You Only Look Once 网络:
统一的，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/

摘 要

我们提出了一种新的物体检测方法 YOLO。先前关于目标检测的工作将分类器重新用于每个物品的检测。相反，我们将目标检测作为一个回归问题，将从空间上分离边界框和相关的类概率。单个神经网络在一次评估中直接从完整图像预先确定边界框和类概率。由于整个检测流水线是单个网络，因此可以直接针对检测性能进行端到端的优化。

我们的统一架构非常快。我们的基础 YOLO 模型以每秒 45 帧的速度实时处理图像。较小版本的网络 Fast YOLO 每秒处理惊人的 155 帧，同时仍然达到其他实时检测器的 mAP 的两倍。与最先进的检测系统相比，YOLO 产生更多的定位误差，但不太可能预测背景上的负样例。最后，YOLO 学习了目标的非常通用的表示。当从自然图像到艺术品等其他领域进行一般化时，它优于其他检测方法，包括 DPM 和 R-CNN。

1. 介绍

人类瞥了一眼图像，立即知道图像中的对象是什么，它们在哪里，以及它们如何相互作用。 人类视觉系统快速而准确，允许我们执行复杂的任务，例如驾驶时几乎不用太专注。 用于物体检测的快速，准确的算法将允许计算机在没有专用传感器的情况下驾驶汽车，使辅助设备能够向人类用户传达实时场景信息，并释放通用响应机器人系统的潜力。

当前的检测系统将分类器重新用于每个对象的检测。为了检测对象，这些系统为该对象采用分类器并在不同位置对其进行评估并在测试图像中进行缩放。 像可变形零件模型（DPM）这样的系统使用滑动窗口方法，其中分类器在整个图像上以均匀间隔的位置运行[10]。

最近的方法如 R-CNN 使用区域提案

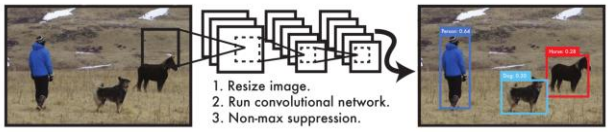


图 1：YOLO 检测系统。 使用 YOLO 处理图像简单明了。 我们的系统（1）将输入图像的大小调整为 448 × 448，（2）在图像上运行单个卷积网络，以及（3）通过模型的置信度对得到的检测进行阈值处理。

方法首先在图像中生成潜在边界框然后在这些建议框上运行分类器。分类后，”后处理”用于细化边界框，消除重复检测，并根据场景中的其他对象重新排列框[13]。这些复杂的过程很慢且难以优化，因为每个单独的组件必须单独训练。

我们将对象检测重新定义为单个回归问题，直接从图像像素到边界框坐标和类概率。使用我们的系统，您只需在图像上查看一次（YOLO）即可预测出现的对象和位置。

YOLO 简单易懂：参见图 1。单个卷积网络同时预测多个边界框和这些框的类概率。 YOLO 训练全图像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 非常快。由于我们将检测框架作为回归问题，因此我们不需要复杂的过程。我们只是在测试时在新图像上运行我们的神经网络来预测检测框和类别。我们的基础网络以每秒 45 帧的速度运行，Titan X GPU 上没有批处理，快速版本的运行速度超过 150 fps。这意味着我们可以实时处理流式视频，延迟时间少于 25 毫秒。此外，YOLO 的平均精度是其他实时系统的两倍多。有关我们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 在全局范围内对图像有所了解当

做出预测的时候。与滑动窗口和基于区域提议的技术不同，YOLO 在训练和测试时段内查看整个图像，因此它隐式编码有关类及其模式的上下文信息。Fast R-CNN 是一种顶级检测方法[14]，但它也会错误地为图像中的背景补丁分类，因为它无法看到更大的上下文。与 Fast R-CNN 相比，YOLO 的背景错误数量不到一半。

第三，YOLO 学习了对象的一般化表示。在对自然图像进行训练并对艺术作品进行测试时，YOLO 的表现优于 DPM 和 R-CNN 等顶级检测方法。由于 YOLO 具有高度可通用性，因此在应用于新域或意外输入时不太可能发生故障。

