

PSRR-MaxpoolNMS: 带关系恢复的金字塔移位 MaxpoolNMS

摘要

非极大值抑制(NMS)是现代卷积神经网络用于目标检测的一个重要后处理步骤。与本质上并行的卷积不同, NMS的实际标准, 即 GreedyNMS, 并行化很困难, 因此它可能成为卷积目标检测管道中的性能瓶颈。MaxpoolNMS 是作为 GreedyNMS 的可并行化替代品引入的, 这反过来使得在精度相当的情况下速度比 GreedyNMS 更快。然而, MaxpoolNMS 只能和 Faster-RCNN 一样在两级检测器的第一阶段替换 GreedyNMS。在最终检测阶段应用 MaxpoolNMS 时, 由于 MaxpoolNMS 无法在边界框选择方面精确逼近 GreedyNMS, 因此精度会显著下降。在这篇论文中, 我们提出了一种通用的、可并行化的、可配置的方法 PSRR-MAXPOOLNMS, 以完全取代所有检测器中所有阶段的 GreedyNMS。通过引入简单的关系恢复模块和金字塔移位的 MaxpoolNMS 模块, 我们的 PSRR MaxpoolNMS 能够比 MaxpoolNMS 更精确地逼近 GreedyNMS。综合实验表明, 我们的方法大大优于 MaxpoolNMS, 并且证明了在同精度条件下比 GreedyNMS 更快。PSRR MaxpoolNMS 首次为定制硬件设计提供了一个完全并行化的解决方案, 可以在任何地方重用该解决方案来加速 NMS。

1. 介绍

目标检测是计算机视觉的关键任务之一, 其目标是对场景中的目标进行定位和分类。在过去的几年中, 深度卷积神经网络已经成为网络的冠军目标检测[9, 21, 19]。卷积目标检测器大致分为 SSD[19]和 YOLO[20]等一级检测器或 Faster-RCNN[21]和 R-FCN[3]等两级检测器, 其中卷积通常占据大部分的计算。另一方面, 通过利用其固有的并行性, 使用更高性能专用硬件例如 GPU 和 Google TPU, 以加速卷积运算方面取得了重大进展[15]。因此, 花在卷积运算上的执

行时间迅速减少, 例如以毫秒为单位。

非极大值抑制(NMS)作为所有卷积目标检测器必须具备的后处理技术, 很可能成为目标检测管道中的性能瓶颈[2]。NMS 的实际标准, 即 GreedyNMS, 由对成千上万个边界框的置信度分数进行排序操作组成, 然后嵌套 for 循环, 贪婪地选择分数高的框, 并移除与所选框明显重叠的框。与本质上是并行的卷积不同, 由于嵌套 for 循环, GreedyNMS 无法轻易并行。因此, GreedyNMS 将逐渐主导卷积对象检测器的执行时间[2], 专用硬件(例如, 从 P100 到 V100 GPU)上的并行性的提升, 才使卷积运行速度更快。

MaxpoolNMS[2]作为 GreedyNMS 的一种可并行化替代方案, 被用来大幅度加速 GreedyNMS, 而不会导致检测精度的损失。MaxpoolNMS 的灵感来源于这样一个观察: 具有高置信度分数的边界框与所谓的置信度分数图上的峰值相关, 其中锚框之间的空间关系保持不变。因此, NMS 可以设计为分数图上的简单最大池, 该图对置信度分数、比例、比率和锚箱的空间位置进行编码(参见下图 1)。在最大池化之后, 仅保留具有峰值分数的框, 而抑制其他框。就执行时间而言, MaxpoolNMS 比 GreedyNMS 运行得快得多, 这主要归功于最大池操作本质上是并行的。然而, MaxpoolNMS 仅用于在两级探测器的第一阶段替换 GreedyNMS, 例如, 在 Faster-RCNN 中区域建议网络之后的 GreedyNMS。在两级检测器的第二级直接应用 MaxpoolNMS 时, 检测精度会显著下降, 例如, 在 faster-RCNN 中检测网络之后的 GreedyNMS(参见图 1 顶部的可视化示例和表 1 中的定量结果)。这降低了 MaxpoolNMS 的价值, 因为 MaxpoolNMS 的定制硬件无法在所有探测器的所有阶段重复使用以取代 GreedyNMS。

