Software Engineering and Applications 软件工程与应用, 2020, 9(1), 36-48
Published Online February 2020 in Hans. http://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2020.91005

Hans汉斯

Object Detection Based on Convolutional Neural Network

Wenxin Zhong

School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou Zhejiang
Email: vincentzho@qq.com

Received: Jan. 14th, 2020; accepted: Jan. 28th, 2020; published: Feb. 4th, 2020

Abstract

Object detection algorithms based on regions proposals have higher detection accuracy, but the detection speed is slower, which cannot achieve the effect of real-time detection. Aiming at this problem, this paper proposes a new type of target detection method based on deep separable convolutional neural network. First, through ResNet-101 and a deep separable convolutional layer, a simplified feature map of the target is extracted to reduce the amount of calculation to increase the detection speed; at the same time, in order to compensate for the accuracy loss caused by the increased detection speed, a key-point-oriented strategy is proposed. Instead of the traditional regression method, this strategy uses the sensitivity of the full convolutional neural network to the position of the object, effectively retains the spatial information of the object, and makes the algorithm locate the target object more accurately. Finally, in order to improve the algorithm's ability to detect small targets, the PS-RoI Align method is used instead of the traditional pooling method to improve the algorithm's ability to detect small targets. Experimental results show that the method can achieve better detection results on the COCO dataset.

Keywords

Target Detection, Deeply Separable Convolution, Full Convolutional Neural Network, Key Points

基于卷积神经网络的目标检测方法

钟文鑫

浙江理工大学信息学院，浙江 杭州
Email: vincentzho@qq.com

收稿日期：2020年1月14日；录用日期：2020年1月28日；发布日期：2020年2月4日

文章引用：钟文鑫. 基于卷积神经网络的目标检测方法[J]. 软件工程与应用, 2020, 9(1): 36-48.
DOI: 10.12677/sea.2020.91005

摘　要

基于候选区域的目标检测算法检测精度较高，但检测速度较慢，无法达到实时检测的效果。针对这一问题，本文提出了一种新型的基于深度可分离卷积神经网络的目标检测方法。首先通过ResNet-101和深度可分离卷积层，提取目标的精简特征图，减少计算量，以提高检测速度；与此同时为了弥补提高检测速度带来的精度损失，提出采用关键点导向的策略，代替传统的回归方法，该策略利用全卷积神经网络对物体位置的敏感特性，有效保留物体的空间信息，使得算法对目标物体定位更精确。最后，为了提高算法对小目标物体的检测能力，使用PS-RoI Align方法代替传统的池化方法，在一定程度上提高算法对小目标物体的检测能力。实验结果表明，在COCO数据集上，该方法能够取得较好的检测效果。

关键词

目标检测，深度可分离卷积，全卷积神经网络，关键点

1. 引言

近些年来，得益于卷积神经网络(**CNN**)技术的不断发展进步，越来越多的学者开始利用 CNN 来进行目标检测方法的研究，并且取得了长足的进步。当前基于 CNN 的目标检测方法主要分为两大类，一类是基于回归的目标检测方法，以 YOLO [1]，YOLOv2 [2]，YOLOv3 [3]，SSD [4]等算法为代表，这类方法的检测速度非常快，而检测精度却一般；另一类是基于候选区域的目标检测方法，以 R-CNN [5]，Fast R-CNN [6]，R-FCN [7]，Faster R-CNN [8]等代表，检测精度较高，但检测速度较慢，本文着眼于提高后者的检测速度。

基于候选区域的目标检测算法，以 R-CNN 为代表，R-CNN 是第一个将 CNN 网络用于目标检测的算法，采用了传统的 Selective Search [9]方法来生成候选区域，对于每张输入图片大约生成 2000 多个候选区域，耗时约 27 s，无法有效满足检测速度需求。因此后来出现了 SPPNet [10]，Fast R-CNN 等对其作出了改进，其中 Fast R-CNN 提出联合训练目标分类和候选框回归，形成了一个多任务训练模型，能够有效提高检测性能，但其仍然使用的是 Selective Search 方法，检测速度仍然不理想，Faster R-CNN 应运而生。Faster R-CNN 最大的改进是引入了区域提取网络(**RPN**)来提取候选区域，这样一来，极大地提高了检测的速度和精度，使得整个模型能够进行端到端的训练。但是基于候选区域的目标检测方法把检测分为两个过程：第一步是生成一系列的候选区域，第二步是对这些区域进行识别，分类和定位。为了得到更高的检测精度，第二步的识别工作往往十分复杂，计算量非常大。例如，Faster R-CNN 和 R-FCN 在 ROI (**感兴趣区域**)的拉伸之前或者之后都有很大的计算量，Faster R-CNN 采用了两个全连接层用于 ROI 的识别，而 R-FCN 则生成了大量的物体得分图，计算量都非常大。因此基于候选区域的目标检测方法通常具有很高的检测精度，但是检测速度较慢。

