SSD：单枪多盒检测器

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公司。[3]谷歌公司。[4]密歇根大学，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，将界线盒的输出空间离散为一组不同长宽比和比例的默认盒，每个特征图位置。在预测时，网络对每个默认框中的每个物体类别的存在产生评分，并对框进行调整以更好地匹配物体形状。此外，该网络结合了来自不同分辨率的多个特征图的预测，以自然地处理各种尺寸的物体。相对于需要物体提议的方法，SSD很简单，因为它完全消除了提议的生成和随后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的准确度与使用额外物体提议步骤的方法相比具有竞争力，而且速度更快，同时为训练和推理提供了一个统一的框架。对于300×300的输入，SSD在Nvidia Titan X上以59 FPS的速度在VOC2007测试中取得了74.3%的mAP[1]在Nvidia Titan X上进行的VOC2007测试中，SSD在59FPS的情况下实现了74.3%的mAP，而对于512×512的输入，SSD实现了76.9%的mAP，超过了同类的最先进的Faster R-CNN模型。与其他单阶段方法相比
即使输入图像尺寸较小，SSD的精确度也要好得多。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 简介

目前最先进的物体检测系统是以下方法的变种：假设边界框，重新取样每个框的像素或特征，并应用高质量的分类器。从Selec- tive Search工作[1]到目前

PASCAL VOC、COCO和ILSVRC检测的领先结果，都是基于Faster R-CNN[2]，尽管有更深的特征，如[3]，这种管道一直在检测基准上占优势。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，对于实时应用也太慢。

[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%。详情请见第3.6节。

这些方法的检测速度通常是以每帧秒数（SPF）来衡量的，即使是最快的高精度检测器，Faster R-CNN，也只能以每秒7帧（FPS）的速度运行。已经有很多人试图通过攻击检测管道的每个阶段来建立更快的检测器（见第4节的相关工作），但到目前为止，速度的大幅提高只是以检测精度的大幅下降为代价。

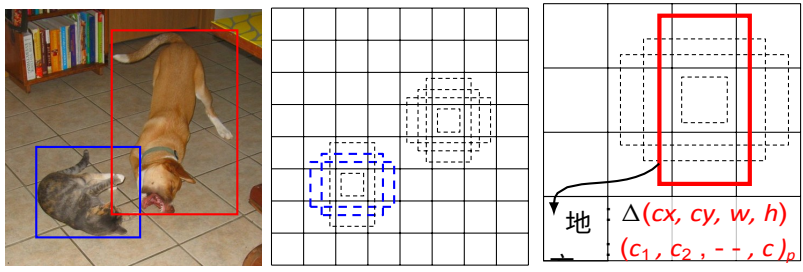
本文提出了第一个基于深度网络的物体检测器，该检测器不对边界盒假设的像素或特征进行重新采样，*并且*与采样的方法一样准确。这使得高精度检测的速度有了明显的提高（在VOC2007测试中，59 FPS，mAP 74.3%，与Faster R-CNN 7 FPS，mAP 73.2%或YOLO 45 FPS，mAP 63.4%）。速度的根本提高来自于消除了边界盒建议和随后的像素或图像重采样阶段。我们不是第一个这样做的人（参见[4,5]），但通过增加一系列的改进，我们设法比以前大大增加了准确性。我们的改进包括使用一个小型卷积滤波器来预测物体类别和边界框位置的偏移，为不同的长宽比检测使用单独的预测器（滤波器），并将这些滤波器应用于网络后期的多个特征图，以进行多尺度的检测。通过这些修改--特别是使用多层预测不同尺度--我们可以使用相对较低的分辨率输入实现高准确度，进一步提高检测速度。虽然这些贡献看起来很小，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到我们的SSD的74.3% mAP。这比最近非常引人注目的关于残余网络的工作[3]在检测精度上有了更大的相对提高。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环境范围。

我们将我们的贡献总结如下：

- 我们介绍了SSD，一个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，事实上与进行明确区域建议和集合的较慢技术（包括Faster 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的默认边界盒的类别分数和盒子偏移。
- 为了达到较高的检测精度，我们从不同尺度的特征图中产生不同尺度的预测，并明确地按长宽比分开预测。
- 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。
- 实验包括对在PASCAL VOC、COCO和ILSVRC上评估的不同输入规模的模型进行时序和精度分析，并与一系列最近的最先进的方法进行比较。

2 单发检测器（SSD）

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



(a) 带有GT盒的图像 (b) 8×8的特征图 (c) 4×4的特征图

图1：**SSD框架。**(a) 在训练过程中，SSD只需要一个输入图像和每个物体的地面真相框。以卷积的方式，我们在几个不同比例的特征图（例如（b）和（c）中的8 8和4 4）中的每个位置评估一小组（例如4）不同长宽比的默认盒子。对于每个默认的盒子，我们预测 - -
形状偏移和所有对象类别的置信度（$(c_1, c_2, , c_p)$）。
在训练时，我们首先将这些默认的盒子与地面真实盒子相匹配。对于
例如，我们将两个默认框与猫匹配，一个与狗匹配，这两个框被视为阳性，其余的被视为阴性。模型损失是定位损失（如Smooth L1[6]）和信心损失（如Softmax）之间的加权和。

