SSD: Single Shot MultiBox Detector

Wei Liu[1] , Dragomir Anguelov[2] , Dumitru Erhan[3] , Christian Szegedy[3] ,
Scott Reed[4] , Cheng-Yang Fu[1] , Alexander C. Berg

[1] UNC Chapel Hill [2]Zoox Inc. [3]Google Inc. [4]University of Michigan, Ann-Arbor
[1]wliu@cs.unc.edu, [2]drago@zoox.com, [3]{dumitru,szegedy}@google.com,[4]reedscot@umich.edu, 1 {cyfu,aberg}@cs.unc.edu

摘要：我们提出了一种使用单个深层神经网络检测图像中对象的方法。我们的方法，名为 SSD，将边界框的输出空间离散化为一组默认框，该默认框在每个特征图位置有不同的宽高比和尺寸。在预测期间，网络针对每个默认框中的每个存在对象类别生成分数，并且对框进行调整以更好地匹配对象形状。另外，网络组合来自具有不同分辨率的多个特征图的预测，以适应处理各种尺寸的对象。我们的 SSD 模型相对于需要 region proposal 的方法是简单的，因为它完全消除了 proposal 生成和后续的像素或特征重采样阶段，并将所有计算封装在单网络中。这使得 SSD 容易训练和直接集成到需要检测组件的系统。PASCAL VOC，MS COCO 和 ILSVRC 数据集的实验结果证实，SSD 与使用额外的 region proposal 的方法具有可比较的准确性，并且速度更快，同时为训练和推理提供统一的框架。与其他单级方法相比，SSD 具有更好的精度，即使输入图像尺寸更小。对 VOC2007，在 300×300 输入，SSD 在 Nvidia Titan X 上 58FPS 时达到 72.1％的 mAP，500×500 输入 SSD 达到 75.1％的 mAP，优于类似的现有技术 Faster R-CNN 模型。代码链接：https://github.com/weiliu89/caffe/tree/ssd。

关键词：实时对象检测; 卷积神经网络

1、 引言

当前，现有对象检测系统是以下方法的变体：假设边界框，对每个框重新取样像素或特征，再应用高质量分类器。选择性搜索[1]方法后，Faster R-CNN[2]在 PASCAL VOC，MS COCO 和 ILSVRC 检测取得领先结果，这种流程成为检测领域的里程碑，具有更深的特征，如[3]所述。尽管准确，但这些方法对于嵌入式系统来说计算量过大，即使对于高端硬件，对于实时或接近实时的应用来说也太慢。 这些方法的检测速度通常以每秒帧数为单位进行测量，高精度检测器(基础 Faster R-CNN）最快仅以每秒 7 帧（FPS）运行。目前，已有广泛的尝试，通过研究检测流程的每个阶段（参见第 4 节中的相关工作）来建立更快的检测器，但是迄今为止，显着增加的速度仅仅是以显着降低的检测精度为代价。

本文提出了第一个基于深层网络的对象检测器，它不会对边界框假设的像素或特征进行重新取样，但和这种做法一样准确。这使高精度检测速度有显着提高（在 VOC2007 测试中, 58 FPS 下 72.1％ mAP，对 Faster R-CNN 7 FPS 下 mAP 73.2％，YOLO 45 FPS 下 mAP 63.4％）。速度的根本改进来自消除边界框 proposal 和随后的像素或特征重采样阶段。这不是第一篇这么做的文章（cf [4,5]），但是通过增加一系列改进，我们设法提高了以前尝试的准确性。我们的改进包括使用不同宽高比检测的单独的预测器（滤波器），预测边界框中的对象类别和偏移，并且将这些滤波器应用于网络后期的多个特征图，以便执行多尺度检测。通过这些修改，我们可以使用相对低分辨率的输入实现高精度检测，进一步提高处理速度。 虽然这些贡献可能独立看起来很小，但我们注意到，所得系统提高了 PASCAL VOC 的高速检测的准确性，从 YOLO 的 63.4％mAP 到我们提出的网络的 72.1％mAP。相比近期工作，这是在检测精度上的较大提高，残差网络上的卓越工作 [3]。 此外，显着提高高质量检测的速度可以拓宽计算机视觉有用使用范围。

总结我们的贡献如下：

- 我们引用了 SSD，一个单次检测器，用于多个类别，比先前技术的单次检测器（YOLO）速度更快，并且更准确很多，实际上和使用 region proposal、pooling 的更慢技术一样准确（包括 Faster RCNN）
- SSD 方法的核心是使用小卷积滤波器来预测特征图上固定的一组默认边界框的类别分数和位置偏移。
- 为了实现高检测精度，我们从不同尺度的特征图产生不同尺度的预测，并且通过宽高比来明确地分离预测。
- 总之，这些设计特性得到了简单的端到端训练和高精度，进一步提高速度和精度的权衡，即使输入相对低分辨率图像。
- 实验包括在 PASCAL VOC，MS COCO 和 ILSVRC 上评估不同输入大小下模型耗时和精度分析，并与一系列最新的先进方法进行比较。