本文致力于提高基于候选区域的目标检测算法的检测速度，提出了一种新的检测方法，采用了深度可分离卷积(**deep separable convolution**)来得到精简的特征图，极大地减小了计算量，提高了检测速度。但精简特征图也会带来检测精度的损失，因此还要进一步提高检测精度。Grid R-CNN [11]中提出了一种

关键点检测的方法，代替了传统的回归定位方法，对目标物体的精确定位有很大帮助，因此本文也采用了关键点导向的方式来确定目标的候选框，以便进一步提高本文算法的检测精度。与传统的回归方法相比，回归方法通过全连接层将特征图映射为一个向量，而本文的方法应用一个全卷积神经网络(**FCN**) [12]来预测目标物体的关键点的位置，来对目标物体进行定位。由于全卷积神经网络结构的位置敏感性，该方法保证了明确的空间信息，并且关键点的位置也能够在像素级获得。当获得了特定位置的一定数量的关键点之后，相应的候选框也就确定了。由关键点作为导向，能够得到比回归方法更精确的目标候选框。同时，为了提高算法对小目标物体的检测能力，使用了 PS-RoI Align 取代传统的池化方法。在 COCO 数据集上，本文算法与其它目标检测算法进行了对比实验，实验结果表明，本文提出的算法取得了较优的检测效果。

2. 本文方法

2.1. 整体结构

本文是基于候选区域的目标检测方法，总体结构模型如图 1 所示，首先主干特征提取网络采用的是残差网络 ResNet-101 [13]的卷积层和深度可分离卷积层，用于提取输入图片的特征图，ResNet 各卷积层具体情况如表 1 所示。ResNet-101 的最后一层共享卷积层(**Conv4**)生成的特征图输入到区域提取网络 RPN (**Region Proposal Network**)中，得到带有分类标签和回归框的候选区域的特征图，再将 RPN 网络输出的特征图和由深度可分离卷积层得到的特征图，输入到 PS-RoI Align 层，进一步做池化操作。这时网络将会分成两部分，一部分用于判断目标物体的类别，一部分用于确定目标物体的位置。

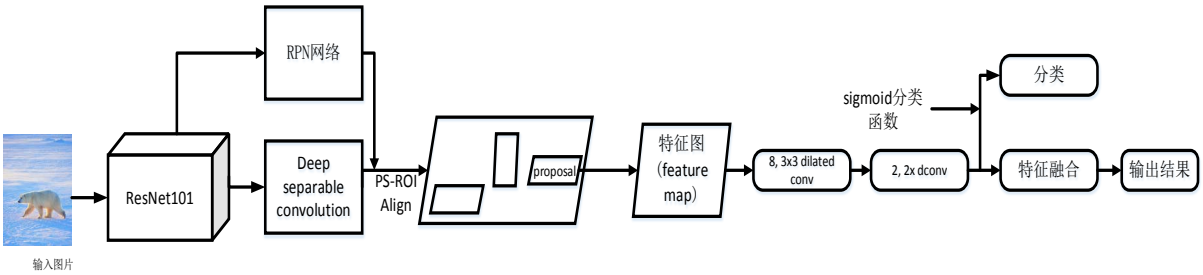

Figure 1. The pipeline of our method
图 1. 本文框架

2.1.1. 残差网络结构

残差网络是由何凯明等人于 2015 年提出的一种卷积神经网络模型，曾在 ILSVRC2015 [14]大赛上获得了图像分类的第一名。与传统的卷积神经网络相比，残差网络在增加了网络深度的同时，有效避免了梯度消失和梯度爆炸、网络泛化的问题，有着更强的泛化能力。在目标识别领域，随着网络深度的增加。深层网络会更加难以训练，模型准确率会受到影响。为了解决这个问题，残差网络模型引入了残差结构和短路连接，如图 2 所示。

残差网络的特点如图 2 所示，每两层增加一个跳跃连接。记输出为 H(x)的话，H(x) = F(x) + x。x 为恒等映射，F(x)为残差映射，这两种映射提供了两种选择方式，用于解决随着网络程度加深导致的模型准确率下降的问题。当模型的准确率已经达到最优的值时，残差映射 F(x)将被置 0，这时网络的输出只有恒等映射 x，此后网络将保持最优状态，从而保证了网络的准确率不会因为网络深度的增加而降低。采用 ResNet-101 可以利用其网络深度高，提取丰富的特征信息。而深度可分离卷积层可以大幅度地降低特征图的维度，极大地减少计算量，降低内存占用，从而提高计算速度，同时又能保证准确性。

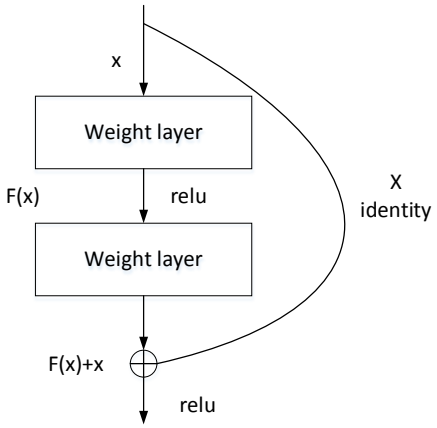

Figure 2. Residual blocks
图 2. 残差块示意图

Table 1. Layers of ResNet-101
表 1. ResNet-101 各个卷积层

Layer name	Output size	101-layer
Conv1	112 × 112	7 × 7, 64, stride = 2
		3 × 3 max pool, stride = 2
Conv2_x	56 × 56	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$
Conv3_x	28 × 28	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$
Conv4_x	14 × 14	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times23$
Conv5_x	7 × 7	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$

