DPNet：融合轻量注意力的双路径实时目标检测网络

摘要—近年来，高精度卷积神经网络(CNN)压缩技术的进步显著推动了实时目标检测的发展。为了加快检测速度，轻量级检测器通常使用卷积层较少的单路径主干网络。然而，单路径架构涉及连续的池化和下采样操作，往往导致特征图过于粗糙且不精确，不利于目标定位。此外，受限于网络容量，现有的轻量级网络对大规模视觉数据的表征能力普遍较弱。为了解决上述问题，本文提出了一种双路径网络 DPNet，该网络集成了轻量注意力机制，用于实时目标检测。双路径的架构能够并行提取高层语义特征和低层目标细节。尽管 DPNet 的结构复杂度近乎是单路径检测器的两倍，但计算开销和模型大小并未显著增加。为了增强网络的表征能力，我们设计了一个轻量级自相关模块(LSCM)用于捕获全局交互关系，仅增加了少量计算开销和模型参数。在颈部网络，LSCM 进一步被扩展为轻量级互相关模块(LCCM)，用以捕获相邻尺度特征之间的相互依赖关系。我们在 MS COCO，Pascal VOC 2007 和 ImageNet 数据集上进行了大量实验。实验结果表明，DPNet 在检测准确率和运行效率之间达到了当前最优的平衡。具体而言， DPNet 在 MS COCO 测试开发集上取得了 31.3%的 AP，在 Pascal VOC 2007 测试集上达到了 82.7%的 mAP，在 ImageNet 的验证集上获得了 41.6%的 mAP，同时模型大小仅为 2.5 M、计算量为 1.04 GFLOPS，对于三个数据集中分辨率为 320×320 的输入图片，推理速度分别为 164 FPS 和 196 FPS。

索引术语—卷积神经网络(CNN)，双路径架构主干网络，轻量注意力，目标检测。

I. 引言

目标检测是计算机视觉领域中一项基础但极具挑战性的任务。它的目标是在输入图像中定位覆盖目标物体的最小边界框，并同时分配对应的语义标签。通常来说，基于卷积神经网络(CNNs)的最新方法可以大致分为双阶段[1],[2]检测器和单阶段[3],[4],[5],[6]检测器。双阶段检测器首先使用区域建议网络生成候选框，随后这些框在下一阶段进行细化。但由于多阶段的设计，其效率通常不高。相比之下，单阶段检测器[3],[4],[5],[6]直接在卷积特征图上预测目标类别并对边界框进行回归预测。得益于整个流程的简化，单阶段目标检测器[3],[4],[5],[6]总是能实现比双阶段检测检测器[1],[2]更快的推理速度。尽管 CNN 在目标检测上取得了显著进步，但绝大多数基于 CNN 的检测器涉及到数百甚至数千个卷积层和特征通道[7],[8]，这使模型大小和计算效率难以满足需要在线评估与实时预测的真实应用场景，如自动驾驶、机器人视觉和虚拟现实等。

为了适应真实世界的场景，大量轻量级网络[11],[12],[13]已被提出用于实时目标检测任务。这些轻量级网络源自用于图像分类的[10],[11],[12]，它们更倾向于在主干网络中直接继承单路径架构并使用轻量化卷积。例如，MobileNet-SSD[11],[12]将 MobileNet 和 SSD 检测头结合。ThunerNet[13]使用 ShuffleNetV2[10]作为主干网络，将 3×3 深度卷积替换为 5×5 深度卷积。Pelee[14]采用密集结构的轻量主干网络，并减小 SSD 检测头的输出尺度以降低计算量。Tiny-DSOD 在主干网络和特征金字塔网络(FPN)中引入了深度卷积。Tiny-YOLO 系列[5],[16],[17]则减少了卷积层的数量或去除了颈部的多尺度输出来实现轻量化。尽管这些先进有效的网络已经取得了很好的检测结果，但它们本质上仍存在以下的局限性：

1) 单路径架构采用激进的下采样策略(如池化和跨步卷积)，长期占据着实时目标检测主干网络设计的主流[12],[17],[18]。然而，随着细粒度特征从浅层到深层卷积逐步丢失，生成的高层特征将阻碍目标精准定位。图 1 给出了两组可视化案例。第一行显示 ShuffleNetV2[10]倾向于从输入图像的周边区域提取特征。尽管轻量级的检测器效仿高精度 CNN 引入 FPN 来缓解这个问题[15]，但通过逐元素相加或拼接方式简单融合浅层至深层的失真特征，反而可能对目标检测不利[19]。
2) 受限于网络容量，现有的轻量级检测器对视觉数据的表征能力可能较弱[20]。如图 1 的第二

