QueryDet:加速高分辨率小物体检测的级联稀疏查询法

杨振红* 爱丁堡大学
chenhongyi.yang@ed.ac.uk

黄泽豪
TuSimple
zehaohuang18@gmail.com

王乃彦 TuSimple
winsty@gmail.com

摘要

虽然在过去的几年里，用深度学习进行的一般物体检测已经取得了巨大的成功，但检测小物体的性能和效率却远远不能令人满意。促进小物体检测的最常见和有效的方法是使用高分辨率的图像或特征图。然而，这两种方法都需要昂贵的计算，因为计算成本随着图像和特征大小的增加而成倍增长。为了获得两个世界的最佳效果，我们提出了QueryDet，它使用一种新的查询机制来加速基于特征金字塔的物体检测器的推理速度。该管道由两个步骤组成：首先预测小物体在低分辨率特征上的粗略位置，然后利用高分辨率特征在这些粗略位置的指导下进行精确检测。通过这种方式，我们不仅可以收获高分辨率特征图的好处，还可以避免对背景区域进行无用的计算。在流行的COCO数据集上，所提出的方法将检测的mAP提高了1.0，mAP-small提高了2.0，高分辨率推理速度平均提高到3.0。在包含更多小物体的VisDrone数据集上，我们创造了一个新的先进技术，同时获得了平均2.3的高分辨率加速。代码见https://github.com/ChenhongyiYang/QueryDet-PyTorch。

1. 简介

随着最近深度学习的进展[15, 53]，虚拟物体检测在性能和速度上都取得了巨大的改进[3, 12, 26, 27, 29, 37, 39, 49]。它已经成为自主驾驶和遥感等广泛应用的基础。然而，检测小物体仍然是一个具有挑战性的问题。在小物体和正常规模的物体之间存在着很大的性能差距。以RetinaNet[27]为例，它是目前最先进的检测技术之一。

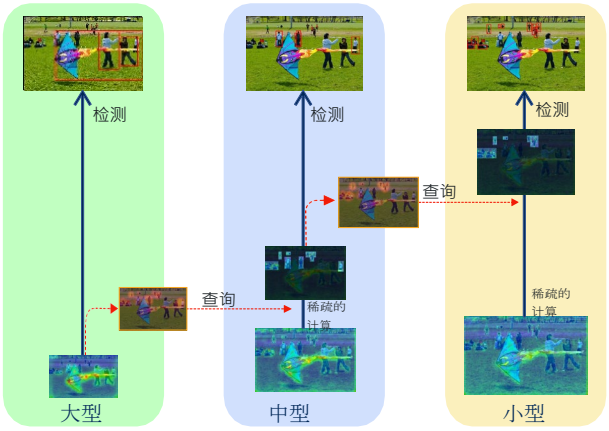

图1.QueryDet在高分辨率特征中实现了高效的小物体检测。首先在低分辨率特征中预测可能存在小物体的位置（查询键），然后利用这些位置的高分辨率特征构建一个稀疏特征图（查询值）。最后，用一个稀疏检测头来输出检测到的盒子。这一范式以级联方式应用，实现了快速而准确的小物体检测。

以COCO[28]*测试开发*集为例，它在大中型物体上实现了44.1和51.2 mAP，但在小型物体上只获得了24.1 mAP。这种退化主要是由三个因素造成的。1）由于卷积神经网络（CNN）主干的下采样操作，突出小物体的特征被消灭了；因此，小物体的特征经常被背景中的噪声所污染。2）如文献[25]所述，低分辨率图像的感受野可能与小物体的尺寸不匹配；3）小物体的定位比大物体的定位更困难，因为边界盒的小扰动可能会对联盟交集（IoU）指标造成重大干扰。

可以通过扩大输入图像的尺寸或降低CNN的下采样率来改进小物体检测，以保持高分辨率的特征，因为他们在

*在TuSimple担任全职研究实习生时完成的工作。

提高所得到的特征图的有效分辨率。然而，仅仅提高特征图的分辨率会产生大量的计算成本。一些工作[1, 26, 29]提出通过重复使用CNN不同层的多尺度特征图来建立一个特征金字塔来解决这个问题。不同尺度的物体被挂在不同的层面上：大型物体往往被检测到高层特征，而小型物体通常被检测到低层。特征金字塔范式节省了在主干中从浅到深维护高分辨率特征图的计算成本。尽管如此，低层特征的检测头的计算复杂性仍然是巨大的。例如，在RetinaNet中增加一个额外的金字塔级别P_2，将使检测头的计算量（FLOPs）和内存成本增加300%；因此，在NVIDIA 2080Ti GPU上，推理速度从13.6 FPS严重降低到4.85 FPS。在本文中，我们提出了一种简单而有效的方法--QueryDet，以节省检测头的计算量，同时提高小物体的性能。其动机来自于两个关键的观察。1）低层次特征的计算是高度冗余的。在大多数情况下，小物体的空间分布是非常稀疏的：它们只占据高分辨率特征图的少数部分；因此，大量的计算被浪费了。2）特征金字塔是高度结构化的。虽然我们不能准确地检测出低分辨率特征图中的小物体，但我们仍然可以推断出它们的存在，并对其进行粗略的分析。准确度很高的地点。

利用这两个观察结果的一个自然的想法是，我们可以只将检测头应用于小物体的空间位置。这种策略需要以低成本定位小物体的大致位置，并在所需的特征图上进行稀疏计算。在这项工作中，我们提出了QueryDet，它是基于一种新颖的查询机制Cascade Sparse Query（CSQ），如图1所示。我们递归地预测小物体（查询）在低分辨率特征图上的粗略位置，并利用它们来指导高分辨率特征图的计算。在稀疏计算[13, 55]的帮助下，我们大大降低了低级特征的检测头的计算成本，同时保持了小物体的检测精度。请注意，我们的方法是为了在空间上节省计算，所以它与其他加速方法兼容，如轻量级骨干[44]、模型修剪[16]、模型量化[51]和知识提炼[5]。

