SSD：单点多盒检测器

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

[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

Abstract. 本文提出一种使用单个深度神经网络检测图像中目标的方法。所提出的方法称为SSD，将边界框的输出空间离散化为一组默认框，这些框具有每个特征地图位置不同的长宽比和比例。在预测时，网络为每个默认框中每个对象类别的存在生成分数，并对框进行调整，以更好地匹配对象形状。此外，该网络结合了来自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对象。与需要目标建议框的方法相比，SSD是简单的，因为它完全消除了建议框的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个网络中。这使得SSD易于训练，并且可以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表明，SSD的准确性与利用额外的目标建议步骤的方法相比具有竞争力，并且速度快得多，同时为训练和推理提供了一个统一的框架。对于300 × 300输入，SSD在VOC2007测试中以59 FPS的速度在Nvidia Titan X上实现了74.3%的mAP [1]，对于512 × 512输入，SSD实现了76.9%的mAP，超过了最先进的Faster R-CNN模型。与其他单阶段方法相比，SSD在输入图像较小的情况下仍具有较高的精度。代码可在：https://github.com/weiliu89/caffe/tree/ssd获得。

Keywords: Real-time Object Detection; Convolutional Neural Network

1 简介

目前最先进的目标检测系统是以下方法的变体：假设边界框，对每个框的像素或特征进行重采样，并应用高质量分类器。自选择性搜索工作以来，该管道一直在检测基准上占主导地位[1]通过PASCAL VOC、COCO和ILSVRC检测的当前领先结果，所有都基于Faster R-CNN [2]，尽管具有更深层次的特征，如[3]。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，对于实时应用来说也太慢。这些方法的检测速度通常以每帧秒数（SPF）来衡量，即使是最快的高精度检测器，Faster R-CNN，也只能运行在每秒7帧（FPS）。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（请参见4节中的相关工作），但到目前为止，显著提高速度的代价是显著降低检测精度。

本文提出了第一个基于深度网络的目标检测器，它不为边界框假设重采样像素或特征，并且与其他方法一样准确。这显著提高了高精度检测的速

[1]在后续的实验中，我们使用改进的数据增强方案取得了更好的结果：在VOC2007上，300 × 300输入的mAP为77.2%,512 × 512输入的mAP为79.8%。详情请见3.6。

度（在VOC2007测试中，mAP为74.3%，帧率为59 FPS，相比之下，Faster R-CNN为7 FPS，mAP为73.2%，YOLO为45 FPS，mAP为63.4%）。速度的根本提升来自于消除边界框建议框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人（cf [4, 5]），但是通过添加一系列改进，我们设法比以前的尝试显著提高了准确性。所提出的改进包括使用一个小型卷积滤波器来预测物体类别和边界框位置中的偏移量，对不同的长宽比检测使用单独的预测器（滤波器），并将这些滤波器应用于网络后期的多个特征图，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，进一步提高检测速度。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实时检测的准确性从YOLO的63.4%mAP提高到了SSD的74.3%mAP。与最近非常引人注目的残差网络工作[3]相比，这是检测精度的一个较大的相对改进。此外，显著提高高质量检测的速度可以扩大计算机视觉有用的设置范围。

我们的贡献总结如下：

- 本文提出SSD，一种用于多个类别的单次检测器，比之前最先进的单次检测器（YOLO）更快，也明显更准确，实际上与执行显式区域建议和池化的较慢技术（包括更快的R-CNN）一样准确。
- SSD的核心是使用应用于特征图的小型卷积滤波器预测一组固定默认边界框的类别分数和框偏移量。
- 为了实现高检测精度，从不同尺度的特征图中产生不同尺度的预测，并按长宽比明确分离预测。
- 这些设计特性导致了简单的端到端训练和高精度，即使在低分辨率的输入图像上，也可以进一步提高速度和精度的权衡。
- 实验包括对具有不同输入大小的模型进行时间和精度分析，在PASCAL VOC、COCO和ILSVRC上进行评估，并与一系列最新的最先进的方法进行比较。

2 SSD （Single Shot Detector）

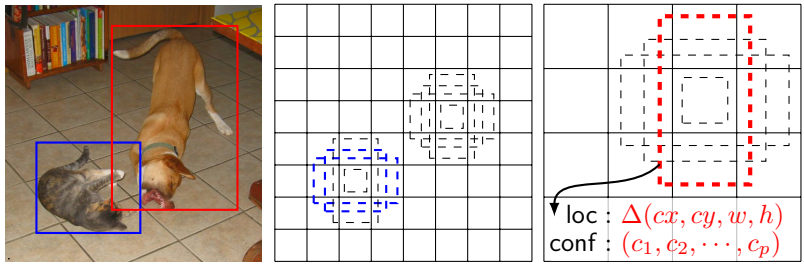
本节描述我们提出的SSD检测框架（2.1节）和相关的培训方法（2.2节）。随后，3节给出了特定数据集的模型细节和实验结果。