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，但它很难精确地定位某些物体，特别是小物体。我们在实验中进一步研究了这些平衡。

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

我们将对象检测的单独组件统一到单个神经网络中。我们的网络使用整个图像中的特征来预测每个边界框。它还可以同时预测所有类中的所有边界框。这意味着我们的网络全面了解整个图像和图像中的所有对象。 YOLO 设计支持端到端训练和实时的速度，同时保持较高的平均精度。

我们的系统将输入图像分成 S × S 网格。如果对象的中心落入网格单元格中，则该网格单元格负责检测该对象。

每个网格单元预测这些框的 B 边界框和置信度分数。这些置信度分数反映了模型对框中包含对象的可信度。

它预测盒子的准确程度也是如此准确。正式地，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对象，则置信度分数应为零。否则，我们希望置信度得分等于预测的盒子和真实标记之间的交并比（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的中心。相对于整个图像预测宽度和高度。最后，置信度预测表示预测框与任何真实标记框之间的 IOU。

每个网格单元还预测 C 条件类概率，$\Pr(\text{Class}_i|\text{Object})$。这些概率适用于包含对象的网格单元。我们只是预测每个网格单元的一组类概率，不管方框 B 的数量。

在测试时，我们将条件类概率与单个盒子置信度预测相乘，

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

这给了我们每个盒子的特定类别的置信度分数。 这些分数编码该类出现在框中的概率以及预测框与该对象的重合程度。

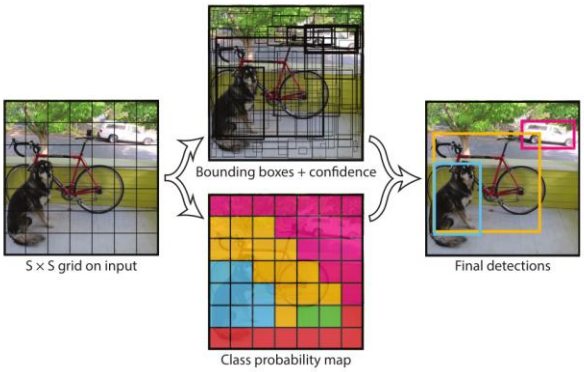


图 2：模型。 我们的系统模型检测作为回归问题。 它将图像划分为 S × S 网格，并为每个网格单元预测 B 边界框，这些框的置信度和 C 类概率。 这些预测被编码为 S×S×(B * 5 + C)张量。

为了评估 POLCAL VOC 上的 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 7 × 7 × 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 检测数据集上进行评估[9]。 网络的初始卷积层从图像中提取特征，而完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[34]。 我们的网络有 24 个卷积层，后面是 2 个完全连接层。 除了 GoogLeNet 使用的初始模块，我们只使用 11 个简化层，然后使用 3 × 3 的卷积层，类似于 Lin 等[22]。完整的网络如图 3 所示。

我们还训练了一个快速版的 YOLO，旨在突破快速物体检测的界限。 快速 YOLO 使用具有较少卷积层（9 而不是 24）的神经网络，并且在这些层中使用较少的滤波器。 除了网络的大小，YOLO 和 Fast YOLO 之间的所有训练和测试参数都是相同的。