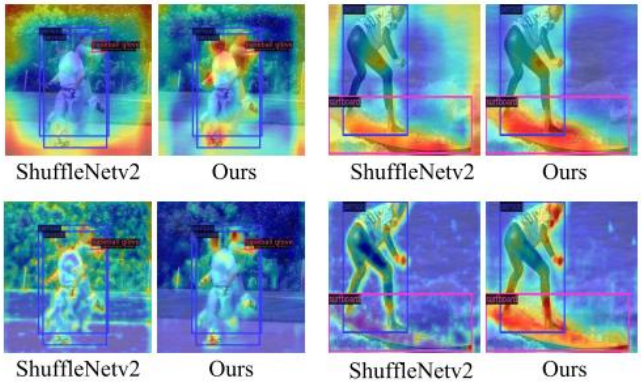

图 1 基于 ShuffleNetV2 和 DPNet 在 MS COCO 验证集上的特征热力图对比(红色表示高响应，蓝色表示低激励；为了清晰起见，检测目标上还叠加了边界框和对应标签；两行分别展示了来自低分辨率和高分辨率路径的特征，与 ShuffleNetV2 相比，DPNet 的热力图更加准确，大多数更高响应的像素在目标区域内，彩色视图最佳)

行所示，高的滤波器响应有时弥散在杂乱的背景中(如树木和海洋)，而包含感兴趣目标的区域却较少被激活。根本原因在于，轻量化卷积感受野有限，难以对全局依赖关系编码[20]。一些网络更倾向于使用大卷积核(如 31×31)[21],[22]，或者自注意力机制[23]；然而，它们涉及的巨额计算开销和模型大小不适合实时目标检测。因此，如何在有限的计算资源下增强特征表达能力仍是轻量目标检测领域的尚未解决的问题。

为了克服上述的缺陷，本文提出了一种双路径网络——DPNet，融合了轻量的注意力设计以实现实时目标检测。如图 2 所示，DPNet 由三个组件构成：主干网络、颈部网络和检测头。为了解决目标细节的丢失问题，与先前的轻量检测网络[10],[11],[12]普遍采用的单路径结构不同，DPNet 采用了并行路径架构，由此构建双分辨率主干网络。具体而言，低分辨率路径(LRP)的分辨率通常逐渐降低，同时对高层次的语义信息编码。相反，高分辨率路径(HRP)的分辨率保持不变，用于提取低层次的空间信息。这两条路径对轻量级的目标检测都至关重要。考虑到这两个子网络的互补特性，我们构建了一个双向特征融合模块(Bi-FM)，用以增强两条路径间的交流，促进不同分辨率特征的信息流通。尽管 DPNet 主干网络的结构复杂度近乎是单路径架构[10],[11],[12]的两倍，但其计算复杂度和网络大小并未显著增加。

为了提高表征能力和计算效率，我们开发了具有轻量级自相关模块(LSCM)的 ShuffleNetV2 单元[10]，该模块模仿了自注意力机制[23],[24]，由此构建出基于注意力的 shuffle 单元(ASU)。LSCM 分为两个步骤进行：注意力计算和特征重加权。与自注意力机制类似，第一步通过计算逐元素相似度来生成注意力图。然而，LSCM 与需要大量计算来探索稠密的像素到像素/通道到通道依赖关系[23],[24]不同，它更加轻量且计算成本更低，因为它在低维嵌入空间中研究稀疏的像素到区域/通道到组通道的互相关联，从而使得计算成本随着输入分辨率呈线性变化。在第二步中，LSCM 采用逐元素重加权的方式进一步降低了计算成本，其中计算得到的注意力图直接与展平后的特征进行相乘，避免了自注意力机制中广泛使用的复杂矩阵乘法。如图 2 所示，为了充分利用颈部网络中不同分辨率的特征，LSCM 被进一步扩展为一个轻量的互相关模块(LCCM)。LCCM 以双向方式工作：自上而下(LCCM-TD)和自下而上(LLCM-BU)。LLCM-TD 引入高层次语义信息以引导低层次特征；相反，LCCM-BU 利用低层次细节来优化高层次特征。如图 1 所示，由于 DPNet 继承了双路径主干网络和轻量注意力设计的优点，目标物体区域(例如人头、手和脚)内的像素无论在高分辨率还是低分辨率特征图中都能被正确激活。简而言之，本文的主要贡献体现在以下三个方面：

1) 与主流轻量检测器采用单路径主干网络的设计不同，DPNet 采用了一种双路径架构，能够同时提取高层语义信息并保留低层细节。不同于检测网络在 FPN(即颈部网络)中融合多分辨率特征的方式，我们的 DPNet 在主干网络中设计了双分辨率路径。据我们所知，这种设计在近期的网络中尚属少见。此外，得益于其简洁高效的双路径架构，DPNet 能够在主干网络中实现传统单路径目标检测器无法完成的特征交互。
2) 我们设计了一种基于注意力机制的轻量化模块 LSCM，该模块兼具高效的实现性能与强大的表征能力。由于其计算复杂度与输入特征的分辨率呈线性关系，LSCM 的计算开销极低。+即便如此，通过捕获全局空间和通道间的交互，LSCM 仍然展现出了强大的特征表达能力。此