2、单次检测器（SSD）

本节介绍我们提出的 SSD 检测架构（第 2.1 节）和相关的训练方法（第 2.2 节）。之后， 第 3 节呈现特定数据集的模型细节和实验结果。

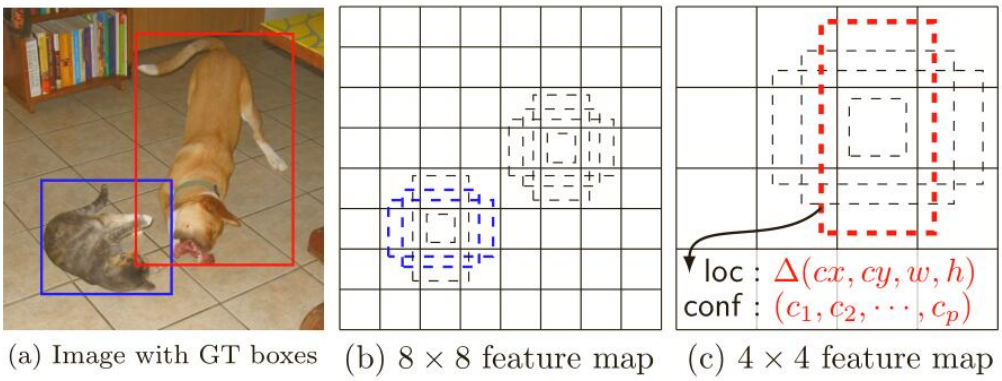

图 1：SSD 架构。（a）SSD 在训练期间仅需要每个对象的输入图像和真实标签框。 卷积处理时，我们在具有不同尺度（例如（b）和（c）中的 8×8 和 4×4）的若干特征图中的每个位置处评估不同横宽比的小集合（例如 4 个）默认框。 对于每个默认框，我们预测对所有对象类别（（c 1，c 2，…，c p））的形状偏移和置信度。在训练时，我们首先将这些默认框匹配到真实标签框。 例如，两个默认框匹配到猫和狗，这些框为正，其余视为负。 模型损失是位置损失（例如平滑 L1 [6]）和置信损失（例如 Softmax）之间的加权和。

2.1 模型

SSD 方法基于前馈卷积网络，其产生固定大小的边界框集合和框中对象类别的分数，接着是非最大化抑制步骤以产生最终检测。早期网络基于高质量图像分类（在任何分类层之前截断*（译者注：特征提取网络，例如：VGG、googlenet、alexnet）*）的标准架构，我们将其称为基础网络（我们的试验中使用了 VGG-16 网络作为基础，其他网络也应该能产生好的结果）。然后，我们向网络添加辅助结构，产生了具有以下主要特征的检测：

多尺度特征图检测：我们将卷积特征层添加到截断的基础网络的末尾。这些层尺寸逐渐减小，得到多个尺度检测的预测值。检测的卷积模型对于每个特征层是不同的（参见在单个尺度特征图上操作的 Overfeat [4]和 YOLO [5]）。

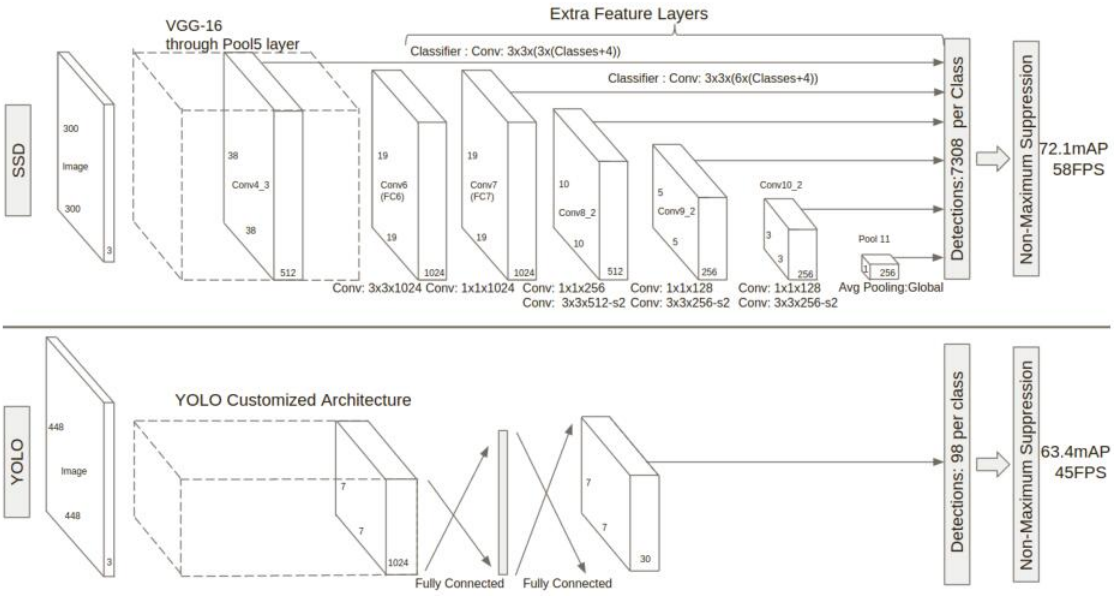

图 2：两个单次检测模型之间的比较：SSD 和 YOLO [5]。我们的 SSD 模型在基础网络的末尾添加了几个特征层，这些层预测了不同尺度和宽高比对默认框的偏移及其相关置信度。 300×300 输入尺寸的 SSD 在 VOC2007 测试中的精度显着优于 448×448 输入的 YOLO 的精度，同时还提高了运行速度，尽管 YOLO 网络比 VGG16 快。

检测的卷积预测器：每个添加的特征层（或可选的基础网络的现有特征层）可以使用一组卷积滤波器产生固定的预测集合。这些在图 2 中 SSD 网络架构顶部已指出。对于具有 p 个通道的大小为 m×n 的特征层，使用 3×3×p 卷积核卷积操作，产生类别的分数或相对于默认框的坐标偏移。在每个应用卷积核运算的 m×n 大小位置处，产生一个输出值。边界框偏移输出值是相对于默认框测量，默认框位置则相对于特征图（参见 YOLO [5]的架构，中间使用全连接层而不是用于该步骤的卷积滤波器）。

