元位置的偏移量，使其也介于 0 和 1 之间。

我们在最后一层使用了线性激活函数，所有其他层都使用了以下泄漏整流线性激活函数：

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

我们对模型输出的平方总误差进行优化。我们使用平方总误差是因为它易于优化，但它与我们最大化平均精度的目标并不完全一致。它将定位误差与分类误差加权相等，这可能并不理想。此外，在每幅图像中，许多网格单元都不包含任何物体。这就会将这些单元格的 "置信度 "分数推向零，往往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，使训练在早期就出现偏差。

为了解决这个问题，我们增加了边界框坐标预测的损失，减少了不包含物体的边界框置信度预测的损失。我们使用两个参数 λ_{coord} 和 λ_{noobj} 来实现这一目标。我们设置 $\lambda_{coord} = 5$ 和 $\lambda_{noobj} = .5$。

总方误差也同样权衡大方框和小方框中的误差。我们的误差指标应该反映出，大方框中的小偏差比小方块中的小偏差更重要。为了部分解决这个问题，我们预测了边框宽

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框。在训练时，我们只希望每个对象由一个边界框预测器负责。我们根据当前与地面实况的 IOU 值最高的预测结果，指定一个预测器 "负责 "预测一个对象。这就导致了边界框预测器之间的专业化。每个预测器都能更好地预测特定尺寸、长宽比或类别的物体，从而提高整体召回率。

在训练过程中，我们对以下多部分损失函数进行优化：

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

其中$\mathbb{1}_i^{\text{obj}}$，表示对象是否出现在单元 i 中，表示单元 i 中的第 j 个边框预测器对该预测 "负责"。

请注意，损失函数只在对象出现在该网格单元时才会对分类错误进行惩罚（因此才有了前面讨论的条件类概率）。此外，只有当预测因子对地面实况框 "负责 "时（即在该网格单元的所有预测因子中具有最高的 IOU），损失函数才会对边界框坐标错误进行惩罚。

我们在 PASCAL VOC 2007 和 2012 的训练和验证数据集上对网络进行了约 135 个历元的训练。在 2012 年进行测试时，我们还使用 VOC 2007 测试数据进行训练。在整个训练过程中，我们使用的批次大小为 64，动量为 0.9，衰减为 0.0005。

我们的学习率安排如下：在最初的历时中，我们将学习率从 10-3 缓慢提高到 10-2。如果我们一开始就采用较高的学习率，我们的模型往往会因梯度不稳定而发散。我们继续以 10-2 的学习率训练 75 个 epochs，然后以 10-3 的学习率训练 30 个 epochs，最后以 10-4 的学习率训练 30 个 epochs。

为了避免过度拟合，我们使用了剔除和广泛的数据扩增。在第一个连接层之后有一个比率为 0.5 的剔除层，可以防止层与层之间的共同适应[18]。在数据增强方面，我们引入了随机缩放和平移，最大缩放和平移量为原始图像大小的 20%。我们还随机调整图像的曝光度和饱和度，在 HSV 色彩空间中最多调整 1.5 倍。

2.3. 推论

与训练一样，预测测试图像的检测结果只需要一次网络评估。在 PASCAL VOC 上，网络可预测每幅图像的 98 个边界框以及每个边界框的类别概率。与基于分类器的方法不同，YOLO 只需要一次网络评估，因此在测试时速度极快。

网格设计强化了边界框预测的空间多样性。通常情况下，一个物体属于哪个网格单元是显而易见的，网络只能为每个物体预测一个边框。然而，一些大型物体或靠近多个单元边界的物体可以被多个单元很好地定位。可以使用非最大抑制来修复这些多重检测。虽然非最大抑制对性能的影响并不像 R-CNN 或 DPM 那样关键，但却能使 mAP 增加 23%。

2.4. YOLO 的局限性

YOLO 对边框预测施加了很强的空间限制，因为每个网格单元只能预测两个边框，并且只能有一个类别。这种空间约束限制了我们的模型所能预测的附近物体的数量。我们的模型很难预测成群出现的小物体，例如鸟群。