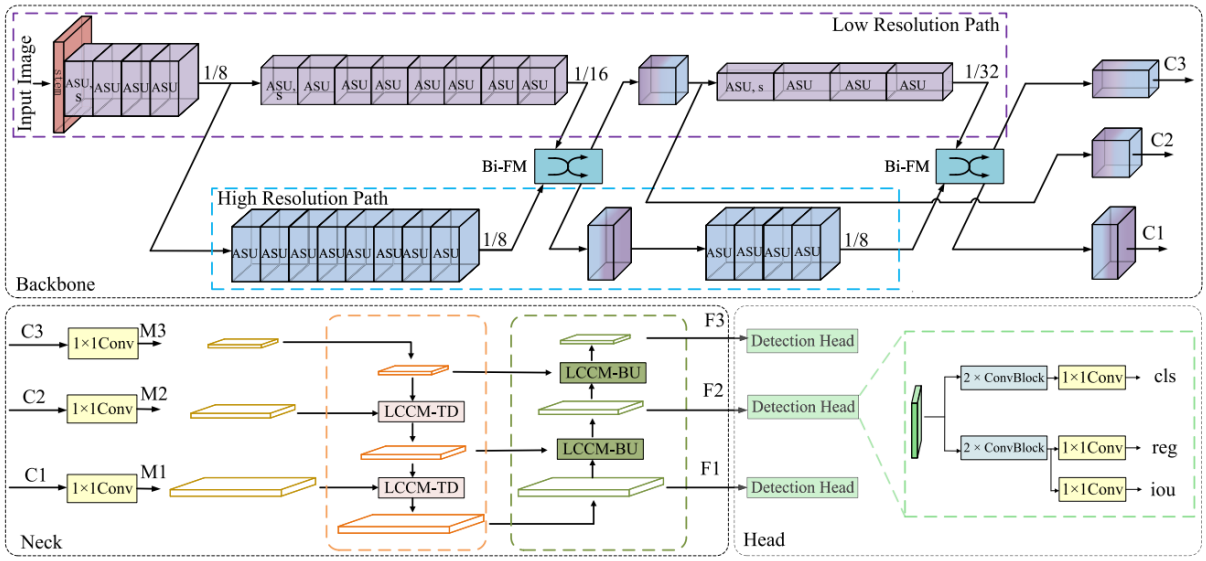

图 2 DPNet 的整体架构(主干网络由一系列 ASU 单元，以及一个干层和两个 Bi-FM 组成；同时具有双路径架构：蓝色和紫色虚线框表示的 HRP 和 LRP；在颈部，LCCM 以双向机制运行，分别用橙色和绿色虚线框表示，用以增强跨尺度相互作用；检测头使用几个轻量的卷积块来进行最后的预测；彩色视图最佳)

外，我们将 LSCM 扩展为应用于颈部网络的 LCCM，通过该模块充分挖掘了不同分辨率的邻近尺度特征之间的相互依赖关系。

3) 我们在三个具有挑战性的数据集上评估了 DPNet 的性能：MS COCO[9]，Pascal VOC 2007[25]和 ImageNet ILSVRC 2017[26]。大量实验表明，我们的方法在检测精度与实现效率间达到了最优的平衡。具体而言，DPNet 在 MS COCO 测试集上取得了 31.3%的 AP，在 Pascal VOC 2017 测试集上取得了 82.7%的 mAP，在 ImageNet 验证集上取得了 41.6%的 mAP。同时 DPNet 的模型大小仅为 2.5 M，计算量为 1.04 GFLOPs，针对三个数据集中 320×320 分辨率的输入图像，推理速度分别为 164 FPS 和 196 FPS。

本文的其余部分组织如下：在第 II 节中简要介绍相关工作后，我们在第 III 节中会详细阐述 DPNet 的实现细节；实验结果在第 IV 节中展示；第 V 节为总结评述并展望未来研究方向。

II. 相关工作

为了适应实时应用的需求，大量研究致力于目标检测器的压缩，主要方法包括量化[27]、剪枝[28]、知识蒸馏[29]以及轻量化模型设计[13],[14],[30]。由于本文方法属于最后一类，我们将简要回顾该方向的相关工作。

A.轻量级设计的实时目标检测

尽管许多最先进的单阶段检测器[3],[4],[5],[6]已经达到了实时推理的速度，但其模型大小与计算成本对于实时应用来说仍然难以接受。为了缓解这些局限，近年来轻量化检测器的设计受到了广泛关注[13],[14],[15],[30]。通常来说，实时目标检测器可大致分为两类：基于 CNN[13],[15],[30]与基于 Transformer[20],[31]的轻量化网络。

第一类网络通常采用紧凑型算子，如深度卷积[11]，分组卷积[12]和分解卷积[32]，来构建其主干网络。例如，MoblieNet-SSD[12]将 MobileNet[11]与 SSD 检测头[4]结合，取得了令人满意的检测结果；ThunderNet[13]采用 ShuffleNetV2[10]作为主干网络并设计了一个空间注意力模块以捕获全局上下文信息；Tiny-DSOD[15]提出了一种高效的深度密集块来替代 DenseNet[33]中的原始块；Pelee[14]则引入了 DenseNet[33]的一种高效变体以实现实时预测。作为最流行且先进的单阶段检测器，YOLO 系列[5],[16]通常被压缩为轻量级版本[17]，通过减少卷积层来压缩模型体积；而 MobileDets[3

