　附表2-1（通用）

南京邮电大学

毕业设计(论文)外文资料翻译

|  |  |
| --- | --- |
| 学 院 | 计算机学院、软件学院、网络空间安全学院 |
| 专　　业 | 数据科学与大数据技术 |
| 学生姓名 | 张笑阳 |
| 班级学号 | B21032229 |
| 外文出处 | Proceedings of the IEEE conference on computer vision and pattern recognition. |

附件：1.外文资料翻译译文；2.外文原文

|  |
| --- |
| 指导教师评价：  1．翻译内容与课题的结合度： ☑ 优 □ 良 □ 中 □ 差  2．翻译内容的准确、流畅： ☑ 优 □ 良 □ 中 □ 差  3．专业词汇翻译的准确性： □ 优 ☑ 良 □ 中 □ 差  4．翻译字符数是否符合规定要求：□ 符合 □ 不符合  　　　　　　　　　　　　　　　　　　　　　　　指导教师签名：  　　　　　　　　　　　　　　　　　　　　　　　　　　 2025年4月14日 |

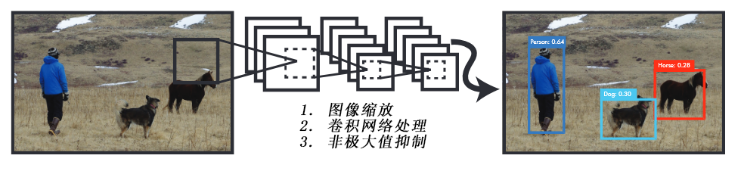
**YOLO：统一的实时目标检测**

摘要——本文介绍了一种新的目标检测方法YOLO。在这之前关于对象检测的工作都是重新调整了分类器的用途使之用于执行检测。取而代之的是，我们将对象检测任务表述为检测框和相关类概率的回归问题。单个神经网络在一次评估中直接从完整图像预测边界框和类概率。由于整个检测流水线是单个网络，引测可以直接对检测性能进行端到端优化。

我们的统一架构速度极快，YOLO基本模型以每秒45帧的速度实时处理图像。小网络版本的Fast YOLO每秒处理155帧，同时其mAP（平均精确度）达到其他实时检测器的两倍。与最先进的检测系统相比，YOLO的定位误差更多，但在不存在的情况下给出错误预测的概率却小得多。最后，YOLO学习的是目标的一般特征。当从自然图像学到的模型用于其它领域如艺术画作时，它的表现都优于包括DPM和R-CNN在内的其它检测方法。

1．前言

人们只需要瞄一眼图像，就能立刻知道图像中的物体是什么，它们在哪里以及它们的空间关系。人类的视觉系统是快速和准确的，使得我们无意中就能够执行复杂的任务，例如驾驶。



**图表 1：YOLO检测系统。**用YOLO检测图像是非常简单且直观的。我们的系统第（1）步先将图片缩放为448×448的大小，第（2）步将其输入一个简单的卷积神经网络，第（3）步根据模型的设定阈值给出检测结果。

快速、准确的目标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，使辅助设备能够向人类传达实时的场景信息，并解锁通用、灵敏的机器人系统。

目前的检测系统通过重用分类器执行检测。为了检测目标，这些系统为该目标提供一个分类器，在测试图像的不同位置和不同尺度是对其进行评估。像deformable parts models（DPM，可变形部分模型）这样的系统使用滑动窗口方法，使分类器在整个图像上的均匀间隔的位置上运行[10]。

最新的方法，如R-CNN使用区域候选策略，首先在图像上生成可能的边界框，然后在这些框上运行分类器。在分类之后，执行用于细化边界框的后处理，消除重复的检测，并根据场景中的其他模板为边界框重新评分[13]。这些复杂的流程是很慢，很难优化的，因为每个独立的部件必须单独进行训练。

我们将目标检测看作是一个单一的回归问题，直接从图像像素得到边界框坐标和类别概率。使用我们的系统——You Only Look Once（YOLO），便能得到图像上的物体是什么和物体的具体位置。

YOLO的结构是前所未有的简单：见**图表 1**。它仅用单个卷积网络就能同时预测多个边界框和它们的类别概率。YOLO在整个图像上训练，并能直接优化检测性能。与传统的目标检测方法相比，这种统一的模型下面所列的一些优点。