我们在COCO检测基准[28]和包含大量小物体的挑战性数据集VisDrone[59]上评估了我们的QueryDet。我们表明我们的方法可以大大加快推理速度，同时提高检测性能。综上所述，我们有两个主要贡献。

- 我们提出了QueryDet，其中有一个简单而有效的设计了级联稀疏查询（CSQ）机制。它可以降低所有基于特征pyramid的物体检测器的计算成本。我们的方法通过有效地利用高分辨率的特征，同时保持快速的推理速度，可以提高对小物体的检测性能。

- 在COCO上，QueryDet通过利用高分辨率特征将RetinaNet基线提高了1.1AP和2.0AP$_s$，而当采用CSQ时，高分辨率检测速度平均提高了3.0。在VisDrone上，我们在检测MAP方面提高了最先进的结果，并增强了高分辨率检测的能力。
分辨率检测速度平均提高了2.3倍。

2. 相关作品

物体检测。基于深度学习的物体检测主要可以分为两类：两阶段检测器[2, 11, 12, 26, 39]和单阶段检测器。由YOLO开创的两阶段方法[17, 29, 35-37, 58]。一般来说，两阶段方法往往比单阶段方法更准确，因为它们使用RoIAlign操作[14]来明确对齐一个物体的特征。然而，这两种流的性能差距最近有所缩小。RetinaNet[27]是第一个基于锚点的单阶段检测器，其性能与两阶段检测器相当。它使用特征金字塔网络（FPN）[26]进行多尺度检测，并提出Focal Loss来处理密集训练中的前景-背景不平衡问题。最近，单阶段无锚检测器[7, 7, 21, 23, 45, 56]由于其简单性而引起了学术界的关注。在本文中，我们基于RetinaNet和FCOS[45]实现了我们的QueryDet，以显示其有效性和泛化能力。

小物体识别。由于低分辨率的特征，小物体识别，如检测和分割，是一项具有挑战性的计算机视觉任务。为了解决这个问题，已经有大量的工作被提出。这些方法主要可分为四种类型。1）提高输入特征的分辨率[1, 10, 22, 24, 26, 29, 41, 48]；2）超采样和强大的数据增强[20, 29, 60]；3）纳入语境信息[4,6,57]，以及4)规模感知的训练摄取[25, 26, 42, 43]。

空间冗余。一些方法使用稀疏计算，以不同的方式利用CNN的空间冗余来节省计算成本。Perforated-CNN[9]用不同的确定性采样方法生成掩码。动态卷积[47]使用一个小的门控网络来预测像素掩码，[54]提出一个随机采样和插值网络。这两个

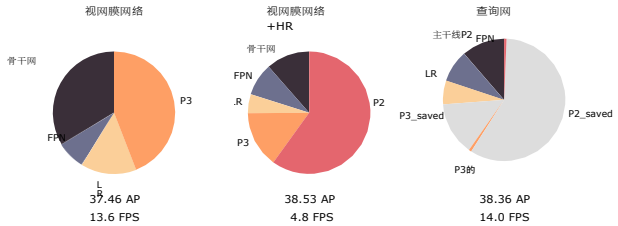

图2.使用ResNet-50骨干网时不同模块的FLOPs分布。在RetinaNet中，高分辨率的P_3的计算成本占总成本的43%；如果加上更高分辨率的P_2，它们总共占总成本的74%。我们的QueryDet可以有效地将这些特征的计算量减少99%，从而获得快速的推理速度并保持较高的检测精度。注：LR代表低分辨率的P_4到P_7。

它们采用Gumbel-Softmax[18]和稀疏损失来训练稀疏掩码。另一方面，空间自适应计算时间（SACT）[8]为每个空间位置预测一个停止得分，该得分由一个亲手摆出的思考成本和特定任务的损失函数监督。SB-Net[38]采用离线路线图或掩码来过滤出被忽略的区域。与这些方法不同的是，我们的QueryDet专注于物体的尺度变化，并简单地采用所提供的地面真实边界盒进行监督。另一个工作流采用了一个两阶段的框架：用于自适应推理的扫视和聚焦。[50]通过强化学习从原始输入图像中选择小区域，并通过动态决策过程来处理这些区域。[46]在物体检测任务上采用了类似的想法。一个与我们的QueryDet类似的工作是AutoFocus[33]。AutoFocus首先预测和裁剪粗略尺度的感兴趣区域，然后按比例放大到较大的分辨率进行最终预测。与AutoFocus相比，我们的QueryDet更有效率，因为"聚焦"操作是在特征金字塔而不是图像金字塔上进行的，这减少了骨干网中的冗余编译。

3. 方法

在这一节中，我们描述了我们的QueryDet用于准确和快速的小物体检测。我们基于RetinaNet[27]来说明我们的方法，这是一个流行的基于锚的密集检测器。请注意，我们的方法并不局限于RetinaNet，因为它可以应用于任何单阶段检测器和带有FPN的两阶段检测器中的re-gion proposal network（RPN）。我们将首先重新审视RetinaNet并分析不同组件的com-putational成本分布。然后，我们将介绍我们如何使用提议的级联稀疏查询来节省推理期间的计算成本。最后，将介绍训练的细节。

3.1. 重新审视视网膜网络

RetinaNet有两部分：一个带有FPN的骨干网络，输出多尺度的特征图；两个检测头，用于分类和回归。当规模输入图像为$H \times W$，FPN特征的大小为$\mathbf{p} = \{p_l \in r^{h' \times w' \times c}\}$。这里$l$表示金字塔的水平，而$(H', W')$通常等于$(\lfloor \frac{H}{2^l} \rfloor, \lfloor \frac{W}{2^l} \rfloor)$在一个

