SSD: Single Shot MultiBox Detector

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

UNC Chapel Hill Zoox Inc. Google Inc. University of Michigan, Ann-Arbor
wliu@cs.unc.edu, drago@zoox.com, {dumitru,szegedy}@google.com,
reedscot@umich.edu, {cucumber}@cs.unc.edu

摘要：我们提出了一种使用单一深度神经网络检测图像中物体的方法。我们的方法被命名为 SSD，它将边界框的输出空间离散化为一组默认框，每个特征图位置都有不同的长宽比和比例。在预测时，网络会对每个默认框中存在的每个物体类别进行评分，并对框进行调整，以更好地匹配物体形状。此外，该网络还能结合来自不同分辨率的多个特征图的预测结果，自然地处理各种尺寸的物体。相对于需要物体建议的方法而言，SSD 非常简单，因为它完全省去了建议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一个网络中。这使得 SSD 易于训练，并可直接集成到需要检测组件的系统中。在 PASCAL VOC、COCO 和 ILSVRC 数据集上的实验结果证实，SSD 的精确度与使用额外对象提议步骤的方法相比具有竞争力，而且速度更快，同时还为训练和推理提供了统一的框架。对于 300×300 输入，SSD 在 Nvidia Titan X 上以 59 FPS 进行的 VOC2007 测试中实现了 74.3% 的 mAP1；对于 512×512 输入，SSD 实现了 76.9% 的 mAP，超过了同类最先进的 Faster R-CNN 模型。与其他单级方法相比，即使输入图像尺寸较小，SSD 的准确率也要高得多。代码见 https://github.com/weiliu89/caffe/tree/ssd .

关键词：实时物体检测，卷积神经网络

1 导言

目前最先进的物体检测系统都是以下方法的变体：假设边界框，对每个边界框的像素或特征进行重采样，然后应用高质量的分类器。

自选择性搜索（Selective Search）[1] 以来，这种方法在检测基准测试中一直占据主导地位，目前在 PASCAL VOC、COCO 和 ILSVRC 检测中取得的领先成果都是基于 Faster R-CNN[2]（尽管具有更深入的特征，如[3]）。这些方法虽然精确，但对于嵌入式系统来说计算量过大，即使使用高端硬件，对于实时应用来说也过于缓慢。

这些方法的检测速度通常以每帧秒数（SPF）为单位，即使是最快的高精度检测器 Faster R-CNN，其检测速度也只有每秒 7 帧（FPS）。人们曾多次尝试通过攻击检测流水线的每个阶段来构建更快的检测器（参见第 4 章中的相关工作），但迄今为止，速度的显著提高只能以检测精度的大幅降低为代价。

本文提出了首个基于深度网络的物体检测器，它无需为边界框假设对像素或特征进行重采样，其准确度不亚于采用重采样的方法。这大大提高了高精度检测的速度（在 VOC2007 测试中为 59 FPS，mAP 为 74.3%，而 Faster R-CNN 为 7 FPS，mAP 为 73.2%，YOLO 为 45 FPS，mAP 为 63.4%）。速度的根本性提高来自于取消了边界框建议和随后的像素或特征重采样阶段。我们并不是第一个这样做的人（参见 [4,5]），但通过一系列改进，我们成功地提高了准确率，大大超过了之前的尝试。我们的改进包括使用小型卷积滤波器来预测物体类别和边界框位置的偏移，使用单独的预测器（滤波器）进行不同长宽比的检测，以及将这些滤波器应用于网络后期阶段的多个特征图，以便在多个尺度上进行检测。通过这些修改，特别是使用多层进行不同尺度的预测，我们可以使用相对较低的分辨率输入实现高精度，从而进一步提高检测速度。虽然这些单独的贡献看起来很小，但我们注意到，由此产生的系统提高了 PASCAL VOC 的实时检测精度，从 YOLO 的 63.4% mAP 提高到我们的 SSD 的 74.3% mAP。这比最近备受瞩目的残差网络工作[3]的检测准确率提高幅度更大。此外，大幅提高高质量检测的速度可以扩大计算机视觉的应用范围。

我们将我们的工作总结如下：

- 我们引入了 SSD，这是一种适用于多个类别的单次检测器，它比以往最先进的单次检测器（YOLO）更快、更准确，实际上与执行显式区域建议和池化的较慢技术（包括更快 R-CNN 技术）一样准确。