默认框与宽高比：我们将一组默认边界框与顶层网络每个特征图单元关联。默认框对特征图作卷积运算，使得每个框实例相对于其对应单元格的位置是固定的。在每个特征映射单元中，我们预测相对于单元格中的默认框形状的偏移，以及每个框中实例的每类分数。具体来说，对于在给定位置的 k 个框中每个框，我们计算 c 类分数和相对于原始默认框的 4 个偏移量。这使得在特征图中的每个位置需要总共（c+4）k 个滤波器，对于 m×n 特征图产生（c+4）kmn 个输出。有关默认框的说明，请参见图 1。我们的默认框类似于 Faster R-CNN [2]中使用的 anchor boxes，但我们将其应用于不同分辨率的特征图中。在多个特征图中使用不同的默认框形状，可以有效地离散可能的输出框形状空间。

2、2 训练

训练 SSD 和训练使用 region proposal、pooling 的典型分类器的关键区别在于，真实标签信息需要被指定到固定的检测器输出集合中的某一特定输出。Faster R-CNN [2]和 MultiBox [7]的 region proposal 阶段、YOLO [5]的训练阶段也需要类似这样的标签。一旦确定了该指定，则端对端地应用损失函数和反向传播。训练还涉及选择用于检测的默认框和尺度集合，以及 hard negative mining 和数据增广策略。

匹配策略：在训练时，我们需要建立真实标签和默认框之间的对应关系。请注意，对于每个真实标签框，我们从默认框中进行选择，这些默认框随位置、纵横比和比例而变化。启始时，我们匹配每个真实标签框与默认框最好的 jaccard 重叠。这是原始 MultiBox [7]使用的匹配方法，它确保每个真实标签框有一个匹配的默认框。与 MultiBox 不同，匹配默认框与真实标签 jaccard 重叠高于阈值（0.5）的默认框。添加

这些匹配简化了学习问题：它使得有多个重叠默认框时网络预测获得高置信度，而不是要求它选择具有最大重叠的那个。

训练：SSD 训练来自 MultiBox[7,8]，但扩展到处理多个对象类别。 以 $x_{ij}^{p} = 1$ 表示第 i 个默认框与类别 p 的第 j 个真实标签框相匹配，相反的$x_{ij}^{p} = 0$。 根据上述匹配策略，我们有$\sum_{i} x_{ij}^{p} \geq 1$，意味着可以有多于一个与第 j 个真实标签框相匹配的默认框。总体目标损失函数是位置损失（loc）和置信损失（conf）的加权和：

$$\mathrm{L(x,c,l,g)} = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{conf}(x,l,g))$$

其中 N 是匹配的默认框的数量，位置损失是预测框（l）和真实标签值框（g）参数之间的平滑 L1 损失[6]。类似于 Faster R-CNN [2]，我们对边界框的中心和其宽度和高度的偏移回归。 我们的置信损失是 softmax 损失对多类别置信（c）和权重项 α 设置为 1 的交叉验证。

选择默认框的比例和横宽比：大多数卷积网络通过加深层数减小特征图的大小。这不仅减少计算和存储消耗，而且还提供一定程度的平移和尺寸不变性。为了处理不同的对象尺寸，一些方法[4,9]建议将图像转换为不同的尺寸，然后单独处理每个尺寸，然后组合结果。然而，通过用单个网络中的若干不同层的特征图来进行预测，我们可以得到相同的效果，同时还在所有对象尺度上共享参数。之前的研究[10,11]已经表明使用来自较低层的特征图可以提高语义分割质量，因为较低层捕获到输入对象的更精细的细节。类似地，[12]表明，添加从高层特征图下采样的全局文本可以帮助平滑分割结果。受这些方法的启发，我们使用低层和高层的特征图进行检测预测。图 1 示出了在框架中使用的两个示例特征图（8×8 和 4×4），当然在实践中，我们可以使用更多具有相对小的计算开销的特征图。

已知网络中不同级别的特征图具有不同的（经验）感受野大小[13]。幸运的是，在 SSD 框架内，默认框不需要对应于每层的实际感受野。我们可以设计平铺，使得特定位置特征图，学习响应于图像的特定区域和对象的特定尺度。假设我们要使用 m 个特征图做预测。每个特征图的默认框的比例计算如下：

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m-1}(\mathrm{k}-1),\ \mathrm{k} \in [1,\mathrm{m}]$$

其中 s_{min} 是 0.2，s_{max} 是 0.95，意味着最低层具有 0.2 的刻度，最高层具有 0.95 的刻度，并且其间的所有层是规则间隔的。我们对默认框施以不同的宽高比，表示为 $a_r \in \{1,2,3,1/2,1/3\}$。我们可以计算每个默认框的宽度（$w_k^a = s_k\sqrt{a_r}$）和高度（$h_k^a = s_k/\sqrt{a_r}$）。对于宽高比为 1，我们还添加了一个缩放为$s_k' = \sqrt{s_k s_{k+1}}$的默认框，从而使每个特征图位置有 6 个默认框。设定每个默认框中心为（$\frac{i+0.5}{|f_k|}$，$\frac{j+0.5}{|f_k|}$），其中$|f_k|$是第 k 个正方形特征图的大小, i,j∈[0，$|f_k|$），随后截取默认框坐标使其 始终在[0，1]内。实际上，可以设计默认框的分布以最佳地拟合特定数据集。