在本文中，我们提出了一种通用方法，即 PSRR MaxpoolNMS，以在所有检测器的所有阶段完全取代 GreedyNMS。关键是尽可能精确地近似 GreedyNMS，这可以通过 GreedyNMS 和近似方法之间选定边界框的重叠率来衡量。正如低重叠率（见表 1）所证明的，MaxpoolNMS 无法近似 GreedyNMS，主要原因是 (A) 置信得分图上的得分图不匹配问题（见图 2）和 (B) 利用置信得分图上的单扫描最大池难以最大化得分图稀疏性（见图 4 和图 5）。PSRR MaxpoolNMS 引入了关系恢复模块和金字塔移位的 MaxpoolNMS 模块，分别用于解决问题 (A) 和 (B)。因此，我们的 PSRR MaxpoolNMS 能够比 MaxpoolNMS 更精确地逼近 GreedyNMS（参见表 1 中的重叠率）

我们将我们的贡献总结如下：

- PSRR MaxpoolNMS 的一般方法，用于在所有卷积目标检测器的所有阶段加速 NMS

- 一个关系恢复模块，用于在将边界框投影到置信度分数图时纠正分数图不匹配，从而实现更精确的比例、纵横比和方框之间的空间关系

- 金字塔移动了置信度得分图上的 MaxpoolNMS，显著增加了得分图的稀疏性，从而消除了更多的重叠框

- 在 PSRR MaxpoolNMS 中，关系恢复和金字塔移位的 MaxpoolNMS 是简单且可并行化的操作。因此，PSRR MaxpoolNMS 首次为定制硬件设计提供了一个完全并行化的解决方案，可以在任何地方重复使用该解决方案来加速 NMS

- 最后，我们的 PSRR MaxpoolNMS 大大优于 MaxpoolNMS。此外，在相同精度下它被证明比 GreedyNMS 更快。

2. 相关工作

2.1 一级和两级目标检测器

卷积目标检测框架大致分为一级检测器和两级检测器。SSD 和 YOLO[19, 20, 17] 等单级检测器通过将整个图像通过单个统一网络直接预测边界框坐标和类概率。两级检测器，如更快的 RCNN[21, 11, 9] 基于类不可知候选区域。候选区域是可能包含目标对象的候选边界框。与以前的 RCNN[10] 或 Fast RCNN[9] 不同，前者采用手工制作的候选区域生成[25]，后者通过训练区域提取网络 (RPN) 生成候选区域。候选区域的特征被输入后续检测网络，以预测每个候选区的最终框坐标和特定类别概率。

2.2 非极大值抑制

目标检测器的最终目标是精确输出一个边界框，以紧密包围每个目标对象。然而，大多数对象检测管道倾向于生成冗余的高度重叠的边界框来包围对象，因此会引入大量误报。

非极大值抑制 (NMS) 是抑制冗余边界框的一个重要步骤。最广泛使用的 NMS 方法是 GreedyNMS[4]。GreedyNMS 首先按置信度得分按降序对框进行排序，然后从其余框中迭代选择最有置信度的预测，并消除与所选框重叠较大的所有其他框。有多种 NMS 可提高检测精度[1、18、13、7、22]。SoftmS[1] 减少要抑制的框的分数，而不是通过硬阈值删除这些框。自适应 NMS[18] 学习根据对象密度自适应设置框选择阈值。Hosang 等人[13] 将 NMS 重新定义为可端到端训练的 ConvNet。可见性引导 NMS[7] 利用对整个对象的检测以及对可见部分的检测来解决高度遮挡的对象检测问题。FeatureNMS[22] 利用特征嵌入距离来确定是抑制还是保留候选框。

硬件感知的 NMS 加速研究较少。MaxpoolNMS[2] 将 NMS 重新格式化为置信度得分映射上的最大池，以删除冗余框。最大池操作本质上是并行的，因此 MaxpoolNMS 比 GreedyNMS 效率更高，因为 GreedyNMS 不容易并行。然而，MaxpoolNMS 仅限于两级探测器的区域提取网络 (RPN)，不能推广到包括一级探测器在内的所有探测器的所有阶段。