第一，YOLO速度非常快。由于我们将检测视为回归问题，所以我们不需要复杂的流程。测试时，我们在一张新图像上简单的运行我们的神经网络来预测检测结果。在Titan X GPU上不做批处理的情况下，YOLO的基础版本以每秒45帧的速度运行，而快速版本运行速度超过150fps。这意味着我们可以在不到25毫秒的延迟内实时处理流媒体视频。此外，YOLO实现了其它实时系统两倍以上的平均精度。关于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：<https://goo.gl/bEs6Cj>。

第二，YOLO是在整个图像上进行推断的。与基于滑动窗口和候选框的技术不同，YOLO在训练期间和测试时都会顾及到整个图像，所以它隐式地包含了关于类的上下文信息以及它们的外观。Fast R-CNN是一种很好的检测方法[14]，但由于它看不到更大的上下文，会将背景块误检为目标。与Fast R-CNN相比，YOLO的背景误检数量少了一半。

第三，YOLO能学习到目标的泛化表征。把在自然图像上进行训练的模型，用在艺术画作进行测试时，YOLO大幅优于DPM和R-CNN等顶级的检测方法。由于YOLO具有高度泛化能力，因此在应用于新领域或碰到意外的输入时不太可能出故障。

YOLO在精度上仍然落后于目前最先进的检测系统。虽然它可以快速识别图像中的目标，但它在定位某些物体尤其是小的物体上精度不高。

我们在实验中会进一步探讨精度／时间的权衡。我们所有的训练和测试代码都是开源的，而且各种预训练模型也都可以下载。

2．统一检测

我们将目标检测的独立部分整合到单个神经网络中。我们的网络使用整个图像的特征来预测每个边界框。它还可以同时预测一张图像中的所有类别的所有边界框。这意味着我们的网络对整张图像和图像中的所有目标进行全局推理。YOLO设计可实现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的网格。如果目标的中心落入某个网格单元中，那么该网格单元就负责检测该目标。

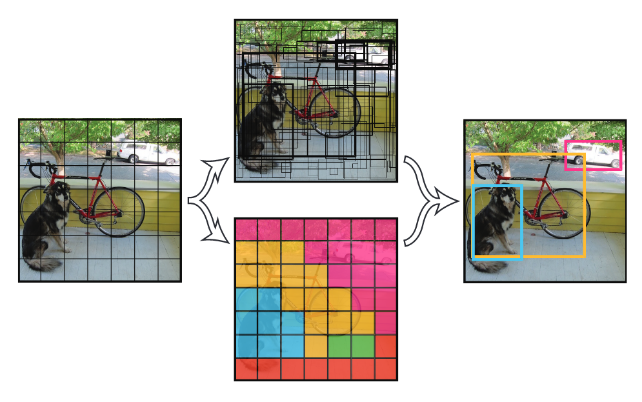
每个网格单元都会预测 B个 边界框和这些框的置信度分数。这些置信度分数反映了该模型对那个框内是否包含目标的置信度，以及它对自己的预测的准确度的估量。在形式上，我们将置信度定义为。如果该单元格中不存在目标，则置信度分数应为 0 。否则，我们希望置信度分数等于预测框与真实标签框之间的交并比（Intersection over Union，IOU）。

每个检测框由5个预测框组成：和置信度。坐标表示相对于栅格单元边界的长方体的中心，宽和高则是相对于整个图片预测的。最后，置信度预测值则是预测框与真值框之间的IOU。

每个网格单元还预测了C类的条件概率，即。这些概率是以包含目标的网格单元为条件的。我们只预测每个网格单元的一组类别概率，而不考虑框B的数量。

在测试时，我们将条件类概率和单个框的置信度预测相乘：

这给我们提供了每个框的特定类别的置信度分数。这些分数既表示了该类出现在框里的概率，也表示了预测的框与目标的框选的匹配程度。



**图表 2：模型。**我们的系统将检测建模为回归问题，它将图像分成均匀网格，同时预测边界框、这些框中的置信度和类别概率，这些预测被编码为S×S×(B×5+C)的张量。

我们使用PASCAL VOC数据集验证YOLO的运行效果，其中S=7，B=2。因为PASCAL VOC数据集有20个类别，故C=20。我们最终预测的是7×7×30张量。

2.1. 设计

我们将此模型作为卷积神经网络实现，并在PASCAL VOC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，而全连接层负责预测输出概率和坐标。

我们的网络架构受图像分类模型GoogLeNet的启发[33]。我们的网络有24个卷积层，后面是2个全连接层。我们只使用降维层，后面是卷积层，这与Lin等人[22]类似，而不是GoogLeNet使用的Inception模块。详细的网络结构请见**图表 3**。