通过组合许多特征图在所有位置的不同尺寸和宽高比的所有默认框的预测，我们具有多样化的预测集合，覆盖各种输入对象尺寸和形状。例如图 1 中，狗被匹配到 4×4 特征图中的默认框，但不匹配到 8×8 特征图中的任何默认框。这是因为那些框具有不同的尺度但不匹配狗的框，因此在训练期间被认为是负样本。

Hard negative mining：在匹配步骤之后，大多数默认框都是负样本，特别是当可能的默认框数量很大时。这导致了训练期间正负样本的严重不平衡。我们使用每个默认框的最高置信度对它们进行排序，并选择前面的那些，使得正负样本之间的比率最多为 3：1，以代替使用所有的负样本。我们发现，这导致更快的优化和更稳定的训练。

数据增广：为了使模型对于各种输入对象大小和形状更加鲁棒，每个训练图像通过以下选项之一随机采样：

- 使用整个原始输入图像

- 采样一个片段，使对象最小的 jaccard 重叠为 0.1,0.3,0.5,0.7 或 0.9。
- 随机采样一个片段

每个采样片段的大小为原始图像大小的[0.1,1]，横宽比在 1/2 和 2 之间。如果真实标签框中心在采样片段内，则保留重叠部分。在上述采样步骤之后，将每个采样片大小调整为固定大小，并以 0.5 的概率水平翻转。

3、实验结果

基础网络：我们的实验基于 VGG16 [14]网络，在 ILSVRC CLS-LOC 数据集[15]预训练。类似于 DeepLab-LargeFOV [16]，我们将 fc6 和 fc7 转换为卷积层，从 fc6 和 fc7 两层采样得到参数，将 pool5 从 2×2-s2 更改为 3×3-s1，并使用 atrous 算法填“洞”。我们删除了所有的 dropout 层和 fc8 层，使用 SGD 对这个模型进行 fine-tune，初始学习率10^{-3}，0.9 momentum, 0.0005 weight decay, batch 大小 32。每个数据集的学习速率衰减策略略有不同，稍后我们将描述详细信息。所有训练和测试代码在 caffe 框架编写，开源地址：https://github.com/weiliu89/caffe/tree/ssd。

3.1 PASCAL VOC2007

在这个数据集上，我们比较了 Fast R-CNN [6]和 Faster R-CNN [2]。所有方法使用相同的训练数据和预训练的 VGG16 网络。特别地，我们在 VOC2007train val 和 VOC2012 train val（16551images）上训练，在 VOC2007（4952 图像）测试。

图 2 显示了 SSD300 模型的架构细节。 我们使用 conv4_3，conv7（fc7），conv8_2，conv9_2，conv10_2 和 pool11 来预测位置和置信度（对 SSD500 模型，额外增加了 conv11_2 用于预测），用“xavier”方法初始化所有新添加的卷积层的参数[18]。由于 conv4_3 的大小较大（38×38），因此我们只在其上放置 3 个默认框 :一个 0.1 比例的框和另外纵横比为 1/2 和 2 的框。对于所有其他层，我们设置 6 个默认框，如第 2.2 节。如[12]中所指出的，由于 conv4_3 与其他层相比具有不同的特征尺度，我们使用[12]中引入的 L2 正则化技术，将特征图中每个位置处的特征范数缩放为 20，并在反向传播期间学习比例。我们使用10^{-3}学习速率进行 40k 次迭代，然后将其衰减到10^{-4}，并继续训练另外 20k 次迭代。表 1 显示，我们的 SSD300 模型已经比 Fast R-CNN 更准确。当以更大的 500×500 输入图像训练 SSD，结果更准确，甚至惊人的超过了 Faster R-CNN 1.9% mAP。

为了更详细地了解我们的两个 SSD 模型的性能，我们使用了来自[19]的检测分析工具。图 3 显示 SSD 可以高质量检测（大、白色区域）各种对象类别。它的大部分置信度高的检测是正确的。召回率在 85-90%左右，并且比“弱”（0.1 jaccard 重叠）标准高得多。与 R-CNN [20]相比，SSD 具有较少的定位误差，表明 SSD 可以更好地定位对象，因为它直接回归对象形状和分类对象类别，而不是使用两个去耦步骤。然而，SSD 对相似对象类别（尤其是动物）有更多的混淆，部分是因为多个类别分享了位置。

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	**74.8**	80.4	70.4
Faster [2]	73.2	76.5	79.0	70.9	**65.5**	**52.1**	83.1	84.7	86.4	52.0	**81.9**	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD300	72.1	75.2	**79.8**	70.5	62.5	41.3	81.1	80.8	86.4	51.5	74.3	**72.3**	83.5	84.6	80.6	74.5	46.0	71.4	73.8	83.0	69.1
SSD500	**75.1**	**79.8**	79.5	**74.5**	63.4	51.9	**84.9**	**85.6**	**87.2**	**56.6**	80.1	70.0	**85.4**	**84.9**	**80.9**	**78.2**	**49.0**	**78.4**	72.4	**84.6**	**75.5**

表 1 ： PASCAL VOC2007 测试集检测结果。Fast 和 Faster R-CNN 输入图像最小尺寸为 600，两个 SSD 模型除了输入图像尺寸（300*300 和 500*500），其他设置与其相同。很明显，较大的输入尺寸得到更好的结果。

图 4 显示 SSD 对边界框尺寸非常敏感。换句话说，它对较小的对象比较大的对象具有更差的性能。这毫不意外，因为小对象在最顶层可能没有任何信息保留下来。增加输入尺寸（例如从 300×300 到

500×500）可以帮助改善检测小对象，但是仍然有很大改进空间。积极的一面是，我们可以清楚地看到 SSD 在大对象上表现很好。并且对于不同的对象宽高比非常鲁棒，因为我们对每个特征图位置使用各种长宽比的默认框。