3. 方法

在本节中，我们首先简要回顾 MaxpoolNMS[2]（第 3.1 节）并分析其局限性（第 3.2 节）。然后介绍我们的 PSRR MaxpoolNMS 以解决这些限制。PSRR MaxpoolNMS 由两个步骤组成：关系恢复（第 3.3 节），然后是金字塔移位的 MaxpoolNMS（第 3.4 节）。

3.1 重看 MaxpoolNMS

MaxpoolNMS[2]是一种有效的NMS方法，专门设计用于在 FasterRCNN 检测管道的第一阶段（即区域提取网络（RPN））移除重叠锚箱。MaxpoolNMS 由两个模块组成。首先，如图 2 所示，它构造了一组置信度得分图，其中每个得分图对应于锚定框比例和比率（即通道 c ）的特定组合，并且得分图上的每个单元对由 PRN 生成的锚箱的目标得分（即单元值）和空间位置（即图上的 x 和 y ）进行编码。例如，如果我们对具有下采样率 β 的 RPN 使用 4 个锚定框标度 {642, 1282, 2562, 5122} 和 3 个锚定框比率 {1:2、1:1、2:1}（例如， $\beta=16$ ），将得到 12 个宽度为 W/β 和高度为 H/β 的置信度得分图，其中 W 和 H 分别表示图像宽度和高度。其次，基于观察到的置信度得分图上的峰值得分对应于目标对象，在该图上进行一个一个简单的最大池操作，以抑制得分较低的锚盒，并仅保留得分最高的锚盒。此外，由于每个分数映射都专用于特定的锚定框大小，因此该映射上最大池的内核大小和池步长由其关联的锚定框大小确定。

图 2. MaxpoolNMS[2]将锚定框（红色虚线）投影到比率为 1:2（珊瑚色）的得分图），不考虑框回归。这会导致得分图不匹配问题，即锚定的回归框（纯绿色）的比率发生变化，因此会以 1:1（蓝灰色）的比率投影到另一个得分图。回归框的投影是正确的，因为它比相应的锚框更准确地包围摩托车。

其中 k_x , k_y 是内核尺寸， s_x , s_y 分别是 x 方向和 y 方向上的池步。 w 、 h 表示特定分数图上的锚箱尺寸（宽度和高度）。 β 是

得分图的下采样率。 α 表示重叠阈值，用于折中考虑精度和召回率。较大的 α 将抑制更多的重叠框（导致更高的精度），但存在漏检的风险（导致更低的召回率）。

此外，MaxpoolNMS 中的最大池有 3 种变体：（1）单通道 MaxpoolNMS 或多尺度 MaxpoolNMS，在每个分数图（通道）上独立应用最大池。（2）交叉比率 MaxpoolNMS 以每个比例的不同比率连接分数图，然后在连接图上连接 3D max pooling。（3）交叉比例 MaxpoolNMS 在每个比率的相邻比例处连接分数图，然后在连接图上连接 3D max pooling。

最后，将分数图上剩余的锚框按分数降序进行组合和排序。仅返回顶框作为最终检测。

3.2 MaxpoolNMS 的局限性

虽然 MaxpoolNMS 的执行速度比 GreedyNMS 快得多，因为它与简单的 max pooling 操作并行，但 MaxpoolNMS 存在一个巨大的缺点，即它只用于在常规两级卷积对象检测器的第一级替换 GreedyNMS。为了保持较高的检测精度，GreedyNMS 仍然是两级探测器的第二级和单级探测器（如 SSD）必不可少的后处理方法。这使得 MaxpoolNMS 的吸引力降低，因为它降低了 MaxpoolNMS 定制硬件的价值，而 MaxpoolNMS 无法在所有探测器的所有阶段轻松重用以加速 NMS。

我们观察到，在两级探测器的第二级应用 MaxpoolNMS 时，检测精度显著下降。具体来说，我们在 Faster-RCNN 的检测网络之后执行 MaxpoolNMS，以移除重叠框，并以 ResNet-50 作为主干。如表 1 所示，在 PASCAL VOC 数据集上，MaxpoolNMS 的性能明显低于 GreedyNMS，mAP 下降超过 50%。如上图 1 顶部所示，可以看到 MaxpoolNMS 的最终选择框与 GreedyNMS 的最终选择框明显不同，这导致我们假设 MaxpoolNMS 的性能差是因为它无法很好地逼近 GreedyNMS。我