- SSD 的核心是利用应用于特征图的小型卷积滤波器，为一组固定的默认边界框预测类别得分和框偏移。
- 为了达到较高的检测精度，我们从不同尺度的特征图中生成不同尺度的预测结果，并根据长宽比明确区分预测结果。
- 这些设计特点使端到端训练变得简单，即使在低分辨率输入图像上也能达到很高的精度，从而进一步改善了速度与精度之间的权衡。
- 实验包括在 PASCAL VOC、COCO 和 ILSVRC 上对不同输入规模的模型进行时序和精度分析，并与一系列最新的先进方法进行比较。

2 The Single Shot Detector (SSD，单发探测器)

本节将介绍我们提出的 SSD 检测框架（第 2.1 节）和相关的训练方法（第 2.2 节）。随后，第 3 节将介绍特定数据集的模型细节和实验结果。

2.1 Model

SSD 方法基于一个前馈卷积网络，该网络生成固定大小的边界框集合，并对这些边界框中是否存在物体类别实例进行评分，然后通过一个非最大抑制步骤生成最终检测结果。早期网络层基于用于高质量图像分类的标准架构（在任何分类层之前截断），我们称之为基础网络。然后，我们为网络添加辅助结构，以产生具有以下主要特征的检测结果：

用于检测的多尺度特征图，我们在截断的基础网络末端添加卷积特征层。这些层的规模逐渐缩小，可以预测多种尺度的探测结果。每个特征层用于预测检测结果的卷积模型都不相同（参照 Overfeat[4] 和 YOLO[5] 在单一尺度特征图上的操作）。

用于检测的卷积预测器，每个添加的特征层（或基础网络中的现有特征层）都可以使用一组卷积滤波器生成一组固定的检测预测结果。图 2 中的 SSD 网络架构顶部显示了这些滤波器。对于具有 p 个通道、大小为 m × n 的特征层来说，预测潜在检测参数的基本要素是一个 3 × 3 × p 的小内核，它可以产生一个类别的分数，也可以产生相对于默认方框坐标的形状偏移。在应用内核的 m × n 个位置上，每个位置都会产生一个输出值。边框偏移输出值是相对

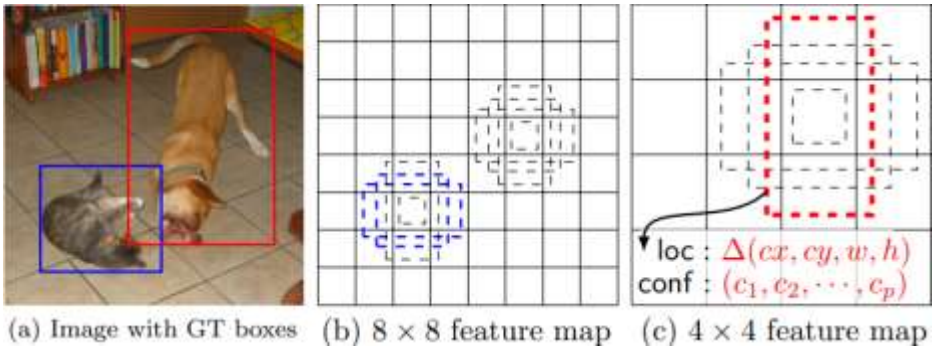

图 1：SSD 框架。(a) 在训练过程中，SSD 只需要输入图像和每个物体的地面实况框。我们以卷积的方式，在不同比例的特征图（如（b）和（c）中的 8 × 8 和 4 × 4）中的每个位置评估一小组（如 4 个）不同长宽比的默认方框。对于每个默认方框，我们都会预测形状偏移和所有对象类别（(c1, c2, - - , cp)）的置信度。在训练时，我们首先将这些默认方框与地面实况方框进行匹配。例如，我们将两个默认方框与猫匹配，一个与狗匹配，这两个方框被视为正方，其余的被视为负方。模型损失是定位损失（如 Smooth L1 [6]）和置信度损失（如 Softmax）的加权和。