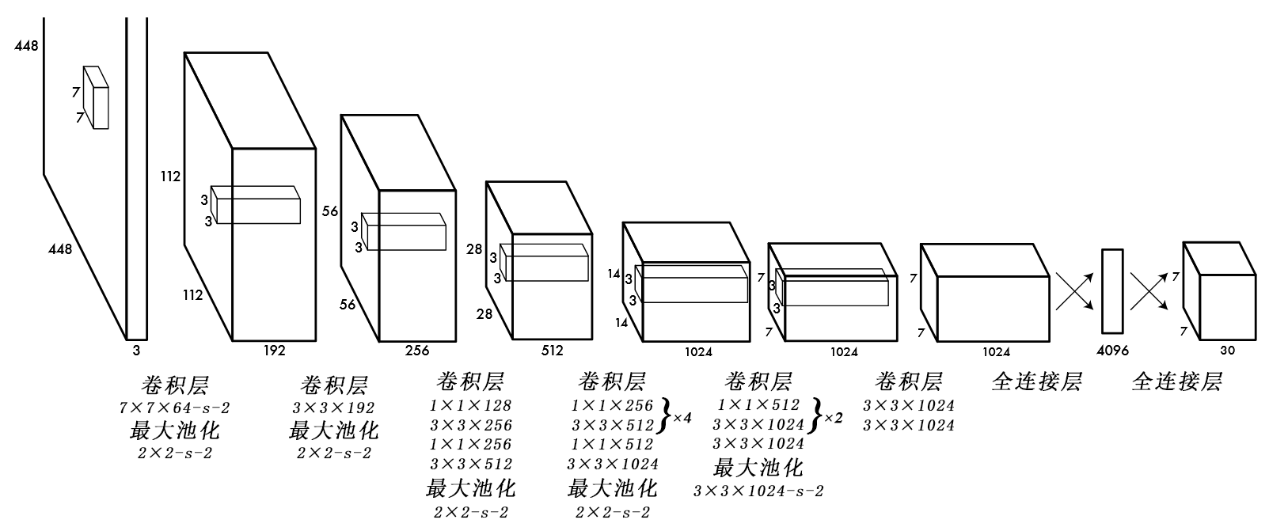
我们还训练了快速版本的YOLO（Fast YOLO），旨在推动快速目标检测的界限。快速YOLO使用较少卷积层（9层而不是24层）的神经网络，在这些层中使用较少的卷积核。除了网络规模之外，基本版YOLO和快速YOLO的所有训练和测试参数都是相同的。

我们网络的最终输出是的预测张量。

2.2. 训练

我们在ImageNet的1000类竞赛数据集[29]上预训练我们的卷积层。对于预训练，我们使用**图表 3**中的前20个卷积层，接着是平均池化层和全连接层。我们对这个网络进行了大约一周的训练，并且在ImageNet 2012验证集上获得了单一裁剪图像88%的top-5准确率，与Caffe模型池中的GoogLeNet模型相当。我们使用Darknet框架进行所有的训练和推断[24]。

然后我们转换模型来执行检测训练。Ren等人表明，预训练网络中增加卷积层和连接层可以提高性能[28]。按照他们的方法，我们添加了四个卷积层和两个全连接层，这些层的权重都用随机值初始化。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改为。



**图表 3：网络架构。**我们的检测网络有24个卷积层，之后是2个全连接层。交替的卷积层减少了来自先前层的特征空间。我们以半分辨率（输入尺寸为）对ImageNet分类任务中的卷积层进行预训练，然后以2倍分辨率用以预测。

模型的最后一层预测类概率和边界框坐标。我们通过图像宽度和高度来规范边界框的宽度和高度，使它们落在0和1之间。我们将边界框和坐标参数化为特定网格单元位置的偏移量，所以它们的值被限定在在0和1之间。

模型的最后一层使用线性激活函数，而所有其它的层使用下面的Leaky-ReLU：

我们对模型输出的平方和误差进行优化。我们选择使用平方和误差，是因为它易于优化，但是它并不完全符合最大化平均精度的目标。它给分类误差与定位误差的权重是一样的，这点可能并不理想。另外，每个图像都有很多网格单元并没有包含任何目标，这将这些单元格的“置信度”分数推向零，通常压制了包含目标的单元格的梯度。这可能导致模型不稳定，从而导致训练在早期就发散。