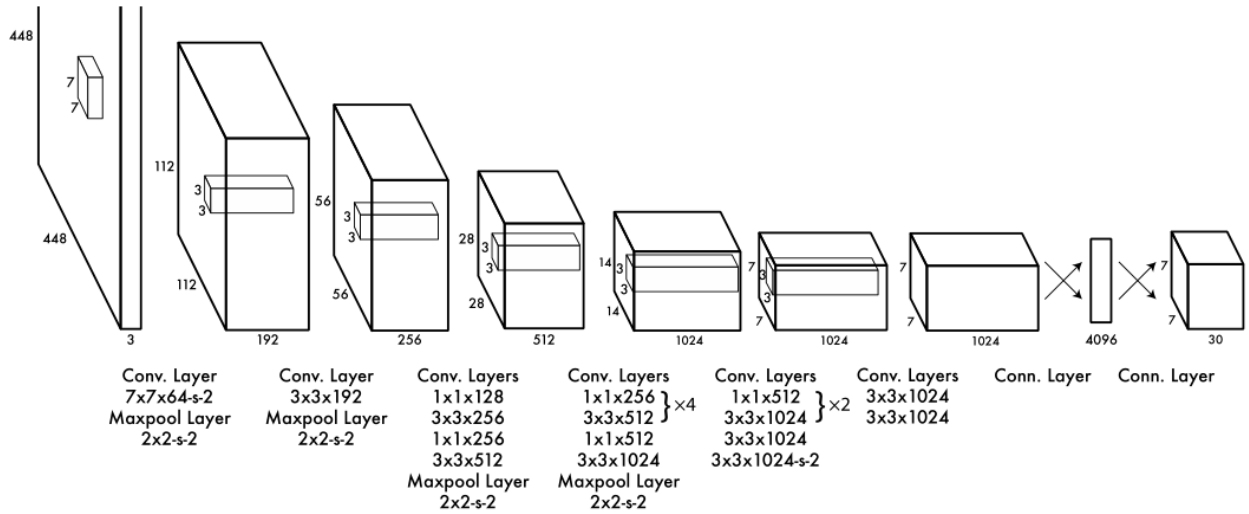


图 3：架构。 我们的检测网络有 24 个卷积层，后面是 2 个完全连接层。 交替的 1 × 1 卷积层减少了前面层的特征空间。 我们在 ImageNet 分类任务上以一半的分辨率（224 × 224 输入图像）预先训练卷积层，然后将分辨率加倍以进行检测。

我们网络的最终输出是 7 × 7 × 30 的张量预测结果。

2.2. 训练

我们在 ImageNet 1000 级竞赛数据集[30]上预先训练我们的卷积层。对于预训练，我们使用图 3 中的前 20 个卷积层，接着是平均池化层和完全连接层。我们训练这个网络大约一周，并在 ImageNet 2012 验证集上实现 88%的单一作物 top-5 精度，与 Caffe Zoo 模型中的 GoogLeNet 相当[24]。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。任等人表明将卷积和连接的层叠加到预训练网络可以提高性能[29]。按照他们的例子，我们添加了四个卷积层和两个完全连接层，随机初始化权重。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了类概率和边界框坐标。我们将边界框宽度和高度标准化为图像宽度和高度，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，因此它们也在 0 和 1 之间。

我们对最终层使用线性激活函数，所有其他层使用以下 Leaky 修正线性激活函数：

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

我们优化了模型输出中的求和平方误差。

我们使用求和平方误差，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。它可以使分类错误同等地定位，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对象。这将这些单元格的“置信度”分数推向零，通常会压制包含对象的单元格的分数渐变。这可能导致模型不稳定，导致训练在早期出现分歧。

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的置信预测损失。我们使用两个参数 λ_{coord} 和 λ_{noobj} 来实现这一目标。我们设置 $\lambda_{\text{coord}} = 5$ 和 $\lambda_{\text{noobj}} = .5$。

求和平方误差也同样对大盒子和小盒子中的错误进行加权。我们的误差度量应该反映出大盒子中的小偏移的影响小于小盒子。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，我们只希望一个边界框预测器负责所有对象。我们根据哪个预测具有最高当前 IOU 和基础事实，将一个预测器指定为“负责”以预测对象。这导致边界框预测变量之间的专门化。每个预测变量都能更好地预测某些大小，宽高比或对象类别，从而提高整体召回率。

在训练期间，我们优化以下多部分

损失函数:

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

其中$\mathbb{1}_i^{\text{obj}}$表示对象是否出现在单元格 i 中，并且$\mathbb{1}_i^{\text{obj}}$表示单元格 i 中的第 j 个边界框预测符对该预测是“负责任的”。

请注意，如果对象存在于该网格单元中，则损失函数仅惩罚分类错误（因此前面讨论过条件类概率）。如果该预测器对真实标签框“负责”（即该网格单元中具有任何预测器的最高 IOU），它也只会限制边界框坐标误差。