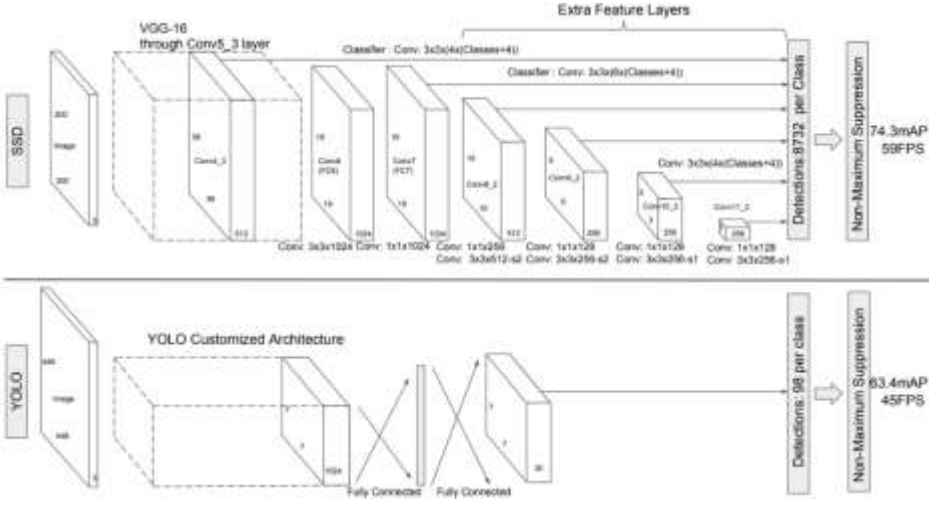

图 2：两种单发检测模型的比较：SSD 和 YOLO [5]。我们的 SSD 模型在基础网络的末端添加了几个特征层，预测不同比例和长宽比的默认方框偏移及其相关的可信度。在 VOC2007 测试中，输入尺寸为 300 × 300 的 SSD 在准确性上明显优于 448 × 448 的 YOLO，同时还提高了速度。

于每个特征图位置的默认边框位置测量的（参考 YOLO[5] 的架构，该架构在此步骤中使用了中间全连接层而非卷积滤波器）。

默认方框和长宽比:对于网络顶部的多个特征图，我们为每个特征图单元关联了一组默认边界框。默认方框以卷积方式平铺特征图，因此每个方框相对于其对应单元的位置都是固定的。在每个特征图单元中，我们都会预测相对于单元中默认方框形状的偏移量，以及表示这些方框中存在类实例的每类分数。具体来说，对于给定位置 k 个方框中的每个方框，我们计算 c 个类别得分和相对于原始默认方框形状的 4 个偏移量。这样，在特征图中的每个位置周围总共会应用 (c + 4)k 个过滤器，从而在 m × n 的特征图中产生 (c + 4)kmn 个输出。默认方框的示意图请参见图 1。我们的默认方框与 Faster R-CNN [2] 中使用的锚方框类似，但我们将其应用于多个不同分辨率的特征图。允许在多个特征图中使用不同的默认方框形状，可以有效地离散化可能的输出方框形状空间。

2.2 Training

训练 SSD 与训练使用区域建议的典型检测器的主要区别在于，需要将地面实况信息分配给固定检测器输出集合中的特定输出。YOLO[5]的训练以及 Faster R-CNN[2] 和 MultiBox[7] 的区域提议阶段也需要某种形式的分配。一旦确定了这一分配，损失函数和反向传播就会被端到端地应用。训练还包括选择一组用于检测的默认盒和尺度，以及硬负挖掘和数据增强策略。

匹配策略：在训练过程中，我们需要确定哪些默认方框与地面实况检测相对应，并据此对网络进行训练。对于每个地面实况框，我们都要从默认框中进行选择，默认框的位置、长宽比和比例各不相同。首先，我们将每个地面实况框与具有最佳 jaccard 重叠的默认框进行匹配（如 MultiBox [7]）。与 MultiBox 不同的是，我们将默认方框与 jaccard 重合度高于阈值（0.5）的任何地面实况进行匹配。这就简化了学习问题，允许网络预测多个重叠默认框的高分，而不是要求网络只选择重叠度最大的默认框。

训练目标，SSD 训练目标源于 MultiBox 目标[7,8]，但被扩展用于处理多个对象类别。假设 xpij = {1,0} 是第 i 个默认方框与第 j 个类别 p 的地面实况方框匹配的指标。总体目标损失函数是定位损失（loc）和置信度损失（conf）的加权和：

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

其中，N 是匹配的默认方框数。定位损失是预测框 (l) 和地面实况框 (g) 参数之间的平滑 L1 损失 [6]。与 Faster R-CNN [2]类似，我们对默认边界框 (d) 的中心 (cx,cy) 及其宽度 (w) 和高度 (h) 的偏移进行回归。

$$L_{loc}(x,l,g) = \sum_{i\in Pos}^{N} \sum_{m\in\{cx,cy,w,h\}} x_{ij}^{k}\text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m)$$
$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h \quad (2)$$
$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