0]作为另一种轻量化神经架构搜索检测器，则在各种移动平台上实现了更优的延迟和兼容性。此外，一些方法通过在轻量级主干网络中设计即插即用模块来实现高效的目标检测，例如 Jin 等人[34]在广泛使用的残差块中提出了一种增强的编码器-解码器路径，DAC[35]则通过在卷积核内部学习注意力机制来加速推理。另一些替代方案则采用联合学习策略，利用额外的学习任务(如多目标跟踪与分割[36]，以及水下场景中的目标检测与色彩转换[37])来进行实时目标检测。

另一方面，Transformer 模型[20],[31],[38]在近年来开始展示其在目标检测中的潜力。由于 Transformer 基于计算量庞大的自注意力机制[39]，研究者通过设计轻量化的 CNN-Transformer 混合架构[20],[31]来压缩模型体积。例如 MobileViT[20]在保留单路径架构的基础上，将 Transformer 嵌入到反向瓶颈模块[12]中以实现实时的目标检测。Lite Transformer[40]采用长短程注意力机制来加速计算，通过局部卷积捕获局部特征，同时利用自注意力提取全局信息；MobileFormer[31]则提出轻量级混合网络架构，其中 CNN 分支负责提取局部特征，而 Transformer 分支则探索全局信息。同时局部特征和全局信息相互作用，提升了模型性能。作为扩展至万亿参数的开创性模型， Switch Transformer[41]用稀疏 FNN 代替了传统的稠密 FNN。除了推理速度之外，FlashAttention[42]在设计高效的 Transformer 时还充分考虑了内存约束。

与上述大多数采用单路径设计的方法不同，我们提出的 DPNet 采用双路径架构。LRP 提取高层的丰富语义，而 HRP 则提取低层的准确细节，这两者对实时目标检测都至关重要。尽管双路径架构已经在轻量级语义分割[43],[44]中被探索过，据我们所知，只有 Mobile-Former[31]采用了双路径架构。然而它仍存在模型体积庞大，并且两条路径的分辨率按常规设计逐渐降低的问题。相比之下，DPNet 作为一个轻量检测器，在整个 HRP 路径上的分辨率都保持恒定。

B. 视觉注意力机制

由于具备强大的全局上下文捕捉能力，视觉注意力机制[39],[45],[46],[47]已经被广泛应用于 CNN 中来推动目标检测的发展。这些网络大致可以分为两类：压缩注意力[45],[48],[49]和自注意力[23],[38],[39]。

第一类方法通过网络学习突出重要的特征通道和位置，即通道注意力和空间注意力。例如，SENet[45]利用池化操作来对全局上下文编码；GSNet[47]探索二阶全局池化来实现特征通道的重加权。除了通道注意力外，部分网络[49],[50]主张学习捕获全局位置信息的空间注意力。例如 CBAM[49]分别在通道与空间维度采用平均池化和最大池化；ECANet[46]用一维卷积层替换了全连接层。尽管通过全局池化捕获上下文信息在计算上较为高效，但它在表征元素间的交互关系方面仍存在不足。

第二类方法通过计算每个图像元素间的相关矩阵来捕获全局上下文信息。作为这类方法的开创者，非局部网络[23]对像素与像素间的关系进行建模，其中每个位置的权重都由其他位置重加权得到；为了降低计算开销，CCNet[51]提出两个连续的交叉注意力来代替密集注意力图；然而 GCNet[48]认为仅通过通道的重加权即可捕获全局上下文；ANNet[52]在非局部网络中引入了空间金字塔池化来加快推理速度。尽管这些先进的网络具备强大的全局上下文捕获能力，其仍存在计算负担过重的问题。

和这些方法不同，我们提出的 DPNet 采用 LSCM 来兼顾表征能力和计算效率。在捕捉全局上下文线索层面，LSCM 同样效仿自注意力机制探索空间与通道间的交互关系，但通过稀疏的像素-区域/通道-通道组互相关机制而非自注意力中广泛使用的密集的像素-像素/通道-通道依赖关系显著降低了计算成本。

C. 多尺度特征融合

直接将卷积特征输入检测头会导致性能较差[11],[12],[14]，因此多尺度卷积特征融合在实时目标检测中至关重要[5],[15],[16]。借鉴了高精度检测器[53],[54]，早期的研究[13],[15],[55]尝试使用 FPN 以自上而下的方式进行特征融合。具体来说，首先对低分辨率特征进行上采样，再通过逐元素相加或拼接与高分辨率特征融合。为了节省计算开销，ThunderNet[13]和 Tiny-YOLOV4[17]都使用小型的 FPN，从而减少了输出的数量。LightDet[55]将 FPN

中的 3×3 标准卷积替换为紧凑型深度可分离卷积。与这种自上而下的方式不同，EfficientDet[56]采用了额外的自下而上的策略，将高分辨率特征下采样后与低分辨率特征进行融合。

与通过简单的加法或堆叠方式融合多尺度特征图不同， LCCM 在颈部网络中专门编码邻近尺度特征间的关联依赖关系。尽管采用了类似于 EfficientDet[56]的双向融合策略，LCCM 凭借它的轻量化设计仅需要很小的计算开销。