2.1 模型

SSD方法是基于一个前馈卷积网络，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。早期的网络层是基于用于高质量图像分类的标准架构（在任何分类层之前被截断），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每个特征层都是不同的（*参考*Overfeat[4]和YOLO[5]，它们在单一尺度特征图上操作）。

用于检测的卷积预测器 每个添加的特征层（或可选择从基础网络中退出的特征层）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2中的SSD网络结构的顶部表示。对于具有p个通道的m n大小的特征层，预测潜在检测参数的基本el-ement是一个3 3 p*的小核*，产生一个类别的分数，或相对于默认框的形状偏移。

坐标。在应用内核的m n个位置，它产生一个输出值。界限盒偏移输出值是相对于默认的

[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。

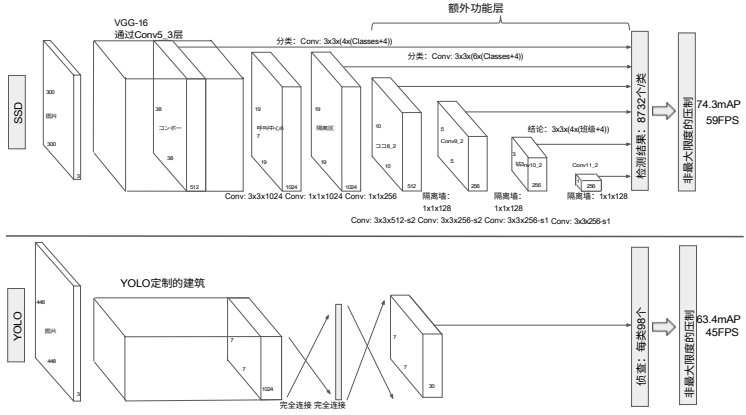


图2：两种单次检测模型的比较：SSD和YOLO[5]。我们的SSD模型在基础网络的末端增加了几个特征层，预测不同比例和长宽比的默认框的偏移量以及它们的相关置信度。在VOC2007测试中，输入规模为300 300的SSD在准确性上明显优于其448 448的YOLO同行，同时也提高了速度。

框的位置相对于每个特征图的位置（*参考*YOLO[5]的架构，该架构使用中间全连接层而不是卷积滤波器来完成这一步）。

默认框和纵横比 我们将一组默认的边界框与每个特征图单元联系起来，用于网络顶部的多个特征图。默认框以卷积方式对特征图进行平铺，因此，每个框相对于其相应单元的位置是固定的。在每个特征图单元，我们预测相对于该单元中的默认盒子形状的偏移量，以及表明这些盒子中的每个类别实例的存在的每类分数。具体来说，对于给定位置的k个盒子中的每一个，我们计算出c个类的分数和相对于原始盒子的4个偏移量。
默认的盒子形状。这就导致了在每一箱的周围总共有$(c+4)k$个过滤器被应用。
在特征图中的位置，对于一个m n的特征图，产生$(c + 4)kmn$的输出。关于默认框的说明，请参考图1。我们的默认框类似于Faster R-CNN[2]中使用的*锚定框*，但是我们将其应用于几个特征
不同分辨率的地图。在几个特征图中允许不同的默认盒形，让我们有效地将可能的输出盒形空间离散化。

2.2 培训

训练SSD和训练一个典型的使用区域提议的检测器之间的关键区别在于，需要将地面真实信息分配给检测器输出的固定集合中的特定输出。在YOLO[5]和

Faster R-CNN[2]和MultiBox[7]的区域提议阶段的训练中，也需要这样的某个版本。一旦这种分配被确定，损失函数和反向传播就会被端到端地应用。训练还包括选择默认的盒子和检测的尺度，以及硬性的负面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与地面真实检测相关，并相应地训练网络。对于每一个地面实况框，我们要从位置、长宽比和比例不同的默认框中选择。我们首先将每个地面实况框与具有最佳jaccard重叠的默认框相匹配（如MultiBox[7]）。与MultiBox不同的是，我们将默认的盒子与任何jaccard重叠度高于阈值（0.5）的地面真相相匹配。这简化了学习问题，允许网络预测多个重叠的默认框的高分，而不是要求它只选择有最大重叠的一个。

训练目标 SSD训练目标来自MultiBox目标。
在这个问题上，我们采用的是 "多维度 "的方法[7,8]，但是我们将其扩展到处理多个对象类别。让$x_{ij}^p = \{1, 0\}$是一个指标，用于将第i个缺省盒与第j个类别的地面真实盒相匹配。
在上述匹配策略中，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）和信心损失（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是匹配的默认盒的数量。定位损失是预测框（l）和地面真实框（g）参数之间的Smooth 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）以及其宽度（w）和高度（h）的偏移进行回归。

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

信任度损失是对多类信任度（c）的softmax损失。

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

并通过交叉验证将权重项α设为1。

选择默认框的比例和长宽比 为了处理不同的物体比例，一些方法[4,9]建议在不同的尺寸下处理图像，然后将结果合并。然而，通过利用单个网络中几个不同层的特征图进行预测，我们可以模仿同样的效果，同时也可以在所有物体尺度