2.1.2. 深度可分离卷积层

深度可分离卷积是 A G Howard 等人在 MobileNet [15]中提出的，它可以大幅度减少计算量，降低内存占用，提高计算速度，同时又能保证准确性。深度可分离卷积将标准卷积分解成深度卷积和逐点卷积(**即 1 × 1 的卷积**)两部分。其中深度卷积指的是对输入图像的单个通道应用单个滤波器进行卷积，然后逐点卷积再应用一个简单的 1 × 1 卷积对深度卷积中的输出进行线性结合，从而生成新的更加精简的特征图。由于深度可分离卷积将卷积过程分成了两步，第一步是对单个通道进行卷积，然后再将其结合，这种分解能够极大地减少计算量以及模型的大小。假设标准卷积的输入为一个特征图 F，维度为 $D_F \times D_F \times M$，与 N 个维度为 $D_K \times D_K \times M$ 的滤波器进行卷积操作，生成一个特征图 G，维度为 $D_G \times D_G \times N$，那么这个过程的计算量为 $D_K \times D_K \times M \times N \times D_F \times D_F$，参数量为 $D_K \times D_K \times M \times N$，由滤波器大小和个数决定，滤波器的通道个数与输入特征图的通道数相同，输出特征图的通道数与滤波器的个数相同。

而如果将上述的特征图输入深度可分离卷积层，它首先会使用一组通道数为 1 的滤波器，每次只与输入特征的一个通道进行卷积，所以这个过程中，滤波器的数量和输入特征图的通道数是相同的，由此输出的特征图，再使用 1 × 1 卷积核进行卷积，将最终输出的通道数变成一个指定的数量。深度卷积和逐点卷积的计算量分别为 $D_K \times D_K \times 1 \times M \times D_F \times D_F$ 和 $1 \times 1 \times N \times M \times D_F \times D_F$，所以深度可分离卷积的计算量为 $D_K \times D_K \times M \times D_F \times D_F + N \times M \times D_F \times D_F$，参数量为 $D_K \times D_K \times M + N \times M$，而标准卷积的计算量为 $D_K \times D_K \times M \times N \times D_F \times D_F$，参数量为 $D_K \times D_K \times M \times N$，所以标准卷积的计算量和深度可分离卷积的计算量对比为

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + N \cdot M \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K \cdot D_K} \tag{1}$$

参数量对比为

$$\frac{D_K \cdot D_K \cdot M + N \cdot M}{D_K \cdot D_K \cdot M \cdot N} = \frac{1}{N} + \frac{1}{D_K \cdot D_K} \tag{2}$$

由此可知，相比于标准卷积过程，深度可分离卷积大大地减少了计算量和参数量，因此提高了整个系统的检测速度。

2.2. 区域提取网络 RPN (Region Proposal Network)

区域提取网络(**RPN**)的输入是主干网络 Conv4 (见表 1)输出的特征图，其输出是一系列目标物体的可能存在的区域(**region proposals**)和目标分类的概率得分值，此时并不输出目标的具体类别，而是输出目标物体分别属于前景和背景的概率值。为了生成候选区域(**region proposals**)，我们在卷积得到的特征图(**feature map**)上滑动一个小的网络，这个特征图也就是 RPN 的输入，是最后一个共享卷积层的输出，也就是本文中 Conv4 的输出。本文中，使用的是 3 × 3 的卷积层，然后加上两个 1 × 1 的卷积层，得到的特征图分别输入两个平行的全连接层，分别进行分类和回归。

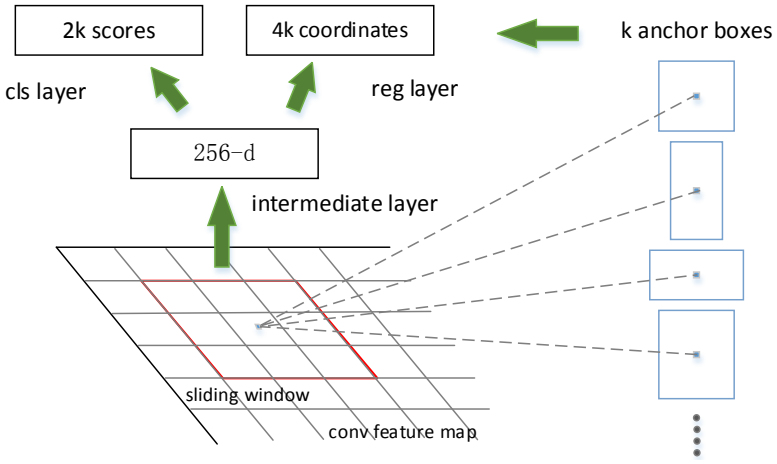

Figure 3. The diagram of anchors
图 3. 锚点示意图

锚点(图 3)：在每个滑动窗口位置，我们需要同时预测多个候选区域(**region proposals**)，其中每个位置最大可能建议的数量表示为 k，回归层的一个回归框需要的数据为(x，y)，w，h，它们分别为回归框的角点的坐标，回归框的宽度和高度，那么总共需要 4 k 个输出。同理，分类层需要 2 k 个输出，因为需要估