置信度损失是多个类别置信度 (c) 的软最大损失。

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i\in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \quad (3)$$

通过交叉验证，权重项 α 设为 1。

为默认方框选择尺度和长宽比:为了处理不同的物体尺度，一些方法 [4,9] 建议以不同的尺寸处理图像，然后将结果合并。然而，通过在单个网络中利用多个不同层的特征图进行预测，我们可以模拟出相同的效果，同时还能在所有物体尺度上共享参数。之前的研究 [10,11] 表明，使用低层的特征图可以提高语义分割质量，因为低层能捕捉到输入对象更多的细节。同样，[12] 的研究表明，添加从特征图中汇集的全局上下文有助于平滑分割结果。受这些方法的启发，我们同时使用下层和上层特征图进行检测。图 1 显示了框架中使用的两个示例特征图（8×8 和 4×4）。实际上，我们可以使用更多的特征图，而且计算开销很小。

众所周知，网络中不同层次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，默认方框并不需要与每一层的实际感受野相对应。我们设计默认方框的平铺，使特定的特征

图学会对物体的特定尺度做出反应。假设我们要使用 m 个特征图进行预测。每个特征图的默认方框比例计算如下
其中 smin 为 0.2，smax 为 0.9，即最低层的刻度为 0.2，最高层的

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1), \quad k \in [1, m] \tag{4}$$

刻度为 0.9，中间所有层的间距都是规则的。我们为默认方框规定了不同的长宽比，并将它们分别记为 $a_r \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$. 我们可以计算出 the width ($w_k^a = s_k\sqrt{a_r}$) and height ($h_k^a = s_k/\sqrt{a_r}$)的默认方框。对于长宽比为 1 的情况，我们还添加了一个默认方框，其比例为$s_k' = \sqrt{s_k s_{k+1}}$, 这样，每个地物图位置就有 6 个默认方框。我们将每个默认方框的中心设置为 $(\frac{i+0.5}{|f_k|}, \frac{j+0.5}{|f_k|})$, 其中，|fk| 是第 k 个方格特征图的大小，i,j ∈ [0,|fk|)。在实践中，我们也可以设计最适合特定数据集的默认方框分布。如何设计最优的平铺也是一个未决问题。

通过将许多特征图中所有位置的不同比例和长宽比的所有默认方框的预测结果结合起来，我们就可以得到一组涵盖各种输入对象尺寸和形状的多样化预测结果。例如，在图 1 中，狗与 4 × 4 特征图中的默认方框相匹配，但与 8 × 8 特征图中的任何默认方框都不匹配。这是因为这些方框的尺度不同，与狗的方框不匹配，因此在训练过程中被视为负值。

负面样本设置：在匹配步骤之后，大部分默认方框都是负面的，尤其是当可能的默认方框数量较多时。这就造成了正负训练示例之间的严重不平衡。我们不使用所有的负面示例，而是使用每个默认框的最高置信度损失对它们进行排序，并挑选出最重要的示例，这样负面示例和正面示例的比例最多为 3:1。我们发现，这样做优化速度更快，训练效果也更稳定。

数据扩增：为了使模型对各种输入物体的大小和形状更加稳健，每张训练图像都会按以下选项之一进行随机抽样：

- 使用整个原始输入图像。
- -对补丁进行采样，使其与对象的最小 jaccard 重叠度分别为 0.1、0.3、0.5、0.7 或 0.9
- 随机取样一个补丁.

每个采样斑块的大小为原始图像大小的 [0.1, 1]，长宽比介于和 2 之间。如果地面实况框的中心在采样斑块中，我们会保留其重叠部分。在上述采样步骤之后，每个采样斑块都会被调整为固定大小，并以 0.5 的概率水平翻转，此外还会应用一些与 [14] 中描述的类似的照片计量失真。

3 实验结果

基础网络：我们的实验全部基于在 ILSVRC CLS-LOC 数据集 [16] 上预先训练过的 VGG16 [15]。与 DeepLab-LargeFOV 算法[17] 类似，我们将 fc6 和 fc7 转换为卷积层，对 fc6 和 fc7 的参数进行子采样，将 pool5 从 2 × 2 - s2 改为 3 × 3 - s1，并使用 a trous` 算法 [18]填补 "漏洞"。我们删除了所有滤波层和 fc8 层。我们使用 SGD 算法对得到的模型进行微调，初始学习率为 10- ，动量为 0.9，权重衰减为 0.0005，批量大小为 32。每个数据集的学习率衰减策略略有不同，我们将在后面详细介绍。完整的训练和测试代码基于 Caffe [19]，开源地址为：https://github.com/weiliu89/caffe/tree/ssd 。