上共享参数。以前的工作[10,11]表明，使用低层的特征图可以提高语义分割的质量，因为低层可以捕获输入物体的更多细节。同样，[12]表明，添加从特征图中汇集的全局背景有助于平滑分割结果。

在这些方法的启发下，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典型的特征图（8 8和4 4）。在实践中，我们可以使用更多的计算开销。

众所周知，来自网络内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，去错盒不一定要与每一层的实际感受野相对应。我们设计了默认框的平铺，使特定的特征图学会对物体的特定比例作出反应。假设我们想使用m个特征图进行预测。每个特征图的默认框的比例计算为：

$$s_k = s_{\text{闽}} + \frac{\text{smax} - \text{smin}}{m - 1}(k) - 1), \quad k \in [1, m] \tag{4}$$

其中$s_{\min}$为0.2，$s_{\max}$为0.9，意味着最低层的比例为0.2，最高层的比例为0.9，中间的所有层都是有规律的间隔。我们为默认的盒子施加不同的长宽比，并将其表示为$a_r \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$。我们可以计算宽度（$w_k^a = s_k\sqrt{a_r}$）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，其比例为$s_k' = \sqrt{s_k s_{k+1}}$，从而使每个特征图的位置有6个默认框。我们设定中心每个默认框的大小为$(\frac{i+0.5}{|f_k|}, \frac{j+0.5}{|f_k|})$，其中$|f_k|$是第k个方形特征的大小。$i, j \in [0, |f_k|)$。在实践中，人们也可以设计一个默认框的分布来最适合特定的数据集。如何设计出最佳的瓦片也是一个开放的问题。

通过结合许多特征图的所有位置的不同比例和长宽比的所有默认框的预测，我们有一个多样化的预测集，涵盖各种输入物体的大小和形状。例如，在图1中，狗与4 4特征图中的默认盒子相匹配，但与8 8特征图中的任何默认盒子都不匹配。这是因为这些盒子有不同的尺度，与狗的盒子不匹配，因此在训练中被认为是负面的。

硬性负面挖掘 在匹配步骤之后，大多数默认的盒子都是负面的，特别是当可能的默认盒子的数量很大时。这就在正面和负面的训练例子之间引入了一个明显的不平衡。我们不使用所有的负面例子，而是使用每个默认框的最高置信度损失对它们进行排序，并挑选出最重要的例子，使负面和正面之间的比例最多为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物体的尺寸和形状更加稳健，每张训练图像都通过以下选项之一进行随机采样：

- 使用整个原始输入图像。
- 对一个补丁进行采样，使其与物体的最小jaccard重叠度为0.1、0.3、0.5、0.7或0.9。

– 随机抽取一个补丁。

每个采样斑块的大小为原始图像大小的[0.1, 1]，长宽比在$\frac{1}{2}$和2之间。如果地面实况框的中心在采样补丁中，我们就保留其重叠部分。在上述取样步骤之后，每个取样斑块被调整为固定大小，并以0.5的概率进行水平翻转，此外还应用一些类似于[14]中描述的照片计量失真。

3 实验结果

基础网络 我们的实验都是基于VGG16[15]，它是在ILSVRC CLS-LOC数据集[16]上预训练的。与DeepLab-LargeFOV[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]来填补 "洞"。我们删除了所有辍学层和fc8层。我们使用SGD对得到的模型进行微调，初始学习率为$10^{-3}$，动量为0.9，权重衰减为0.0005，批次大小为32。每个数据集的学习率衰减策略都略有不同，我们将在后面描述细节。完整的训练和测试代码建立在Caffe[19]之上，并在以下网站开放源代码：`https://github.com/weiliu89/caffe/tree/ssd` 。

3.1 PASCAL VOC2007

在这个数据集上，我们与VOC2007上的快速R-CNN[6]和快速R-CNN[2]进行了比较。测试（4952张图片）。所有方法都在相同的预训练的VGG16网络上进行微调。

图2显示了SSD300模型的结构细节。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上设置了默认的尺度为0.1的盒子。[3].我们用 "xavier "方法来初始化所有新加入的卷积层的参数[20]。对于conv4 3、conv10 2和conv11 2，我们只在每个特征图的位置上关联4个默认盒子 - 省略长宽比为$\frac{1}{3}$ 和 3。对于所有其他层，我们把6个默认的盒子作为2.2节中描述的。正如文献[12]所指出的，与其他层相比，conv4 3有不同的特征尺度，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常模缩放为20，并学习
在反向传播过程中的规模。我们使用10^{-3} 的学习率进行40k迭代，然后用10^{-4} 和10^{-5} 继续训练10k迭代。当在VOC2007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。当我们在更大的512 512输入图像上训练SSD时，它甚至更准确，超过Faster R-CNN 1.7% mAP。如果我们用更多的图片来训练SSD
(即07+12）数据，我们看到SSD300已经比Faster R-CNN好了1.1%，SSD512好了3.6%。如果我们采用第3.4节中描述的在COCO `trainval35k`上训练的模型，并在07+12数据集上用SSD512对其进行微调，我们会取得最佳结果：81.6%的mAP。