计每个区域(proposals)是否是目标物体的概率。锚点的位置是在滑动窗口中的，和候选框面积大小和纵横比相关。本文中，我们使用的纵横比有三种{1:2，1:1，2:1}，面积种类有五种，分别是{32^2，64^2，128^2，256^2，512^2}，因此每个滑动窗口每次滑动产生 15 个锚点。对于 W × H 大小的特征图，锚点个数为 W × H × 15。

2.3. PS-RoI Align (位置敏感的候选区域对齐)

本文中，使用 PS-RoI Align 来对齐特征图，而不用传统的池化方法。PS-RoI Pooling 是 RFCN 中提出的，一般情况下，网络越深，其具有的平移不变性越强，这个性质对保证分类模型的鲁棒性具有积极意义。然而在检测问题中，对物体的定位任务要求模型对位置信息具有良好的感知能力，过度的平移旋转不变性会削弱这一性能。研究表明，对于较深的卷积神经网络(**Inception** [16]、**ResNet** 等)，Faster R-CNN 检测框架存在一个明显的缺陷：检测器对物体的位置信息的敏感度下降，检测准确度降低。所以一般会采取将 RPN 的位置向浅层移动，本文中是将 RPN 嵌入到 Conv4_x 的位置，但是这样的话，也明显增加了后续的计算量，使得检测速度明显变得很慢。所以本文中采用 PS-RoI Align。也就是将 RoI Align 移植到 PS-RoI Pooling 中，主要的改进就是两次量化的取消：ROI 的边界坐标值和每个 ROI 中所有矩形单元的边界值保持浮点数形式，在每个矩形单元中计算出固定数量的采样点的像素值作平均池化。PS-ROI Align 对模型的检测性能有提升，对小物体的感知能力有明显改善。

2.4. 基于关键点的目标定位和分类

本文方法的整体框架如图 1 所示。在 R-CNN 之后出现的目标检测方法，都是在 R-CNN 上进行改进的。比如，在 COCO 数据集上检测精度非常高的 Faster R-CNN 网络，它的 R-CNN 子网络采用的是两个非常大的全连接层，来作为分类器，这样一来，当候选区域的数量比较庞大的时候，该网络的计算量也是巨大的。当 Faster，Zeming Li 等人提出的 Light-Head R-CNN [17]方法，是一种基于 Faster R-CNN 的改进方法。Faster R-CNN 基于候选区域的方式由一个 CNN 主干网络得到相应的特征图，然后从这些特征图中提取出每一个 ROI 的特征。随后我们就可以利用这些特征对相应的候选区域进行分类和定位。和 Faster R-CNN 相比，受到 Grid R-CNN 的启发，我们采用了一种关键点机制取代传统的边框回归来定位，并利用了其中的一些方法，如点的预测定位机制、点与点之间进行特征融合使得点的定位更加精准、以及如何定位不在目标物体上的点。关键点的预测采用了一个全卷积网络。它能够输出一个概率热图(**heatmap**)，从中我们可以获取与目标物体对应的候选框的网格点的位置。最后，对于这些网格点，我们通过一个特征图级的信息融合方法来得到精确的物体的候选框。

2.4.1. 基于关键点的候选框定位

传统的基于深度学习的目标检测方法中，通常使用一些全连接层作为一个回归器，得到左上角点的坐标(x，y)和 width (**宽度**)，height (**高度**)四个值，来进一步预测出目标物体位置的边框。然而，我们采用的是一个全卷积网络来预测一些关键点的位置，然后利用这些点来确定精确的目标物体的候选框。我们设计了一种 3×3 的网格形式，与目标物体的候选框相对应。如图 4 所示是一个 3 × 3 的网格的例子，总共 9 个点，分别是四个角点，四条边的中点和中心点。将由 RPN 提取到的候选区域 ROI 的特征，输入到 PS-ROI Align 中，PS-RoI Align 大小固定为 14 × 14，生成分辨率为 14 × 14 的特征图，对生成的特征图进行连续的八个 3 × 3 膨胀卷积(**针对大接收野的**)。然后连接两个连续的 2 × 反卷积层，所以每个候选区域(**region proposal**)生成分辨率为 56 × 56 的热图(heatmap)。在每个 heatmap 上，我们使用了像素级的 sigmoid 函数，以便得到最终的输出概率图。每一个 heatmap 都有一个相应的监督图，以每 5 个十字交叉形状的像素点都会标记为目标点的位置。本文使用了二进制交叉熵损失函数来对结果进行优化，在检测中选取每个 heatmap 中最大值点作为相应的关键点。