3.2 模型分析

为了更好地理解 SSD，我们还进行了几个人为控制的实验，以检查每个组件如何影响最终性能。对于所有以下实验，我们使用完全相同的设置和输入大小（300×300），除了变动的组件。

	SSD300					
more data augmentation?		✔	✔	✔	✔	✔
use conv4_3?	✔		✔	✔	✔	✔
include $\{\frac{1}{2}, 2\}$ box?	✔	✔		✔	✔	✔
include $\{\frac{1}{3}, 3\}$ box?	✔	✔			✔	✔
use atrous?	✔	✔	✔	✔		✔
VOC2007 `test` mAP	65.4	68.1	69.2	71.2	71.4	72.1

表 2： 不同选择和组件对 SSD 表现的影响

关键的数据增广 Fast 和 Faster R-CNN 使用原始图像和水平翻转（0.5 概率）图像训练。我们使用更广泛的采样策略，类似于 YOLO [5]，但它使用了我们没有使用的光度失真。表 2 显示，我们可以用这个抽样策略提高 6.7%的 mAP。我们不知道我们的采样策略将对 Fast 和 Faster R-CNN 提升多少，但可能效果不大，因为他们在分类期间使用了 pooling，比人为设置更鲁棒。

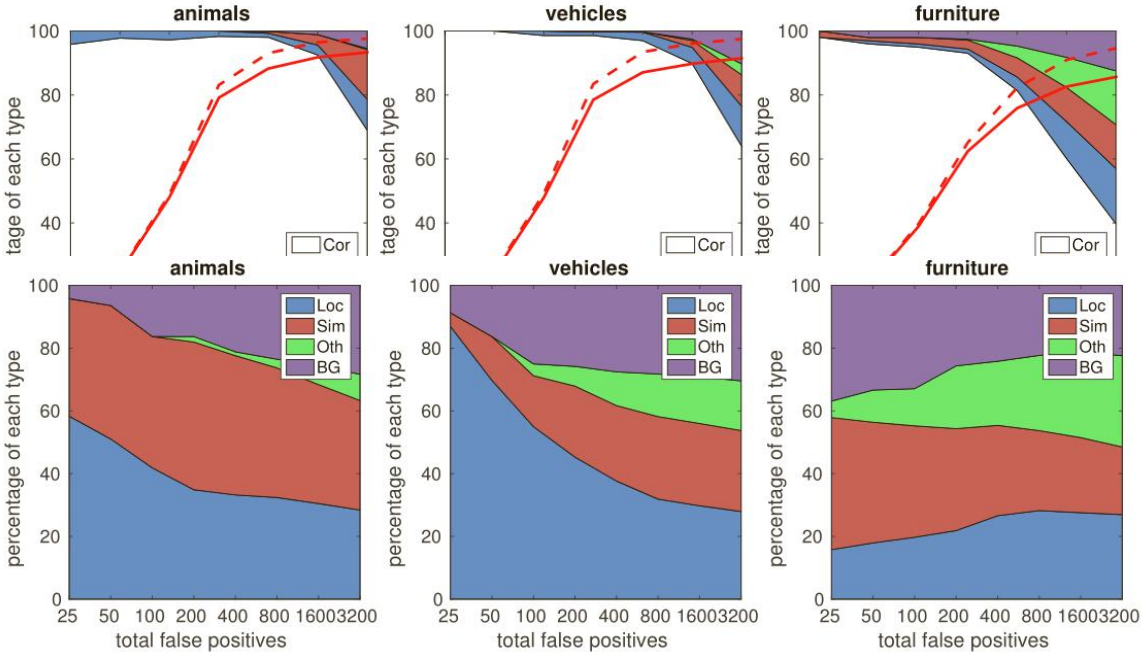

图 3：VOC2007 测试集上 SSD 500 对动物、车辆和家具性能的可视化。 第一行显示由于定位不良（Loc），与类似类别（Sim）、其他类别（Oth）或背景（BG）混淆的正确检测（Cor）、假阳性检测的累积分数。 红色实线反映了随着检测次数的增加，“强”标准（0.5 jaccard 重叠）的召回率变化。 红色虚线使用“弱”标准（0.1 jaccard 重叠）。 底行显示排名靠前的假阳性类型的分布。

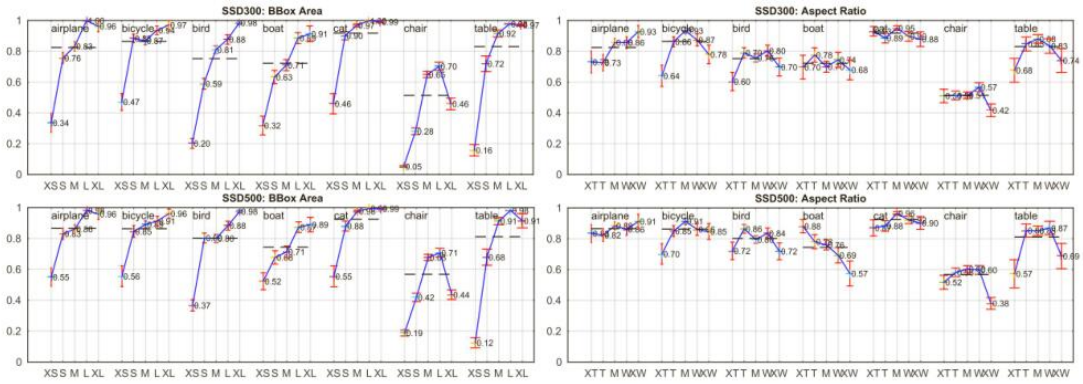

Fig. 4: **Sensitivity and impact of different object characteristics on VOC2007 `test` set.** Each plot shows the normalized AP [19] with standard error bars (red). Black dashed lines indicate overall normalized AP. The plot on the left shows the effects of BBox Area per category, and the right plot shows the effect of Aspect Ratio. Key: BBox Area: XS=extra-small; S=small; M=medium; L=large; XL =extra-large. Aspect Ratio: XT=extra-tall/narrow; T=tall; M=medium; W=wide; XW =extra-wide.