为了更详细地了解我们两个SSD模型的性能，我们使用了[21]的检测分析工具。图3显示，SSD可以高质量地检测各种物体类别（大的白色区域）。它的大

部分自信检测都是正确的。召回率约为85-90%，在 "弱"（0.1 jaccard重叠）标准下，召回率要高得多。与R-CNN[22]相比，SSD的定位错误较少，这表明SSD可以更好地定位物体，因为它直接学习回归物体形状和分类物体类别，而不是使用两个解耦步骤。然而，固态硬盘在类似的物体类别上有更多的混淆（特别是对于动物），部分原因是我们共享多个类别的位置。图4显示，SSD对边界盒的大小非常敏感。换句话说，它在较小的物体上的性能要差得多。

---

[3]对于SSD512模型，我们增加了额外的conv12 2进行预测，将$_{smin}$设置为0.15，并在conv4 3上设置0.07。

方法	数据	MAP	航空	自行车	鸟	船	瓶子	公交车	猫	椅子	牛	桌子	狗	马	mbike	人	植物	羊	沙发	火车	电视机
快速[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
快速[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
更快 [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
更快 [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
更快 [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**

表1：**PASCAL VOC2007测试检测结果。**快速和快速的R-CNN都使用最小尺寸为600的输入图像。两个SSD模型的设置完全相同，只是它们的输入尺寸不同（300 300 vs. 512 512）。很明显，更大的输入尺寸会带来更好的结果，更多的数据总是有帮助。数据："07":VOC2007训练值，"07+12"：VOC2007和VOC2012训练值的结合。"07+12+COCO"：首先在COCO训练值35k上训练，然后在07+12上微调。

比起大的物体，小的物体更容易被发现。这并不奇怪，因为那些小物体在最顶层甚至可能没有任何信息。增加输入大小（例如从300 300增加到512 512）可以帮助改善对小物体的检测，但仍有很大的改进空间。在积极的一面，我们可以清楚地看到，SSD在大型物体上的表现非常好。而且它对不同物体的长宽比非常稳健，因为我们在每个特征图的位置使用了不同长宽比的默认盒子。

3.2 模型分析

为了更好地了解SSD，我们进行了控制性实验，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尺寸（300×300），除了对设置或组件的特定变化。

	SSD300				
更多的数据增强？包括		C	C	C	C
{1/2, 2}箱？	C		C	C	C
包括 {1/3, 3}箱？	C			C	C
使用 ATROS？	C	C	C		C
	65.5	71.6	73.7	74.2	**74.3**

表2：**各种设计选择和组件对SSD性能的影响。**

数据增强是至关重要的。快速和较快的R-CNN使用原始图像和水平翻转来进行训练。我们使用一个更广泛的采样策略，类似于YOLO[5]。表2显示，我们用这

种采样策略可以提高8.8%的mAP。我们不知道我们的抽样策略对Fast和Faster R-CNN有多大好处，但它们可能受益较少，因为它们在分类过程中使用了一个特征池步骤，在设计上对物体翻译相对稳健。

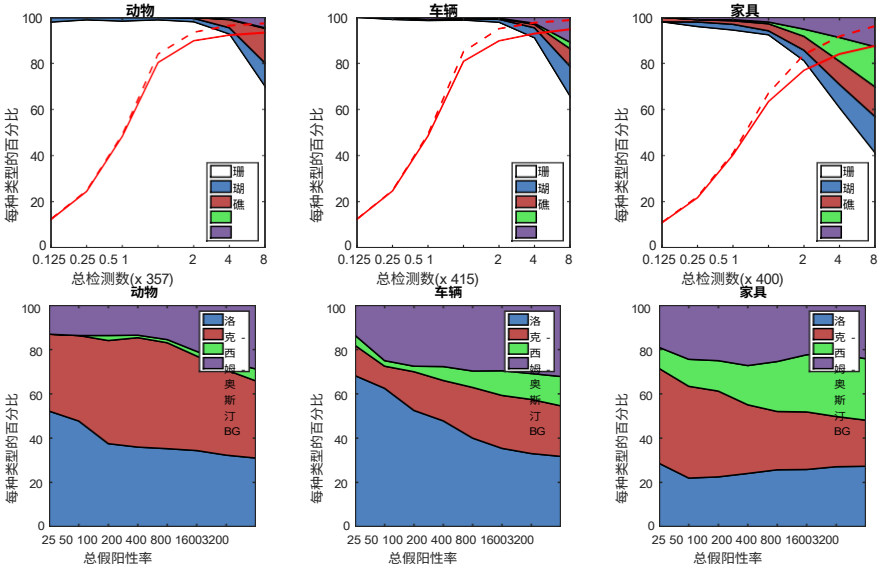


图3：**VOC2007测试中SSD512对动物、车辆和毛皮的性能可视化。**最上面一行显示了由于定位不良（Loc）、与类似类别（Sim）、与其他类别（Oth）或与背景（BG）混淆而导致的正确（Cor）或假阳性检测的累积比例。红色实线反映了随着检测数量的增加，强标准（0.5 jaccard重叠）的召回率的变化。红色虚线是使用弱标准（0.1 jaccard重叠）。底部一行显示了排名靠前的假阳性类型的分布。