我们在 PASCAL VOC 2007 和 2012 的训练和验证数据集上训练了大约 135 个 epoch（迭代）的网络。在 VOC 2012 数据集上测试时，我们还把 VOC 2007 的测试集加入用于训练。在整个训练过程中，我们使用的批量大小为 64，动量为 0.9，衰减率为 0.0005。

我们的学习率计划如下：对于第一个时期，我们将学习率从10^{-3}缓慢提高到10^{-2}。如果我们从高学习率开始，我们的模型通常由于不稳定的梯度而发散。我们继续训练10^{-2}的 75 个时期，然后10^{-3}的 30 个时期，最后10^{-4}的 30 个时期。

为避免过拟合，我们使用了 DropOut 和大量数据扩增的方法。在第一个连接层之后，速率= .5 的 DropOut 层阻止了层之间的共适应[18]。对于数据增强，我们引入了高达原始图像大小的 20%的随机缩放和转置。我们还在 HSV 颜色空间中随机调整图像的曝光和饱和度达 1.5。

2.3. 推论

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 98 个边界框和每个框的类概率。YOLO 在测试时非常快，因为它只需要一个网络评估，不像基于分类器的方法。

网格设计在边界框预测中强制实施空间多样性。 通常很清楚一个对象落入哪个网格单元，并且网络仅为每个对象预测一个框。 但是，附近有一些大型物体或物体靠近多个单元格的边界可以被多个单元同时定位到。非最大值抑制可用于修复这些多个检测的问题。虽然对于 R-CNN 或 DPM 的性能并不重要，但非最大值抑制在 mAP 中增加 2-3%。

2.4. YOLO 的局限性

YOLO 对边界框预测施加了强烈的空间约束，因为每个网格单元只预测两个框，并且只能有一个类。这个空间约束限制了我们的模型可以预先指定的附近物体的数量。我们的模型与群体中出现的小物体斗争，例如成群的鸟类。

由于我们的模型学习从数据中预测边界框，因此很难在新的或不寻常的宽高比或配置中推广到对象。我们的模型还使用相对粗糙的特征来预测边界框，因为我们的体系结构具有来自输入图像的多个下采样层。

最后，当我们基于一个损失函数训练一个近似符合检测性能的模型时，我们的损失函数对待小边界框和大边界框中误差是相同的。大盒子中的小错误通常是良性的，但小盒子中的小错误对 IOU 的影响要大得多。我们的主要错误来源是定位上的错误。

3.与其他检测系统的比较

对象检测是计算机视觉中的核心问题。检测流水线通常首先从输入图像中提取一组稳健特征（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]用于识别特征空间中的对象。这些分类器或定位器可以在整个图像中以滑动窗口方式运行，也可以在图像中的某些区域子集上运行[35,15,39]。我们将 YOLO 检测系统与几个顶级检测框架进行了比较，突出了主要的相似点和不同点。

可变形零件模型。可变形零件模型（DPM）使用滑动窗口方法进行物体检测[10]。 DPM 使用不相交的过程来提取静态特征，对区域进行分类，预测高分区域的边界框等。我们的系统用单个卷积神经网络替换所有这些不同的部分。网络同时执行特征提取，边界框预测，非最大值抑制和上下文推理。网络不是静态功能，而是在线训练功能并针对检测任务对其进行优化。我们的统一架构使得模型比 DPM 更快，更准确。

R-CNN。 R-CNN 及其变体使用区域提议而不是滑动窗口来查找图像中的对象。可选择性

搜索[35]生成了潜在的边界框，卷积网络提取特征，SVM 对框进行评分，线性模型调整边界框，非最大值抑制消除重复检测。这个复杂过程的每个阶段必须独立精确调整，导致系统非常慢，在测试时每个图像需要超过 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。但是，我们的系统对网格单元提议设置了空间约束，这有助于减轻同一对象的多次检测。我们的系统还提出了更少的边界框，每个图像只有 98 个，而选择性搜索只有 2000 个。最后，我们的系统将这些单独的组件组合成一个联合优化的模型。