3.1 PASCAL VOC2007

在这个数据集上，我们在 VOC2007 测试（4952 幅图像）上与 Fast R-CNN [6] 和 Faster R-CNN [2] 进行了比较。所有方法都在相同的预训练 VGG16 网络上进行微调。

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 conv11 2 预测位置和可信度。我们在 conv4 33 上设置了比例为 0.1 的默认方框。我们使用 "xavier "方法[20]初始化所有新添加卷积层的参数。对于 conv4 3、conv10 2 和 conv11 2，我们只在每个特征图位置设置了 4 个默认方框--省略了和 3 的纵横比。对于所有其他图层，我们按照 2.2 节所述设置了 6 个默认方框。正如文献[12]所指出的，conv4 3 的特征尺度与其他层不同，因此我们使用文献[12]中介绍的 L2 归一化技术，将特征图中每个位置的特征法线缩放为 20，并在反向传播过程中学习该尺度。我们使用 10-3 学习率进行 40k 次迭代，然后以 10-4 和 10-5 继续进行 10k 次迭代训练。表 1 显示，在 VOC2007 trainval

上进行训练时，低分辨率 SSD300 模型的准确度已经高于快速 R-CNN 模型。当我们在更大的 512×512 输入图像上训练 SSD 时，它的准确度更高，比快速 R-CNN 高出 1.7% mAP。如果我们用更多（即 07+12）的数据来训练 SSD，我们会发现 SSD300 已经比 Faster R-CNN 高出 1.1%，而 SSD512 则高出 3.6%。如果我们采用第 3.4 节所述的在 COCO trainval35k 上训练的模型，并用 SSD512 在 07+12 数据集上对其进行微调，我们将获得最佳结果：81.6% mAP。

为了更详细地了解两种 SSD 模型的性能，我们使用了 [21] 中的检测分析工具。图 3 显示，SSD 可以高质量地检测到各种物体类别（大面积白色区域）。其大部分有把握的检测都是正确的。召回率约为 85-90%，在 "弱"（0.1 jaccard 重叠）标准下召回率更高。与 R-CNN [22] 相比，SSD 的定位误差更小，这表明 SSD 能够更好地定位物体，因为它直接学习回归物体形状和分类物体类别，而不是使用两个分离的步骤。然而，固态定位系统对相似物体类别（尤其是动物）的混淆较多，部分原因是我们共享多个类别的位置。图 4 显示，SSD 对边框大小非常敏感。换句话说，它在处理较小物体时的表现要比处理较大物体时差很多。这并不奇怪，因为这些小物体在最顶层可能甚至没有任何信息。增大输入尺寸（例如从 300×300 增大到 512×512）有助于改善对小物体的检测，但仍有很大的改进空间。从积极的一面来看，我们可以清楚地看到 SSD 在大型物体上的表现非常出色。而且，由于我们在每个特征地图位置使用了不同长宽比的默认方框，因此它对不同物体的长宽比都非常稳定。

3.2 模型分析

为了更好地理解 SSD，我们进行了对照实验，以检查每个组件对性能的影响。在所有实验中，我们使用相同的设置和输入大小（300 × 300），除了对设置或组件进行特定的更改。

数据扩增至关重要。快速和更快 R-CNN 使用原始图像和水平翻转图像进行训练。我们使用了一种更广泛的采样策略，类似于 YOLO [5]。表 2 显示，使用这种采样策略，我们可以提高 8.8% 的 mAP。我们不知道我们的采样策略对 Fast 和 Faster R-CNN 有多大益

处，但它们的益处可能较小，因为它们在分类过程中使用了一个特征池步骤，该步骤在设计上对物体平移具有相对的鲁棒性。

默认方框形状越多越好。如第 2.2 节所述，我们默认每个位置使用 6 个默认方框。如果去掉两个和三个长宽比的方框，性能会下降 0.6%。如果再去掉两个和两个长宽比的方框，性能又会下降 2.1%。使用各种默认方框形状似乎能让网络更轻松地完成预测方框的任务。

Atrous 更快。如第 3 节所述，我们按照 DeepLab-LargeFOV [17]，使用了子采样 VGG16 的 atrous 版本。如果我们使用完整的 VGG16，保留 2×2-s2 的 pool5，不使用 fc6 和 fc7 的子采样参数，并添加 conv5 3 进行预测，结果大致相同，而速度则要慢 20%。