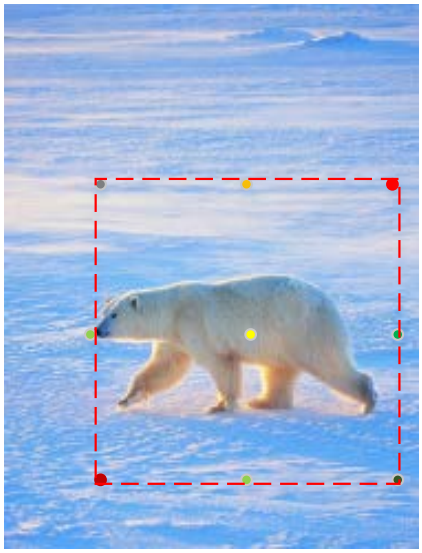

Figure 4. The example of 3 × 3 key points location
图 4. 3 × 3 关键点定位示例

由于生成的 9 个关键点都位于 heatmap 之中，所以为了对目标进行定位，就必须将这些点映射回原图像中，如图 5 所示：

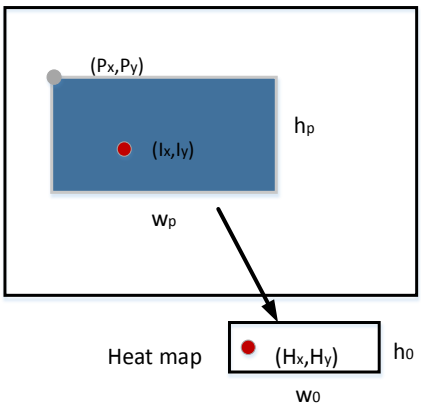

Figure 5. Key points map
图 5. 关键点映射图

图 5 中最大的矩形框表示输入图像，蓝色的框表示候选区域，下边的矩形框表示相应的 heatmap，heatmap 中的红色的点表示 heatmap 中值最大的点(H_x，H_y)，点(I_x，I_y)表示该最大值点映射回原图像中的原始点，(P_x, P_y)表示候选区域的左上角坐标点。将 heatmap 中的最大值点映射回原图中，公式如式(3)所示。

$$\begin{aligned} I_x &= P_x + \frac{H_x}{\omega_o}\omega_p \\ I_y &= P_y + \frac{H_y}{h_o}h_p \end{aligned} \tag{3}$$

其中(P_x，P_y)是输入图片中候选区域的左上角的点的位置，ω_p 和 h_p 分别是候选区域的宽度和高度，ω_o 和 h_o 分别是输出的 heatmap 的宽度和高度。通过式(3)，将这 9 个关键点从 heatmap 映射回了原图中，接下

来就要根据这些点确定目标物体的坐标框。将物体候选框四条边的坐标记为 $B = (x_l, y_u, x_r, y_b)$，分别表示左、上、右、下四条边。将第 j 个点的坐标记为 $g_j = (x_j, y_j)$，该点在 heatmap 中的概率值记为 p_j。定义 E_i 为第 i 条边上的点的索引集合，如点 $g_j = (x_j, y_j)$位于候选框的第 i 条边上，那么 $j \in E_i$。如图 6 所示。

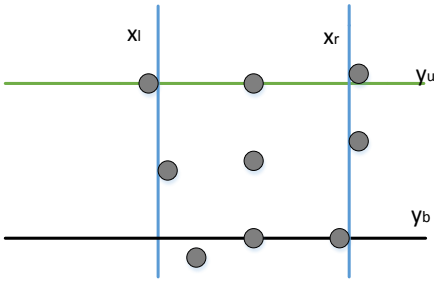

Figure 6. Prediction of bounding box
图 6. 回归框的预测

根据图 6，我们采用了加权求和的方式，来求坐标 B。例如，以最左边的边界 x_l 为例，选择最左边的三个点的坐标，并计算它们三个点的 x 坐标的加权和，权重就是相应的各点在 heatmap 中的概率值 p_j。计算公式如式(4)所示。

$$x_l = \frac{1}{N}\sum_{j\in E_1} x_j p_j, y_u = \frac{1}{N}\sum_{j\in E_2} y_j p_j$$
$$x_r = \frac{1}{N}\sum_{j\in E_3} x_j p_j, y_b = \frac{1}{N}\sum_{j\in E_4} y_j p_j \quad (4)$$

2.4.2. 点的特征融合

特征融合：如果仅仅按照上文所描述的方法对目标物体进行定位的话，还存在一定的问题。由于上文的方法只利用了一个 heatmap 来生成一个点，在这种情况下，如果某个点存在背景区域，那么该区域获得的信息是不足以精确地定位目标物体的边界位置的，这样就需要融合周边的点对应的 heatmap 来对其进行校正。融合方法如图 7 所示。