们将近似的质量度量为 MaxpoolNMS 和 GreedyNMS 之间选定边界框的重叠率。如表 1 所示, mAP 随着重叠率的增加而增加, 但 MaxpoolNMS 的重叠率较低。

我们发现有两个关键因素导致 MaxpoolNMS 的低重叠率, 置信度得分图上的得分图不匹配, 以及在置信度得分图上单次扫描最大化得分图稀疏性的困难。

得分图不匹配发生在置信度得分图的构建过程中。MaxpoolNMS 在不考虑框回归的情况下, 将锚定框投影到地图上进行评分。如果与锚定框相对应的回归框在位置、比例或纵横比上发生了显著变化, 则会导致分数图不匹配问题。图 2 示出了比率变化的一个示例。这种不匹配会导致分数图上出现错误的方框投影, 进而对以下最大池操作产生负面影响分数图上的低稀疏性。由于 MaxpoolNMS 仅在置信度分数图上进行单扫描最大池操作, 因此在密集分数图上很难实现高稀疏性, 这意味着池化后仍保留大量高重叠盒, 如图 4 左侧所示 (即, 仅使用单通道的最大池化)。此外, 置信度得分图上的单个扫描最大池将导致边缘效应。如图 5 所示, 即使相邻单元的两个框其中一个被视为重复, 但在最大池化后都被保留下来了。

3.3 关系恢复

不是将锚框投影到置信度得分图, 而是将回归框投影到信任度得分图, 从而解决了得分图不匹配的问题。在框回归的帮助下, 回归框在空间位置、大小和形状 (即比例和纵横比) 方面通常比相应的锚框更准确地包围对象。因此, 由回归框投影的置信度得分图能够更好地反映场景中对象之间的实际空间和通道 (比例和比率的组合) 关系。具体地说, 关系恢复模块由三部分组成: 空间和通道恢复, 用于识别回归框的空间位置 (X, Y) 和通道 $(C(s, r))$, 然后是确定映射中每个单元的置信度分数的分数分配 (见图 3)。

空间恢复。MaxpoolNMS 将锚定框投影到

得分图上错误的空间位置, 这是由于框回归后位置发生了明显变化。为了解决此位置不匹配问题, 给定输入图像中回归框的中心位置 $[x_c, y_c]$, 空间恢复将其映射到得分图上的空间索引 $[X, Y]$, $X = \lfloor x_c / \beta \rfloor$, $Y = \lfloor y_c / \beta \rfloor$, 其中 β 是得分图的下采样率。

通道恢复。MaxpoolNMS 仅基于锚盒的默认比例 (s_0) 和比率 (r_0) 将锚盒投影到分数图的通道 $(C(s_0, r_0))$ 。同样, 如果相应的回归框在比例和/或比率上发生了显著变化, 则通道投影可能是错误的。为了解决这个信道失配问题, 给定一个大小为 $w \times h$ 的回归盒, 信道恢复基于欧几里德距离计算最接近的标度 s 到 $w \times h$, 以及最接近的比率 r 到 h/w , 选择 $C(s, r)$ 作为框的投影通道。

分数分配。确定所有框的空间位置和通道后, 图中的每个单元可以有多个框投影到其中。因此, 引入了分数分配, 只保留每个单元格中分数最高的框。可能会注意到, 分数分配基本上是分数映射的每个单元格中的 1×1 最大池, 因此可以将其视为用于移除易于识别的重叠框的预过滤步骤。

评价。关系恢复方法的所有操作都简单且高度并行。此外, 关系恢复方法是无锚的。换句话说, 作为 PSRRMaxpoolNMS 的第一步, 关系恢复为将 PSRRMaxpoolNMS 从基于锚定的一级或两级卷积对象检测器扩展到无锚定卷积对象检测器提供了可能性 [16, 5], 因为置信度得分图的构建根本不会取决于锚定框。相反, 它只需要回归框的位置和大小, 在无锚探测器中也可以访问回归框。我们将把它作为我们今后的工作。