2.1 模型

SSD方法基于一个前馈卷积网络，该网络生成一个固定大小的边界框集合，并对这些框中存在的对象类实例进行分数，然后通过一个非最大抑制步骤来产生最终检测。早期的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），我们将其称为基础网络[2]。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：

用于检测的多尺度特征图我们将卷积特征层添加到截断的基础网络的末端。这些层的大小逐渐减小，并允许在多个尺度上进行检测预测。用于预测检测

[2] 我们使用VGG-16网络作为基础，但其他网络也应该产生良好的结果。



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

Fig. 1: **SSD**框架。 (a) SSD在训练期间只需要输入图像和每个对象的真实值框。以卷积方式评估了几个具有不同尺度的特征图中每个位置上不同长宽比的默认框的小集合（例如4）。8×8 而且4×4 （b)及(c)）。对于每个默认框，我们预测所有对象类别的形状偏移量和置信度($(c_1, c_2, \cdots, c_p)$)。在训练时，我们首先将这些默认框与真实框进行匹配。例如，我们将两个默认框与猫和狗进行了匹配，它们被视为正类，其余的被视为负类。模型损失是定位损失之间的加权和(例如，平滑L1 [6])和信心损失（例如Softmax）。

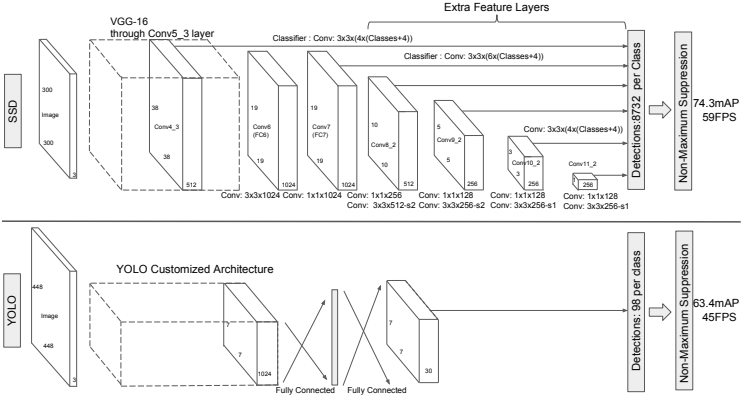


Fig. 2: SSD和YOLO两种单发检测模型的对比[5]。我们的SSD模型在基本网络的末端添加了几个特征层，用于预测不同尺度和纵横比的默认框的偏移量及其相关置信度。在VOC2007测试中，300×300输入尺寸的SSD在准确性方面明显优于448×448 YOLO，同时也提高了速度。

的卷积模型对于每个特征层都是不同的（在单尺度特征图上操作的*cf* Overfeat[4]和YOLO [5]）。

用于检测的卷积预测器每个添加的特征层（或来自基本网络的现有特征层）都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2中SSD网络架构的顶部显示。对于具有p通道的$m \times n$大小的特征层，用于预测潜在检测参数的基本元素是$3 \times 3 \times p$小内核，该内核可以生成类别的分数，或相对于默认框坐标的形状偏移。在应用内核的每个$m \times n$位置，它都会产生一个输出值。边界框偏移输出值是相对于相对于每个特征图位置的默认框位置进行测量的（参考YOLO [5]的架构，在此步骤中使用中间全连接层而不是卷积滤波器）。

默认框和纵横比我们将一组默认边界框与每个特征图单元关联，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，因此每个框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，我们预测相对于单元中默认框形状的偏移量，以及每个框中表示类实例存在的每个类分数。具体来说，对于在给定位置k外的每个框，我们计算c类分数和相对于原始默认框形状的偏移量4。这将总共产生$(c+4)k$过滤器，这些过滤器应用于特征映射中的每个位置，为$m \times n$特征映射产生$(c+4)kmn$输出。有关默认框的说明，请参阅图. 1。我们的默认框类似于Faster R-CNN [2]中使用的锚框，但我们将它们应用于不同分辨率的几个特征图。允许在几个特征映射中使用不同的默认框形状，可以有效地离散可能输出框形状的空间。

2.2 培训

训练SSD和训练使用区域建议框的典型检测器之间的关键区别是，真实值信息需要分配给固定检测器输出集中的特定输出。YOLO [5]的训练以及Faster R-CNN [2]和MultiBox [7]的区域建议阶段也需要一些版本。一旦确定了这个分配，损失函数和反向传播就会被端到端地应用。训练还包括为检测选择一组默认框和尺度，以及困难的负向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框对应于真实值检测，并相应地训练网络。对于每个真实值框，我们从随位置、宽高比和比例变化的默认框中进行选择。我们首先匹配每个真实值框与默认框的最佳杰卡德重叠（如在MultiBox [7]）。与MultiBox不同的是，我们将默认框与任何杰卡德重叠度高于阈值（0.5）的真实值进行匹配。这简化了学习问题，允许网络预测多个重叠默认框的高分，而不是要求它只选择重叠最大的一个。