为了弥补平方和误差的缺陷，我们增加了边界框坐标预测的损失，并减少了不包含目标的框的置信度预测的损失。 我们使用两个参数，和，其中，。

平方和误差对大框和小框的误差权衡是一样的，而我们的错误指标应该要体现出，大框的小偏差的重要性不如小框的小偏差的重要性。为了部分解决这个问题，我们直接预测边界框宽度和高度的平方根，而不是宽度和高度。

YOLO为每个网格单元预测多个边界框。在训练时，每个目标我们只需要一个边界框预测器来负责。若某预测器的预测值与目标的实际值的IOU值最高，则这个预测器被指定为“负责”预测该目标。这导致边界框预测器的专门化。每个预测器可以更好地预测特定大小，方向角，或目标的类别，从而改善整体召回率。

在训练期间，我们优化以下多部分损失函数：

其中，表示对象是否出现在了单元中，而表示单元中的第号检测器是否“负责”此次预测。

注意，如果目标存在于该网格单元中（前面讨论的条件类别概率），则损失函数仅惩罚分类错误。如果预测器“负责”实际边界框（即该网格单元中具有最高IOU的预测器），则它也仅惩罚边界框坐标错误。

我们用Pascal VOC 2007和2012的训练集和验证数据集进行了大约 135个epoch 的网络训练。因为我们仅在Pascal VOC 2012上进行测试，所以我们的训练集里包含了Pascal VOC 2007的测试数据。在整个训练过程中，我们使用：batch size为64，momentum为0.9，decay为0.0005。

我们的学习率计划如下：在第一个epoch中，我们将学习率从慢慢地提高到。如果从大的学习率开始训练，我们的模型通常会由于不稳定的梯度而发散。我们继续以进行75个周期的训练，然后以进行30个周期的训练，最后以进行30个周期的训练。

为避免过拟合，我们使用了Dropout和大量的数据增强。 在第一个连接层之后的dropout层的丢弃率设置为0.5，以防止层之间的相互适应[18]。 对于数据增强，我们引入高达20％的原始图像大小的随机缩放和平移。我们还在 HSV 色彩空间中以高达 1.5 的因子随机调整图像的曝光度和饱和度。

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，每张图像上网络预测 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。

网格设计强化了边界框预测中的空间多样性。通常一个目标落在哪一个网格单元中是很明显的，而网络只能为每个目标预测一个边界框。然而，一些大的目标或接近多个网格单元的边界的目标能被多个网格单元定位。非极大值抑制(Non-maximal suppression，NMS)可以用来修正这些多重检测。非最大抑制对于YOLO的性能的影响不像对于R-CNN或DPM那样重要，但也能增加2−3%的mAP。

2.4. YOLO的局限性

由于每个格网单元只能预测两个框，并且只能有一个类，因此YOLO对边界框预测施加了很强的空间约束。这个空间约束限制了我们的模型可以预测的邻近目标的数量。我们的模型难以预测群组中出现的小物体（比如鸟群）。

由于我们的模型学习是从数据中预测边界框，因此它很难泛化到新的、不常见的长宽比或配置的目标。我们的模型也使用相对较粗糙的特征来预测边界框，因为输入图像在我们的架构中历经了多个下采样层。

最后，我们的训练基于一个逼近检测性能的损失函数，这个损失函数无差别地处理小边界框与大边界框的误差。大边界框的小误差通常是无关要紧的，但小边界框的小误差对IOU的影响要大得多。我们的主要错误来自于不正确的定位。

3．与其他目标检测算法的比较

目标检测是计算机视觉中的核心问题。检测流程通常是首先从输入图像上提取一组鲁棒特征（Haar [25]，SIFT [23]，HOG [4]，卷积特征 [6]）。然后，分类器[35][21][13][10]或定位器[1][31]被用来识别特征空间中的目标。这些分类器或定位器或在整个图像上或在图像中的一些子区域上以滑动窗口的方式运行[34][15][38]。我们将YOLO检测系统与几种顶级检测框架进行比较，突出了关键的相似性和差异性。

**Deformable parts models**。可变形部分模型（DPM）使用滑动窗口方法进行目标检测[10]。DPM使用不相交的流程来提取静态特征，对区域进行分类，预测高评分区域的边界框等。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。网络同时进行特征提取，边界框预测，非极大值抑制和上下文推理。网络的特征feature是在线训练出来的而不是静态，因此可以根据特定的检测任务进行优化。我们的统一架构比DPM更快，更准确。

