SSD: Single Shot MultiBox Detector

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy,
Scott Reed, Cheng-Yang Fu, Alexander C. Berg

摘要

我们提出了一种在图片中使用单个深度神经网络进行物体检测的方法。我们的方法，名为 SSD，将边界框的输出空间离散为存在于特征图每个位置的一组具有不同长宽比和尺度的默认框。在预测时，网络为每个默认框中的每个物体类别的存在生成分数，并对框进行调整以更好的匹配物体形状。此外，该网络结合了来自具有不同分辨率的多个特征图的预测，以自然地处理各种大小的物体。相对于需要物体候选的方法，SSD 很简单，因为它完全消除了候选生成和后续像素重采样或特征重采样阶段，并将所有计算封装在单个网络中。这使得 SSD 易于训练并可以直接明了地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 与使用额外物体候选步骤的方法的准确性可以相提并论，同时 SSD 速度要快得多，同时为训练和推理提供统一的框架。SSD 在 VOC2007 *text* 上，以 300×300 的图片作为输入，在 Nvidia Titan X 上以 59 FPS 的速度达到了 74.3%mAP 的准确率，以 512×512 的图片作为输入，达到了 76.9%mAP 的准确率，优于同类最先进的 Faster R-CNN 模型。与其他单阶段方法相比，即使输入图像尺寸更小，SSD 依然具有更好的准确性。代码位于：https://github.com/weiliu89/caffe/tree/ssd。

1 简介

当前最先进的物体检测系统是如下方法的变体：假设边界框，为每个框重新采样像素或特征，并应用高质量分类器。自选择性搜索 [2] 以来，该管道一直在检测基准上占上风，截至目前在 PASCAL VOC、COCO 和 ILSVRC 检测上的领先结果，都是基于 Faster R-CNN[9] 的，尽管有更深的特征，如 [10]。虽然准确，但这些方法对于嵌入式系统来说计算量太大，即使使用高端硬件，对于实时应用程序来说也太慢。这些方法的检测速度通常以每帧秒数 (SPF) 为单位，甚至是最快的高精度检测器 Faster R-CNN 的运行速度

仅为每秒 7 帧 (FPS)。已经有许多尝试通过攻击检测管道的每个阶段来构建更快的检测器（参见第 4 节中的相关工作），但到目前为止，显著提高速度只是以显著降低检测精度为代价。

本文提出了第一个基于深度网络的物体检测器，该检测器不对假设的边界框像素或特征进行重采样，同时与重采样的方法一样准确。这使得高准确率检测的速度有了明显的提高（在 VOC2007 *test* 上，SSD 以 59 FPS 的速度达到了 74.3% 的 mAP，而 Faster R-CNN 为 7 FPS，mAP 为 73.2%，YOLO 为 45 FPS, mAP 63.4%）。速度的根本提高来自于消除了边界框候选和随后的像素或特征重采样阶段。我们不是第一个这样做的（参见 [1, 11]），但通过增加一系列的改进，我们的方法的准确率较以前的尝试有了显著提高。我们的改进包括使用小卷积核来预测物体类别和边界框位置的偏移，为不同的长宽比检测使用单独的预测器（核），并将这些核应用于网络后期的多个特征图，来在多个尺度上进行检测。通过这些修改——特别是在不同尺度上使用多层预测——我们可以使用相对较低的分辨率输入实现高精确度，并进一步提高了检测速度。虽然这些贡献独立来看可能很小，但我们注意到所产生的系统将 PASCAL VOC 的实时检测精度从 YOLO 的 63.4% mAP 到我们的 SSD 的 74.3% mAP。这比最近非常引人注目的关于残差网络 [10] 的工作在检测精度上有着更大的相对改善。此外，高质量检测的速度的显著提高可以扩大计算机视觉的使用范围。