3.4 金字塔移位 MaxpoolNMS

我们提出金字塔移位 MaxpoolNMS 来移除置信度得分图上的重叠框, 其中金字塔移位 MaxpoolNMS 旨在彻底抑制跨通道 (比例和比率) 的重叠框, 而移位的 MaxpoolNMS 旨在通过解决边缘效应问题, 有效消除空间域中的重叠盒。在金字塔移位的 MaxpoolNMS 之

后，分数贴图变得非常稀疏，只有少量非零单元格。非零单元格中的方框作为最终检测返回。

金字塔 MaxpoolNMS。一方面，MaxpoolNMS 仅在置信度得分图上运行单扫描最大池。另一方面，MaxpoolNMS 假设重叠框仅存在于分数图上具有相邻比例（或比率）的通道中，这并不总是正确的，因为重叠框可以以任意比例/比率分布（例如，被卡车遮挡的迷你库珀）。因此，具有无效假设的单扫描最大池不足以有效抑制重叠框，从而导致池化后分数图上的低稀疏性。可以增加等式 1 中的重叠阈值 α ，以得到更高的稀疏性，但有漏检的风险 [2]。

我们建议金字塔 MaxpoolNMS 通过在具有不同信道组合的置信度得分图上一个接一个地执行最大池序列来逐步得到得分图稀疏性，如图 4 所示。最大池的顺序从单通道最大池开始，然后是交叉比率和交叉比例最大池，最后是交叉所有通道最大池。如第 3.1 节所述，单通道最大池在单分数图上独立运行，而交叉比率和交叉比例最大池通过在相邻比率/比例处连接通道在多数分数图上运行。此外，我们引入了跨所有通道的最大池，它在所有通道上操作池。通过这种方式，我们的金字塔 MaxpoolNMS 逐渐将池算子的“感受野”从局部（单分数图）增加到全局（所有图），因此不需要对重叠框的分布进行任何假设。

当在单个通道上独立操作最大池时，每个通道的内核大小和步长设置为等式 1。在跨多个通道操作最大池时，内核大小（或步长）设置为连接通道的内核大小（或步长）的最小值。首先，如果内核大小大于最小值，则它可能会抑制以前的单通道最大池检测到的真值。其次，标度/比率之间的差距越大，重叠框的可能性就越小，因此，较小的内核大小（或步长）可以降低抑制真值的风险。

移位的 MaxpoolNMS。移位的 MaxpoolNMS 可以进一步增加分数图的稀疏性，从而更有

效地消除空间域 (X, Y) 中的重叠框。这是通过在置信度得分图上引入具有空间移位的额外最大池来实现的，这反过来解决了边缘效应问题，如图 5 所示。具体地说，给定一个内核大小 k ，移位的最大池在边界周围填充了 k^2 的零的分数图上进行操作。最后，可以在金字塔 MaxpoolNMS 序列中的每个池步骤之后追加移位的最大池。

4. 实验

4.1 实验设置

我们仅在 Faster-RCNN [21] 和 SSD [19] 的预测阶段评估不同的 NMS 方法。(1) RCNN 是两级卷积对象。我们使用 ResNet-50、ResNet-101 和 ResNet152 [12] 作为骨干网络架构。对于 FasterRCNN 的训练，我们使用了 PyTorch 公开的默认训练参数。由于 MaxpoolNMS [2] 可视为我们的 PSRR MaxpoolNMS 的简化版本，因此我们简单地用多尺度（或单通道）MaxpoolNMS 代替 GreedyNMS，作为 Faster-RCNN 第一阶段的后处理技术，该技术实现了相同的精度，但运行速度远快于 GreedyNMS。(2) SSD [19] 是一种单级卷积目标检测器。我们使用 VGG-16 [24]、MobileNet v1 [14]、MobileNet-v2 [23] 作为主干。我们使用 PyTorch 提供的预培训模型评估 NMS。在预处理阶段，SSD 首先过滤出每个类得分 < 0.01 的边界框。对于 GreedyNMS，它进一步从通过预处理阶段的框中选择 200 个得分最高的框。我们的 PSRR MaxpoolNMS 将所有通过预处理的框作为输入。