典型的FPN实现。探测头包括
四个3.3卷积层，然后是一个额外的3.3卷积层用于最终预测。为了提高参数效率，不同特征层共享相同的检测头（参数）。然而，不同层的计算成本是高度不平衡的：从P_7到P_3，检测头的FLOPs按特征分辨率的比例以二次方的顺序增加。如图2所示，P_3头占据了近一半的FLOPs，而低分辨率特征P_4到P_7的成本只占15%。因此，如果我们想把FPN扩展到P_2，以获得更好的小物体performance，其成本是无法承受的：高分辨率的P_2和P_3将占据总体成本的75%。在下文中，我们将描述我们的QueryDet如何减少对高分辨率特征的计算，并促进RetinaNet的推理速度，即使有一个额外的高分辨率P_2。

3.2. 通过稀疏查询加快推理速度

在现代基于FPN的检测器的设计中，小物体往往从高分辨率的低级特征图中被检测出来。然而，由于小物体在空间中通常是稀疏的，高分辨率特征图上的密集计算范式是非常低效的。受此启发，我们提出了一种从粗到细的方法来降低低水平金字塔的计算成本。首先，在粗特征图上预测小物体的粗略位置，然后在细特征图上密集地计算对应的位置。这个过程可以看作是一个查询过程：粗略位置是查询键，用于检测小物体的高分辨率特征是查询值；因此我们把我们的方法称为**QueryDet**。
我们的方法的整个管道在图3中显示。

为了预测小物体的粗略位置，我们增加了一个查询头，与分类头和注册头并行。查询头接收特征图P_l，其跨度为2^l，并输出热图$V_l \in R^{H' \times W'}$，其跨度为2。$V^{i,j}$表示网格(i, j)的概率。
含有一个小物体。在训练过程中，我们定义小物体为在每个层次上的目标都是尺度小于预先定义的阈值s_l。为了简单起见，我们将s_l设为P_l上的最小锚点尺度，对于无锚检测器，则设为P_l上的最小回归范围。对于一个小对象o，我们通过计算其中心位置(x_o, y_o)与特征图上每个位置之间的距离，为查询头编码目标图。

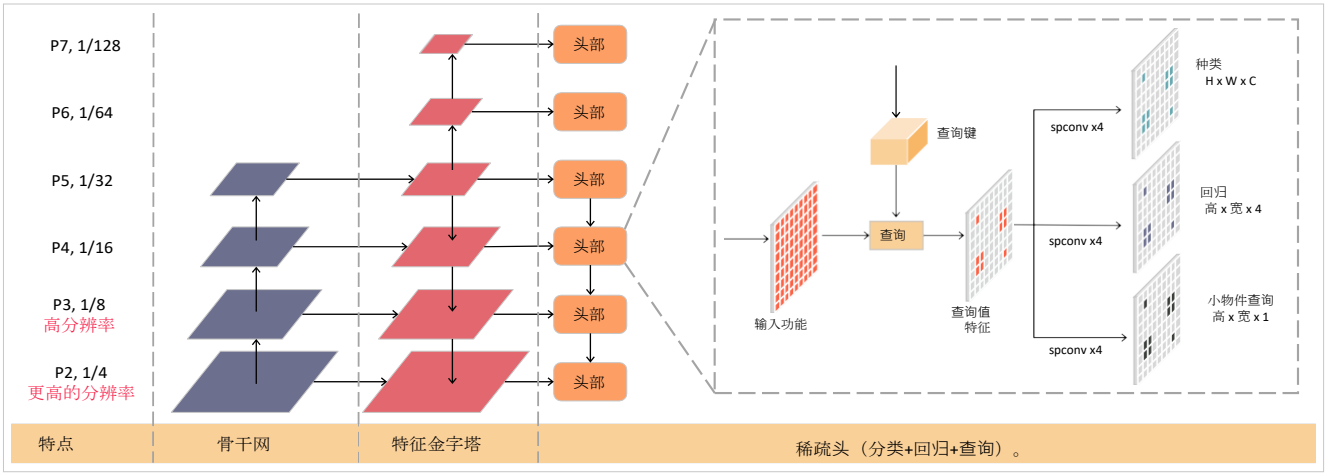

图3.所提出的QueryDet的整个流水线。

图像被送入骨干网和特征金字塔网络（FPN），产生一系列不同分辨率的特征图。从查询起始层（本图中的P_5）开始，每一层都从上一层接收一组关键位置，并应用查询操作来生成稀疏值特征图。然后，稀疏检测头和稀疏查询头为下一层预测相应比例和关键位置的检测盒。

距离小于s_l，则为1，否则为0。那么Querycenters $\{(x^o, y^o)\}$。头部是用FocalLoss[27]训练的。在推理过程中，我们选择那些预测分数大于一个阈值σ作为查询。然后，q^o将被映射到其四个P上的最近的邻居$_{l-1}$作为关键位置$\{k^o\}$。

$$\{k_{l-1}^o\} = \{(2x_l + o_q, 2y_l + j), \forall i, j \in \{0, 1\}\}。 \quad (1)$$

P_{l-1}上的所有$\{k^o\}$被收集起来，形成关键位置设置$\{k_{l-1}\}$。然后，三个头将只处理这些位置来检测物体并计算下一级的查询。具体来说，我们从P_{l-1}中提取特征，使用$\{k_{l-1}\}$作为指数来构建一个稀疏张量P_{l-1}，我们称之为价值特征。然后，稀疏卷积（spconv）[13]核子使用4-*conv*密集头的权重来计算ll层的结果。

为了最大限度地提高推理速度，我们应用在以级联的方式。特别是，对P的查询$_{l-2}$会只能从k_{l-1}。我们将这种范式命名为**级联稀疏查询（CSQ）**，如图1所示。我们的CSQ的好处是，我们可以避免从一个单一的P_l生成查询q_l，这将导致相应的关键位置k_l的大小随着l的减少而呈指数级增长。

3.3. 培训

我们保持分类和回归头的训练与原始RetinaNet[27]相同。对于查询头，我们使用FocalLoss[27]和生成的二元目标图来训练它。让P_l上的小物体o的真实边界盒为$b^o = (x^o, y^o, w^o, h^o)$。我们首先计算$P_l$上每个特征位置$(x, y)$与所有小的地面实况之间的最小距离图$D_l$。

$$D_l[x][y] = \min_o \{\sqrt{(x - x_l^o)^2 + (y - y_l^o)^2}\}, \quad (2)$$

那么，基础事实查询图V_l^*被定义为

$$V_l^*[x][y] = \begin{cases} 1 & \text{if } D_l[x][y] < s_l \\ 0 & \text{如果} D_l[x][y] \geq s_l, \text{则为} \end{cases}。 \quad (3)$$