由于我们的模型是从数据中学习预测边界框的，因此它很难泛化到新的或不寻常的长宽比或配置的物体上。我们的模型还使用了相对粗糙的特征来预测边界框，因为我们的架构对输入图像进行了多层降采样。

最后，虽然我们使用近似检测性能的损失函数进行训练，但我们的损失函数对小包围盒和大包围盒中的错误一视同仁。大包围盒中的小误差通常是无害的，但小包围盒中的小误差对 IOU 的影响要大得多。我们的主要错误来源是不正确的定位。

3. 与其他检测系统的比较

物体检测是计算机视觉的核心问题。检测管道一般首先从输入图像中提取一组鲁棒特征（Haar [25]、SIFT [23]、HOG [4]、卷积特征[6]）。然后，分类器 [36, 21, 13, 10] 或定位器 [1, 32] 被用来识别特征空间中的物体。这些分类器或定位器要么以滑动窗口方式在整个图像上运行，要么在图像中的某些区域子集上运行 [35、15、39]。我们将 YOLO 检测系统与几种顶级检测框架进行了比较，突出了主要的异同点。

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物体检测[10]。DPM 使用一个互不关联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取代了所有这些不同的部分。该网络可同时执行特征提取、边界框预测、非最大抑制和上下文推理。该网络不使用静态特征，而是在线训练特征，并针对检测任务对其进行优化。与 DPM 相比，我们的统一架构能带来更快、更准确的模型。

R-CNN. R-CNN 及其变体使用区域建议而不是滑动窗口来查找图像中的物体。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。这一复杂流水线的每个阶段都必须独立进行精确调整，由此产生的系统速度非常慢，测试时每幅图像需要 40 多秒钟[14]。

YOLO 与 R-CNN 有一些相似之处。每个网格单元都会提出潜在的边界框，并利用卷积特征对这些边界框进行评分。不过，我们的系统对网格单元的提议施加了空间限制，这有助于减少同一物体的多次检测。我们的系统提出的边界框数量也要少得多，每幅图像只需 98 个，而选择性搜索则需要约 2000 个。最后，我们的系统将这些单独的组件整合到一个联合优化的模型中。

其他快速检测器：快速和更快 R-CNN 的重点是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快 R-CNN 框架的速度 [14] [28]。与 R-CNN 相比，它们在速度和准确性上都有所提高，但仍达不到实时性能。

许多研究工作的重点是加速 DPM 管道 [31] [38] [5]。它们加快了 HOG 计算速度，使用级联，并将计算推向 GPU。然而，真正实时运行的 DPM 只有 30Hz [31]。

YOLO 并不试图优化大型检测管道中的单个组件，而是完全抛弃了管道设计，因此速度非常快。

针对人脸或人物等单一类别的检测器可以进行高度优化，因为它们需要处理的变化要少得多 [37]。YOLO 是一种通用检测器，可同时检测多种物体。

深度多盒：与 R-CNN 不同，Szegedy 等人训练一个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 还可以用单类预测取代置信度预测，从而进行单个物体检测。不过，MultiBox 无法进行一般物体检测，仍然只是更大的检测管道中的一个环节，需要进一步的图像斑块分类。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。

OverFeat：Sermanet 等人训练了一个卷积神经网络来执行定位，并调整该定位器来执行检测[32]。OverFeat 可以高效地执行滑动窗口检测，但它仍然是一个不连贯的系统。OverFeat 优化的是定位，而不是检测性能。与 DPM 一样，定位器在进行预测时只能看到本地信息。OverFeat 无法推理全局上下文，因此需要进行大量的后处理才能产生一致的检测结果。

多重抓取：我们的工作在设计上与 Redmon 等人的抓取检测工作类似[27]。我们的网格边界框预测方法是基于 MultiGrasp 系统的抓取回归。然而，抓取检测是一项比物体检测简单得多的任务。MultiGrasp 只需预测包含一个物体的图像中的单一可抓取区域。它不需要估计物体的大小、位置或边界，也不需要预测物体的类别，只需要找到适合抓取的区域即可。YOLO 可以预测图像中多个物体的边界框和类别概率。