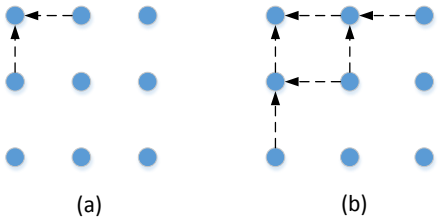

Figure 7. Feature fusion between key points
图 7. 关键点之间的特征融合

如图 7 所示，以左上角的点为例，假设其对应的 heatmap 为 F_i，和左上角的点距离为一个单位的点称为源点，源点构成的集合设为 S_i，假设某个源点的 heatmap 为 F_j，对 F_j 进行连续三次的卷积运算，卷积核大小为 5 × 5，得到 F_j'，然后将源点集合中，所有的源点对应的 heatmap 进行上述运算之后，与 F_i 相加得到融合后的 F_j'，如图 7(a)所示，有两个点与目标点进行了特征融合，这是距离为一个单位长度的特征融合情况，称为一阶融合。进行完一阶融合后，还需要进行二阶融合，如图 7(b)所示，对距离目标点两个单位长度的点进行特征融合。融合的公式如式 3 所示。

$$F'_j = F_i + \sum_{j \in S_i} T_{j \to i}(F_j) \tag{5}$$

$T_{j \to i}$ 是表示将源点的 heatmap 进行三次 5 × 5 卷积操作的函数。这样一来，经过两次融合之后得到的 heatmap，再通过上述的点的定位方法，确定关键点的精确位置，生成对应的边界框，从而提高边界框的定位精度。

区域拓展：上文的方法还不能完全实现对点的准确定位，因为上文没有考虑到一种情况，就是有一些候选区域覆盖的区域比较小，其可能存在和 ground truth (**真值框**)的重合度较小的情况。如图 8 所示。

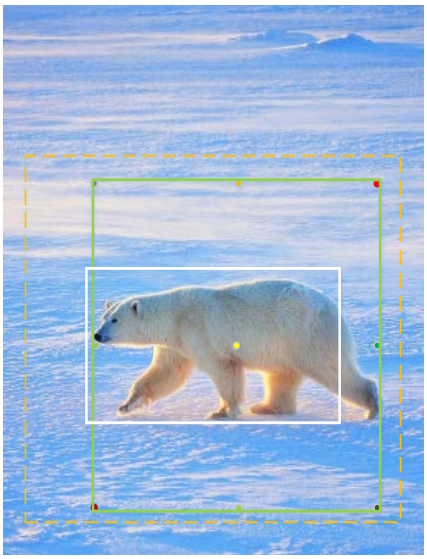

Figure 8. Extended region map
图 8. 扩展区域映射图

如图 8 所示，白色框为候选区域，绿色框为 ground truth 区域，这种情况下，只有左边的中点和中心点是在 proposal 中的，这会导致在这些关键点的训练中，会缺少这些点的标签，难以对其进行有效地监督训练，导致训练样本的利用率降低。而在预测阶段，如果只是简单地选择 heatmap 中的最大值点，出现如图 8 的情况，将不能预测位于候选区域之外的那些点的正确位置。为了解决这个问题，采用了以下的解决方案。本文采用了扩展 heatmap 区域的方法，具体来说，即不改变候选区域的范围，仍然从原来的 proposal 提取特征，但是在将输出的 heatmap 的区域映射回到原始图像时，映射到一个更大的区域，如图 8 的虚线部分。映射公式如式(6)所示。

$$\begin{aligned} I'_x &= P_x + \frac{H_x}{\omega_o}\omega_p + \left(\frac{H_x}{\omega_o} - \frac{1}{2}\right)\omega_p \\ I'_y &= P_y + \frac{H_y}{h_o}h_p + \left(\frac{H_y}{h_o} - \frac{1}{2}\right)h_p \end{aligned} \tag{6}$$

新的映射关系是在原来的映射关系上添加了一个修正项，这样一来，当 heatmap 中选择的点在左侧时，映射回原始图像的点会向左移动，在右侧是向右移动，在 y 方向也是同样的映射关系。通过映射的修正，就解决了 proposal 与 ground truth 重合过少难以训练的问题，又避免了因扩大 proposal 带来的背景特征混入的问题。这样的话，proposal 中所有的关键点(**与 ground truth 的重叠 IoU 大于 0.5 的区域**)都将被相应区域的 heatmap 覆盖。

3. 实验配置及过程分析

实验的硬件配置为：CPU：Intel(R) Core(TM) i7-4790K CPU@ 4.00GHz；运行内存：32G 显卡：Nvidia Geforce GTX1070。

网络环境及配置：实验的所有模型都在 Ubuntu16.04 环境下完成，使用开源框架 Pytorch 实验，CUDA 版本为 8.0，使用 Python3.6 进行编程实现。