其他快速检测器。Fast 和 Faster R-CNN 专注于加速 R-CNN 框架，通过共享计算和使用神经网络来提出区域而不是选择性搜索[14] [28]。虽然它们提供了比 R-CNN 更快的速度和精度，但两者仍然没有达到实时性能。

许多研究工作都集中在加速 DPM 过程 [31] [38] [5]。它们加速 HOG 计算，使用级联，并将计算推送到 GPU。但是，只有 30Hz 的 DPM [31]才算实际上达到了实时运行。

YOLO 不是试图优化大型检测过程的各个组件，而是完全抛出复杂过程并且设计快速通道。

单个类（如面部或人）的探测器可以高度优化，因为它们必须处理更少的变化[37]。 YOLO 是一种通用探测器，可以学习同时探测各种物体。

Deep MultiBox。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 还可以通过用单个类预测替换置信度预测来执行单个对象检测。但是，Multi-Box 无法执行常规对象检测，并且仍然只是更大的检测过程中的一部分，需要进一步的图像补丁分类。 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 等人训练卷积神经网络以执行定位并调整定位器以执行检测[32]。 OverFeat 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 优化了定位，而不是检测性能。与 DPM 一样，定位程序在进行预测时仅查看定位信息。 OverFeat 不能推断全局背景，因此需要进行大量的后处理以产生相干检测。

MultiGrasp。我们的工作在设计上类似于工作 Redmon 等人的检测[27]。我们对边界框预测的网格方法基于 MultiGrasp 系统进行回归以抓取到。然而，抓取检测是比对象检测简单得多的任务。 MultiGrasp 只需要为包含一个对象的图像预测单个可抓取区域。它不必估计对象的大小，位置或边界或预测它的类，只找到适合抓取的区域。 YOLO 预测图像中多个类的多个对象边界框和类概率。

4.实验

首先，我们将 PASCAL VOC 2007 上的 YOLO 与其他实时检测系统进行比较。为了理解 YOLO 和 R-CNN 变体之间的差异，我们探讨了 YOLO 和 Fast R-CNN 对 VOC 2007 的误差，这是性能最高的版本之一的 R-CNN [14]。基于不同的错误配置文件，我们显示 YOLO 可用于重新调整 Fast R-CNN 检测并减少背景误报的错误，从而显著提升性能。我们还介绍了 VOC 2012 上的结果，并将 mAP 与当前最先进的方法进行了比较。最后，我们展示了 YOLO 比两个艺术品数据集上的其他检测器能更好地推广到新域。

4.1.与其他实时系统的比较

目标检测方面的许多研究工作都集中在快速制定标准检测过程上 [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 GPU 实现进行比较。虽然其他工作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，以检查对象检测系统中可用的准确性 - 性能权衡。

Fast YOLO 是 PASCAL 上最快的物体检测方法;据我们所知，它是现存最快的物体探测器。使用 52.7%的 mAP，它比以前的实时检测工作精确度高两倍。 YOLO 将 mAP 推至 63.4%，同时仍保持实时性能。

我们还使用 VGG-16 训练 YOLO。这个模型比 YOLO 更准确但也明显更慢。与其他依赖 VGG-16 的检测系统进行比较是有用的，但由于它比实时慢，本文的其余部分主要关注我们的快速模型。

最快的 DPM 有效地加速了 DPM 而没有牺牲太多的 mAP，但它仍然错过了实时性能 2 倍[38]。与神经网络方法相比，它还受到 DPM 检测精度相对较低的限制。

R-CNN 减去 Region 用静态边界框提议取代选择性搜索[20]。虽然比 R-CNN 快得多，

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：PASCAL VOC 2007 上的实时系统。比较快速检测器的性能和速度。 Fast YOLO 是 PASCAL VOC 检测中记录最快的检测器，其精度仍然是其他任何实时检测器的两倍。 YOLO 比快速版本更精确约 10 mAP，同时仍然远高于实时速度。