4. 实验

首先，我们在 PASCAL VOC 2007 上比较了 YOLO 和其他实时检测系统。为了了解

YOLO 和 R-CNN 变体之间的差异，我们探讨了 YOLO 和 Fast R-CNN（R-CNN 性能最高的版本之一[14]）在 VOC 2007 上的误差。根据不同的误差情况，我们发现 YOLO 可用于对 Fast R-CNN 检测进行重新评分，减少背景误报带来的误差，从而显著提高性能。我们还介绍了 VOC 2012 的结果，并将 mAP 与当前最先进的方法进行了比较。最后，我们还在两个艺术作品数据集上展示了 YOLO 在新领域的通用性优于其他检测器。

4.1. 与其他实时系统的比较

物体检测领域的许多研究工作都集中在使标准检测管道快速运行上。[5] [38] [31] [14] [17] [28] 然而，只有 Sadeghi 等人真正开发出了实时（每秒 30 帧或更好）运行的检测系统[31]。我们将 YOLO 与他们在 GPU 上实现的 DPM 进行了比较，后者的运行频率为 30Hz 或 100Hz。虽然其他研究没有达到实时的里程碑，但我们也比较了它们的相对 mAP 和速度，以考察物体检测系统在精度和性能之间的权衡。

快速 YOLO 是 PASCAL 上最快的物体检测方法；据我们所知，它也是目前最快的物体检测器。它的 mAP 为 52.7%，比之前的实时检测准确率高出一倍多。YOLO 在保持实时性能的同时，将 mAP 提高到了 63.4%。

我们还使用 VGG-16 训练 YOLO。该模型更准确，但速度也明显慢于 YOLO。它有助于与其他依赖 VGG-16 的检测系统进行比较，但由于它的速度比实时速度慢，本文的其余部分将重点讨论我们速度更快的模型。

最快 DPM 可以在不牺牲太多 mAP 的情况下有效地加快 DPM 的速度，但其实时性仍然要差 2 倍[38]。与神经网络方法相比，DPM 的检测精度相对较低，这也限制了它的使用。

R-CNN minus R 用静态边界框建议取代了选择性搜索 [20]。虽然它比 R-CNN 快得多，但仍达不到实时性，而且由于没有好的建议，准确性也会受到很大影响。

快速 R-CNN 加快了 R-CNN 的分类阶段，但它仍然依赖于选择性搜索，每幅图像生成边界框提议需要大约 2 秒钟的时间。因此，它具有较高的 mAP，但 0.5 fps 的速度离实时还很远。

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
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

表 1：2007 年 PASCAL VOC 实时系统。比较快速检测器的性能和速度。快速 YOLO 是 PASCAL VOC 检测记录中速度最快的检测器，其准确度仍然是其他实时检测器的两倍。YOLO 的精确度比快速版本高 10 mAP，但速度仍远远高于实时版本。

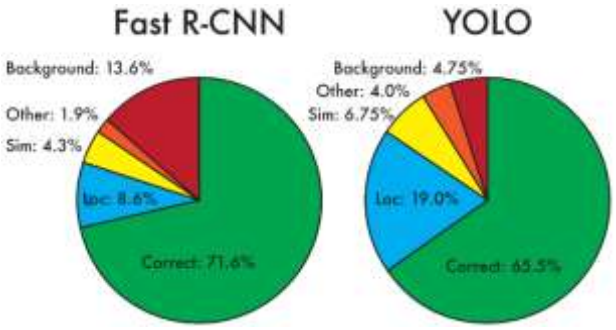


图 4：误差分析：快速 R-CNN 与 YOLO 的对比 这些图表显示了不同类别（N = 该类别中的物体数量）中前 N 个检测中定位和背景错误的百分比。

最近推出的 Faster R-CNN 用神经网络取代了选择性搜索来提出边界框，这与 Szegedy 等人的做法类似[8]。在我们的测试中，他们最精确的模型达到了 7 fps，而一个较小但不太精确的模型运行速度为 18 fps。VGG-16 版本的 Faster R-CNN 比 YOLO 高 10 mAP，但速度也慢 6 倍。ZeilerFergus 的 Faster R-CNN 只比 YOLO 慢 2.5 倍，但精确度也较低。