培养目标 SSD训练目标源自MultiBox目标[7, 8]，但被扩展为处理多个对象类别。让$x_{ij}^p = \{1, 0\}$作为一个指示器，用于将i -th default框与类别p的j -th ground truth框进行匹配。在上面的匹配策略中，我们可以有$\sum_i x_{ij}^p \geq 1$。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：

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

其中N是匹配的默认框的数量。如果是$N = 0$，将loss设置为0。定位损失是预测框（l）和真实框（g）参数之间的平滑L1损失[6]。与Faster R-CNN [2]类似，

我们回归默认边界框（d）的中心（cx, cy）及其宽度（w）和高度（h）的偏移量。

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

置信度损失是多个类别置信度上的softmax损失（c）。

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

通过交叉验证将权重项α设置为1。

为默认框选择比例和宽高比 为了处理不同的物体尺度，有些方法[4, 9]建议以不同的尺寸处理图像，然后合并结果。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，同时在所有对象尺度上共享参数。之前的工作[10, 11]表明，使用来自较低层的特征图可以提高语义分割质量，因为较低层捕获了输入对象的更多精细细节。类似地，[12]表明添加来自特征图的全局上下文池可以帮助平滑分割结果。在这些方法的激励下，我们同时使用下特征图和上特征图进行检测。图1显示了框架中使用的两个范例特征图（和）。在实践中，我们可以以较小的计算开销使用更多。

已知网络中不同层次的特征图具有不同的（经验的）感受野大小[13]。幸运的是，在SSD框架中，默认框不必对应于每个层的实际感受野。设计了默认框的平铺，使特定的特征图学会对对象的特定比例做出响应。假设我们想使用特征映射进行预测。每个特征图默认框的比例计算如下：

其中为0.2,为0.9，意味着最低层的尺度为0.2，最高层的尺度为0.9，在这之间的所有层都是规则间隔的。我们为默认框施加不同的纵横比，并将它们表示为。我们可以计算每个默认框的宽度（）和高度（）。对于1的宽高比，我们还添加了一个比例为的默认框，导致每个特征地图位置有6个默认框。我们将每个默认框的中心设置为，其中是 -第一个方形特征图的大小。在实践中，人们还可以设计一个默认框的分布来最适合特定的数据集。如何设计最优的分块也是一个开放问题。

通过结合来自许多特征图所有位置的具有不同尺度和宽高比的所有默认框的预测，我们有一组不同的预测，涵盖了各种输入对象大小和形状。例如，在图1中，狗被匹配到特征映射中的默认框，但没有匹配到特征映射中的任何默认框。这是因为这些框具有不同的尺度，与狗框不匹配，因此在训练过程中被视为消极因素。

硬负挖掘 在匹配步骤之后，大多数默认框都是负数，特别是在可能的默认框数量很大的情况下。这导致了正负训练样本之间的显著不平衡。我们不是使用所有的负例，而是对每个默认框使用置信度损失最高的示例进行排序，并选择最高的示例，以便负例和正例之间的比例不超过3:1。我们发现这导致了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的鲁棒性更强，每个训练图像都通过以下选项之一随机采样：

- 使用整个原始输入图像。
- 对一个样本块进行采样，使其与物体的最小jaccard重叠为0.1、0.3、0.5、0.7或0.9。
- 随机抽取一个补丁。

每个采样块的大小为原始图像大小的[0.1,1]，长宽比在和2之间。如果真实值框的中心在采样块中，则保留其重叠部分。在上述采样步骤之后，每个采样块都被调整为固定大小，并以0.5的概率水平翻转，此外还应用了一些类似于[14]中描述的图像测量失真。

3 实验结果

基本网络 实验均基于在ILSVRC CLS-LOC数据集[16]上进行预训练的VGG16 [15]。类似于DeepLab-LargeFOV [17]，我们将fc6和fc7转换为卷积层，从fc6和fc7子采样参数，将pool5从更改为，并使用*à trous*算法[18]来填补“空洞”。我们移除所有的dropout层和fc8层。使用SGD对产生的模型进行微调，初始学习率、动量0.9、权重衰减0.0005和批量大小32。每个数据集的学习率衰减策略略有不同，稍后我们将详细描述。完整的训练和测试代码构建在Caffe [19]上，并在上开源。

3.1 PASCAL voc2007

在该数据集上，我们在VOC2007测试中与Fast R-CNN [6]和Faster R-CNN [2]（4952张图像）进行了比较。所有方法都在同一个预训练的VGG16网络上进行微调。

图2显示了SSD300模型的体系结构细节。我们使用conv4_3，conv7 (fc7), conv8_2, conv9_2，conv10_2和conv11_2来预测位置和置信度。我们在conv4_3[3]上设置缩放为0.1的默认框。我们使用“xavier”方法初始化所有新添加的卷积层的参数[20]。对于conv4_3，conv10_2和conv11_2，我们只在每个特征图位置关联4个默认框——省略$\frac{1}{3}$和3的纵横比。对于所有其他层，我们放置了6个默认框，如2.2节所述。由于，如[12]中指出的，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归一化技术将特征图中每个位置的特征范数缩放到20，并在反向传播过程中学习尺度。我们使用10^{-3}学习率进行40k迭代，然后使用10^{-4}和10^{-5}继续训练10k迭代。当在VOC2007 `trainval`上进行训练时，表1显示我们的低分辨率SSD300模型已经比Fast R-CNN更准确。当我们在一个更大的512 × 512输入图像上训练SSD时，它甚至更加准确，比Faster