更多特征图的提升 受许多语义分割工作启发[10,11,12]，我们也使用底层特征图来预测边界框输出。我们比较使用 conv4_3 预测的模型和没有它的模型。从表 2，我们可以看出，通过添加 conv4_3 进行预测，它有明显更好的结果（72.1% vs 68.1%）。这也符合我们的直觉，conv4_3 可以捕获对象更好的细粒度，特别是细小的细节。

更多的默认框形状效果更好 如第 2.2 节所述，默认情况下，每个位置使用 6 个默认框。如果我们删除具有 1/3 和 3 宽高比的框，性能下降 0.9%。通过进一步移除 1/2 和 2 纵横比的框，性能再下降 2%。使用多种默认框形状似乎使网络预测任务更容易。

Atrous 算法更好更快 如第 3 节所述，我们使用了 VGG16 的 atrous 版本，遵循 DeepLabLargeFOV[16]。如果我们使用完整的 VGG16，保持 pool5 与 2×2-s2，并且不从 fc6 和 fc7 的采集参数，添加 conv5_3，结果稍差（0.7%），而速度减慢大约 50%。

3.3 PASCAL VOC2012

采用和 VOC2007 上一样的设置，这次，用 VOC2012 的训练验证集和 VOC2007 的训练验证集、测试集（21503 张图像）训练，在 VOC2012 测试集（10991 张图像）测试。由于有了更多的训练数据，模型训练时以10^{-3}学习率进行 60K 次迭代，再减小到10^{-4}继续迭代 20K 次。

表 3 显示了 SSD300 和 SSD500 模型的结果。我们看到与我们在 VOC2007 测试中观察到的相同的性能趋势。我们的 SSD300 已经优于 Fast R-CNN，并且非常接近 Faster R-CNN（只有 0.1%的差异）。通过将

Method	*mAP*	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	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
Faster [2]	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
YOLO [5]	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
SSD300	70.3	84.2	76.3	69.6	53.2	40.8	78.5	73.6	88.0	50.5	73.5	**61.7**	85.8	80.6	81.2	77.5	44.3	73.2	**66.7**	81.1	65.8
SSD500	**73.1**	**84.9**	**82.6**	**74.4**	**55.8**	**50.0**	**80.3**	**78.9**	88.8	**53.7**	76.8	59.4	**87.6**	**83.7**	**82.6**	**81.4**	**47.2**	**75.5**	65.6	**84.3**	**68.1**

Table 3: **PASCAL VOC2012 `test` detection results.** Fast and Faster R-CNN use images with minimum dimension 600, while the image size for YOLO is 448 × 448.

训练和测试图像大小增加到 500×500，我们比 Faster R-CNN 高 2.7%。与 YOLO 相比，SSD 显著更好，可能是由于使用来自多个特征图的卷积默认框和训练期间的匹配策略。

3.4 MS COCO

为了进一步验证 SSD 架构，我们在 MS COCO 数据集上训练了我们的 SSD300 和 SSD500 模型。由于 COCO 中的对象往往较小，因此我们对所有图层使用较小的默认框。我们遵循第 2.2 节中提到的策略，但是现在我们最小的默认框具有 0.1 而不是 0.2 的缩放比例，并且 conv4_3 上默认框的缩放比例是 0.07（例如，对应于 300×300 图像的 21 个像素）。

我们使用 trainval35k [21]来训练我们的模型。由于 COCO 有更多的对象类别，开始时的梯度不稳定。我们首先用 8×10^{-4}的学习率迭代 4K 次训练模型，接着以10^{-3}学习率进行 140K 次迭代，再以10^{-5}学习率迭代 60K 次，10^{-5}学习率迭代 40K 次。表 4 显示了 test-dev2015 上的结果。与我们在 PASCAL VOC 数据集上观察到的类似，SSD300 在 mAP@0.5 和 mAP@[0.5：0.95]中优于 Fast R-CNN，在 mAP @ [0.5：0.95]与 Faster R-CNN 接近。然而，mAP@0.5 更糟，我们推测，这是因为图像尺寸太小，这阻止了模型精确定位许多小对象。通过将图像大小增加到 500×500，我们的 SSD500 在两个标准中都优于 Faster R-CNN。此外，我们的 SSD500 模型也比 ION[21]更好，它是一个多尺寸版本的 Fast R-CNN，使用循环网络显式模拟上下文。在图 5 中，我们展示了使用 SSD500 模型在 MS COCO test-dev 的一些检测示例。

Method	data	Average Precision		
		0.5	0.75	0.5:0.95
Fast R-CNN [6]	train	35.9	-	19.7
Faster R-CNN [2]	train	42.1	-	21.5
Faster R-CNN [2]	trainval	42.7	-	21.9
ION [21]	train	42.0	23.0	23.0
SSD300	trainval35k	38.0	20.5	20.8
SSD500	trainval35k	43.7	24.7	24.4