4.2. VOC 2007 误差分析

为了进一步研究 YOLO 与最先进的检测器之间的差异，我们查看了 VOC 2007 的详细结果。我们将 YOLO 与 Fast RCNN 进行了比较，因为 Fast R-CNN 是 PASCAL 上性能最高的检测器之一，而且它的检测结果是公开的。

我们使用 Hoiem 等人的方法和工具[19]。对于测试时的每个类别，我们会查看该类别的前 N 个预测。每个预测要么正确，要么根据错误类型进行分类：

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 [11]	**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

表 3：PASCAL VOC 2012 排行榜。截至 2015 年 11 月 6 日，YOLO 与完整的 comp4（允许外部数据）公开排行榜进行了比较。显示了各种检测方法的平均精度和每类平均精度。YOLO 是唯一的实时检测器。Fast R-CNN + YOLO 是得分第四高的方法，比 Fast R-CNN 提高了 2.3%。

- 其他：类别错误，IOU > .1
- 背景：任何对象的 IOU < .1

图 4 显示了所有 20 个班级中每种错误类型的平均值。

Y OLO 难以正确定位对象。在 YOLO 的错误中，定位错误所占的比例比所有其他来源的错误总和还要多。快速 R-CNN 的定位错误要少得多，但背景错误却多得多。它的最高检测结果中有 13.6% 是不包含任何物体的误报。快速 R-CNN 预测背景检测的可能性几乎是 YOLO 的 3 倍。

4.3. 结合快速 R-CNN 和 YOLO

与快速 R-CNN 相比，YOLO 犯的背景错误要少得多。通过使用 YOLO 来消除快速 R-CNN 的背景检测，我们的性能得到了显著提升。对于 R-CNN 预测的每一个边界框，我们都会检查 YOLO 是否预测了类似的边界框。如果有，我们就会根据 YOLO 预测的概率和两个方框之间的重叠情况对该预测进行提升。

	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**

表 2：2007 年 VOC 的模型组合实验。我们研究了将各种模型与最佳版本的快速 R-CNN 结合起来的效果。其他版本的快速 R-CNN 只带来了很小的好处，而 YOLO 则带来了显著的性能提升。

在 VOC 2007 测试集上，最佳快速 R-CNN 模型的 mAP 为 71.8%。当与 YOLO 结合使用时，其 mAP 增加了 3.2%，达到 75.0%。我们还尝试将顶级快速 R-CNN 模型与其他几个版本的快速 R-CNN 模型相结合。这些组合使 mAP 略有增加，增幅在 0.3% 到 0.6% 之间，详见表 2。

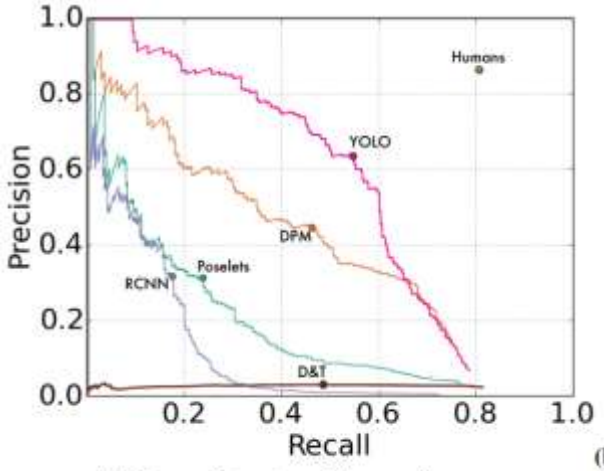
YOLO 带来的提升并不只是模型集合的副产品，因为将不同版本的 Fast R-CNN 结合在一起几乎没有什么好处。相反，正是因为 YOLO 在测试时会犯不同类型的错误，所以它才能如此有效地提升 Fast R-CNN 的性能。