对于 faster-RCNN 和 SSD，我们的 PSRR MaxpoolNMS 用于抑制最终边界框预测。我们将 α 的值固定为 0.75。在通道恢复步骤中，我们将锚定设置为 $\{642, 1282, 2562, 5122\}$ 以及 $[0.5, 1, 2]$ 的比率。对于交叉比率 MaxpoolNMS，每个标度的所有 3 个比率都连接起来以实现最大池化。对于跨尺度 MaxpoolNMS，对于最大池化的每个步骤，我们仅将两个具有相邻尺度的通道连接起来。我们将单通道 MaxpoolNMS、交叉比率 MaxpoolNMS 和交叉比例 MaxpoolNMS 分别表

示为单通道、比率和比例。我们在 PASCAL VOC[6]和 KITTI[8]数据集上进行了实验。对于 PASCAL VOC, 我们使用 2007 和 2012 trainval 数据集训练 FasterRCNN 检测模型, 并在 2007 测试数据集上进行评估。我们得到了 PASCAL VOC 数据集的平均精度 (mAP)。对于 KITTI, 我们将数据集随机分成 5611 张训练图像和 1870 张测试图像。我们在 KITTI 上得到了从易到难的难度级别的 mAP。

4.2 与 MaxpoolNMS 的比较

我们在 FasterRCNN 的第二阶段将 MaxpoolNMS[2]与我们的 PSRRMaxpoolNMS 方法进行比较。PASCAL VOC 数据集的结果如表 1 所示。首先, 我们观察到 MaxpoolNMS 的性能很差, 例如 MaxpoolNMS 单个 33%, 而 GreedyNMS 为 78.1%。正如预期的那样, 尽管检测 mAP 随着重叠率的增加而增加, 但 MaxpoolNMS 和 GreedyNMS 之间选定框的重叠率仍然非常低。其次, 我们的 PSRR MaxpoolNMS 更接近 GreedyNMS, 这可以通过大重叠率和与 GreedyNMS 相当的检测精度 (mAP 下降小于 1%) 来证明。值得注意的是, 与[2]类似, 我们的方法需要设置的唯一参数是重叠阈值 α 。因此, 我们的方法 PSRR MaxpoolNMS 不引入额外的参数调整工作负载, 而比 MaxpoolNMS 有很大的优势。

4.3 与 GreedyNMS 的比较

我们在各种数据集和卷积目标检测器上将 PSRR MaxpoolNMS 与 GreedyNMS 进行比较。首先, 我们使用 Faster-RCNN 检测器在 KITTI 数据集上进行实验, 并在表 2 中显示了检测结果。我们观察到, 无论使用何种主干模型, 我们的方法在大多数操作点都能达到与 KITTI 上的 GreedyNMS 相当的检测精度。其次, 我们使用两级 (即 Faster-RCNN) 和一级 (即 SSD) 检测器在 PASCAL VOC 数据集上进行实验。如表 3 所示, 我们可以看到, 我们的方法的性能稍差 (大多数操作点低于 1%) 于具有不同的主干模型和不同的对象检测器的 GreedyNMS。使用 Faster-RCNN 检测器和 ResNet-152 作为主干,

PSRR MaxpoolNMS 和 GreedyNMS 之间的 mAP 性能差距仅为 0.3%。还值得注意的是, 我们的 PSRR MaxpoolNMS 方法适用于各种对象检测管道。

4.4 效率

我们对我们的方法 PSRRMaxpoolNMS 的计算效率进行了理论和实验分析。表 4 提供了 GreedyNMS 和我们的方法的时间复杂度的理论分析。给定输入框的数量 N , 关系恢复和金字塔移位 MaxpoolNMS 的时间复杂度都是 $O(N)$, 这比 GreedyNMS $O(N \log N) + O(N^2)$ 的时间复杂度小得多。此外, 关系恢复和金字塔移位的 MaxpoolNMS 都可以很容易地并行化, 从而进一步缩短 PSRR MaxpoolNMS 的执行时间。