我们将我们的贡献总结如下：

- 我们介绍了 SSD，一种多类别的单次检测器，它比以前的单次检测器（YOLO）更快，而且明显更准确，实际上与进行显示区域候选和集合的较慢技术一样准确（包括 Faster R-CNN）。
- SSD 的核心是使用应用于特征图上的小卷积核预测类别分数和固定的默认边界框的偏移量。
- 为了达到较高的检测精度，我们从不同尺度的特征图中产生不同尺度的预测，并显式按长宽比分开预测。
- 这些设计特点导致了简单的端到端训练和高精确度，甚至在低分辨率的输入图像上也是如此，进一步改善了速度与精确度的权衡。
- 实验包括在 PASCAL VOC、COCO 和 ILSVRC 上对不同输入尺寸的模型进行时间和精度分析，并与一系列最新的最先进的方法进行比较。

2 The Single Shot Detector (SSD)

本节将介绍我们提出的 SSD 检测框架（2.1节）和相关的训练方法（2.2节）。之后，第 3 节将介绍了对应数据集的具体模型细节和实验结果。

2.1 模型

SSD 方法基于一个前向传播卷积网络，它产生一个固定大小的边界框集合，并对这些框中存在的物体类别实例进行评分，然后通过一个非最大抑制步骤来产生最终的检测结果。前期的网络层基于用于高质量图像分类的标准结构（在分类层之前截断），我们将其称之基础网络。然后，我们向网络添加辅助结构，以产生具有以下关键特征的检测结果：

用于检测的多尺度特征图 我们在截断的基础网络的末端添加卷积特征层。这些层的大小逐渐减少，并允许在多个尺度上预测检测。预测检测的卷积模型对于每个特征层都是不同的（参考 Overfeat[1] 和 YOLO[11]，它们在单一尺度的特征图上进行操作）。

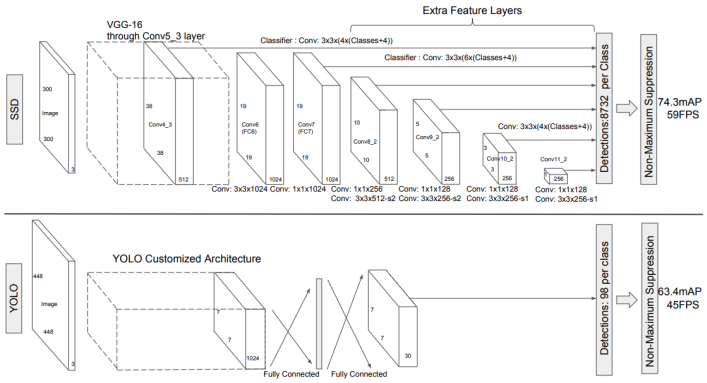
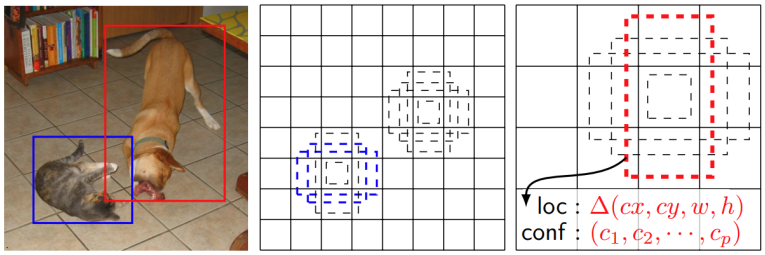


图 1: 两个单次检测模型之间的比较：SSD 和 YOLO[11]。我们的 SSD 模型在基础网络的末端添加了几个特征层，用于预测不同尺度和长宽比的默认框的偏移量及其相关的置信度。具有 300×300 输入大小的 SSD 在 VOC2007$test$ 中的准确性显着优于其对应的 448×448YOLO，同时还提高了速度。

用于检测的卷积预测器 每个添加的特征层（或来自基础网络的现有特征层）可以使用一组卷积滤波器产生一组固定的检测预测。这些在图1中 SSD 网络架构的顶部表示。对于具有 p 个通道的大小为 $m \times n$ 的特征层，预测潜在检测参数的基本元素是一个 $3 \times 3 \times p$ 小卷积核，它产生类别的分数，或相对于默认框坐标的形状偏移。在应用卷积核的 $m \times n$ 位置中的每一个位置，它都会产生一个输出值。边界框偏移量输出值是相对于每个特征图位置的默认框位置测量的（参见 YOLO[11] 的架构，该架构在此步骤中使用中间全连接层而不是卷积滤波器）。