但它仍然没有实时性，并且由于没有好的提议而受到很大的准确性影响。

Fast R-CNN 加速了 R-CNN 的分类阶段，但仍然依赖于选择性搜索，每个图像大约需要 2 秒才能生成边界框提议。因此它具有高 mAP 但是在 0.5 fps 时它仍然远非实时。

最近的 Fast R-CNN 用神经网络取代选择性搜索以提出边界框，类似于 Szegedy 等人 [8]。在我们的测试中，他们最精确的模型达到了 7 fps，而更小、更不精确的模型以 18 fps 运行。 Faster R-CNN 的 VGG-16 版本高出 10 mAP，但也比 YOLO 慢 6 倍。 Zeiler-Fergus 的 Faster R-CNN 仅比 YOLO 慢 2.5 倍，但也不太准确。

4.2. VOC 2007 错误分析

为了进一步研究 YOLO 和最先进的探测器之间的差异，我们将详细分析 VOC 2007 的结果。我们将 YOLO 与 Fast R-CNN 进行比较，因为 Fast R-CNN 是性能最高的探测器之一。 PASCAL 和它的检测是公开可用的。

我们使用 Hoiem 等人的方法和工具[19]对于测试时的每个类别，我们查看该类别的前 N 个预测。 每个预测都是正确的或者能根据错误类型进行分类：

- 正确：正确的类并且 IOU > .5
- 定位：正确的类，.1 < IOU < .5
- 类似：类是相似的，IOU > .1

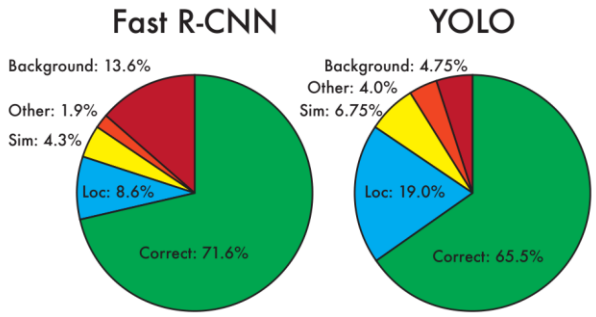


图 4：错误分析：Fast R-CNN 与 YOLO。这些图表显示了各种类别的前 N 个检测中的定位和背景错误的百分比（N = 类别中的 # 个对象）。

- 其他: 类是错的, IOU > .1
- 背景: IOU < .1 对任何物体

图 4 显示了所有 20 个类中平均每种错误类型的细分。

YOLO 努力正确地定位对象。 与所有其他来源相结合，定位错误占 YOLO 错误的比重更多。 Fast R-CNN 使定位错误少得多，但背景错误要多得多。 其中 13.6%的顶级检测方法的结果是误报，不包含任何对象。 与 YOLO 相比，Fast R-CNN 预测背景检测错误的可能性几乎高出 3 倍。

4.3. 结合 Fast R-CNN 与 YOLO

与 Fast R-CNN 相比，YOLO 的背景错误要少得多。 通过使用 YOLO 消除 Fast R-CNN 的背景检测，我们可以显著提升性能。 对于 R-CNN 预测的每个边界框，我们检查 YOLO 是否预测了类似的框。 如果确实如此，我们会根据 YOLO 预测的概率和两个框之间的重叠来推进该预测。

最佳的 Fast R-CNN 型号在 VOC 2007 测试装置上实现了 71.8%的 mAP。 当与 YOLO 结合使用时，它的

	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：VOC 2007 的模型组合实验。我们研究了将各种模型与 Fast R-CNN 的最佳版本组合的效果。 其他版本的 Fast R-CNN 仅提供小的优势，而 YOLO 提供显著的性能提升。

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 [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）公共排行榜进行了比较。显示了各种检测方法的平均精度和每级平均精度。 YOLO 是唯一的实时探测器。Fast R-CNN+YOLO 是第四高得分方法，比 Fast R-CNN 提高 2.3%。

mAP 提升 3.2%至 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。这些结合使 mAP 的小幅增加在.3 和.6%之间，详见表 2。