不同分辨率的多个输出层效果更好。SSD 的一个主要贡献是在不同的输出层上使用不同比例的默认方框。为了衡量所获得的优势，我们逐步移除图层并比较结果。为了进行公平比较，每移除一层，我们都会调整默认方框的平铺方式，以保持方框总数与原始方框数（8732）相似。具体做法是在剩余图层上堆叠更多尺度的方框，并在必要时调整方框的尺度。我们并没有针对每种设置对平铺进行详尽的优化。表 3 显示，层数越少，准确率越低，从 74.3 单调下降到 62.4。当我们在一个图层上堆叠多个尺度的方框时，很多方框都位于图像边界上，需要小心处理。我们尝试了 Faster R-CNN [2] 中使用的策略，即忽略边界上的方框。我们观察到一些有趣的趋势。例如，如果我们使用非常粗糙的特征图（如 conv11 2 (1 × 1) 或 conv10 2 (3×3)），会大大降低性能。原因可能是剪枝后，我们没有足够的大方框来覆盖大型物体。当我们主要使用分辨率更高的地图时，性能又开始提高，因为即使在剪枝后，仍然有足够数量的大方框。如果我们只使用 conv7 进行预测，性能则最差，这更加说明了将不同尺度的方框分散到不同图层是至关重要的。此外，由于我们的预测并不像文献[6]中那样依赖于 ROI 池，因此我们不会遇到低分辨率特征图中的折叠箱问题[23]。SSD 架构结合了来自不同分辨率特征图的预测结果，在使用较低分辨率输入图像的同时，实现了与 Faster R-CNN 相媲美的准确性。

3.3 PASCAL VOC2012

除了使用 VOC2012 trainval 和 VOC2007 trainval 及 test（21503 幅图像）进行训练，并在 VOC2012 test（10991 幅图像）上进行测试外，我们使用了与上述 VOC2007 基本实验相同的设置。我们以 10-3 的学习率对模型进行 60k 次迭代训练，然后以 10-3 的学习率对模型进行 20k 次迭代训练。表 4 显示了 SSD300 和 SSD5124 模型的结果。我们看到了与 VOC2007 测试相同的性能趋势。与 Fast/Faster RCNN 相比，我们的 SSD300 提高了准确性。将训练和测试图像的大小增加到 512×512 后，我们的准确率比 Faster R-CNN 提高了 4.5%。与 YOLO 相比，SSD 的准确度明显更高，这可能是由于使用了来自多个特征图的卷积缺省盒以及我们在训练过程中的匹配策略。根据 COCO 训练的模型进行微调后，我们的 SSD512 达到了 80.0% 的 mAP，比 Faster R-CNN 高出 4.1%。

3.4 COCO

为了进一步验证 SSD 框架，我们在 COCO 数据集上训练了 SSD300 和 SSD512 架构。由于 COCO 中的对象往往比 PASCAL VOC 中的小，因此我们在所有层中都使用了较小的默认框。我们沿用了第 2.2 节中提到的策略，但现在最小的默认方框的比例为 0.15，而不是 0.2，而 conv4 3 默认方框的比例为 0.07（例如，300 × 300 图像的比例为 21 像素）。

我们使用 trainval35k [24] 进行训练。我们首先以 10-3 的学习率对模型进行 16 万次迭代训练，然后以 10-4 和 10-5 的学习率分别继续训练 4 万次和 4 万次。表 5 显示了 test-dev2015 的结果。与我们在 PASCAL VOC 数据集上观察到的结果类似，SSD300 在 mAP@0.5 和 mAP@[0.5:0.95]方面都优于快速 R-CNN。SSD300 的 mAP@0.75 与 ION [24] 和 Faster R-CNN [25] 相似，但在 mAP@0.5 中表现较差。将图像大小增加到 512 × 512 后，我们的 SSD512 在这两个标准上都优于 Faster R-CNN [25]。有趣的是，我们发现 SSD512 在 mAP@0.75 中比 Faster R-CNN 好 5.3%，但在 mAP@0.5

中只比 Faster R-CNN 好 1.2%。我们还观察到，SSD512 在大型物体的 AP（4.8%）和 AR（4.6%）方面的表现要好得多，但在小型物体的 AP（1.3%）和 AR（2.0%）方面的改进相对较少。与 ION 相比，大型物体和小型物体在 AR 方面的改进更为相似（5.4% 对 3.9%）。我们推测，由于 Faster R-CNN 在 RPN 部分和快速 R-CNN 部分都执行了两个方框细化步骤，因此它在较小物体上与 SSD 相比更具竞争力。在图 5 中，我们展示了使用 SSD512 模型在 COCO test-dev 上的一些检测示例。