对于每个级别P_l，损失函数定义如下

$$L_l(U_l, R_l, V_l) = L_{FL}(U_l, U_l^*) + L_r(R_l, R_l^*) + L_{FL}(V_l, V_l^*) \quad (4)$$

其中，U_l，R_l，V_l是分类输出，regressor out--放和查询得分输出，U_l，R_l，V_l是它们对应的地面实况图；L_{FL}是焦点损失，L_r是边界盒回归损失，它是平滑的l_1损失[11]的原始RetinaNet。总的损失是

$$L_{所有} = \beta_l * L_l. \quad (5)$$

这里我们通过β_l来重新平衡每一层的损失。其原因是，当我们加入像P_2这样的高分辨率特征时，训练样本的分布发生了明显的变化。P_2上的训练样本总数甚至大于跨P_3到P_7的训练样本总数。如果我们不降低它的权重，训练将被小物体所主导。因此，我们需要重新平衡不同层的损失，使模型同时从所有层学习。

3.4. 与相关工作的关系

请注意，尽管我们的方法与使用RPN的两阶段物体检测器有一些相似之处，但它们的区别在于

的问题有以下几个方面。1），我们在粗略预测中只计算分类结果，而RPN同时计算分类和回归。2）、RPN是在所有级别的全特征图上计算的，而我们的QueryDet的计算是稀疏的和选择性的。3）、两阶段方法依赖于RoIAlign[14]或RoIPooling[11]等操作，以使特征与第一阶段的提议相一致。尽管如此，在我们的方法中并没有使用它们，因为我们在粗略的预测中没有箱体输出。值得注意的是，我们提出的方法与基于FPN的RPN兼容，因此QueryDet可以被纳入两阶段检测器以加速提案的生成。

另一项密切相关的工作是PointRend[19]，它使用极少数自适应选择的点计算高分辨率的分割图。我们的QueryDet和PointRend的主要区别在于。1）查询是如何产生的；2）稀疏计算是如何应用的。对于第一个区别，PointRend根据每个位置的预测得分选择最不确定的区域，而我们直接添加一个辅助损失作为监督。我们的经验表明，这种简单的方法可以产生高召回率的预测并提高最终性能。至于第二种方法，PointRend使用多层感知器进行每像素分类。它只需要来自高分辨率特征图中的单个location的特征，因此可以很容易地进行分批，以获得高效率。另一方面，由于物体检测需要更多的上下文信息来进行准确的预处理，因此，PointRend采用的是多层感知器。

我们使用3×3核的稀疏卷积法。

4. 实验

我们在两个物体检测数据集上进行了定量实验。COCO[28]和VisDrone[59]。

COCO是最广泛使用的一般物体检测数据集；VisDrone是专门用于无人机拍摄的图像检测的数据集，其中小物体在规模分布中占主导地位。

4.1. 实施细节

我们在PyTorch[34]和Detectron2工具包[52]的基础上实现我们的方法。所有的模型都在8个NVIDIA 2080Ti GPU上训练。对于COCO，我们遵循常规的训练方法。我们采用Detectron2中的标准1调度和默认的数据增强功能。批量大小被设置为16，初始学习率为0.01。用于重新平衡不同层次的损失的权重β_l 被设置为在P_2到P_7之间从1到3线性增长。对于VisDrone，按照[30]，我们将一幅图像平均分成四个不重叠的斑块，并在训练期间独立处理它们。我们在初始学习率为0.01的情况下对网络进行了5万次迭代训练，并在3万和4万次迭代时将学习率递减10。重新平衡权重β_l被设置为从1到2.6线性增长。对于这两个数据集，我们在训练期间冻结了骨干网络中所有的批处理正常化（BN）层，并且我们做了

方法	CSQ	美联社	美联社$_{50}$	美联社$_{75}$	美联社$_S$	美联社$_M$	美联社$_L$	FPS
视网膜网络	-	37.46	56.90	39.94	22.64	41.48	48.04	13.60
视网膜网络 (3x)	-	38.76	58.27	41.24	22.89	42.53	50.02	13.83
查询数据		38.53	59.11	41.12	24.64	41.97	49.53	4.85
查询数据		38.36	58.78	40.99	24.33	41.97	49.53	14.88
询问Det (3x)		39.47	59.93	42.11	25.24	42.37	51.12	4.89
询问Det (3x)		39.34	59.69	41.98	24.91	42.38	51.12	15.94

表1.我们的QueryDet和基线RetinaNet在COCO *mini-val*集上的准确性（AP）和速度（FPS）的比较。

方法	CSQ	美联社	AP50	AP75	AR1	AR10	AR100	AR500	FPS
视网膜网络	-	26.21	44.90	27.10	0.52	5.35	34.63	37.21	2.63
查询数据		28.35	48.21	28.78	0.51	5.96	36.48	39.42	1.16
查询数据		28.32	48.14	28.75	0.51	5.96	36.45	39.35	2.75

表2.我们的QueryDet和基线RetinaNet在VisDrone验证集上的检测精度（AP）和速度（FPS）的比较。

在检测头中不增加BN层。在所有的实验中都使用了混合精度训练[32]以节省GPU内存。查询阈值σ被设置为0.15，我们从P_4开始查询。在没有特别说明的情况下，我们的方法是在具有ResNet-50主干的RetinaNet上构建的。

4.2. 我们方法的有效性

在表1中，我们比较了我们的方法和COCO上的基线RetinaNet之间的平均精度（mAP）和平均每秒一帧（FPS）。基准线以13.6 FPS运行，得到37.46的总体AP和22.64的AP$_S$，这比原始论文[27]中的结果略高。在高分辨率特征的帮助下，我们的方法获得了38.53的AP和22.64的AP。24.64个AP$_S$，将AP和AP$_S$，提高了1.1和2.0。这些结果显示了在检测小物体时使用高分辨率特征的重要性。然而，加入这样一个高分辨率的特征图，大大降低了推理速度，达到4.85FPS。当采用我们的级联稀疏查询（CSQ）时，推理速度提高到14.88FPS，甚至比基础查询还要快。