III. 我们的方法

在本节中，我们首先描述 DPNet 的整个轻量级双分辨率架构，然后详细解析主干网络中的 LSCM 模块和颈部网络中的 LCSM 模块的细节。

A. DPNet

DPNet 的整体架构如图 2 所示。具体而言，它由三个部分组成：主干网络、颈部网络和检测头。下面我们将详细介绍每个部分。

1) 主干网络：DPNet 主干网络的详细结构见表 I。具体而言，DPNet 采用了双分辨率主干结构，形成了并行路径架构：LRP 和 HRP。两条路径主要由一系列的 ASU 单元组成。与传统的单路径检测器[13],[14],[15]相似，LRP 采用了一个干层和多个步长为 2 的 ASU 模块，逐步生成分辨率分别为输入图像的(1/2)、(1/4)、(1/8)、(1/16)及(1/32)的卷积特征图。注意到干层包含一个 3×3 的跨步卷积层和一个最大池化层，直接将输入图像的分辨率缩小了四倍。而为了获得高质量的目标细节，HRP 则保持相对于 LRP 更高的分辨率，其特征图分辨率始终保持在输入图像大小的(1/8)。在两条路径间两个 Bi-FM 模块用于强化跨分辨率融合与交互。最终如图 2 所示，整合生成的特征图{$C1$，$C2$，$C3$}，它们的尺寸分别为{40×40×128、20×20×256、10×10×512}。然后它们作为多尺度输入送入颈部网络，用于帮助挖掘特征之间的互相关关系。接下来，我们将分别介绍 ASU 和 Bi-FM 的细节。

a) ASU：如图 3(a)所示，ASU 采用分离-变换-合并结构，兼具残差连接和轻量化特征提取的作用。在每个 ASU 的初始阶段，输入特征首先被分为两个低维部分，分别是变换分支和恒等分支，各占输入通道数的一半。变换分支充当残差函数，而恒等分支用于优化模型训练。摒弃了使用 3×3 的深度卷积[10]，变换分支依次使用更大核的深度卷积(如 5×5)和 LCSM 模块来提取更显著的特征。随后，通过拼接操作合并两分支的输出，确保通道数与输入的一致。最终执行通道混洗操作，实现两个分支间的信息交互。混洗完成后就开始下一个 ASU 单元的处理。图 3(b)展示了 ASU 的另一个跨步版本，用于减小特征图的分辨率，其变换分支和恒等分支分别采用 5×5 的跨步深度卷积。

b) Bi-FM：Bi-FM 模块在主干网络中充当 HRP 和 LRP 之间的通信接口，其详细结构如图 3(c)所示。$F_i^h \in \mathbb{R}^{H\times W\times C}$ 和 $F_i^l \in \mathbb{R}^{(H/m)\times(W/m)\times mC}$ 作为 Bi-FM 的输入，其中 $m \in \{2,4\}$，而 $F_o^h \in \mathbb{R}^{H\times W\times C}$ 和 $F_o^l \in \mathbb{R}^{(H/m)\times(W/m)\times mC}$ 作为输出，$H\times W$ 代表输入的分辨率，C 表示通道数量。具体来说，F_i^l 首先通过 1×1 的卷积处理，然后以相同的维度进行下采样，用于和 F_i^h 融合。在另一边，为了得到 F_o^l，F_i^h 被送入到 5×5 的跨步深度卷积中后，再以相同的维度进行下采样，用于接下来与 F_i^l 的特征融合。实际上，HRP 和 LRP 之间的相互作用可以被视为跨分辨率的残差函数，这有助于以端到端的方式训练 Bi-FM。

2) 颈部网络：检测颈部，也称为 FPN[53]，是当前最先进检测器中用于多尺度特征融合的一个基本组件。之前的方法[53],[54]使用了一种简单的融合策略，即通过双线性插值或逐元素加法，通常忽略了不同分辨率特征间的相互依赖关系。为此，LCCM 被引入到 DPNet 的颈部，用于融合来自不同卷积层的跨分辨率特征。

颈部网络的详细架构如图 2 所示。需要注意的是，LCCM 以双向机制工作：自上而下(LCCM-TD)和自下而上(LCCM-BU)方向。LCCM-TD 用于提取高层语义信息以优化类别识别，而 LCCM-BU 则侧重强化底层细节特征来提升目标定位精度。具体而言，颈部网络以主干网络生成的特征图{$C1$，$C2$，$C3$}作为输入，在首阶段通过一系列 1×1 的卷积操作，生成具有相同通道数但分辨率不同的特征图。这些中间特征，记作{$M1$，$M2$，$M3$}，首先通过两个 LCCM-TD 进行自上而下的融合，随后通过两个 LCCM-BU 进行自