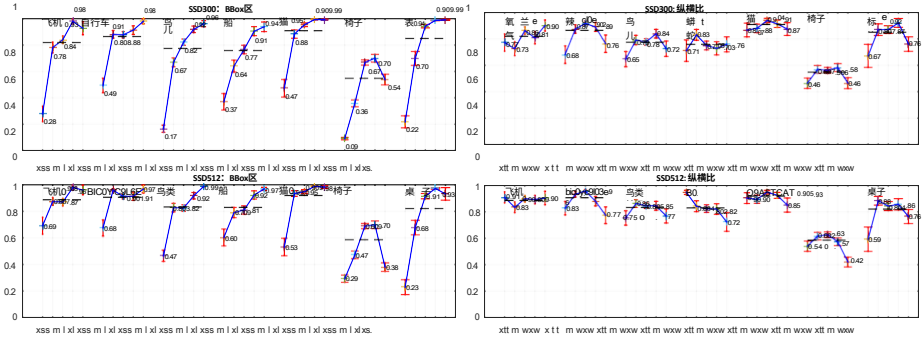


图4：**使用[21]对VOC2007测试集的不同物体特征的敏感度和影响。**左边的图显示了每个类别的BBox Area的影响，右边的图显示了Aspect Ratio的影响。关键：BBox面积：XS=超小；S=小；M=中；L=大；XL=超大。纵横比：XT=超高/窄；T=高；M=中；W=宽；XW=超宽。

更多的默认盒子形状是更好的。如第2.2节所述，默认情况下，我们使用6个每个位置的默认盒子。如果我们删除带有^{1} 和3长宽比的盒子，则性能下降了0.6%。通过进一步删除具有$_{2}$ 和2长宽比的盒子，性能又下降了2.1%。使用各种默认的盒子形状似乎可以使网络预测盒子的任务更容易。

Atrous更快。如第3节所述，我们按照DeepLab-LargeFOV[17]，使用了子采样VGG16的ATRUS版本。如果我们使用完整的VGG16，保持pool5与2 2 s2，不对fc6和fc7的参数进行子采样，并加入conv5 3进行预测，结果大致相同，而速度却慢了20%左右。

预测源层来自： conv7						mAP 使用边界是	mAP 箱子？没有	# 盒子
conv4 3	conv8 2	conv9 2	conv10 2	conv11 2				
C	C	C	C	C	C	74.3	63.4	8732
C	C	C	C	C		**74.6**	63.1	8764
C	C	C	C			73.8	68.4	8942
C	C	C				70.7	69.2	9864
C	C					64.2	64.4	9025
	C					62.4	64.0	8664

表3：**使用多个输出层的效果。**

不同分辨率的多个输出层是更好的。SSD的一个主要贡献是在不同的输出层上使用不同尺度的默认框。为了衡量所获得的优势，我们逐步移除各层并比较结果。为了进行公平的比较，每当我们移除一个层时，我们都会调整默认的盒子编排，以保持盒子的总数与原来的相似（8732）。这是通过在剩余的层上堆叠更多尺度的盒子，并在需要时调整盒子的尺度来实现的。我们没有详尽地优化每个设置的拼合。表3显示，随着层数的减少，准确率下降，从74.3单调地下降到62.4。当我们在一个图层上堆叠多个尺度的盒子时，很多盒子都在图像边界上，需要小心处理。我们尝试了Faster R-CNN[2]中使用的策略，忽略了边界上的盒子。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)），就会在很大程度上损害性能。或conv10 2（3×3）。原因可能是，我们没有足够的大箱子来修剪后覆盖大型物体。当我们主要使用较细的分辨率地图时，在性能又开始增加，因为即使在修剪之后，仍然有足够数量的大盒子。如果我们只用conv7进行预测，性能是最差的，这就加强了一个信息，即把不同规模的盒

子分散到不同的层上是至关重要的。此外，由于我们的预测不依赖于[6]中的ROI池，我们没有低分辨率特征图中的*塌陷箱*问题[23]。SSD架构结合了不同分辨率的特征图的预测，以达到与Faster R-CNN相当的准确性，同时使用较低分辨率的输入图像。

3.3 PASCAL VOC2012

我们使用与上述VOC2007基本实验相同的设置，只是我们使用VOC2012 trainval和VOC2007 trainval和test（21503张图片）进行训练，并在VOC2012 test（10991张图片）上测试。我们用10^{-3}的学习率训练模型，进行60k次迭代，然后用10^{-4}进行20k次迭代。表4显示了我们的SSD300和SSD512[4]模型的结果。我们看到与我们在VOC2007测试中观察到的性能趋势相同。我们的SSD300比Fast/Faster R- CNN提高了准确性。通过增加训练和测试图像大小到512 512，我们比Faster R-CNN的准确率高4.5%。与YOLO相比，SSD明显更准确，这可能是由于在训练期间使用了来自多个特征图的卷积默认框和我们的匹配策略。当从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高4.1%。

方法	数据	mAP	航空	自行车	鸟	船	瓶子	公交车	猫	椅子	牛	桌子	狗	马	mbike	人	植物	羊	沙发	火车	电视机
快速[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
更快[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
更快[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**

表4：**PASCAL VOC2012测试检测结果。**快速和快速R-CNN使用最小维度为600的图像，而YOLO的图像大小为448 448。 数据："07++12"：VOC2007训练值和测试值与VOC2012训练值的结合。"07++12+COCO"：首先在COCO trainval35k上训练，然后在07++12上微调。

3.4 COCO

为了进一步验证SSD框架，我们在COCO数据集上训练了我们的SSD300和SSD512架构。由于COCO中的物体往往比PASCAL VOC小，我们对所有层使用较小的默认框。我们遵循第2.2节中提到的策略，但现在我们最小的默认框的比例是0.15，而不是0.2，conv4 3的默认框的比例是0.07（例如，300 300的图像的21像素）。[5].