[3] 对于SSD512模型，我们添加额外的conv12_2用于预测，将$s_{\min}$设置为0.15，在conv4_3上设置为0.07。

R-CNN高出1.7%mAP。如果我们用更多的数据（即07+12）训练SSD，我们看到SSD300已经比Faster R-CNN提高了1.1%，SSD512提高了3.6%。如果我们使用3.4节中描述的COCO `trainval35k`训练的模型，并使用SSD512在07+12数据集上对其进行微调，我们可以取得最好的结果：81.6%的mAP。

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
Fast [6]	07+12	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]	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
Faster [2]	07+12	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
Faster [2]	07+12+COCO	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9
SSD300	07	68.0	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
SSD300	07+12	74.3	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
SSD300	07+12+COCO	79.6	80.9	86.3	79.0	**76.2**	57.6	87.3	88.2	88.6	60.5	85.4	**76.7**	**87.5**	**89.2**	84.5	81.4	55.0	81.9	**81.5**	85.9	78.9
SSD512	07	71.6	75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0	67.4	82.3	83.9	79.4	76.6	44.9	69.9	69.1	78.1	71.8
SSD512	07+12	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
SSD512	07+12+COCO	**81.6**	**86.6**	**88.3**	**82.4**	76.0	**66.3**	**88.6**	**88.9**	**89.1**	**65.1**	**88.4**	73.6	86.5	88.9	**85.3**	**84.6**	**59.1**	**85.0**	80.4	**87.4**	**81.2**

Table 1: **PASCAL VOC2007**测试检测结果。Fast和Faster R-CNN都使用最小维度为600的输入图像。除了输入大小不同（300 × 300 vs. 512 × 512）外，这两个SSD模型具有完全相同的设置。很明显，更大的输入规模带来更好的结果，更多的数据总是有帮助的。数据："07"：VOC2007 `trainval`，"07+12"：VOC2007和VOC2012 `trainval`的结合。"07+12+COCO"：首先在COCO `trainval35k`上进行训练，然后在07+12上进行微调。

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[21]的检测分析工具。图3显示SSD可以高质量地检测各种对象类别（大面积白色区域）。它的大多数可信检测都是正确的。召回率约为85-90%，并且在'weak '（0.1 jaccard overlap）条件下更高。相比R-CNN [22]，SSD的定位误差更小，说明SSD由于直接学习回归物体形状和分类物体类别，而不是使用两个解耦的步骤，因此可以更好地定位物体。然而，SSD与相似的对象类别（特别是动物）有更多的混淆，部分原因是我们为多个类别共享位置。图4显示SSD对边界框大小非常敏感。换句话说，它在较小对象上的性能比在较大对象上的性能差得多。这并不奇怪，因为这些小对象甚至可能在最顶层没有任何信息。增加输入大小（例如从300 × 300到512 × 512）可以帮助提高对小目标的检测，但仍有很多改进的空间。从积极的方面来看，我们可以清楚地看到SSD在大型对象上的性能非常好。它对不同对象的长宽比非常鲁棒，因为我们在每个特征地图位置使用不同长宽比的默认框。

3.2 模型分析

为了更好地理解SSD，我们进行了对照实验，以检查每个组件如何影响性能。对于所有的实验，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指定的更改。

数据增强至关重要。越来越快的R-CNN使用原始图像和水平翻转来训练。我们使用了更广泛的采样策略，类似于YOLO[5]。表2显示了我们可以使用这种采样策略改进8.8%的mAP。我们不知道我们的采样策略将在多大程度上受益于快速