**R-CNN**。R-CNN及其变体使用区域候选而不是滑动窗口来查找图像中的目标。选择性搜索[34]生成潜在的边界框，卷积网络提取特征，SVM对框进行评分，线性模型调整边界框，非最大抑制消除重复检测。 这个复杂流水线的每个阶段都必须独立地进行精确调整，所得到的系统非常缓慢，在测试时间每个图像需要超过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。然而，我们的系统对网格单元的候选框施加空间限制，这有助于缓解对同一目标的多次检测的问题。 我们的系统还生成了更少的边界框，每张图像只有98个，而选择性搜索则有约2000个。最后，我们的系统将这些单独的组件组合成一个单一的、共同优化的模型。

**其它快速检测器**。 Fast R-CNN 和 Faster R-CNN 通过共享计算和使用神经网络替代选择性搜索[14][27]来提出候选区域来加速 R-CNN 框架。虽然它们提供了比 R-CNN 更快的速度和更高的准确度，但仍然不能达到实时性能。

许多研究工作集中在加快DPM流程上[37][30][5]。它们加速HOG计算，使用级联，并将计算推动到（多个）GPU上。但是，实际上只有30Hz的DPM可以实时运行。

YOLO并没有试图优化大型检测流程的单个组件，相反，而是完全抛弃了大型检测流程，并通过设计来提高速度。

像人脸或行人等单个类别的检测器可以高度优化，因为他们只需处理较少的多样性[36]。YOLO是一种通用的检测器，它可以同时检测多个目标。

**Deep MultiBox**。与R-CNN不同，Szegedy等人[8]训练一个卷积神经网络来预测感兴趣的区域(regions of interest，ROI)，而不是使用选择性搜索。 MultiBox还可以通过用单个类别预测替换置信度预测来执行单个目标检测。 但是，MultiBox无法执行一般的目标检测，并且仍然只是较大检测流水线中的一部分，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等人训练了一个卷积神经网络来执行定位，并使该定位器进行检测[31]。OverFeat高效地执行滑动窗口检测，但它仍然是一个不相交的系统(disjoint system)。OverFeat优化了定位功能，而不是检测性能。像DPM一样，定位器在进行预测时只能看到局部信息。OverFeat无法推断全局上下文，因此需要大量的后处理来产生连贯的检测。

**MultiGrasp**。我们的系统在设计上类似于Redmon等[26]的抓取检测。 我们的网格边界框预测方法基于MultiGrasp系统进行回归分析。 然而，抓取检测比物体检测要简单得多。 MultiGrasp只需要为包含一个目标的图像预测一个可抓取区域。 它不必估计目标的大小，位置或边界或预测它的类别，只需找到适合抓取的区域就可以了。 而YOLO则是预测图像中多个类的多个目标的边界框和类概率。

4．实验

首先，我们通过PASCAL VOC 2007数据集比较YOLO和其他实时检测系统。为了理解YOLO和R-CNN变体之间的差异，我们研究了YOLO和Fast R-CNN（R-CNN的最高性能版本之一[14]）在VOC 2007上产生的误差。基于不同的错误分布，我们显示YOLO可用于重新搜索Fast R-CNN的检测结果并减少背景误报导致的错误，从而显著提高 性能。我们还在VOC 2012上进行了测试，并将mAP与当前最先进的方法进行了比较。最后，我们在两个艺术画作数据集上证明了YOLO的泛化性能其他检测器更强。

4.1. 推论

目标检测方面的许多研究工作都集中在使标准的检测流程更快[5][37][30][14][17][27]。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DPM的GPU实现进行了比较，其在30Hz或100Hz下运行。虽然其它的算法没有达到实时性的标准，我们也比较了它们的mAP和速度的关系，从而探讨目标检测系统中精度和性能之间的权衡。

Fast YOLO是PASCAL上最快的目标检测方法；据我们所知，它是现有的最快的目标检测器。具有52.7%的mAP，实时检测的精度是以前的方法的两倍以上。普通版YOLO将mAP推到63.4%的同时保持了实时性能。

我们还使用VGG-16训练YOLO。 这个模型比普通版YOLO更精确，但也更慢。 它的作用是与依赖于VGG-16的其他检测系统进行比较，但由于它比实时更慢，所以本文的其他部分将重点放在我们更快的模型上。

最快的DPM可以在不牺牲太多mAP的情况下有效加速DPM，但仍然会将实时性能降低2倍[38]。与神经网络方法相比，DPM的检测精度相对较低，这也是限制它的原因。

R-CNN minus R用静态侯选边界框取代选择性搜索[20]。虽然速度比R-CNN更快，但它仍然无法实时，并且由于该方法无法找到好的边界框，准确性受到了严重影响。