在不使用高分辨率的P_2，而性能损失可以忽略不计的情况下，线RetinaNet。此外，图2显示了我们的CSQ是如何节省计算成本的。

与使用高分辨率P_2的RetinaNet相比，其中P_3和P_2占总FLOPs的74%，我们的CSQ成功地将这些成本降低到1%左右。其原因是，在QueryDetall中，对高分辨率的P_3和P_2的计算是在稀疏分布的小物体周围的位置进行的。这些结果充分证明了我们方法的有效性。我们还在表1中×显示了3个训练计划的结果。更强的基线并没有削弱我们的改进，而是带来了更明显的加速。我们把

它归功于更强的查询头，因为小物体的估计变得更加准确。在VisDrone中，如表2所示，发现的情况是

人	篮	QH	CSQ	美联	美联	美联	美联	美联	美联	FPS
				37.46	56.90	39.94	22.64	41.48	48.04	13.60
				36.10	56.39	38.17	21.94	39.91	45.25	4.83
				37.66	57.57	40.37	22.03	41.86	49.10	13.60
				38.11	58.48	40.85	23.06	41.53	49.36	4.83
				38.53	59.11	41.12	24.64	41.97	49.53	4.85
				38.36	58.78	40.99	24.33	41.97	49.53	14.88

表3.对COCO*小值*集的消融研究。**HR**代表使用高分辨率特征；**RB**代表FPN层之间的损失再平衡；**QH**代表是否添加QueryHead，提供额外的客观性监督。

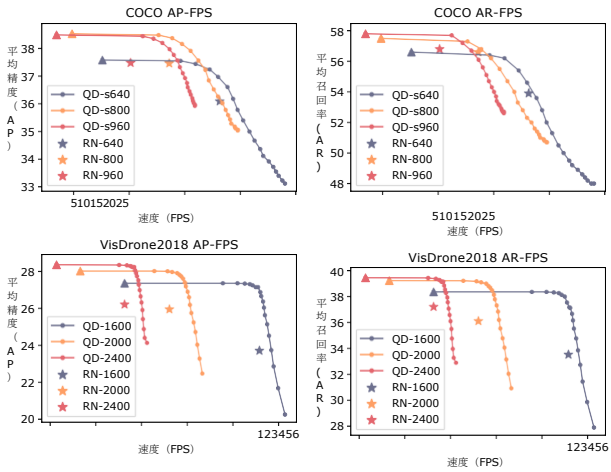

图4.在COCO和VisDrone上输入不同尺寸的图像时，速度和精度（AP和AR）的权衡。该权衡由查询阈值σ控制。每条曲线中最左边的标记（▲标记）代表没有应用级联稀疏查询时的结果。QD代表QueryDet，RN代表RetinaNet。

开始层	美联社	美联社$_{50}$	美联社$_{75}$	美联社$_S$	美联社$_M$	美联社$_L$	FPS
没有查询	38.53	59.11	41.12	24.64	41.97	49.53	4.86
P_6	37.91	57.98	40.51	23.18	42.02	49.53	13.42
P_5	38.22	58.55	40.86	23.65	42.00	49.53	13.92
P_4	38.36	58.78	40.99	24.33	41.97	49.53	14.88
P_3	38.45	58.94	41.07	24.50	41.93	49.52	11.51

表4.对我们的CSQ在MS-COCO*迷你瓦尔*集的最佳起始层的调查。

在这个小的面向对象的数据集上，我们将总体AP提高了2.1，AP提高了3.2。在这个面向对象的小数据集上，我们将总体AP提高了2.1，AP$_{50}$ 提高了3.2。推理速度从2.75 FPS的1.16 FPS提高到2.3倍。

4.3. 消融研究

我们在COCO*迷你瓦尔*集上进行消融研究，分析每个组件如何影响检测精度和速度，见表3。我们重新训练的RetinaNet实现了37.46 AP。当我们把高分辨率的P_2，AP的dra-。矩阵下降了1.34。正如我们在第3.3节中所讨论的，这个问题是由加入P_2后的训练样本的分布变化引起的。然后我们重新平衡这些层的损失。结果被提高到38.11，主要是为了解决这个问题。有趣的是，当采用原始基线时，重新平衡策略只给我们一个小的AP增强（0.2），这表明在高分辨率情况下，损失的重新平衡更为关键。然后，我们在网络中加入我们的查询头，通过它，我们得到了进一步的性能提升，即0.42的AP和0.2的AP。1.58个AP$_S$，将总的AP和AP$_S$推到38.53和24.64，验证了额外客观性超视觉的有效性。最后，使用CSQ，检测速度从4.85 FPS大幅提高到14.88 FPS，检测AP的0.17损失可以忽略不计。

4.4. 讨论

查询阈值的影响。在这里，我们研究了我们的级联稀疏查询中的准确性-速度权衡。我们测量了不同查询阈值σ下的检测精度（AP）和检测速度（FPS），其作用是确定输入图像中的网格（低分辨率特征位置）是否包含小物体。直观地讲，提高这个阈值会减少小物体的召回率，但由于考虑的位置较少，会加快推理的速度。图4显示了不同输入尺寸下的精度-速度权衡。我们在一条曲线中对相邻的数据标记依次增加σ，最左边的标记表示不应用CSQ时的性能。我们观察到，即使是一个非常低的阈值（0.05）也能给我们带来巨大的速度提升。这一观察验证了我们方法的有效性。另一个观察是关于不同输入分辨率的AP上界和下界之间的差距。这个差距对于大尺寸的图像来说是很小的，但是对于小尺寸的图像来说是很大的，这表明对于高分辨率的输入，即使查询阈值设置得很高，我们的CSQ也能保证一个好的AP下限。