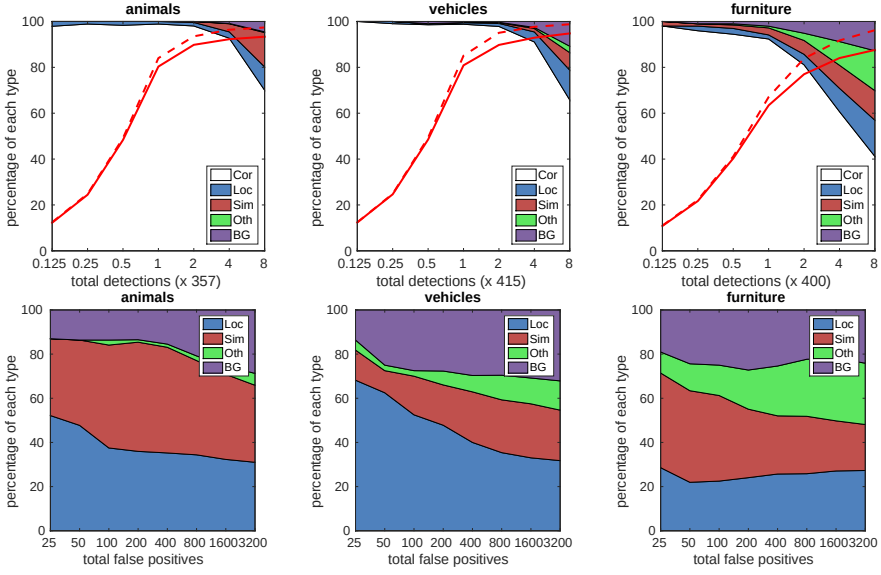


Fig. 3: 可视化**SSD512**在动物，车辆和家具上的性能从**VOC2007**测试。最上面一行显示了由于定位不良（Loc）、与相似类别（Sim）、与其他类别（Oth）或与背景（BG）混淆而正确（Cor）或假阳性检测的累积分数。红色实线反映了随着检测数量的增加，使用âstrongâ标准（0.5 jaccard重叠）的召回率的变化。红色虚线使用âweakâ标准（0.1杰卡尔德重叠）。底部一行显示了排名靠前的假阳性类型的分布。

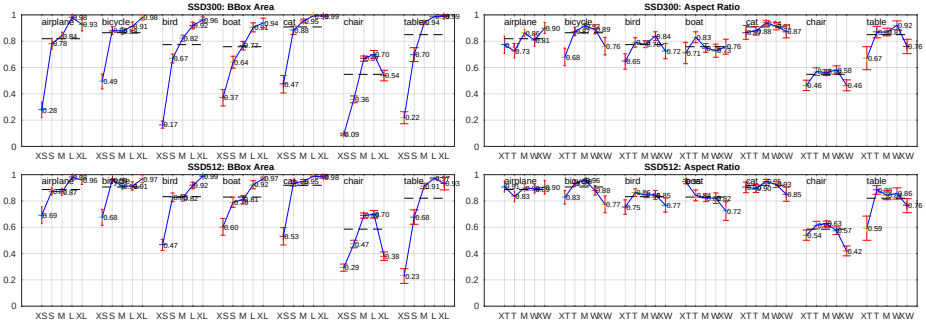


Fig. 4: 不同对象特征对**VOC2007**的敏感性及影响测试 设置使用**[21]**。左边的图显示了每个类别的BBox面积的影响，右边的图显示了长宽比的影响。键：BBox面积：XS=extra-small；S=小；M=中等；L=大；XL =特大号。宽高比：XT=超高/窄；T=高；M=中等；W=宽；XW =超宽。

	SSD300				
more data augmentation?		✔	✔	✔	✔
include $\{\frac{1}{2}, 2\}$ box?	✔		✔	✔	✔
include $\{\frac{1}{3}, 3\}$ box?	✔			✔	✔
use atrous?	✔	✔	✔		✔
VOC2007 `test` mAP	65.5	71.6	73.7	74.2	**74.3**

Table 2: 各种设计选择和组件对**SSD**性能的影响。

和更快的R-CNN，但它们可能受益较少，因为它们在分类过程中使用了特征池化步骤，从设计上来说，该步骤对目标转换相对鲁棒。

默认盒子形状越多越好。如2.2节所述，默认情况下，每个位置使用6个默认框。如果我们移除$\frac{1}{3}$和3个纵横比的盒子，性能会下降0.6%。通过进一步移除$\frac{1}{2}$和2个长宽比的盒子，性能又下降了2.1%。使用各种默认框形状似乎可以使网络更容易地预测框的任务。

Atrous更快。如3节所述，我们使用了subsampled VGG16的atrous版本，然后是DeepLab-LargeFOV [17]。如果我们使用完整的VGG16，保持$2\times2-s2$的pool5，而不是从fc6和fc7采样参数，并添加conv5_3进行预测，结果几乎相同，但速度大约慢20%。

Prediction source layers from:						mAP use boundary boxes?		# Boxes
conv4_3	conv7	conv8_2	conv9_2	conv10_2	conv11_2	Yes	No	
✔	✔	✔	✔	✔	✔	74.3	63.4	8732
✔	✔	✔	✔	✔		**74.6**	63.1	8764
✔	✔	✔	✔			73.8	68.4	8942
✔	✔	✔				70.7	69.2	9864
✔	✔					64.2	64.4	9025
	✔					62.4	64.0	8664

Table 3: 使用多个输出层的效果。

不同分辨率的多个输出层更好。SSD的一个主要贡献是在不同的输出层上使用不同规模的默认框。为了衡量获得的优势，逐步删除层并比较结果。为了进行公平的比较，每次我们删除一个图层时，我们都调整默认的方块平铺，以保持方块的总数与原始的（8732）相似。这是通过在剩余的层上堆叠更多的盒子比例，并在需要时调整盒子的比例来实现的。我们没有对每个设置进行彻底的优化。表3显示了层数更少时精度的下降，单调地从74.3下降到62.4。当我们将多个尺度的盒子堆叠在一个图层上时，许多在图像边界上，需要小心处理。我们尝试了在Faster R-CNN [2]中使用的策略，忽略边界上的框。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映射（例如conv11_2 (1×1)或conv10_2 (3×3)），则会在很大程度上损害性能。原因可能是在修剪之后，我们没有足够的大盒子来覆盖大型对象。当我们主要使用更精细的分辨率映射时，性能再次开始提高，因为即使修剪后仍然有足够数量的大框。如果我们只使用conv7进行预测，性能是最差的，这强化了将不同尺度的框分布在不同层上