我们使用trainval35k[24]进行训练。我们首先用10^{-3}学习率为160k迭代，然后继续训练40k迭代，10^{-4}，40k迭代，10^{-5}。表5显示的是test-dev2015的结果。与我们在PASCAL VOC数据集上观察到的情况类似，SSD300在mAP@0.5 和mAP@[0.5:0.95]上都优于Fast R-CNN。SSD300与ION[24]和Faster R-CNN[25]有相似的mAP@0.75，但在mAP@0.5。通过增加image大小到512 512，我们的SSD512在两个标准上都比Faster R-CNN [25]好。有

趣的是，我们观察到SSD512在mAP@0.75，有5.3%的优势，但在mAP@0.5，只有1.2%的优势。我们还观察到，对于大型物体，它的AP（4.8%）和AR（4.6%）要好得多，但对于大型物体，它的AP（1.3%）和AR（2.0%）的改进相对较少。

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进行预测，将$_{smin}$设置为0.1，并在conv4 3上设置0.04。

方法	数据	平均值。精度，IoU：0.5:0.95	0.5	0 . 75	平均。精度，面积：S	M	L	平均。召回率，#Dets：1	10	100	平均。召回，面积：S	M	L
快速[6]	训练	19.7	35.9	-	-	-	-	-	-	-	-	-	-
快速[24]	训练	20.5	39.9	19.4	4.1	20.0	35.8	21.3	29.5	30.1	7.3	32.1	52.0
更快 [2]	培训	21.9	42.7	-	-	-	-	-	-	-	-	-	-
离子[24]	训练	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
更快 [25]	培训	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	训练35k	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	训练35k	**26.8**	**46.5**	**27.8**	**9.0**	**28.9**	**41.9**	**24.8**	**37.5**	**39.8**	**14.0**	**43.5**	**59.0**

表5：**COCO `test-dev2015`检测结果。**

小物体。与ION相比，大物体和小物体的AR改进更为相似（5.4% vs. 3.9%）。我们猜想，Faster R-CNN在小型物体上与SSD相比更有竞争力，因为它在RPN部分和Fast R-CNN部分都执行了两个箱体细化步骤。在图5中，我们展示了一些使用SSD512模型在COCO `test-dev`上的检测实例。

3.5 ILSVRC的初步结果

我们将用于COCO的相同网络结构应用于ILSVRC DET数据集[16]。我们使用ILSVRC2014 DET训练和`val1`训练一个SSD300模型，如[22]中使用的那样。我们首先以10^{-3} 的学习率训练模型，进行32万次迭代，并且
然后在10个$^{-4}$ ，继续训练80k迭代，在10个$^{-5}$ ，继续训练40k迭代。我们
在`val2`集上可以达到43.4mAP[22]。这再次验证了SSD是一个通用的
高质量实时检测的框架。

3.6 小物体精度的数据增强

如果没有像Faster R-CNN那样的后续特征重采样步骤，小型物体的分类任务对SSD来说是相对困难的，这在我们的分析中得到了证明（见图4）。第2.2节中描述的数据增强策略有助于大幅提高性能，特别是在小数据集上，如PASCAL VOC。该策略产生的随机作物可以被认为是一种 "放大 "操作，可以产生许多更大的训练实例。为了实现 "放大 "操作，产生更多的小的训练例子，我们首先将
一张图片随机地放在16个的画布上。
在我们做任何随机裁剪操作之前，原始图像的大小充满了平均值。
因为我们通过引入这种新的 "扩展 "数据增强技巧，有了更多的训练图像，我们必须将训练迭代次数增加一倍。如表6所示，我们看到多个数据集的mAP持续增加2%-3%。具体来说，图6显示，新的增强技巧大大改善了小物体的性能。这一结果强调了数据增强策略对最终模型准确性的重要性。

改善SSD的另一个方法是设计一个更好的默认盒的平铺，使其位置和比例与特征图上每个位置的接受域更好地对齐。我们把这个问题留给未来的工作。

图5：**使用SSD512模型在COCO测试开发中的检测实例。**我们展示了分数高于0.6的检测结果。每种颜色对应于一个物体类别。

方法	VOC2007测试		VOC2012测试		COCO test-dev2015 trainval35k		
	07+12	07+12+COCO	07++12	07++12+COCO	0.5:0.95	0.5	0.75
	0.5	0.5	0.5	0.5			
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**