哪一层开始查询？在我们的级联稀疏查询中，我们需要决定起始层，在该层之上，我们运行传统的卷积，以获得以下的检测结果

查询方法	美联社	美联社$_{50}$	美联社$_{75}$	美联社$_S$	美联社$_M$	美联社$_L$	FPS
没有查询	38.53	59.11	41.12	24.64	41.97	49.53	4.86
CQ	38.31	58.73	40.98	24.25	41.98	49.53	10.49
CCQ	38.32	58.75	40.98	24.26	41.98	49.53	8.73
CSQ (我们的)	38.36	58.78	40.99	24.33	41.97	49.53	14.88

表5.不同查询方法在COCO*小值*集上的比较。我们比较了我们的CSQ和Crop Query（CQ）以及Complete Convolution Query（CCQ）。

背景介绍	美联社	美联社$_{50}$	美联社$_{75}$	美联社$_S$	美联社$_M$	美联社$_L$	FPS
没有查询	38.53	59.11	41.12	24.64	41.97	49.53	4.86
1x1	38.25	58.60	40.87	23.88	41.97	49.53	14.09
3x3	38.30	58.66	40.94	24.14	41.97	49.53	14.06
5x5	38.36	58.72	40.98	24.18	41.97	49.53	14.00
7x7	38.37	58.73	40.98	24.30	41.97	49.53	13.77
9x9	38.37	58.73	40.98	24.30	41.97	49.53	13.42
11x11	38.38	58.755	40.99	24.33	41.97	49.53	13.11

表6.在MS-COCO*小数值*集上使用不同数量的上下文信息时的检测AP和速度比较。上下文被定义为查询位置周围不同大小的补丁。

骨干网	模型	CSQ	美联社	AP50	AP75	*APS*	*APM*	*APL*	FPS
移动网络V2	护理人员	-	26.72	43.17	28.17	15.27	29.28	34.51	17.75
	QD		29.16	46.20	30.95	16.14	31.26	38.66	5.31
	QD		28.94	45.79	30.71	15.74	31.26	38.66	21.66
ShuffleNet V2	护理人员	-	23.04	38.32	23.75	12.01	25.50	35.16	17.45
	QD		26.07	42.34	27.30	13.20	28.03	36.23	5.26
	QD		25.85	41.96	27.08	12.81	28.05	36.23	20.02

表7.不同骨干网的结果。**RN**和**QD**分别代表RetinaNet和QueryDet。

	CSQ	美联社	AP50	AP75	*APs*	*ÃÃÃ*	*ÃÃÃ*	FPS
基金会	-	38.37	57.63	41.03	22.34	41.95	48.96	17.06
查询Det (FCOS)		40.05	58.69	43.46	25.52	43.43	50.69	7.92
查询Det (FCOS)		39.49	57.97	42.82	24.81	43.45	50.69	14.40

表8.我们的QueryDet(FCOS)及其基线模型在COCO*小数值*集上的性能和速度。

大型物体。

我们不从最低分辨率层开始CSQ的原因有两个方面。1）普通的卷积操作对于低分辨率的特征来说是非常快的，因此CSQ所节省的时间不能弥补构建稀疏特征图所需的时间；2）在分辨率很低的特征图上很难区分小物体。结果列于表4。我们发现，获得最高推理速度的层是P_4，这验证了从非常高级的层（如P_5和P_6）查询会导致速度损失。我们观察到，随着起始层的升高，AP损失逐渐增加，这表明网络在非常低的分辨率层中寻找小物体的难度。

使用查询的最佳方式是什么？我们展示了我们的级联稀疏查询的高效率。我们提出

CSQ	美联社	美联社$_{50}$	美联社$_{75}$	美联社$_s$	美联社$_m$	美联社$_l$	FPS
	38.47	59.44	41.73	22.98	41.90	49.55	17.57
	38.20	58.88	41.50	22.23	41.91	49.55	19.03

表9.在Faster R-CNN中使用我们的CSQ的性能和速度在COCO*小数值*集上。

两种可供选择的查询操作进行比较。第一种裁剪查询（CQ），即从高分辨率特征中裁剪出查询所指示的相应区域用于后续计算。注意这种类型的查询类似于AutoFocus[33]的方法。另一种是完全卷积查询（CCQ），我们使用常规的卷积来计算每一层的完整特征图，但只从被查询的位置提取结果用于后处理。对于CQ，我们从特征图中裁剪出一个11 11的补丁，该补丁被选为适合检测头中5个3 3连续卷积的接受域。我们在表5中列出了结果。一般来说，这三种方法都能成功地加速推理，而AP损失可以忽略不计。其中，我们的CSQ可以达到最快的推理速度。

我们需要多大的背景？为了应用我们的CSQ，我们需要构建一个稀疏的特征图，其中只有小物体的位置被激活。我们还需要激活小物体周围的上下文区域，以避免降低准确性。然而，在实践中，我们发现过多的上下文不能提高检测AP，而只能降低检测速度；另一方面，太少的上下文会严重降低检测AP。在这一节中，我们将探讨我们需要多少上下文来平衡速度-准确度的权衡。在这里，上下文被定义为在被查询的位置周围有不同大小的补丁，我们的稀疏检测头也会处理补丁内的特征。结果在表6中报告。从中我们得出结论，一个5x5的补丁可以为我们带来足够的上下文来检测一个小物体。虽然更多的上下文带来了小的AP改进，但我们的CSQ的加速效果受到了负面影响，而更少的上下文不能保证高的检测AP。

轻量级骨干网的结果。正如我们在第1节中所说的，我们的方法可以与轻量级骨干网结合，以获得更多的速度改进。同时，由于我们的CSQ旨在加速检测头的计算，所以当使用这种骨干网时，整体加速更明显，因为骨干网的推理时间变得更少。我们在表7中报告了使用不同轻量级骨干网的结果。用MobileNet V2[40]进行高分辨率检测时，速度平均提高到4.1**倍**，而用Shuf-V2[40]则提高到3.8**倍**。