的重要性。此外，由于我们的预测不像[6]那样依赖于ROI池化，我们在低分辨率特征图[23]中没有坍缩箱子问题。SSD架构结合了来自各种分辨率特征图的预测，以实现与Faster R-CNN相当的精度，同时使用较低分辨率的输入图像。

3.3 PASCAL voc2012

除了使用VOC2012 训练、VOC2007 训练和测试（21503张图片）进行训练，使用VOC2012 测试（10991张图片）进行测试之外，我们使用的设置与上面的基本VOC2007实验相同。我们用10^{-3}学习率进行60k次迭代训练模型，然后用10^{-4}进行20k次迭代。表4显示了SSD300和SSD512 [4]模型的结果。我们看到了与VOC2007测试相同的性能趋势。SSD300比Fast/Faster R-CNN提高了精度。通过将训练和测试图像大小增加到512×512，我们比Faster R-CNN准确率提高了4.5%。与YOLO相比，SSD的准确性要高得多，这可能是由于在训练过程中使用了多个特征图的卷积默认框和我们的匹配策略。当对COCO上训练的模型进行微调时，我们的SSD512达到了80.0%的mAP，比Faster R-CNN提高了4.1%。

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast[6]	07++12	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]	07++12	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
Faster[2]	07++12+COCO	75.9	87.4	83.6	76.8	62.9	59.6	81.9	82.0	91.3	54.9	82.6	59.0	89.0	85.5	84.7	84.1	52.2	78.9	65.5	85.4	70.2
YOLO[5]	07++12	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	07++12	72.4	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0	60.8	87.0	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5
SSD300	07++12+COCO	77.5	90.2	83.3	76.3	63.0	53.6	83.8	82.8	92.0	59.7	82.7	63.5	89.3	87.6	85.9	84.3	52.6	82.5	**74.1**	**88.4**	74.2
SSD512	07++12	74.9	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0
SSD512	07++12+COCO	**80.0**	**90.7**	**86.8**	**80.5**	**67.8**	**60.8**	**86.3**	**85.5**	**93.5**	**63.2**	**85.7**	**64.4**	**90.9**	**89.0**	**88.9**	**86.8**	**57.2**	**85.1**	72.8	**88.4**	**75.9**

Table 4: **PASCAL voc2012** 测试检测结果。Fast and Faster R-CNN使用最小尺寸600的图像，而YOLO的图像大小是448×448。资料来源："07++12"：VOC2007 训练、测试、VOC2012 训练联盟。"07++12+COCO"：首先在COCO `trainval35k`上训练，然后在07++12上进行微调。

3.4 椰子树

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中的对象往往小于PASCAL VOC，我们为所有层使用较小的默认框。我们遵循2.2节中提到的策略，但现在我们最小的默认框的比例为0.15，而不是0.2，conv4_3上的默认框的比例为0.07（例如300×300图像的21像素）[5]。

我们使用`trainval35k`[24]进行训练。我们首先用10^{-3}学习率训练模型进行160k次迭代，然后用10^{-4}和10^{-5}继续训练40k次迭代。表5显示了`test-dev2015`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都优于Fast R-CNN。SSD300有类似的mAP@0.75与ION [24]和Faster R-CNN [25]，但在mAP@0.5更差。通过将图像大小增加到512×512，我们的SSD512在这两个标准上都优于Faster R-CNN [25]。有趣的是，我们观察到SSD512在mAP@0.75中

[4] `http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?cls=mean&challengeid=11&compid=4`

[5] 对于SSD512模型，我们添加额外的conv12_2用于预测，将$s_{\min}$设置为0.1，在conv4_3上设置为0.04。

Method	data	Avg. Precision, IoU:			Avg. Precision, Area:			Avg. Recall, #Dets:			Avg. Recall, Area:		
		0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
Fast [6]	train	19.7	35.9	-	-	-	-	-	-	-	-	-	-
Fast [24]	train	20.5	39.9	19.4	4.1	20.0	35.8	21.3	29.5	30.1	7.3	32.1	52.0
Faster [2]	trainval	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [24]	train	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
Faster [25]	trainval	24.2	45.3	23.5	7.7	26.4	37.1	23.8	34.0	34.6	12.0	38.5	54.4
SSD300	trainval35k	23.2	41.2	23.4	5.3	23.2	39.6	22.5	33.2	35.3	9.6	37.6	56.5
SSD512	trainval35k	**26.8**	**46.5**	**27.8**	**9.0**	**28.9**	**41.9**	**24.8**	**37.5**	**39.8**	**14.0**	**43.5**	**59.0**