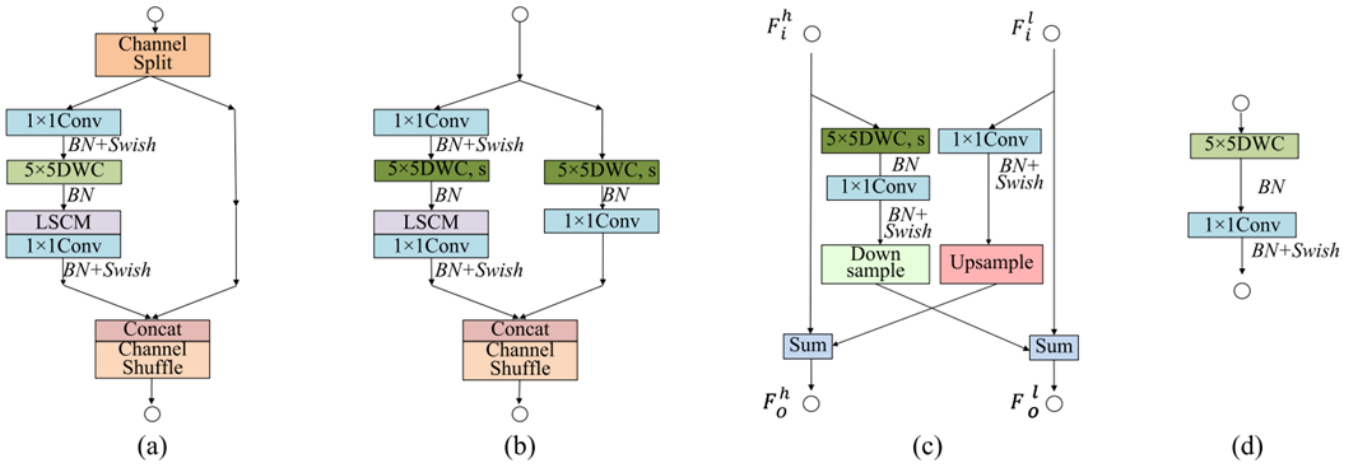

图 3 在主干网络和检测头中使用的单元概览 (a) ASU. (b) 步长为 2 的 ASU. (c) Bi-FM. (d) 卷积块 其中卷积为标准卷积，DWC 为深度卷积，BN 为批归一化，Swish 为激活函数(彩色视图最佳)

表 I

DPNet 主干网络的详细结构(s 表示步长为 2，LRP 和 HRP 分别表示低分辨率和高分辨率路径)

层	LRP	HRP	操作	
1	160×160×24		3×3 卷积,s	
2	80×80×24		2×2 最大池化	
3	40×40×128		ASU,s	
4-6	40×40×128		[ASU]×3	
7	20×20×256	40×40×128	ASU,s	ASU
8-14	20×20×256	40×40×128	ASU×7	ASU×7
15	20×20×256	40×40×128	Bi-FM	
16	10×10×512	40×40×128	ASU,s	ASU
17-19	10×10×512	40×40×128	ASU×3	ASU×3
20	10×10×512	40×40×128	Bi-FM	

上而下的聚合。最终，生成的输出特征$\{F1, F2, F3\}$被输入到轻量检测头，它们邻近尺度特征图之间的相互关联经过了充分的融合。

3) 检测头：检测头负责学习如何将特征映射到最终预测结果。部分检测网络[11],[12]虽采用轻量主干网络，然而其中的 SSD 检测头[4]对预测任务来说过于复杂。有一些研究[13],[14],[15]设计了轻量化的检测头来减小模型体积。同样，DPNet 也采用了轻量化的检测头来加快推理速度。如图 3(d)所示，DPNet 使用了大核的紧凑卷积(如 5×5)代替 3×3 的深度卷积[13],[14],[15]来扩展感受野，这仅仅微量增加了模型体积。检测头的详细结构见图 2。颈部网络输出的特征图$\{F1, F2, F3\}$经过两个连续的卷积块处理。最终通过 1×1 卷积生成预测结果，由对应的真实标注图进行监督训练。

B. LSCM 和 LCCM

1) LSCM：上下文信息建模的任务是提取周边信息，该任务通常由全局池化[46],[49],[50]完成。这些方法虽能生成表达图像整体信息的高层特征，但其在提供元素级交互关系上的表征能力较弱。有很多替代方案[23],[38],[39]致力于通过密集注意力图捕获全局上下文，其中每个独立像素的重要性由其他剩余像素编码。然而，这些方法需要大量的计算资源。作为 ASU 的核心单元，LCSM 在计算效率和表征能力间实现了平衡。降低计算开销的方法有两种：减少参与计算的元素数量和降低特征维度。下面我们将介绍 LCSM 如何在这两方面实现高效计算。

LCSM 的详细结构如图 4(a) 所示。令$F \in \mathbb{R}^{C\times H\times W}$为输入特征，其中$W$,$H$和$C$分别代表输入的宽度，高度和通道数。为了减少计算元素的数量，我们首先对输入特征F执行池化操作，生成一个紧凑的表征$R \in \mathbb{R}^{C\times k\times k}$,$k^2 \ll W \times H$，该表征内的每个元素对应$F$中一个包含$W$ H/k^2像素的局部区域。随后，将特征F和R都展平为两个二维序列$X \in \mathbb{R}^{HW\times C}$和$X' \in \mathbb{R}^{k^2\times C}$，便于后续空间和通道注意力的计算。