YOLO 的推动不仅仅是模型集成的副产品，因为组合不同版本的 Fast R-CNN 几乎没有什么好处。相反，正是因为 YOLO 在测试时犯了不同类型的错误，才能提升 Fast R-CNN 的性能。

不幸的是，这种组合并没有受益于 YOLO 的速度，因为我们单独运行每个模型然后结合结果。然而，由于 YOLO 如此之快，与 Fast R-CNN 相比，它不会增加任何显著的计算时间。

4.4. VOC 2012 结果

在 VOC 2012 测试集中，YOLO 的分数为 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，参见表 3。与最接近的竞争对手相比，我们的系统与小物体相比更加困难。在瓶子，绵羊和电视/显示器等类别中，YOLO 比 R-CNN 或 Feature Edit 低 8-10%。然而，在其他类别如猫和火车上，YOLO 实现了更高的性能。

我们的 Fast R-CNN + YOLO 组合模型是性能最高的检测方法之一。Fast R-CNN 从与 YOLO 的组合中获得了 2.3%的提升，使其在公共排行榜上增加了 5 个点。

4.5. 泛化性：艺术作品中的人物检测

用于对象检测的学术数据集从同一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和测试数据可能与系统所见的不同[3]。我们将 YOLO 与 Picasso 数据集[12]和人物艺术数据集[3]上的其他检测系统进行比较，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 与其他检测方法之间的比较性能。作为参考，我们在 VOC 2007 检测人类别（person）的 AP 值，其中所有模型仅针对 VOC 2007 数据进行训练。在 Picasso 模型上接受了 VOC 2012 的训练，而在 People-Art 上，他们接受了 VOC 2010 的训练。

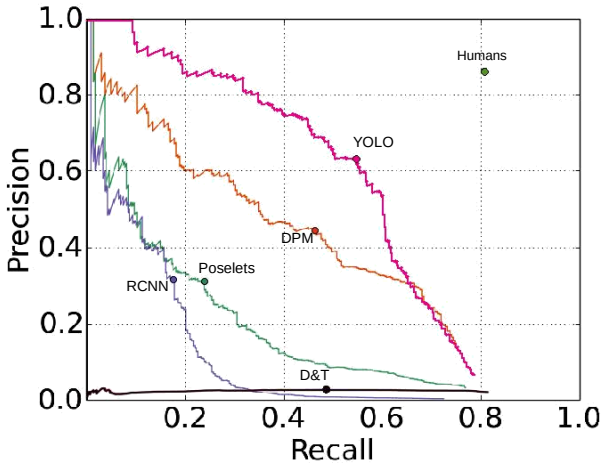
R-CNN 在 VOC 2007 上具有很高的 AP。然而，当应用于艺术品时，R-CNN 显著下降。 R-CNN 使用选择性搜索来调整自然图像的边界框提议。 R-CNN 中的分类器步骤只能看到小区域，需要很好的提议。

当应用于艺术品时，DPM 可以很好地维护其 AP。之前的工作认为 DPM 表现良好，因为它具有强大的对象形状和布局的空间模型。尽管 DPM 不像 R-CNN 那样降低比较快，但它直接从较低的 AP 开始。

YOLO 在 VOC 2007 上具有良好的性能，并且当应用于艺术品时，其 AP 降低值比其他方法更少。与 DPM 一样，YOLO 模拟对象的大小和形状，以及对象和对象通常出现的位置之间的关系。图像和自然图像在像素级别上是非常不同的，但它们在对象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 是一款快速，精确的物体探测器，非常适合计算机视觉应用。 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，



（a）Picasso 数据集精确回忆曲线。

	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）VOC 2007，Picasso 和 People-Art 数据集的定量结果。 Picasso 数据集评估 AP 和最佳 F1 得分。