实验数据集以及实现过程：为了评估本算法的检测性能，本实验采用了 COCO [18]数据集，在 COCO 数据集中有 80 种目标物体。实验采用的训练集为 80 k，验证集为 40 k，其中的 35 k 作为验证集(mini-validation)，5 k (mini-test)用于测试。测试集的大小为 20 k。本文采用的主干网络框架是 ResNet，深度为 101 层，并且去掉所有的全连接层。使用 RPN 网络来提取候选区域，在此过程中对每张图片进行采样，每张图片采样 256 个锚点，正负锚点的比例为 1:1，锚点的尺度涵盖三种纵横比，分别是{1:2，1:1，2:1}，五种面积，分别是$\{32^2, 64^2, 128^2, 256^2, 512^2\}$。我们规定 IOU(候选框和 groundtruth 的重合面积/groundtruth) > 0.7 的锚点为正样本，低于 0.3 的为负样本。实验中，我们采用了 PS-ROI Align 来完成池化的操作，定位时用的池化层的大小为 14 × 14，分类时为 7 × 7。我们使用 SGD [19] (**随机梯度下降**)方法来训练损失函数，寻找最优解，momentum = 0.9，weight decay = 0.0001，网络框架是在预训练的 ImageNet [20]模型上进行初始化的。同时，我们还对数据进行了水平翻转，以便增强模型的鲁棒性。我们将图片的最小像素和最大像素分别设置为 800 和 1200，2000 个 RoI 用作训练，1000 个用作测试。在预测阶段，我们将 RoI 的特征图(feature map)输入 PS-RoI Align 进行处理，然后将得到的特征图输入到分类分支，得到一个类别置信度得分，然后对其使用 NMS [21] (**非极大值抑制**)算法，去掉 IoU < 0.5 的区域。随后，从保留的符合要求的 RoI 中，挑选出得分值最大的前 125 个 RoI，并将它们的 PS-RoI Align 特征输入到上文中提到的关键点导向的分支，进行目标物体的定位工作。

评价指标：mAP (**Mean Average Precision**)和检测速度 FPS (**Frame Per Second**)。AP 是判断检测器检测每一类目标性能的标准，而 mAP 则多用来评价多目标检测条件下检测器的平均检测精度，其大小为 AP 值得平均值，以此作为目标检测中衡量检测精度的指标。在同一个公共数据集上计算不同的检测算法的 mAP 值，可以分析出检测算法的性能，具体计算公式如下

$$\begin{aligned} AP &= \int_0^1 p(r)\,dr \\ mAP &= \frac{\sum_{i=1}^{C} AP(i)}{C} \end{aligned} \tag{7}$$

式中，AP 代表 PR 围成的面积，p(r)是 PR 曲线，mAP 是对 AP 进行加权平均。C = 20，代表 20 类检测目标。

仅仅依靠检测精度对一个目标检测网络模型进行评估，显然是不够的。因此，本文的另一个重要参考指标是检测速度。评估速度的常用指标是帧率(Frame Per Second, FPS)，通过模型每秒处理的图片数量来判定模型检测性能的好坏。只有在保证了检测精度的同时还能兼顾检测速度，才能说明目标检测模型的性能优劣。

4. 结果与分析

由于通常评价目标检测算法的性能，主要是从检测精度和检测速度这两方面来进行评估的，因此，我们针对本算法的检测精度和检测速度分别与其他的经典目标检测算法做了对比。

1) 检测精度对比

如表 2 所示，为在 COCO 测试集上，本算法与其他一些经典算法的检测精度对比。

表 2 中，mAP 表示 IoU 在[0.5:0.95]之间的平均精度，AP_S，AP_M，AP_L 分别为各个算法对小目标，中等目标和大目标的检测精度。由表 2 可知，本文的算法的 mAP 与表中次优的 Grid R-CNN 相当，在不同大小的物体检测方面也有小幅度的提升。得益于本文采用的 PS-RoI Align 和关键点导向的策略，提升了本算法的目标检测精度。

Table 2. Comparison of detection accuracy between this paper and other classic algorithms
表 2. 本文与其它经典算法的检测精度对比

模型	主干网络	mAP@[0.5:0.95]	AP_S	AP_M	AP_L
R-FCN [7]	ResNet-101	32.1	12.8	32.2	47.4
Faster R-CNN [8]	ResNet-101	30.3	9.9	32.2	47.4
Mask R-CNN [22]	ResNet-101	38.2	20.1	41.1	50.2
Light-head R-CNN [17]	ResNet-101	39.5	21.8	43.0	50.7
Grid R-CNN [11]	ResNet-101	41.5	23.3	44.9	53.1
Deformable [23]	ResNet-101	34.5	14.0	37.7	50.3
RetinaNet [24]	ResNet-101	37.8	20.2	41.1	49.2
FPN [25]	ResNet-101	36.2	18.2	39.0	48.2
本文方法	ResNet-101	41.9	24.1	45.3	53.7

2) 检测速度对比

Table 3. Comparison of detection speed between our algorithm and other algorithms
表 3. 本文算法与其它算法的检测速度对比

模型	主干网络	检测速度(**fps**)	mAP@[0.5:0.95]
YOLOv2 [2]	Darknet-19	40	21.6
YOLOv3 [3]	Darknet-53	78	28.2
SSD [4]	ResNet-101	16	28.0
R-FCN [7]	ResNet-101	11	29.9
Light-head R-CNN [17]	ResNet-101	95	39.5
DSSD [26]	ResNet-101	8	28.2
本文算法	ResNet-101	43	41.9