3.5 ILSVRC 初步结果

我们在 ILSVRC DET 数据集 [16] 中采用了与 COCO 相同的网络架构。我们使用 ILSVRC2014 DET train 和 val1 训练 SSD300 模型，与 [22] 中使用的一样。我们首先以 10-3 的学习率对模型进行 320k 次迭代训练，然后以 10-4 的学习率继续训练 80k 次迭代，以 10-5 的学习率继续训练 40k 次迭代。我们可以在 val2 集上实现 43.4 mAP [22]。这再次验证了 SSD 是一种用于高质量实时检测的通用框架。

3.6 提高小物体精度的数据扩增

如果没有像 Faster R-CNN 那样的后续特征重采样步骤，对于 SSD 来说，小物体的分类任务相对较难，我们的分析也证明了这一点（见图 4）。第 2.2 节中描述的数据增强策略有助于显著提高性能，尤其是在 PASCAL VOC 等小型数据集上。该策略生成的随机作物可以看作是一种 "放大 "操作，可以生成许多更大的训练示例。为了实现 "缩小 "操作，生成更多小的训练示例，我们在进行任何随机裁剪操作之前，首先将图像随机放置在一个原始图像大小为 16× 的画布上，画布上填满了平均值。由于引入了这种新的 "扩展 "数据增强技巧，我们获得了更多的训练图像，因此训练迭代次数必须增加一倍。如表 6 所示，在多个数据集中，我们看到 mAP 持续增加了 2%-3%。具体而言，图 6 显示，新的扩展技巧显著提高

了小物体的性能。这一结果凸显了数据增强策略对最终模型准确性的重要性。

改进 SSD 的另一种方法是设计更好的默认方框平铺，使其位置和比例与特征图上每个位置的感受野更好地匹配。我们将这一问题留待今后研究。

3.7 推理时间

考虑到我们的方法会产生大量的方框，因此在推理过程中必须有效地执行非最大抑制（nms）。通过使用 0.01 的置信度阈值，我们可以过滤掉大部分方框。然后，我们在每个类别中应用 jaccard 重叠率为 0.45 的 nms，并保留每幅图像中前 200 个检测结果。在 SSD300 和 20 个 VOC 类别中，每个图像的这一步骤耗时约为 1.7 毫秒，接近所有新添加层的总耗时（2.4 毫秒）。我们使用配备英特尔至强 E5-2667v3@3.20GHz 的 Titan X 和 cuDNN v4 测量了批量大小为 8 的速度。

表 7 显示了 SSD、Faster R-CNN[2] 和 YOLO[5] 之间的比较。我们的 SSD300 和 SSD512 方法在速度和准确度上都优于 Faster R-CNN。虽然 Fast YOLO[5] 可以以 155 FPS 的速度运行，但它的精确度却低了近 22% mAP。据我们所知，SSD300 是第一个达到 70% 以上 mAP 的实时方法。需要注意的是，大约 80% 的前向时间都花在了基础网络上（在我们的案例中为 VGG16）。因此，使用更快的基础网络可以进一步提高速度，从而有可能使 SSD512 模型也具有实时性。

4 相关工作

图像中的物体检测有两类成熟的方法，一类基于滑动窗口，另一类基于区域建议分类。在卷积神经网络出现之前，这两种方法--可变形部件模型（DPM）[26] 和选择性搜索[1]--的性能相当。然而，R-CNN[22]将选择性搜索区域建议和基于卷积网络的后分类相结合，带来了巨大的改进，之后区域建议物体检测方法开始盛行。

最初的 R-CNN 方法已经过多种改进。第一套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，既

昂贵又耗时。SPPnet [9] 显著加快了原始 R-CNN 方法的速度。它引入了一个空间金字塔汇集层，对区域大小和尺度具有更强的鲁棒性，并允许分类层重复使用在多个图像分辨率下生成的特征图上计算的特征。快速 R-CNN [6] 对 SPPnet 进行了扩展，使其可以通过最小化信度和边界框回归的损失，对所有层进行端到端的微调。