图 6。我们的方法和 GreedyNMS 的执行时间 (以毫秒为单位), 作为正在处理的边界框数量的函数。我们还报告了我们的方法 (关系恢复 (RR) 和金字塔移位 (PS) MaxpoolNMS) 的时序分解。这两种方法都在 CPU 上运行。

我们还测量了 GreedyNMS 和我们的方法在 Intel (R) Core (TM) i9-10900X CPU 上的执行时间, 并处理了不同数量的边界框。我们在 PASCAL VOC 数据集上对以 VGG-16 为主干的 SSD 进行了实验。为了公平比较, 我们删除了分数阈值步骤, 以便设置每个类中每个图像处理的输入框数量。结果如图 6 所示。首先, 随着输入边界框数量的增加, 我们的 PSRR MaxpoolNMS 比 GreedyNMS 的效率越来越高。当框数增加到 8000 时, PSRR MaxpoolNMS 的执行时间为 89ms, 而 GreedyNMS 的执行时间为 512ms, 几乎比我们方法慢了 6 倍。其次, 我们研究 PSRRMaxpoolNMS 的时序细分。我们观察到, 金字塔移位的 MaxpoolNMS 的执行时间几乎是恒定的, 而关系恢复的执行时间随着盒数的增加而线性增加, 这是因为我们自己的关系恢复实现目前没有并行化。对于金字塔移位的 MaxpoolNMS, 我们依赖于已经在 CPU 上并行化的 PyTorch Maxpool 和

MaxUnpool API。

4.5 消融研究

在本节中，我们进行消融研究，以评估我们方法中的不同成分，即关系恢复和金字塔移位的 NMS。所有实验结果均在 PASCAL VOC 数据集上报告，使用的是 ResNet-50 作为主干的 Faster-RCNN 检测器

4.5.1 关系恢复

空间和通道恢复。我们分析了金字塔 MaxpoolNMS 的每个组件对空间和信道恢复的影响。结果见表 5。可以看出，与缺少恢复关系的基线（因为它基于锚箱投影）相比，无论后续最大池阶段中使用的通道组合如何，空间和通道恢复步骤都可以缓解分数图不匹配问题，并大大提高基线的检测精度。分数分配。如前所述，对于分数映射的每个单元格中的分数分配，我们只保留分数最高的框，这可以视为基于每个单元格中最大池的预过滤步骤（最大分配）。我们还研究了最大分配之外的备选方案，即随机分配和总和分配。”“随机分配”是指随机选择投影到单元格的框（及其分数值） $\sum \text{assign}$ ；是将投影到单元格的所有框的得分值相加。与最大分配一样，随机分配和总和分配都可以轻松并行。表 6 报告了具有不同分数分配变量的检测图。我们观察到最大分配性能最好，其次是和分配和随机分配。

4.5.2 棱锥变换的 MaxpoolNMS

棱锥变换的 MaxpoolNMS。我们评估了序列中信道组合数量以及金字塔 MaxpoolNMS 定义的序列中的执行顺序的影响。表 7 报告了检测结果。首先，单扫描最大池（即，单扫描、比率、缩放、全部）的性能始终低于序列中预定义的多扫描最大池（例如，序列单个+比率具有 2 个最大池），这意味着金字塔 MaxpoolNMS 的必要性。其次，如果我们以相反的顺序执行序列（例如，对于序列 single+ratio，先执行 ratio，然后执行 single），则性能会比原始执行顺序稍差。用于边缘效果的移位 MaxpoolNMS。我们对移

位的 MaxpoolNMS 进行了研究，并在表 8 中显示了结果。结果表明，在不增加移位的 MaxpoolNMS 来缓解边缘效应的情况下，检测图明显下降约 3.4%。

5. 结论

在本文中，我们提出 PSRR MaxpoolNMS 作为 GreedyNMS 的并行替代方案，用于所有卷积目标检测器中的重叠盒去除。利用所提出的关系恢复模块和金字塔移位的 MaxpoolNMS 模块，我们解决了分数图池化后分数图不匹配和低稀疏性的问题。综合实验表明，PSRR MaxpoolNMS 的检测精度与 GreedyNMS 相当，但在执行时间上有更高的加速比。