遗憾的是，这种组合无法从 YOLO 的速度中获益，因为我们是先单独运行每个模型，然后再合并结果。不过，由于 YOLO 的速度非常快，因此与快速 R-CNN 相比，并不会增加大量的计算时间。

4.4. VOC 2012 Results

在 VOC 2012 测试集上，YOLO 的 mAP 得分为 57.9%。这低于目前的技术水平，更接近于使用 VGG-16 的原始 R-CNN，见表 3。与最接近的竞争对手相比，我们的系统在处理小物体时比较吃力。在瓶子、羊和电视/显示器等类别上，YOLO 的得分比 R-CNN 或 Feature Edit 低 8-10%。不过，在猫和火车等其他类别上，YOLO 的性能更高。

我们的快速 R-CNN + YOLO 组合模型是性能最高的检测方法之一。快速 R-CNN 通过



(a) Picasso Dataset precision-recall curves.

	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) Quantitative results on the VOC 2007, Picasso, and People-Art Datasets. The Picasso Dataset evaluates on both AP and best F_1 score.

图 5：毕加索和人物-艺术数据集的泛化结果。

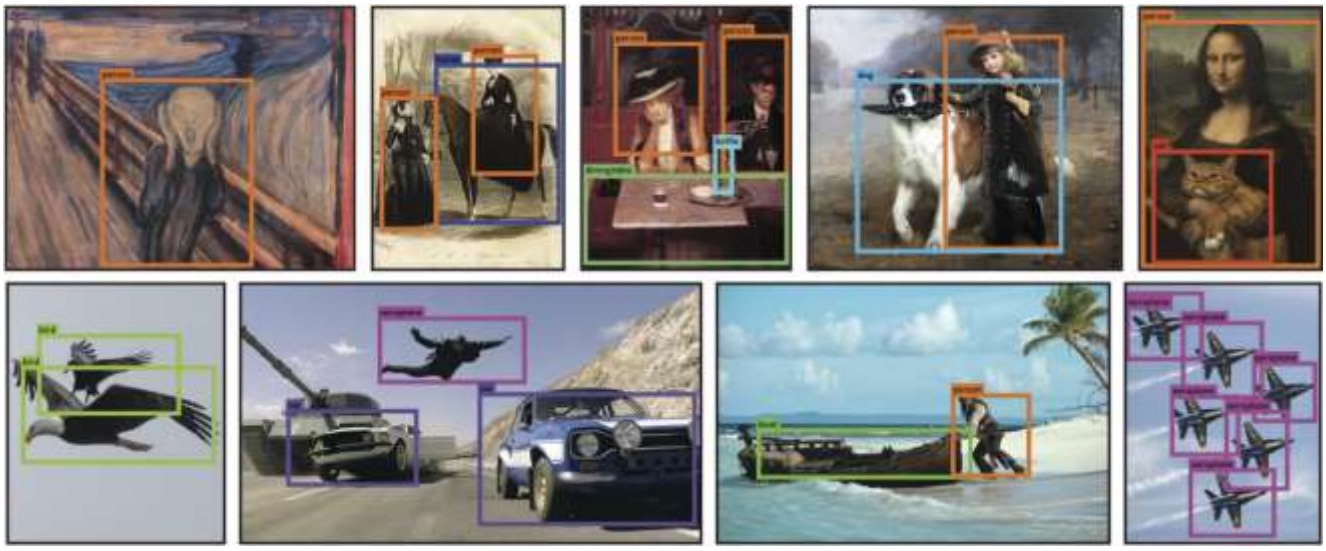


图 6：定性结果。YOLO 在样本艺术品和互联网自然图像上运行。虽然它认为一个人是飞机，但基本准确。

与 YOLO 的结合提高了 2.3%，在公众排行榜上提升了 5 位。

4.5. 通用性：艺术作品中的人物检测

用于物体检测的学术数据集从相同的分布中提取训练数据和测试数据。在实际应用中，很难预测所有可能的使用情况，测试数据也可能与系统之前看到的数据不同[3]。我们在毕加索数据集[12]和人物-艺术数据集[3]上将 YOLO 与其他检测系统进行了比较。