我们将本文算法和其他一些检测速度较高的目标检测算法在 COCO 测试集上进行了对比实验，表 3 中，检测速度 fps 为帧每秒，由表 3 可知，本文算法的检测速度与 YOLOv2 相当，低于 YOLOv3 和 Light-head R-CNN，但是检测速度也很快。而且检测精度 mAP 是高于其它的目标检测算法的。

5. 结语

本文提出了一种新型的基于卷积神经网络的目标检测方法，在现在的双阶段目标检测方法中，发现各种其他检测模型的优势，利用 ResNet-101 和 Deep separable convolution 来提取图片特征，大大地降低了提取到的特征图维度，减少了计算量。池化层选择 PS-ROI Align，提高了检测性能，以及对小目标的检测能力，同时为了提高检测精度，我们采用了关键点导向的方法来替代传统的回归框方法来对目标物体进行定位。在未来的工作中，我们将继续优化网络模型，从而进一步提高目标检测的精度和检测速度。

参考文献

[1] Redmon, J., Divvala, S., Girshick, R., *et al*. (2016) You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

[2] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 7263-7271. https://doi.org/10.1109/CVPR.2017.690

[3] Redmon, J. and Farhadi, A. (2018) Yolov3: An Incremental Improvement.

[4] Liu, W., Anguelov, D., Erhan, D., *et al*. (2016) SSD: Single Shot Multibox Detector. In: *European Conference on Computer Vision*, Springer, Cham, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[5] Girshick, R., Donahue, J., Darrell, T., *et al*. (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 580-587. https://doi.org/10.1109/CVPR.2014.81

[6] Girshick, R. (2015) Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[7] Dai, J., Li, Y., He, K., *et al*. (2016) R-FCN: Object Detection via Region-Based Fully Convolutional Networks. In: *Proceedings of the* 30*th International Conference on Neural Information Processing Systems*, Curran Associates Inc., Red Hook, 379-387.

[8] Ren, S., He, K., Girshick, R., *et al*. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Proceedings of the* 28*th International Conference on Neural Information Processing Systems*, Volume 1, 91-99.

[9] Uijlings, J.R.R., Van De Sande, K.E.A., Gevers, T., *et al*. (2013) Selective Search for Object Recognition. *International Journal of Computer Vision*, **104**, 154-171. https://doi.org/10.1007/s11263-013-0620-5

[10] He, K., Zhang, X., Ren, S., *et al*. (2015) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

[11] Lu, X., Li, B., Yue, Y., *et al*. (2019) Grid R-CNN. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, 15-21 June 2019, 7363-7372.

[12] Long, J., Shelhamer, E. and Darrell, T. (2015) Fully Convolutional Networks for Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 3431-3440. https://doi.org/10.1109/CVPR.2015.7298965

[13] He, K., Zhang, X., Ren, S., *et al*. (2016) Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[14] Russakovsky, O., Deng, J., Su, H., *et al*. (2015) Imagenet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, **115**, 211-252. https://doi.org/10.1007/s11263-015-0816-y

[15] Howard, A.G., Zhu, M., Chen, B., *et al*. (2017) Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

[16] Szegedy, C., Liu, W., Jia, Y., *et al*. (2015) Going Deeper with Convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 1-9. https://doi.org/10.1109/CVPR.2015.7298594

[17] Li, Z., Peng, C., Yu, G., *et al*. (2017) Light-Head R-CNN: In Defense of Two-Stage Object Detector.

[18] Lin, T.Y., Maire, M., Belongie, S., *et al*. (2014) Microsoft Coco: Common Objects in Context. In: *European Conference on Computer Vision*, Springer, Cham, 740-755. https://doi.org/10.1007/978-3-319-10602-1_48

[19] Goyal, P., Dollár, P., Girshick, R., *et al*. (2017) Accurate, Large Minibatch SGD: Training Imagenet in 1 Hour.

[20] Deng, J., Dong, W., Socher, R., *et al*. (2009) Imagenet: A Large-Scale Hierarchical Image Database. *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, 20-25 June 2009, 248-255. https://doi.org/10.1109/CVPR.2009.5206848

[21] Hosang, J., Benenson, R. and Schiele, B. (2017) Learning Non-Maximum Suppression. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 4507-4515. https://doi.org/10.1109/CVPR.2017.685

[22] He, K., Gkioxari, G., Dollár, P., *et al*. (2017) Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 October 2017, 2961-2969. https://doi.org/10.1109/ICCV.2017.322

[23] Dai, J., Qi, H., Xiong, Y., *et al*. (2017) Deformable Convolutional Networks. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 October 2017, 764-773. https://doi.org/10.1109/ICCV.2017.89

[24] Lin, T.Y., Goyal, P., Girshick, R., *et al*. (2017) Focal Loss for Dense Object Detection. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 October 2017, 2980-2988. https://doi.org/10.1109/ICCV.2017.324

[25] Lin, T.Y., Dollár, P., Girshick, R., *et al*. (2017) Feature Pyramid Networks for Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 2117-2125. https://doi.org/10.1109/CVPR.2017.106

[26] Fu, C.Y., Liu, W., Ranga, A., *et al*. (2017) DSSD: Deconvolutional Single Shot Detector.