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

图 2: **SSD 框架**。(a) SSD 在训练期间只需要每个对象的输入图像和 ground truth 框。以卷积方式，我们在不同尺寸的几个特征图中（例如（b）和（c）中的 8×8 和 4×4）的每个位置评估一小组（例如 4 个）不同长宽比的默认框。对于每个默认框，我们预测形状偏移和所有物体类别（$(c_1, c_2, \ldots, c_p)$）的置信度。在训练时，我们首先将这些默认框与 ground truth 框匹配。例如，我们将两个默认框与猫匹配，一个与狗匹配，将它们视为正例，其余视为负例。模型损失是定位损失（例如 Smooth L1 [5]）和置信度损失（例如 Softmax）的加权和。

默认框和长宽比 我们将一组默认边界框与每个特征图单元相关联，用于网络顶部的多个特征图。默认框以卷积方式平铺特征图，这样每个框相对于其对应单元格的位置是固定的。在每个特征图单元格中，我们预测相对于单元格中默认框形状的偏移量，以及每类分数，这些分数表明每个框内是否存在该类的实例。具体来说，对于给定位置的 k 个框中的每个框，我们计算 c 类分数和相对于原始默认框形状的 4 个偏移量。这导致总共 $(c+4)k$ 个卷积核应用于特征图中的每个位置，为 $m \times n$ 的特征图产生 $(c+4)kmn$ 个输出。

有关默认框的说明，请参阅图2。我们的默认框类似于 Faster R-CNN[9] 中使用的锚框，但是我们将它们应用于多个不同分辨率的特征图。在几个特征图中允许不同的默认框形状让我们有效地离散化可能的输出框形状的空间。

2.2 训练

训练 SSD 与训练使用区域候选的典型检测器之间的主要区别在于，需要将 ground truth 信息分配给固定检测器输出集合中的特定输出。YOLO[11] 中的训练以及 Faster R-CNN[9] 和 MultiBox[3] 的区域候选阶段也需要某种版本的分配。一旦确定了此分配，就可以端到端地应用损失函数和反向传播。训练还涉及选择一组默认框和尺度进行检测，以及难负例挖掘和数据增强策略。

匹配策略 在训练期间，我们需要确定哪些默认框对应于 ground truth 检测并相应地训练网络。对于每个 ground truth 框，我们从位置、长宽比和比例不同的默认框中进行选择。我们首先将每个 ground truth 框与具有最佳 jaccard 重叠的默认框匹配（如 MultiBox[3]）。与 MultiBox 不同，我们接下来将默认框匹配到任何具有高于阈值 (0.5) 的 jaccard 重叠的 ground truth。这简化了学习问题，允许网络为多个重叠默认框预测高分，而不是要求它只选择重叠最大的一个。

训练目标 SSD 训练目标源自 MultiBox 目标 [3, 4]，但扩展到处理多个对象类别。让 $x_{ij}^{p} = \{1, 0\}$ 是将第 i 个默认框与类别为 p 的第 j 个 ground truth 框匹配的指示符。在上面的匹配策略中，我们可以有 $\sum_i x_{ij}^{p} \geq 1$。整体目标损失函数是定位损失（loc）和置信损失（conf）的加权和：

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

其中 N 是匹配了的默认框数目。如果 $N = 0$，我们将损失设置为 0。定位损失是预测框 (l) 和地面实况框 (g) 参数之间的 Smooth L1 损失 [5]。与 Faster R-CNN [9] 类似，我们回归到默认边界框 (d) 的中心 (xc, cy) 及其宽