Table 4: **MS COCO test-dev2015 detection results.**

3.5 ILSVRC 初步结果

我们将我们用于 MS COCO 的相同的网络架构应用于 ILSVRC DET 数据集[15]。我们使用 ILSVRC2014 DET train 和 val1 来训练 SSD300 模型，如[20]中所使用。我们首先以 8×10^{-4}的学习率迭代 4K 次训练模型，再用10^{-3}学习率进行 320k 次迭代训练该模型，然后用10^{-4}进行 100k 次迭代和10^{-5}继续训练 60k 次迭代。我们可以在 val2 集上实现 41.1mAP[20]。再一次的，它验证 SSD 是高质量实时检测的一般框架。

3.6 推理期间

考虑到从我们的方法生成的大量框，有必要在推理期间有效地执行非最大抑制（nms）。通过使用 0.01 的置信度阈值，我们可以过滤掉大多数框。然后，我们使用 Thrust CUDA 库进行排序，使用 GPU 计算所有剩余框之间的重叠，对 jaccard 重叠为 0.45 的每个类应用 nms，并保存每个图像的前 200 个检测。对于 20 个 VOC 类别的 SSD300，每个图像该步花费大约 2.2 毫秒，这接近在所有新添加的层上花费的总时间。

表 5 显示了 SSD、Faster R-CNN [2]和 YOLO [5]之间的比较。Faster R-CNN 对 region proposal 使用额外的预测层，并且需要特征下采样。相比之下，我们的 SSD500 方法在速度和精度上优于 Faster R-CNN。 值得一提的是，我们的方法 SSD300 是唯一的实时实现 70％以上 mAP 的方法。 虽然快速 YOLO [5]可以运行在 155 FPS，但精度只有差不多 20％的 mAP。

Method	*mAP*	FPS	# Boxes
Faster R-CNN [2](VGG16)	73.2	7	300
Faster R-CNN [2](ZF)	62.1	17	300
YOLO [5]	63.4	45	98
Fast YOLO [5]	52.7	155	98
SSD300	72.1	58	7308
SSD500	**75.1**	23	20097

Table 5: **Results on Pascal VOC2007 `test`.** SSD300 is the only real-time detection method that can achieve above 70% mAP. By using a larger input image, SSD500 outperforms all methods on accuracy while maintaining a close to real-time speed. The speed of SSD models is measured with batch size of 8.

4、相关工作

目前有两种已建立的用于图像中对象检测的方法，一种基于滑动窗口，另一种基于 region proposal 分类。在卷积神经网络出现之前，用于检测的两种方法 Deformable Part Model（DPM）[22]和选择性搜索[1]性能接近。然而，在 R-CNN[20]带来的显着改进之后，其结合了选择性搜索 region proposal 和基于卷积网络的后分类，region proposal 对象检测方法变得普遍。

原始的 R-CNN 方法已经以各种方式进行了改进。第一组方法提高了后分类的质量和速度，因为它需要对成千上万的图像作物进行分类，这是昂贵和耗时的。SPPnet[9]对原始的 R-CNN 方法大大提速。它引入了空间金字塔池化层，其对区域大小和尺度更加鲁棒，并且允许分类层重用在若干图像分辨率生成的特征图特征。Fast R-CNN[6]扩展了 SPPnet，使得它可以通过最小化置信度和边界框回归的损失来对所有层进行端对端微调，这在 MultiBox[7]中首次引入用于学习对象。

第二组方法使用深层神经网络提高 proposal 生成的质量。在最近的工作中，例如 MultiBox[7,8]，基于低层图像特征的选择性搜索 region proposal 被直接从单独的深层神经网络生成的 proposal 所替代。这进一步提高了检测精度，但是导致了一些复杂的设置，需要训练两个神经网络及其之间的依赖。Faster R-CNN[2]通过从 region proposal 网络（RPN）中学习的方案替换了选择性搜索 proposal，并且引入了通过微调共享卷积层和两个网络的预测层之间交替来集成 RPN 与 Fast R-CNN 的方法。用这种方式 region proposal 池化中层特征图，最终分类步骤更快速。我们的 SSD 与 Faster R-CNN 中的 region proposal 网络（RPN）非常相似，因为我们还使用固定的（默认）框来进行预测，类似于 RPN 中的 achor 框。但是，不是使用这些来池化特征和评估另一个分类器，我们同时在每个框中为每个对象类别产生一个分数。因此，我们的方法避免了将 RPN 与 Fast R-CNN 合并的复杂性，并且更容易训练，更易于集成到其他任务中。

另一组方法与我们的方法直接相关，完全跳过 proposal 步骤，直接预测多个类别的边界框和置信度。OverFeat[4]是滑动窗口方法的深度版本，在知道基础对象类别的置信度之后直接从最顶层特征图的每个位置预测边界框。YOLO [5]使用整个最高层特征图来预测多个类别和边界框（这些类别共享）的置信度。我们的 SSD 方法属于此类别，因为我们没有提案步骤，但使用默认框。然而，我们的方法比现有方法更灵活，因为我们可以在不同尺度的多个特征图中的每个特征位置上使用不同宽高比的默认框。如果顶层特征图每个位置只使用一个默认框，我们的 SSD 将具有与 OverFeat[4]类似的架构;如果我们使用整个顶层特征图并且添加一个全连接层用于预测而不是我们的卷积预测器，并且没有明确考虑多个宽高比，我们可以近似地再现 YOLO[5]。

5、结论

本文介绍了 SSD，一种用于多个类别的快速单次对象检测器。我们的模型的一个关键特点是使用多尺度卷积边界框输出附加到网络顶部的多个特征图。这种表示允许我们有效地模拟可能的框形状空间。我们实验验证，给定适当的训练策略，更大量的仔细选择的默认边界框得到了性能的提高。我们建立 SSD 模型，与现有方法相比，至少相差一个数量级的框预测位置，规模和纵横比[2,5,7]。