Table 5: **COCO`test-dev2015`**检测结果。

提高了5.3%，但在mAP@0.5中仅提高了1.2%。我们还观察到，对于大目标，它有更好的AP（4.8%）和AR(4.6%)，但对于小目标，AP（1.3%）和AR（2.0%）的改进相对较少。与ION相比，增强现实在大小目标上的改进更为相似（5.4%vs. 3.9%）。我们推测，Faster R-CNN在带有SSD的较小物体上更具竞争力，因为它在RPN部分和Fast R-CNN部分都执行了两个盒子细化步骤。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的初步结果

我们将与COCO相同的网络架构应用于ILSVRC DET数据集[16]。我们使用在[22]中使用的ILSVRC2014 DET `train`和`val1`来训练SSD300模型。我们首先用10^{-3}学习率训练模型进行320k次迭代，然后用10^{-4}和10^{-5}继续训练80k次迭代和40k次迭代。我们可以在`val2`集[22]上实现43.4 mAP。再次验证了SSD是一个通用的高质量实时检测框架。

3.6 小目标精度的数据增强

如果没有像Faster R-CNN那样的后续特征重采样步骤，SSD的小目标分类任务相对困难，如我们的分析所示（见图4）。2.2节中描述的数据增强策略有助于显著提高性能，特别是在PASCAL VOC等小型数据集上。该策略生成的随机裁剪可以被认为是一个“放大”操作，可以生成许多更大的训练样本。为了实现“缩小”操作，以创建更多的小训练示例，我们首先在进行任何随机裁剪操作之前，将原始图像大小的图像随机放置在16×的画布上，并填充平均值。因为通过引入这个新的“扩展”数据增强技巧，我们有了更多的训练图像，我们必须将训练迭代次数加倍。在多个数据集上，我们看到mAP持续增长了2%-3%，如表6所示。具体来说，图6显示了新的增强技巧显著提高了小对象的性能。这个结果强调了数据增强策略对最终模型精度的重要性。

改进SSD的另一种方法是设计更好的默认框平铺，以便其位置和比例与特征图上每个位置的感受野更好地对齐。我们把这个留给以后的工作。

3.7 推理时间

考虑到从该方法产生的大量框，在推理过程中有效地执行非极大值抑制（nms）是至关重要的。通过使用置信阈值0.01，我们可以过滤掉大多数框。然后，

Fig. 5: **COCO**的检测实例测试开发 采用**SSD512**模型。我们的检测分数高于0.6。每种颜色对应一个对象类别。

Method	VOC2007 test		VOC2012 test		COCO test-dev2015		
	07+12	07+12+COCO	07++12	07++12+COCO	trainval35k		
	0.5	0.5	0.5	0.5	0.5:0.95	0.5	0.75
SSD300	74.3	79.6	72.4	77.5	23.2	41.2	23.4
SSD512	76.8	81.6	74.9	80.0	26.8	46.5	27.8
SSD300*	77.2	81.2	75.8	79.3	25.1	43.1	25.8
SSD512*	**79.8**	**83.2**	**78.5**	**82.2**	**28.8**	**48.5**	**30.3**

Table 6: 当我们添加图像扩展数据增强技巧时在多个数据集上的结果。SSD300*和SSD512*是使用新数据增强训练的模型。

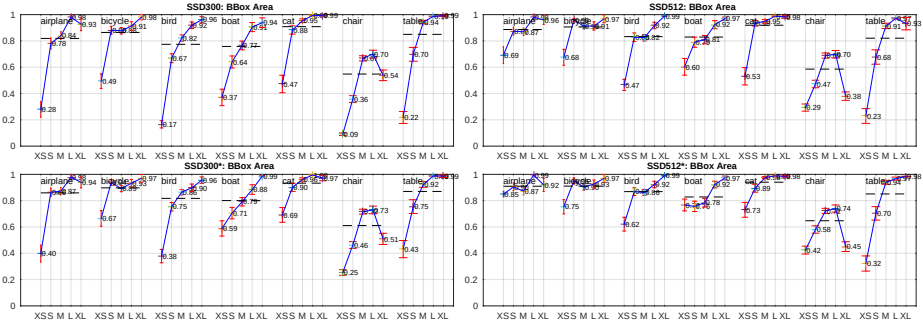


Fig. 6: **VOC2007**中新数据增强对目标大小的敏感性和影响测试 设置使用**[21]**。顶部行显示了每个类别的BBox区域对原始SSD300和SSD512模型的影响，底部行对应于使用新的数据增强技巧训练的SSD300*和SSD512*模型。很明显，新的数据增强技巧有助于显著地检测小目标。

我们应用nms，每个类的jaccard重叠为0.45，并保持每个图像的前200个检测结果。对于SSD300和20个VOC类，此步骤每个图像大约花费1.7 msec，这接近于所有新添加层所花费的总时间（2.4 msec）。我们使用Titan X和英特尔Xeon E5-2667v3@3.20GHz的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]之间的对比。SSD300和SSD512方法在速度和精度方面都优于更快的R-CNN。尽管Fast YOLO [5]可以以155 FPS的速度运行，但它的准确率低了近22%。据我们所知，SSD300是第一个实现70%以上mAP的实时方法。请注意，大约80%的转发时间花费在基础网络上（在我们的例子中是VGG16）。因此，使用更快的基础网络甚至可以进一步提高速度，这可能使SSD512模型也具有实时性。