第二类方法是利用深度神经网络提高建议生成的质量。在 MultiBox [7,8]等最新成果中，基于低层图像特征的 "选择性搜索 " 区域建议被由单独的深度神经网络直接生成的建议所取代。这进一步提高了检测的准确性，但也导致了一定程度的复杂设置，需要训练两个神经网络，且它们之间存在依赖关系。更快的 R-CNN [2] 用从区域建议网络（RPN）中学习到的建议取代了选择性搜索建议，并引入了一种方法，通过交替微调这两个网络的共享卷积层和预测层，将 RPN 与快速 R-CNN 整合在一起。这样，区域建议被用来汇集中层特征，而最终分类步骤的成本则更低。我们的 SSD 与 Faster R-CNN 中的区域建议网络（RPN）非常相似，我们也使用一组固定的（默认）框进行预测，这与 RPN 中的锚框类似。但是，我们并不使用这些框来汇集特征和评估另一个分类器，而是同时为每个框中的每个对象类别生成一个分数。因此，我们的方法避免了将 RPN 与快速 R-CNN 合并的复杂性，而且更易于训练，速度更快，并可直接集成到其他任务中。

另一组方法与我们的方法直接相关，它们完全跳过了提议步骤，直接预测多个类别的边界框和可信度。OverFeat [4] 是滑动窗口方法的一个深度版本，它在知道底层对象类别的可信度后，直接从最顶部特征图的每个位置预测边界框。YOLO [5]使用整个最顶部特征图来预测多个类别的可信度和边界框（这些类别共享边界框）。我们的 SSD 方法属于这一类，因为我们没有提议步骤，而是使用默认的框。不过，我们的方法比现有方法更加灵活，因为我们可以在每个特征位置上使用不同长宽比的默认框，这些默认框来自不同比例的多个特征图。如果我们只在最顶层特征图的每个位置使用一个默认框，那么我们的固态硬盘就会具有与 OverFeat [4] 相似的架构；如果我们使用整个最顶层特征图，并添加一个全连接层来进行预测，而不是我们的卷积预测器，并且不明确考虑多种纵横比，那么我们就可以近似重现 YOLO [5]。

5 结论

本文介绍的 SSD 是一种适用于多个类别的快速单次物体检测器。我们模型的一个主要特点是使用多尺度卷积边界框输出，并将其连接到网络顶部的多个特征图上。通过这种表示方法，我们可以对可能的框形状空间进行有效建模。我们通过实验验证了在适当的训练策略下，精心选择更多的默认边界框可以提高性能。与现有方法 [5,7] 相比，我们建立的 SSD 模型对方框位置、比例和长宽比的预测采样至少多了一个数量级。我们证明，在相同的 VGG-16 基本架构下，SSD 在准确性和速度方面都优于最先进的物体检测器。在 PASCAL VOC 和 COCO 上，我们的 SSD512 模型在准确性方面明显优于最先进的 Faster R-CNN [2]，同时速度快 3 倍。我们的实时 SSD300 模型以 59 FPS 的速度运行，比当前的实时 YOLO [5] 更快，同时检测准确率也明显更高。

除了其独立的实用性之外，我们还相信，我们的单一且相对简单的 SSD 模型为采用物体检测组件的大型系统提供了一个有用的构件。未来一个很有前景的方向是探索将其作为使用递归神经网络的系统的一部分，以同时检测和跟踪视频中的物体。

6 致谢

这项工作始于谷歌的一个实习项目，后在 UNC 继续进行。我们要感谢亚历克斯-托舍夫（Alex Toshev）在讨论中提供的帮助，并对谷歌的图像理解团队和 DistBelief 团队表示感谢。我们还要感谢 Philip Ammirato 和 Patrick Poirson 提供的有益意见。我们感谢英伟达™（NVIDIA®）提供 GPU，并感谢国家自然科学基金 1452851、1446631、1526367 和 1533771 的支持。

References

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., Arbelaez, 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. Howard, A.G.: Some improvements on deep convolutional neural network based image classification. arXiv preprint arXiv:1312.5402 (2013)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: NIPS. (2015)
16. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV (2015)
17. 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)
18. Holschneider, M., Kronland-Martinet, R., Morlet, J., Tchamitchian, P.: A real-time algorithm for signal analysis with the help of the wavelet transform. In: Wavelets. Springer (1990) 286–297
19. 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. (2014)
20. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS. (2010)
21. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: ECCV 2012. (2012)
22. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)
23. Zhang, L., Lin, L., Liang, X., He, K.: Is faster r-cnn doing well for pedestrian detection. In: ECCV. (2016)
24. 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)
25. COCO: Common Objects in Context. http://mscoco.org/dataset/#detections-leaderboard (2016) [Online; accessed 25-July-2016].
26. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR. (2008)