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，每个图像需要大约2秒才能生成边界候选框。因此，它虽然具有较高的mAP，但的速度是0.5 fps，仍然远未达到实时。

|  |  |  |  |
| --- | --- | --- | --- |
| 实时检测器 | 数据集 | mAP | 帧数 |
| 100Hz DPM [30] | 2007 | 16 | 100 |
| 30Hz DPM [30] | 2007 | 26.1 | 30 |
| Fast YOLO | 2007+2012 | 52.7 | 155 |
| YOLO | 2007+2012 | 63.4 | 45 |
| 检测速度过慢 |  |  |  |
| Fastest DPM [37] | 2007 | 30.4 | 15 |
| R-CNN Minus R [20] | 2007 | 53.5 | 6 |
| Fast R-CNN [14] | 2007+2012 | 70 | 0.5 |
| Faster R-CNN VGG-16 [27] | 2007+2012 | 73.2 | 7 |
| Faster R-CNN ZF [27] | 2007+2012 | 62.1 | 18 |

**表格 1：各实时检测系统在PASCAL VOC 2007数据集上的运行结果。**比较了快速探测器的性能和速度。Fast YOLO 是 PASCAL VOC 检测有记录以来最快的检测器，其精确度是其他实时检测器的两倍。YOLO比快速版本精确10 mAP，且检测速度对于实时任务依然绰绰有余。

因此，Fast R-CNN虽然在0.5 fps的速度下有很高的检测精度，但该模型不适用于实时任务。

最近的Faster R-CNN用神经网络替代了选择性搜索来候选边界框，类似于Szegedy等人[8]的方法。在我们的测试中，他们最准确的模型达到了 7fps，而较小的、不太准确的模型以18 fps运行。 Faster R-CNN的VGG-16版本比YOLO高出10mAP，但比YOLO慢了6倍。 Zeiler-Fergus 版本的Faster R-CNN只比YOLO慢2.5倍，但也不如YOLO准确。

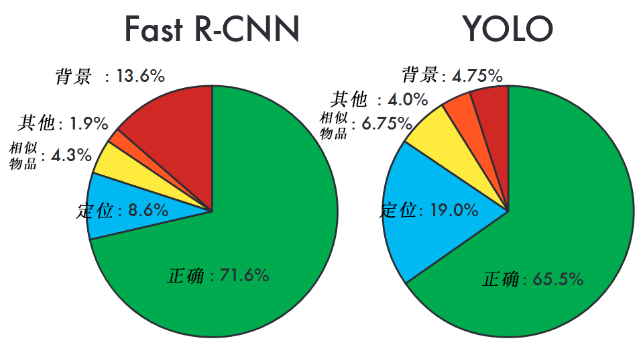
4.2. VOC 2007数据集误差分析

为了进一步研究YOLO和最先进的检测器之间的差异，我们详细分析了VOC 2007的分类结果。我们将YOLO与Fast R-CNN进行比较，因为Fast R-CNN是PASCAL上性能最高的检测器之一并且它的检测代码是开源的。

我们使用Hoiem等人的方法和工具[19]，对于测试的每个类别，我们查看该类别的前N个预测。每个预测都或是正确的，或是根据错误的类型进行分类：

* 正确：分类正确，
* 定位：分类正确，
* 相似物品：类似的分类，
* 其他：分类错误，
* 背景：对于的任何物体

**图表 4**展示了在20个分类上各种误差的详细数据。



**图表 4：误差分析：Fast R-CNN与YOLO。**这些图显示了各种类别的前N个预测中定位错误和背景错误的百分比（N = #表示目标在那个类别中）。

YOLO难以正确地定位目标，因此定位错误比YOLO的所有其他错误总和都要多。Fast R-CNN定位错误更少，但把背景误认成目标的错误比较多。它的首选检测结果中有13.6％是不包含任何目标的误报。 Fast R-CNN把背景误认成目标的概率比YOLO高出3倍。

4.3. 把Fast R-CNN与YOLO结合

|  |  |  |  |
| --- | --- | --- | --- |
|  | mAP | 结合后 | 提升 |
| Fast R-CNN | - | 71.8 | - |
| Fast R-CNN (2007 data) | 66.9 | 72.4 | 0.6 |
| Fast R-CNN (VGG-M) | 59.2 | 72.4 | 0.6 |
| Fast R-CNN (CaffeNet) | 57.1 | 72.1 | 0.3 |
| YOLO | 63.4 | 75 | 3.2 |