在空间注意力中，首先学习两个线性投影$\{W_{\text{sp}}^k, W_{\text{sp}}^q\} \in \mathbb{R}^{C\times C/r}$，将输入序列$X$和$X'$映射至

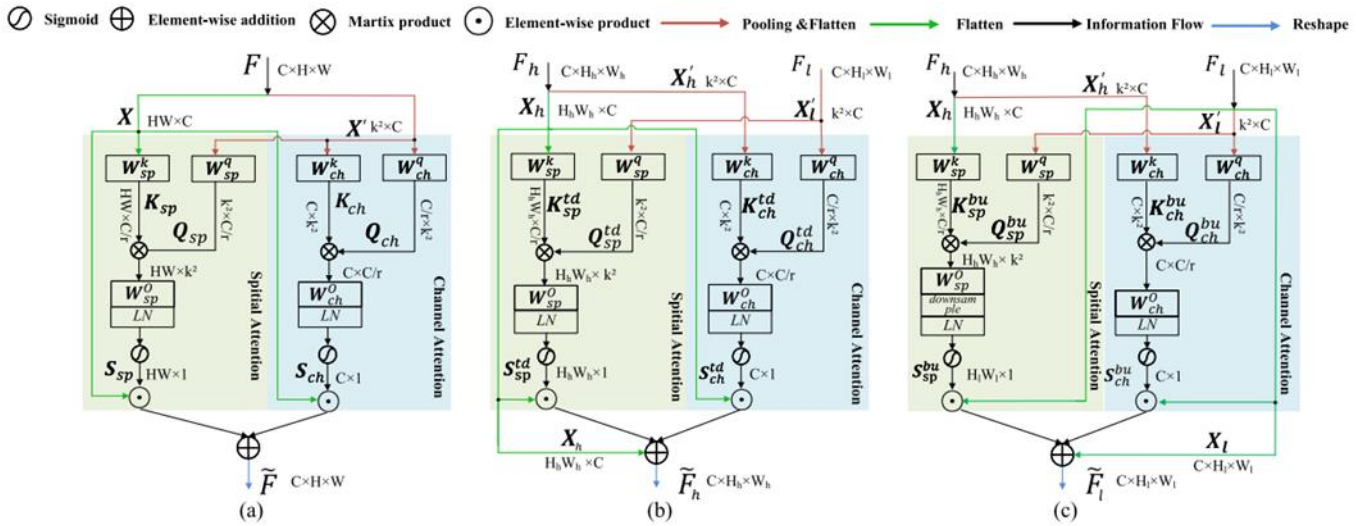

图 4 在主干和颈部网络中使用的轻量注意力概览 (a) LSCM. (b) LCCM-TD. (c) LCCM-BU. (彩色视图最佳)

表 II

自注意力，高效注意力和 LSCM 的计算复杂度对比 ($n=W\times H$,c 表示通道数量,s 表示堆叠的池化特征数量)

方法	相似度计算	重加权计算	总计算
Non-local	$\mathcal{O}(n^2c)$	$\mathcal{O}(n^2c)$	$\mathcal{O}(2n^2c)$
DANet	$\mathcal{O}(n^2c)$	$\mathcal{O}(n^2c)$	$\mathcal{O}(2n^2c)$
CCNet	$\mathcal{O}(n(H+W)c)$	$\mathcal{O}(n(H+W)c)$	$\mathcal{O}(2n(H+W)c)$
ANN	$\mathcal{O}(nsc)$	$\mathcal{O}(nsc)$	$\mathcal{O}(2nsc)$
LSCM	$\mathcal{O}\left(nc\frac{k^2}{r}\right)$	$\mathcal{O}(nsc)$	$\mathcal{O}\left(nc\left(1+\frac{k^2}{c}\right)\right)$

低维嵌入空间$K_{sp}\in\mathbb{R}^{HW\times C/r}$和$Q_{sp}\in\mathbb{R}^{k^2\times C/r}$，其中$r$是用于控制特征压缩比的非负比例因子。

$$K_{\mathrm{sp}}=XW_{\mathrm{sp}}^{k},\quad Q_{\mathrm{sp}}=X'W_{\mathrm{sp}}^{q}. \tag{1}$$

然后，通过计算K_{sp}和Q_{sp}的矩阵乘积来生成空间的像素-区域互相关关系，再依次通过线性投影$W_{\mathrm{sp}}^{O}\in\mathbb{R}^{k^2\times 1}$，层归一化$LN(\cdot)$和 sigmoid 函数$\sigma(\cdot)$的处理，最终输出空间注意力图$S_{\mathrm{sp}}\in\mathbb{R}^{HW\times 1}$

$$S_{\mathrm{sp}}=\sigma\left(LN\left(K_{\mathrm{sp}}Q_{\mathrm{sp}}^{\top}W_{\mathrm{sp}}^{O}\right)\right). \tag{2}$$