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	$\sim 1000 \times 600$
Fast YOLO	52.7	155	1	98	448×448
YOLO (VGG16)	66.4	21	1	98	448×448
SSD300	74.3	46	1	8732	300×300
SSD512	76.8	19	1	24564	512×512
SSD300	74.3	59	8	8732	300×300
SSD512	76.8	22	8	24564	512×512

Table 7: **Pascal VOC2007**的结果测试。SSD300是目前唯一能够实现70%以上mAP的实时检测方法。通过使用更大的输入图像，SSD512在保持接近实时速度的同时，精度优于所有方法。

4 相关工作

现有的图像目标检测方法分为两类：基于滑动窗口的方法和基于区域建议分类的方法。在卷积神经网络出现之前，这两种方法的最先进技术——可变形部件模型（DPM）[26]和选择性搜索[1]——具有相当的性能。然而，在R-CNN [22]结合了选择性搜索区域建议和基于卷积网络的后分类带来的显著改进之后，区域建议目标检测方法开始流行。

原始的R-CNN方法经过了多种方式的改进。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]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池化层，该层对区域大小和尺度更鲁棒，并允许分类层重用在几种图像分辨率生成的特征图上计算的特征。Fast R-CNN [6]扩展了SPPnet，以便它可以通过最小化置信度和边界框回归的损失来微调端到端的所有层，这是首次在MultiBox [7]中引入的用于学习目地性。

第二组方法使用深度神经网络提高了候选框生成的质量。在最近的工作中，如MultiBox [7, 8]，基于低级图像特征的选择性搜索区域建议框被直接从单独的深度神经网络生成的建议框所取代。这进一步提高了检测精度，但导致了一些复杂的设置，需要训练两个神经网络，它们之间存在依赖关系。Faster R-CNN [2]用从区域候选网络（RPN）学习到的候选框替换选择性搜索框，并介绍了一种通过交替微调这两个网络的共享卷积层和预测层来集成RPN和Fast R-CNN的方法。这样，区域建议用于合并中层特征，并且最终分类步骤的开销更小。我们的SSD与Faster R-CNN中的区域建议网络（RPN）非常相似，因为我们也使用一组固定的（默认）框进行预测，类似于RPN中的锚框。但是，我们不是使用这些特征来汇集特征并评估另一个分类器，而是同时为每个框中的每个对象类别产生一个分数。所提出方法避免了将RPN与Fast R-CNN合并的复杂性，更容易训练，更快，并直接集成到其他任务中。

另一组与我们的方法直接相关的方法，完全跳过了建议步骤，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后，直接从最顶层特征图的每个位置预测一个边界框。YOLO [5]使用整个最顶层的特征图来预测多个类别的置信度和边界框（这些类别共享）。我们的SSD方法属于这一类，因为我们没有建议步骤，而是使用默认框。然而，所提出方法比现有方法更灵活，因为可以在不同尺度的多个特征图的每个特征位置上使用不同长宽比的默认框。如果我们只从最顶层的特征地图的每个位置使用一个默认框，我们的SSD将具有类似的架构OverFeat [4]；如果我们使用整个最顶层的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，并且不明确考虑多个长宽比，我们可以近似地复现YOLO [5]。

5 结论

本文提出了SSD，一种面向多类别的快速单点目标检测器。我们模型的一个关键特征是使用连接到网络顶部多个特征图的多尺度卷积边界框输出。这种表示方式使我们能够有效地对可能的盒子形状空间进行建模。通过实验验证了，给定适当的训练策略，精心选择的默认边界框数量更多，可以提高性能。构建的SSD模型比现有方法至少多一个数量级的框预测采样位置、尺度和长宽

比[5, 7]。实验表明，在相同的VGG-16基础架构下，SSD在精度和速度方面都与最先进的目标检测器相媲美。我们的SSD512模型在PASCAL VOC和COCO上的精度方面明显优于最先进的Faster R-CNN [2]，同时速度更快3×。所提出的实时SSD300模型以59 FPS的速度运行，比当前的实时YOLO [5]更快，同时产生明显优越的检测精度。

除了其独立的实用功能，我们相信我们的单体和相对简单的SSD模型为采用目标检测组件的大型系统提供了一个有用的构建块。一个有希望的未来方向是探索将其作为使用递归神经网络同时检测和跟踪视频中的目标的系统的一部分。

6 鸣谢

这项工作开始于谷歌的实习项目，并在UNC继续进行。我们要感谢Alex Toshev的讨论，感谢谷歌的图像理解和怀疑团队。我们也感谢Philip Ammirato和Patrick Poirson的有用评论。我们感谢NVIDIA提供gpu，并感谢NSF 1452851, 1446631, 1526367,1533771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., 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. 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)