**表格 2：VOC 2007模型组合实验。**我们检验了各种模型与Fast R-CNN最佳版本结合的效果。Fast R-CNN的其它版本只提供很小的提升，而YOLO则提供了显著的性能提升。

YOLO把背景误认为目标的情况比Fast R-CNN少得多。 通过使用YOLO消除Fast R-CNN的背景检测误差，我们获得了显著的性能提升。 对于R-CNN预测的每个边界框，我们检查YOLO是否预测了一个相似的框。 如果确实如此，那么我们会根据YOLO预测的概率和两个框之间的重叠情况提高预测值。

最好的Fast R-CNN模型在VOC 2007测试集中达到了 71.8％ 的mAP。 当与YOLO合并时，其mAP增加了 3.2％ 至 75.0％。 我们还尝试将顶级Fast R-CNN模型与其他几个版本的Fast R-CNN结合起来。 这写的结合的平均增长率在 0.3％ 至 0.6％ 之间。

结合YOLO后获得的性能提高不仅仅是模型集成的副产品，因为结合不同版本的Fast R-CNN几乎没有什么益处。 相反，正是因为YOLO在测试时出现了各种各样的错误，所以它在提高Fast R-CNN的性能方面非常有效。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 [27] | 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 [28] | 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 [32] | 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 |

**表格 3：PASCAL VOC 2012排行榜。**截至2015年11月6日，YOLO与完整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均值和每类的平均精度。YOLO是唯一的实时检测器。Fast R-CNN + YOLO是评分第四高的方法，比Fast R-CNN提升了2.3％。

不幸的是，这种组合不会从YOLO的速度中受益，因为我们分别运行每个模型，然后合并结果。 但是，由于YOLO速度如此之快，与Fast R-CNN相比，它不会增加任何显著的计算时间。

4.4. VOC 2012上的运行结果

在VOC 2012测试集中，YOLO的mAp得分是57.9％。这比现有最先进的技术水平低，更接近使用VGG-16的原始的R-CNN，见**表格 3**。与其最接近的竞争对手相比，我们的系统很难处理小物体上。在瓶子、羊、电视等类别上，YOLO得分比R-CNN和Feature Edit低8-10％。然而，在其他类别，如猫和火车YOLO取得了更好的表现。

我们的Fast R-CNN + YOLO模型组合是性能最高的检测方法之一。 Fast R-CNN与YOLO的组合提高了2.3％，在公共排行榜上提升了5个排名。

4.5. 泛化能力

用于目标检测的学术数据集以相同分布获取训练和测试数据。在现实世界的应用中，很难预测所有可能的用例，而且测试数据可能与系统之前看到的不同[3]。我们在毕加索（Picasso）数据集上[12]和People-Art数据集[3]上将YOLO与其它的检测系统进行比较，这两个数据集用于测试艺术画作中的行人检测。

**图表 5**显示了YOLO和其它检测方法之间的比较性能。作为参考，我们在person上提供VOC 2007的检测AP，其中所有模型仅在VOC 2007数据上训练。在毕加索数据集上的模型在VOC 2012上训练，而人物艺术（People-Art）数据集上的模型则在VOC 2010上训练。

值得注意的是，YOLO在VOC 2007上表现出色，其应用于艺术画作时的AP衰减程度远低于其他方法。

R-CNN在VOC 2007上有高AP。然而，当应用于艺术画作时，R-CNN明显下降。R-CNN使用选择性搜索来调整自然图像的边界框提出。R-CNN中的分类器步骤只能看到小区域，并且对提出的边界框质量要求较高。

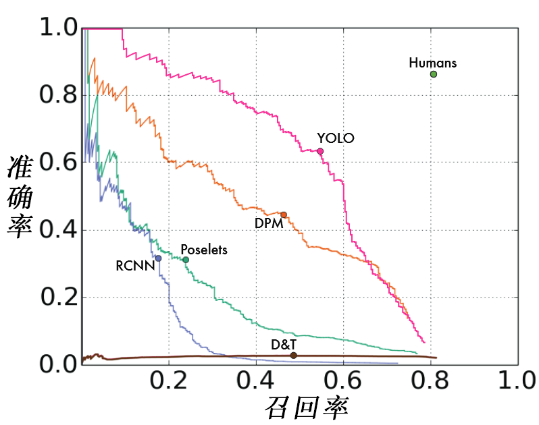
DPM在应用于艺术画作时保持了其AP。之前的工作认为DPM表现良好，因为它具有目标形状和布局的强大空间模型。虽然DPM不会像R-CNN那样退化，但它开始时的AP较低。