图 5 显示了 YOLO 与其他检测方法的性能比较。作为参考，我们给出了 VOC 2007 对人物的检测结果，其中所有模型都只在 VOC 2007 数据基础上进行了训练。关于毕加索的模型是在 VOC 2012 数据基础上训练的，而关于人物-艺术的模型是在 VOC 2010 数据基础上训练的。

R-CNN 在 2007 年的挥发性有机化合物检测中具有很高的 AP 值。不过，当 R-CNN 应用于艺术品时，其性能会大幅下降。R-CNN 使用选择性搜索（Selective Search）来提出边界框建议，这种方法适用于自然图像。R-CNN 中的分类器步骤只能看到很小的区域，因此需要很好的建议。

DPM 在应用于艺术品时能很好地保持其 AP 性能。先前的研究理论认为，DPM 表现出色是因为它对物体的形状和布局具有强大的空间模型。虽然 DPM 不会像 R-CNN 那样退化，但它的起始 AP 较低。

YOLO 在 VOC 2007 上有良好的表现，在应用于艺术品时，其 AP 的退化程度低于其他方法。与 DPM 一样，YOLO 对物体的大小和

形状、物体之间的关系以及物体经常出现的位置进行建模。艺术品和自然图像在像素层面上有很大不同，但在物体的大小和形状方面却很相似，因此 YOLO 仍能预测出良好的边界框和检测结果。

5. 野外实时检测

YOLO 是一种快速、准确的物体检测器，是计算机视觉应用的理想选择。我们将 YOLO 连接到网络摄像头，并验证它是否能保持实时性能，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互动性和吸引力。YOLO 可以单独处理图像，而当连接到网络摄像头时，它的功能就像一个跟踪系统，可以检测物体的移动和外观变化。系统演示和源代码可在我们的项目网站 http://pjreddie.com/yolo/ 上找到。

6. 结论

我们介绍了用于物体检测的统一模型 YOLO。我们的模型构建简单，可直接在完整图像上进行训练。与基于分类器的方法不同，YOLO 是根据与检测性能直接对应的损失函数进行训练的，而且整个模型是联合训练的。

快速 YOLO 是文献中速度最快的通用对象检测器，YOLO 推动了实时对象检测技术的发展。YOLO 还能很好地扩展到新的领域，因此非常适合依赖于快速、稳健物体检测的应用。

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

参考文献

[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. 8

[3] H. Cai, Q. Wu, T. Corradi, and P. Hall. The crossdepiction problem: Computer vision algorithms for recognising objects in artwork and in photographs. *arXiv preprint arXiv:1505.00110*, 2015. 7

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 4, 8

[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. 5

[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. 7

[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. 5, 6

[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 multiregion & semantic segmentation-aware CNN model. *CoRR*, abs/1505.01749, 2015. 7

[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. 7

[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, 7

[14] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. 2, 5, 6, 7

[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. Arbelaez, R. Girshick, and J. Malik. Simul- ´ taneous detection and segmentation. In *Computer Vision–ECCV 2014*, pages 297–312. Springer, 2014. 7

[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. 5

[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. 4

[19] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *Computer Vision–ECCV 2012*, pages 340–353. Springer, 2012. 6

[20] K. Lenc and A. Vedaldi. R-cnn minus r. *arXiv preprint arXiv:1506.06981*, 2015. 5, 6

[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. 3

[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. 3

[27] J. Redmon and A. Angelova. Real-time grasp detection using convolutional neural networks. *CoRR*, abs/1412.3128, 2014. 5

[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. 5, 6, 7

[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. 3, 7

[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. 3

[31] M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In *Computer Vision–ECCV 2014*, pages 65–79. Springer, 2014. 5, 6

[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, 5

[33] Z. Shen and X. Xue. Do more dropouts in pool5 feature maps for better object detection. *arXiv preprint arXiv:1409.6911*, 2014. 7

[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. 5

[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. 5, 6

[39] C. L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014. 4