(a) COCO检测 (b) COCO查询热图 (c) VisDrone检测 (d) VisDrone查询热图

图5.我们的QueryDet在MS-COCO和VisDrone2018数据集上对小物体的检测结果和查询热图的可视化。我们删除了VisDrone2018的类标签，以更好地区分小边界框。

fleNet V2 [31]，这验证了我们的方法可以部署在边缘设备上，用于实时应用，如自动驾驶车辆的有效小物体检测。

无锚检测器上的结果。QueryDet可以应用于任何基于FPN的检测器，以加速高分辨率检测。因此，我们将QueryDet应用于FCOS，一个最先进的无锚检测器，并在表8中报告COCO的结果。可以得出结论，QueryDet在高分辨率特征的帮助下证明了AP，当采用级联稀疏查询（CSQ）时，高分辨率速度平均提高了1.8，验证了所提方法的通用性。 ×

对两阶段检测器的有效性
我们的CSQ也可以应用于基于FPN的两阶段检测器，以减少RPN中高分辨率层的computaion成本。为了验证这一说法，我们将CSQ应用于Faster R-CNN检测器[39]。在我们的实现中，RPN的输入是从P_2到P_6，我们从P_4开始查询。我们修改了RPN的结构，让它有3个*说服层*，而不是普通实现中的1个层，接下来是3个分支，分别是客观性分类、边界盒回归和查询密钥计算。前两个分支是按照通常的做法[39]训练的，而查询分支是通过Focal Loss训练的，$\gamma = 1.2$，$\alpha = 0.25$。在推理过程中，我们将查询线程hold设置为0.15。如表9所示，我们的Faster R-CNN实现了38.47的总体AP和22.98的AP_S，17.57的FPS。当利用CSQ时，推理速度提高到19.03 FPS，但AP的损失很小。这些结果验证了我们的方法在加速两阶段检测器方面的有效性。

请注意，在两阶段检测中，我们的CSQ不仅可以节省RPN中密集计算的时间，而且可以减少进入第二阶段的RoI的数量。

4.5. 可视化和失败案例

在图5中，我们可视化了COCO和VisDrone上小物体的检测结果和查询热图。从热图中可以看出，我们的查询头可以成功地找到小物体的粗略位置，使我们的CSQ能够有效地检测它们。此外，通过纳入高分辨率的特征，我们的方法可以非常准确地检测出小物体。

我们还展示了我们方法的两个典型失败案例。1）即使查询头正确地提取了小物体的珊瑚位置，检测头也可能无法定位它们（VisDrone的第二张图片）；2）大物体的位置被错误地激活，导致检测头处理无用的位置，从而降低了速度（COCO的第一张图片）。

5. 总结

我们提出了QueryDet，它使用一种新的查询机制Cascade Sparse Query（CSQ）来加速基于特征金字塔的密集物体检测器的inference。QueryDet使物体检测器能够以较低的成本检测小的物体，并且容易部署，使其在实时应用（如自动驾驶）上的部署变得实用。在未来的工作中，我们计划将QueryDet扩展到更具挑战性的三维物体检测任务，该任务以LiDAR点云为输入，其中三维空间通常比二维图像更稀疏，计算资源对昂贵的三维卷积操作来说更为紧张。

参考文献

[1] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos.一个统一的多尺度深度卷积神经网络用于快速物体检测。In *ECCV*.Springer, 2016.2

[2] 蔡兆伟和Nuno Vasconcelos.Cascade r-cnn:深入研究，实现高质量的物体检测。In *CVPR*, 2018.2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko.结束语
用变压器进行末端物体检测。*ECCV*, 2020.1

[4] Chennyi Chen, Ming-Yu Liu, Oncel Tuzel, and Jianxiong Xiao.用于小物体检测的R-CNN。在*ACCV*中。Springer, 2016.2

[5] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan
Chandraker.通过知识提炼学习高效的物体检测模式
els。In *NeurIPS*, 2017.2

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille.Deeplab:用深度卷积网、无序卷积和完全连接的crfs进行语义图像分割。*IEEE transactions on pattern analysis and machine intelligence*, 40（4）, 2017.2

[7] 段开文，白松，谢灵溪，齐洪刚，黄庆明，田琦.Centernet: 物体检测的关键点三联体。In *ICCV*, 2019.2

[8] Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. 残差网络的空间自适应计算时间。In *CVPR*, 2017.3

[9] Mikhail Figurnov, Aizhan Ibraimova, Dmitry P Vetrov, and Pushmeet
Kohli.Perforatedcnns:通过消除冗余卷积的加速， 。In *NeurIPS*, 2016.2

[10] 傅成洋，刘伟，Ananth Ranga，Ambrish Tyagi，和Alexander C Berg。DSSD:去卷积单次拍摄检测器。*arXiv预印本arXiv:1701.06659*, 2017。2

[11] Ross Girshick.Fast r-cnn. In *ICCV*, 2015.2, 4, 5

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik.丰富的特征层次用于准确的物体检测
和语义分割。在*CVPR*, 2014.1, 2

[13] Benjamin Graham and Laurens van der Maaten.亚流形稀疏卷积网络。*arXiv预印本 arXiv:1706.01307*, 2017。2, 4

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick.Mask r-cnn.In *ICCV*, 2017.2, 5

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.用于图像识别的深度残差学习。In *CVPR*, 2016.1

[16] Yihui He, Xiangyu Zhang, and Jian Sun.加速超深度神经网络的通道修剪。In *ICCV*, 2017.2

[17] 黄立超，杨毅，邓亚峰，于英楠.DenseBox:统一地标定位与端到端物体
检测。*arXiv预印本arXiv:1509.04874*，2015。2

[18] Eric Jang, Shixiang Gu, and Ben Poole.Categorical reparameterization with gumbel-softmax.*ICLR*, 2017.3

[19] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick.Pointrend:作为渲染的图像分割。In *CVPR*, 2020.5

[20] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho.Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*, 2019.2