YOLO在VOC 2007上有很好的表现并且泛化性。像DPM一样，YOLO建模目标的大小和形状，以及目标和目标通常出现的位置之间的关系。艺术画作和自然图像在像素级别上有很大不同，但是它们在目标的大小和形状方面是相似的，因此YOLO仍然可以预测好的边界框和检测结果。

5．自然环境下的实时检测

YOLO是一款快速，精确的物体检测器，非常适合计算机视觉应用。 我们将YOLO连接到网络摄像头，并验证它是否保持实时性能，包括从摄像头获取图像并显示检测结果的时间。

由此产生的系统是交互性的。 虽然YOLO单独处理图像，但当连接到网络摄像头时，它的功能类似于跟踪系统，可在目标移动并在外观上发生变化时检测目标。 系统演示和源代码可在我们的项目网站上找到：<https://goo.gl/bEs6Cj>

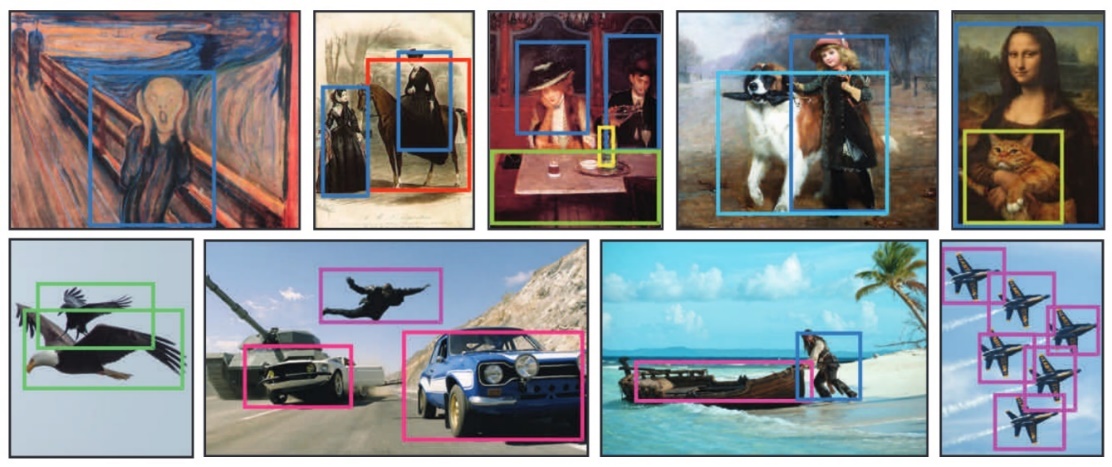


**图 (a)** : 毕加索数据集的P-R曲线

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | VOC 2007 | | 毕加索 | | 人物艺术 | |
|  |  | 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 |  |  |

**图 (b)** : VOC 2007、毕加索和人物艺术数据集的定量结果。毕加索数据集对AP和 最佳F1评分进行评价。

**图表 5：毕加索数据集与人物艺术数据集上的归纳结果。**



**图表 6：定性结果。**YOLO在网络采样的艺术画作和自然图像上的运行结果。虽然它将人误检成了飞机，但它大部分上是准确的。

6. 结论

我们详细介绍了YOLO——一种用于物体检测的统一模型。 我们的模型构造简单，可以直接在完整图像上训练。 与基于分类器的方法不同，YOLO是通过与检测性能直接对应的损失函数进行训练的，并且整个模型是一起训练的。

快速YOLO是文献中最快的通用目标检测器，YOLO推动实时对象检测的最新技术。 YOLO还能很好地推广到新领域，使其成为快速，高可靠性应用的理想选择。

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. 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>. 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 and A. Angelova. Real-time grasp detection using convolutional neural networks. *CoRR*, abs/1412.3128, 2014. 5
27. 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
28. 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
29. 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
30. M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In *Computer Vision–ECCV 2014*, pages 65–79. Springer, 2014. 5, 6
31. 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
32. Z. Shen and X. Xue. Do more dropouts in pool5 feature maps for better object detection. *arXiv preprint arXiv:1409.6911*, 2014. 7
33. 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
34. 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, 5
35. P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 4:34–47, 2001. 4
36. P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 5
37. 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
38. 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