表6：**当我们加入图像扩展数据增强技巧时，多个数据集的结果。**SSD300*和SSD512*是用新的数据增强法训练的模型。

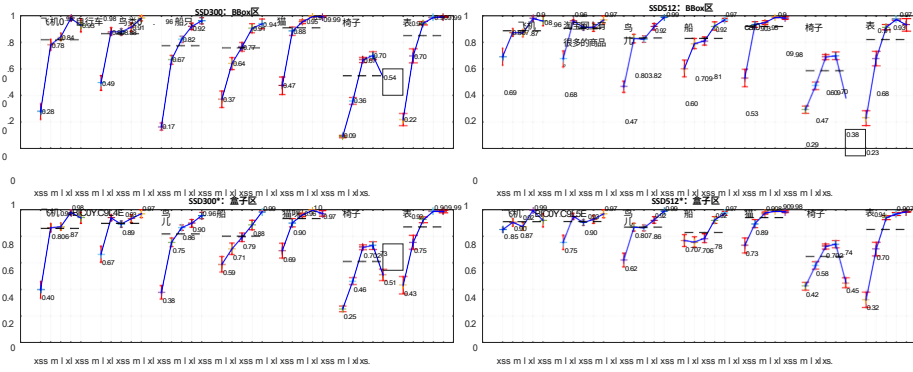


图6：**使用[21]对VOC2007测试集进行的新数据增强的敏感性和物体大小的影响**。最上面一行显示了原始SSD300和SSD512模型的BBox Area的影响，下面一行对应的是用新数据增强技巧训练的SSD300*和SSD512*模型。很明显，新的数据增强技巧对检测小物体有很大帮助。

3.7 推断时间

考虑到我们的方法产生了大量的盒子，在推理过程中有效地进行非最大抑制（nms）是非常重要的。通过使用0.01的置信度阈值，我们可以过滤掉大多数盒子。然后，我们用每类0.45的jaccard重叠来应用nms，并保留每幅图像的前200个检测结果。这一步对于SSD300和20个VOC类来说，每幅图像花费大约1.7毫秒，这接近于所有新增加的层所花费的总时间（2.4毫秒）。我们使用Titan X和cuDNN v4与英特尔至强E5-2667v3@3.20GHz，测量批处理量为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方法在速度和准确性方面都超过了Faster R-CNN。尽管Fast YOLO[5]可以以155 FPS的速度运行，但它的准确率却低了近22% mAP。就我们所知，SSD300是第一个达到70%以上mAP的实时方法。请注意，大约80%的前进时间是花在基础网络上（在我们的例子中是VGG16）。因此，使用更快的基础网络可以进一步提高速度，这有可能使SSD512模型也变得实时。

4 相关工作

有两类既定的方法用于图像中的物体检测，一类是基于滑动窗口，另一类是基于区域建议分类。在卷积神经网络出现之前，这两种方法的技术水平--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]

带来的巨大改进之后，它结合了选择性搜索区域提议和基于卷积网络的后分类，区域提议物体检测方法开始盛行。

最初的R-CNN方法已经通过各种方式得到了改进。第一组方法提高了后分类的质量和速度，因为它需要

方法	ÄÄÄ	FPS	批量大小	# 盒子	输入分辨率
更快的R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000 × 600
快速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

表7：**Pascal VOC2007测试的结果。**SSD300是唯一能达到70%以上mAP的实时检测方法。通过使用更大的输入图像，SSD512在准确性上胜过所有方法，同时保持了接近实时的速度。

对数以千计的图像作物进行分类，这既昂贵又费时。SPPnet[9]大大加快了原有的R-CNN方法。它引入了一个空间金字塔池层，对区域大小和比例更加稳健，并允许分类层重新使用在几个图像分辨率下生成的特征图上计算的特征。快速R-CNN[6]扩展了SPPnet，使其能够通过最小化置信度和边界盒回归的损失来对所有层进行端到端的微调，这在MultiBox[7]中首次引入，用于学习对象性。

第二组方法是使用深度神经网络来提高提案生成的质量。在MultiBox[7,8]等最新的作品中，基于低层次图像特征的选择性搜索区域建议被直接由一个单独的深度神经网络生成的建议所取代。这进一步提高了检测的准确性，但导致了一个有点复杂的设置，需要训练两个神经网络并在它们之间建立联系。Faster R-CNN[2]用从区域建议网络（RPN）中学习的建议取代了选择性搜索建议，并介绍了一种方法，通过交替微调这两个网络的共享卷积层和预测层来整合RPN和Fast R-CNN。这样，区域建议被用来汇集中层特征，最后的分类步骤就不那么昂贵了。我们的固态硬盘与Faster R-CNN中的区域提议网络（RPN）非常相似，因为我们也使用一组固定的（默认）盒子进行预测，类似于RPN中的锚盒。但我们不是用这些来汇集特征并评估另一个分类器，而是同时为每个盒子中的每个物体类别产生一个分数。因此，我们的方法避免了将RPN与快速R-CNN合并的复杂性，而且更容易训练，速度更快，并可直接整合到其他任务中。

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盒和置信度。OverFeat[4]是滑动窗口法的一个深度版本，在知道底层物体类别的置信度后，直接从最顶层特征图的每个位置预测一个边界框。YOLO[5]使用整个最顶层特征图来预测多个类别的置信度和边界框（这些类别的边界框是共享的）。我们的SSD方法属于这一类，因为我们没有提议步骤，