我们证明，给定相同的 VGG-16 基础架构，SSD 在精度和速度方面胜过最先进的对象检测器。我们的 SSD500 型号在 PASCAL VOC 和 MS COCO 的精度方面明显优于最先进的 Faster R-CNN [2]，速度快了 3 倍。我们的实时 SSD300 模型运行在 58 FPS，这比当前的实时 YOLO[5]更快，同时有显著高质量的检测。

除了它的独立实用程序，我们相信，我们的完整和相对简单的 SSD 模型为使用对象检测组件的大型系统提供了一个伟大的组成块。一个有希望的未来方向，是探索其作为使用循环神经网络的系统一部分，用以检测和跟踪视频中对象。

6、致谢

这个项目是在谷歌开始的实习项目，并在 UNC 继续。 我们要感谢亚历克斯·托舍夫有用的讨论，并感谢谷歌的 Image Understanding 和 DistBelief 团队。 我们也感谢菲利普·阿米拉托和帕特里克·波尔森有益的意见。我们感谢 NVIDIA 提供 K40 GPU 并感谢 NSF 1452851 的支持。

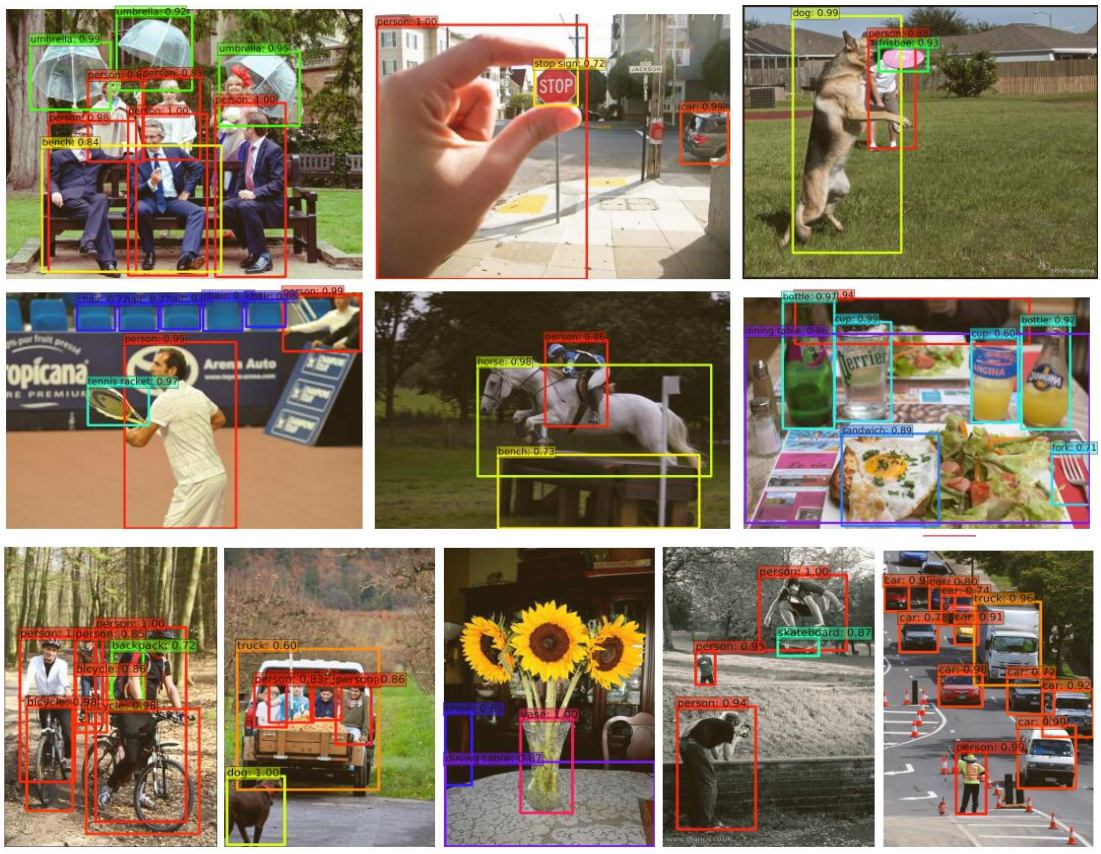

Fig. 5: Detection examples on MS COCO `test-dev` with SSD500 model. We show detections with scores higher than 0.6. Each color corresponds to an object category.

引用

1. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV (2013)
2. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR.(2016)
4. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR. (2014)
5. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016)
6. Girshick, R.: Fast R-CNN. In: ICCV. (2015)
7. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR. (2014)
8. Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441 v3 (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV. (2014)
10. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)
11. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR. (2015)
12. Liu, W., Rabinovich, A., Berg, A.C.: ParseNet: Looking wider to see better. In: ILCR. (2016)
13. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. In: ICLR. (2015)
14. Simonyan,K.,Zisserman,A.: Verydeepconvolutionalnetworksforlarge-scaleimagerecog-nition. In: NIPS. (2015)
15. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Li, F.F.: Imagenet large scale visual recognition challenge. IJCV (2015)
16. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR. (2015)
17. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: MM, ACM (2014)
18. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS. (2010)
19. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: ECCV 2012. (2012)

20. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)
21. Bell, S., Zitnick, C.L., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: CVPR. (2016)
22. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR. (2008)

（菜鸟水平有限，错误之处难免，忘大家多多拍砖，才能更快进步。

2016.11.1 翻译完成

王广胜）