图 5：Picasso 和 People-Art 数据集的泛化结果。

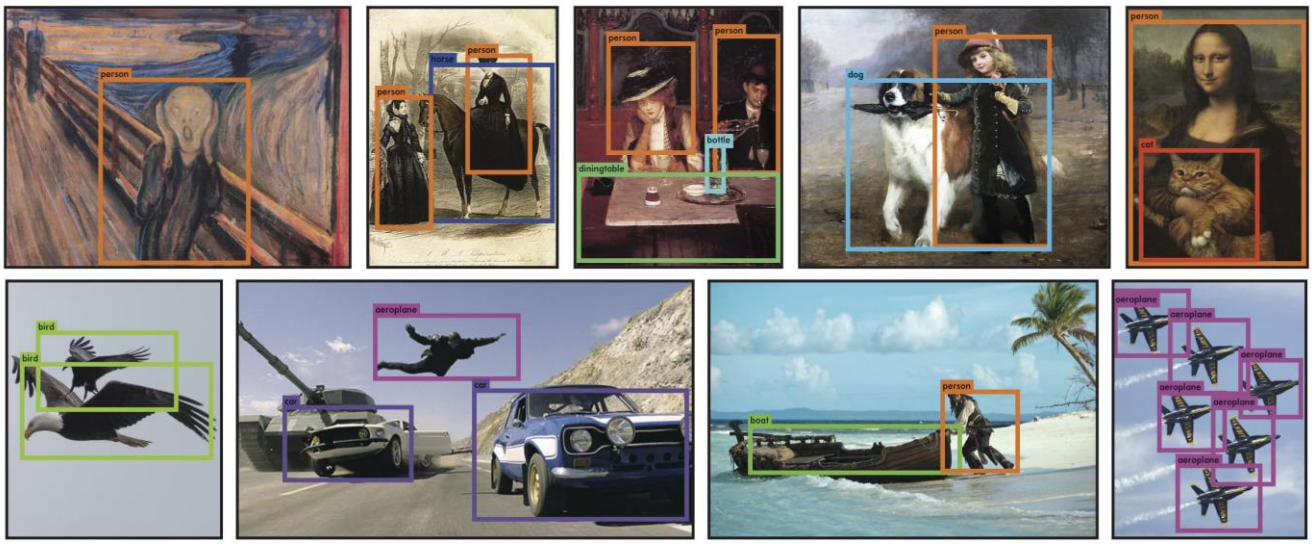


图 6：定性结果。 YOLO 运行样本艺术作品和来自互联网的自然图像。 虽然它确实认为一个人是飞机，但它大多是准确的。

包括从相机获取图像和显示检测的时间。

由此产生的系统是有互动性和吸引人的。 虽然 YOLO 可以单独处理图像，但当连接到网络摄像头时，它的功能就像跟踪系统一样，可以在物体移动和外观变化时检测对象。 可在我们的项目网站上找到系统演示和源代码：http://pjreddie.com/yolo/.

6. 结论

我们介绍 YOLO，一个用于对象检测的统一模型。 我们的模型构造简单，可以直接训练完整的图像。 与基于分类器的方法不同，YOLO 针对与检测性能直接对应的损失函数进行训练，并且整个模型被联合训练。

Fast YOLO 是文献中最快的通用对象检测器，YOLO 推动了最先进的实时对象检测。 YOLO 还可以很好地推广到新域，使其成为依赖快速，强大的对象检测的应用程序的理想选择。

致谢：ONR N00014-13-1-0720，NSF IIS-1338054 和艾伦杰出研究员奖部分支持这项工作。

参考文献

[1] M. B. Blaschko and C. H. Lampert. Learning to localize ob-jects 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 cross-depiction problem: Computer vision algorithms for recog-nising 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 Confer-ence 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 acti-vation 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 Confer-ence 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 ob-ject 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. Ra-manan. 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. 7

[12] S. Ginosar, D. Haas, T. Brown, and J. Malik. Detecting peo-ple 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 fea-ture 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 segmenta-tion 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 pre-venting 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 fea-tures 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. Ac-cessed: 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: To-wards real-time object detection with region proposal net-works. 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, localiza-tion 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. Inter-national 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 Pat-tern Recognition (CVPR), 2014 IEEE Conference on, pages 2497–2504. IEEE, 2014. 5, 6

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