度 (w) 和高度 (h) 的偏移量。

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

置信度损失是多类置信度 (c) 上的 softmax 损失。

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

并且权重项 α 通过交叉验证设置为 1。

为默认框选择尺度和长宽比 为了处理不同的对象尺度，一些方法 [1, 6] 建议处理不同的尺寸图像，然后组合结果。然而，通过在单个网络中利用来自多个不同层的特征图进行预测，我们可以模拟相同的效果，同时还可以在所有对象尺度上共享参数。以前的工作 [8] 表明，由于较低层捕获了输入对象的更多细节，所以使用来自较低层的特征图可以提高语义分割质量。类似地，[7] 表明添加从特征图中汇集的全局上下文可以帮助平滑分割结果。受这些方法的启发，我们使用底层和高层特征图进行检测。图2显示了框架中使用的两个示例特征图（8×8 和 4×4）。在实践中，我们可以以小计算开销来使用更多的层。

已知来自网络内不同级别的特征图具有不同的（经验上来说）感受野大小 [13]。幸运的是，在 SSD 框架内，默认框不需要对应每一层的实际感受野。我们设计了默认框的平铺，以便特定的特征图学会对对象的特定尺寸做出响应。假设我们要使用 m 个特征图进行预测。每个特征图的默认框的尺寸计算如下：

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

其中 $s_{\min}$ 为 0.2，$s_{\max}$ 为 0.9，这意味着最低层的尺度为 0.2，最高层的尺度为 0.9，并且中间的所有层都是均匀间隔的。我们对默认框施加不同的长

宽比，并将它们表示为 $a_r \in \left\{1, 2, 3, \frac{1}{2}, \frac{1}{3}\right\}$。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（$h_k^a = s_k\sqrt{a_r}$）。对于长宽比为 1 的情况，我们还添加了一个默认框，其尺寸为 $s'_k = \sqrt{s_k s_{k+1}}$，从而导致特征图的每个位置有 6 个默认框。我们将每个默认框的中心设置为 $\frac{i+0.5}{|f_k|}, \frac{j+0.5}{|f_k|}$，其中 $|f_k|$ 是第 k 个正方形特征图的大小，$i, j \in [0, |f_k|)$。在实践中，还可以设计默认框的分布以最适合特定数据集。如何设计最佳平铺也是一个悬而未决的问题。

通过结合来自多个特征图的所有位置的具有不同尺寸和长宽比的所有默认框的预测，我们有一组多样化的预测，涵盖各种输入对象大小和形状。例如，在图2中，狗与 4×4 特征图中的默认框匹配，但没有与 8×8 特征图中的任何默认框匹配。这是因为这些框具有不同的尺度并且与狗框不匹配，因此在训练期间被视为负例。

难负例挖掘 在匹配步骤之后，大多数默认框都是负例，当可能的默认框数量很大时尤其如此。这引入了正负训练样例之间的显著不平衡。我们没有使用所有的负例，而是使用每个默认框的最高置信度损失对它们进行排序，并选择最高的那些，以便负例和正例之间的比率最多为 3:1。我们发现这会导致更快的优化和更稳定的训练。

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

- 使用全部原始输入图片。
- 采样得到补丁，使得与物体的最小 jaccard 重叠为 0.1、0.3、0.5、0.7 或 0.9。
- 随机采样一个补丁。

每个采样补丁的大小是原始图像大小的 $[0.1, 1]$，长宽比在 $\frac{1}{2}$ 和 2 之间。如果 ground truth 框的中心在采样块中，我们保留它的重叠部分。在上述采样步骤之后，除了应用一些类似于 [14] 中描述的光度失真之外，每个采样的补丁都被重塑为固定大小并以 0.5 的概率水平翻转。

References

[1] Pierre Sermanet et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *arXiv preprint arXiv:1312.6229* (2013).

[2] Jasper RR Uijlings et al. "Selective search for object recognition". In: *International journal of computer vision* 104.2 (2013), pp. 154–171.

[3] Dumitru Erhan et al. "Scalable object detection using deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 2147–2154.

[4] Christian Szegedy et al. "Scalable, high-quality object detection". In: *arXiv preprint arXiv:1412.1441* (2014).

[5] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[6] Kaiming He et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), pp. 1904–1916.

[7] Wei Liu, Andrew Rabinovich, and Alexander C Berg. "Parsenet: Looking wider to see better". In: *arXiv preprint arXiv:1506.04579* (2015).

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[9] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99.

[10] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[11] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.