而是使用默认的盒子。然而，我们的方法比现有的方法更灵活，因为我们可以使用不同方面的默认框。

对来自不同尺度的多个特征图的每个特征位置的比率。如果我们只使用来自最上面的特征图的每个位置的一个默认框，我们的SSD将具有类似于OverFeat[4]的架构；如果我们使用整个最上面的特征图，并添加一个全连接层来预测，而不是我们的卷积预测器，并且不明确考虑多个长宽比，我们可以大约重现YOLO[5]。

5 结论

本文介绍了SSD，一种快速的多类别单次物体检测器。我们模型的一个关键特征是使用多尺度卷积边界盒输出，连接到网络顶部的多个特征图。这种表示方法使我们能够有效地对可能的盒子形状的空间进行建模。我们通过实验验证，在适当的训练策略下，更多的精心选择的默认边界框会使性能得到改善。与现有的方法[5,7]相比，我们建立的SSD模型至少有一个数量级的盒子预测采样位置、比例和长宽比。我们证明，在相同的VGG-16基础架构下，SSD在准确性和速度方面都优于其最先进的物体检测器。我们的SSD512模型在PASCAL VOC和COCO的准确度方面明显优于最先进的Faster R-CNN [2]，同时也是
3× 更快。我们的实时SSD300模型以59FPS的速度运行，比目前的实时YOLO[5]替代方案要快，同时产生了明显优越的检测精度。除了其独立的实用性，我们相信我们的单片机和相对模拟的恳求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建块。一个有希望的未来方向是探索它作为使用递归神经网络的系统的一部分来同时检测和跟踪视频中的物体。

6 鸣谢

这项工作是作为谷歌的一个实习项目开始的，在UNC继续进行。我们要感谢Alex Toshev的有益讨论，并感谢谷歌的Im- age Understanding和DistBelief团队。我们也感谢Philip Ammirato和Patrick Poirson的有益评论。我们感谢NVIDIA提供的GPU，并感谢NSF 1452851, 1446631, 1526367, 1533771的支持。

参考文献

1. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: 选择性搜索的对象 识别

。IJCV (2013)

2. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN：用区域建议网络实现实时物体检测。在：NIPS.(2015)
3. He, K., Zhang, X., Ren, S., Sun, J.: 图像识别的深度残差学习。In：CVPR.(2016)
4. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.：Overfeat：使用卷积网络进行综合识别、定位和检测。在：ICLR.(2014)

5. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once：统一、实时的 物体检测。In：CVPR。(2016)
6. Girshick, R.: 快速R-CNN。In：ICCV.(2015)
7. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: 使用深度 神经网络进行可扩展的物体检测。In：CVPR.(2014)
8. Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: 可扩展的高质量物体检测。 arXiv preprint arXiv:1412.1441 v3 (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: 用于视觉识别的深度卷积网络中的空间金字塔池 。In：ECCV.(2014)
10. Long, J., Shelhamer, E., Darrell, T.: 用于语义分割的完全卷积网络。In：CVPR.(2015)
11. Hariharan, B., Arbela´ez, P., Girshick, R., Malik, J.：用于物体分割的超柱 和细粒度的定位。在：CVPR。(2015)
12. Liu, W., Rabinovich, A., Berg, A.C.: ParseNet：看得更广，看得更清楚。在：ILCR.(2016)
13. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.：物体检测器出现在深度 场景cnns。在：ICLR.(2015)
14. Howard, A.G.: 基于深度卷积神经网络的图像的一些改进 分类。arXiv预印本 arXiv:1312.5402 (2013)
15. Simonyan, K., Zisserman, A.: 用于大规模图像识别的非常深入的卷积网络 nition。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大尺度视觉识别 挑战。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。在：ICLR.(2015)
18. Holschneider, M., Kronland-Martinet, R., Morlet, J., Tchamitchian, P. ：在小波变换的帮助下进行信号分析的实时算法。在：小波。Springer (1990) 286-297
19. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe：用于快速特征嵌入的卷积架构。在：MM.(2014)
20. Glorot, X., Bengio, Y.：了解训练深度前馈神经 网络的难度。In：AISTATS.(2010)
21. Hoiem, D., Chodpathumwan, Y., Dai, Q.: 诊断物体检测器的错误。In：ECCV 2012。(2012)
22. Girshick, R., Donahue, J., Darrell, T., Malik, J.：用于准确的物体 检测和语义分割的丰富特征层次。In：CVPR.(2014)
23. Zhang, L., Lin, L., Liang, X., He, K.: 更快的r-cnn在行人检测中表现良好。In：ECCV.(2016)
24. Bell, S., Zitnick, C.L., Bala, K., Girshick, R.: Inide-outside net：用跳过集合和循环神经网络检测上下文中的物体 。In：CVPR.(2016)
25. COCO: Common Objects in Context .http://mscoco.org/dataset/ #detections-leaderboard (2016) [在线；2016年7月25日访问] 。

26. Felzenszwalb, P., McAllester, D., Ramanan, D.： 一个经过辨别训练的、多尺度的、可变形的零件模型。In：CVPR.(2008)