[21] 孔涛，孙福春，刘华平，姜玉宁，李磊，石建波。FoveaBox:基于Beyound锚点的物体检测。*IEEE Transactions on Image Processing*, 29, 2020.2

[22] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun.Hypernet:迈向准确的区域建议生成和
联合物体检测。In *CVPR*, 2016.2

[23] 罗曦和邓佳。CornerNet:检测对象为 成对的关键点。In *ECCV*, 2018.2

[24] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan.感知生成对抗性网络的小物体检测。In *CVPR*, 2017.2

[25] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang.用于物体检测的规模感知三叉戟网络。
In *ICCV*, 2019.1, 2

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie.用于物体检测的特征金字塔 网络。在*CVPR*, 2017.1, 2

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár.密集物体检测的焦点损失。在*ICCV*, 2017。1, 2, 3, 4, 5

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick.Microsoft coco:上下文中的通用对象。在*ECCV*。Springer, 2014.1, 2, 5

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg.Ssd：单镜头多箱检测器。在*ECCV*中。
Springer, 2016.1, 2

[30] 刘子明，高光宇，孙琳，和方志远.HRDNet:小型目标的高分辨率检测网络 jects. *arXiv preprint arXiv:2006.07607*, 2020.5

[31] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun.ShuffleNet V2:高效CNN Architecture Design的实用指南。In *ECCV*, 2018.8

[32] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training.*ICLR*, 2018.5

[33] Mahyar Najibi, Bharat Singh, and Larry S Davis.自动对焦。高效的多尺度推理。In *ICCV*, 2019.3, 7

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch。一个命令式的高性能深度学习库。在
NeurIPS，2019年。5

[35] 约瑟夫-雷德蒙、桑托什-迪瓦拉、罗斯-吉希克和阿里-法哈迪。你只看一次。统一、实时的对象检测。In *CVPR*, 2016.2

[36] 约瑟夫-雷德蒙和阿里-
法哈迪。YOLO9000：更好，更快， 强。In *CVPR*,

2017.2

[37] Joseph Redmon 和 Ali Farhadi.Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.1, 2

[38] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun.SBNet:用于快速推理的稀疏块状网络。在 *CVPR*，2018年6月。3

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun.Faster r-cnn: Towards real-time object detection with region proposal networks.In *NeurIPS*, 2015.1, 2, 8

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zh- moginov, and Liang-Chieh Chen.Mobilenetv2:倒置的残差和线性瓶颈。In *CVPR*, 2018.7

[41] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta.Beyond skip connections:*arXiv preprint arXiv:1612.06851*, 2016.2

[42] Bharat Singh and Larry S Davis.An analysis of scale invariance in object detection snip.In *CVPR*, 2018.2

[43] 巴拉特-辛格，马海尔-纳吉比，和拉里-S-戴维斯。狙击手。高效的多尺度训练。In *NeurIPS*, 2018.2

[44] Mingxing Tan, Ruoming Pang, and Quoc V Le.EfficientDet:可扩展和高效的物体检测。在*CVPR*, 2020.2

[45] 田智，沈春华，陈浩，何彤.Fcos:完全卷积的单阶段物体检测。In *ICCV*, 2019.2

[46] Burak Uzkent, Christopher Yeh, and Stefano Ermon.使用深度强化学习（ment）在大型图像中进行高效的物体检测。In *WACV*, 2020.3

[47] Thomas Verelst and Tinne Tuytelaars.Dynamic Convolutions:利用空间稀疏性实现更快的推理。在 *CVPR*，2020年6月。2

[48] 王敬东，孙科，程天恒，姜波瑞，邓超瑞，赵阳，刘东，穆亚东，谭明奎，王兴刚，等.视觉识别的深度高分辨率代表学习.*IEEE transactions on pattern analysis and machine intelligence*, 2020.2

[49] 王新江，张世龙，于卓然，冯立彤，和张伟.用于物体检测的尺度均衡金字塔卷积 。In *CVPR*, 2020.1

[50] Yulin Wang, Kangchen Lv, Rui Huang, Shiji Song, Le Yang, and Gao Huang.Glance and Focus: a Dynamic Approach to Reducing Spatial Redundancy in Image Classification. *NeurIPS*, 2020.3

[51] Yi Wei, Xinyu Pan, Hongwei Qin, Wanli Ouyang, and Junjie Yan.量化模仿。Towards very tiny cnn for object detection.In *ECCV*, 2018.2

[52] 吴玉新，Alexander Kirillov，Francisco Massa，Wan-Yen Lo，和Ross Girshick。Detectron2. https://github.com/facebookresearch/detectron2, 2019.5

[53] Saining Xie, Ross Girshick, Piotr Dar, Zhuowen Tu, and Kaiming He.深度 神经网络的聚合残差转换。在*CVPR*, 2017.1

[54] Zhenda Xie, Zheng Zhang, Xizhou Zhu, Gao Huang, and Stephen Lin.带有随机特征采样和插值的空间自适应推理。*ECCV*, 2020.2

[55] Yan Yan, Yuxing Mao, and Bo Li.第二：疏散嵌入-ded卷积检测。*Sensors*, 18(10), 2018.2

[56] 杨泽、刘绍辉、胡瀚、王立伟、林志强。Reppoints:用于物体检测的点集表示。In *ICCV*, 2019.2

[57] Fisher Yu和Vladlen Koltun.*arXiv预印本 arXiv:1511.07122*,
2015年，通过扩张的卷积进行多尺度上下文聚合。2

[58] 张世峰，池成，姚永强，雷震，和Stan Z Li.通过自适应训练样本选择弥合基于锚点和无锚点检测之间的差距。
在*CVPR*, 2020.2

[59] 朱鹏飞，文龙吟，杜大伟，卞晓，凌海斌，胡清华，聂勤勤，程浩，刘晨峰，刘晓宇，等. Visdrone-det2018:视觉满足无人机图像中的物体检测挑战结果。
In *ECCV*, 2018.2, 5

[60] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le.Learning data augmentation strategies for object detection. *arXiv preprint arXiv:1906.11172*, 2019.2