在通道注意力的计算中，为了降低特征的维度，首先学习线性投影$W_{\mathrm{ch}}^{q}\in\mathbb{R}^{C/r\times C}$将输入序列$X'$映射至低维嵌入空间$Q_{\mathrm{ch}}\in\mathbb{R}^{C/r\times k^2}$，其中$Q_{\mathrm{ch}}$中的每个通道表示$X'$中的$r$个通道。同时，另一个线性投影$W_{\mathrm{ch}}^{k}\in\mathbb{R}^{C\times C}$将$X'$映射至$K_{\mathrm{ch}}\in\mathbb{R}^{C\times k^2}$

$$K_{\mathrm{ch}}=W_{\mathrm{ch}}^{k}X'^{\top},\quad Q_{\mathrm{ch}}=W_{\mathrm{ch}}^{q}X'^{\top}. \tag{3}$$

接着，类似空间注意力的计算，使用K_{ch}和Q_{ch}的矩阵乘积来计算通道-通道组的互相关关系，再依次通过线性投影$W_{\mathrm{ch}}^{O}\in\mathbb{R}^{C/r\times 1}$，层归一化$LN(\cdot)$和 sigmoid 函数$\sigma(\cdot)$的处理，最终输出通道注意力图$S_{\mathrm{ch}}\in\mathbb{R}^{C\times 1}$

$$S_{\mathrm{ch}}=\sigma\left(LN\left(K_{\mathrm{ch}}Q_{\mathrm{ch}}^{\top}W_{\mathrm{ch}}^{O}\right)\right). \tag{4}$$

最后，学习到的空间注意力图S_{sp}和通道注意力图S_{ch}分别用于对输入序列X重加权，随后用元素加法融合，生成融合后的特征$\widetilde{X}\in\mathbb{R}^{HW\times C}$

$$\widetilde{X}=\left(S_{\mathrm{sp}}\odot X\right)\oplus\left(S_{\mathrm{ch}}\odot X\right). \tag{5}$$

其中$\oplus$和$\odot$分别是元素加法和乘法。需要注意的是，两个注意力图S_{sp}和S_{ch}分别通过列方向重加权和行方向重加权与输入序列X相乘。生成的序列$\widetilde{X}$最终被重构为$\widetilde{F}\in\mathbb{R}^{C\times H\times W}$,与输入的特征$F$维度相同，该输出用于后续的 1×1 卷积，如图 3(a)所示。

我们比较了 LSCM，自注意力[23],[24]和高效注意力[51],[52]的计算复杂度，三者均具备强大的全局上下文信息表达能力。尽管这些方法都能计算空间和通道注意力，但两者的计算复杂度相近。因此，表 II 仅展示了空间注意力的对比结果。先前的

方法和我们提出的 LSCM 均包含两个计算步骤：元素相似度计算和输入特征重加权。在自注意力机制[23],[24]中，计算密集空间注意力和特征重加权都需要n^2c次运算，导致计算复杂度随输入分辨率呈平方增长。在高效注意力机制[51],[52]中，CCNet[51]将自注意力分解为两次连续十字交叉注意力。因此，计算复杂度与特征的高度与宽度之和呈线性关系。另一方面，ANNet[52]希望降低矩阵乘法的计算成本，从而使得计算复杂度与堆叠池化特征数呈线性关系。相比之下，我们提出的 LSCM 模块仅需要$nc(k^2/r)$次运算，其计算复杂度同样与输入分辨率呈线性关系，这得益于全局池化对特征元素的显著压缩。尤为关键的是，特征重加权无需矩阵乘法，只需要nc次运算，远远小于高效注意力中的$\mathcal{O}(nsc)$和$\mathcal{O}(n(H+W)c)$，以及自注意力中的nc^2。

2) LCCM：本节将 LSCM 扩展为多输入的版本，即 LCCM，它被用于在颈部网络中融合多尺度特征。LCCM 以双向机制运行：自上而下和自下而上，分别由 LCCM-TD 和 LCCM-BU 表示。鉴于两者工作方式相似，本节只详细介绍 LCCM-TD，并指出其与 LCCM-BU 的主要差异。

LCCM-TD 的详细结构如图 4(b)所示。总的来说，LCCM-TD 与 LSCM 结构相似，但具有两个不同分辨率的输入。设高分辨率输入特征为$F_h \in \mathbb{R}^{C\times H_h\times W_h}$，低分辨率输入特征为$F_l \in \mathbb{R}^{C\times H_l\times W_l}$。由于两者源自相邻尺度的卷积层，满足$H_h=2H_l$，$W_h=2W_l$。为了挖掘跨层的交互关系并节省计算成本，同步对$F_h$和$F_l$执行全局池化来压缩分辨率，继而将它们分别展平为两个二维序列$X'_h \in \mathbb{R}^{k^2\times C}$和$X'_l \in \mathbb{R}^{k^2\times C}$，其中$k^2 \ll H_l\times W_l < H_h\times W_h$。同时，输入特征$F_h$也被展平为二维序列$X_h \in \mathbb{R}^{H_hW_h\times C}$，用于参与后续空间和通道注意力的计算。

在空间注意力中，输入的特征X_h和X'_l分别经过两个线性投影矩阵$\{W^k_{\text{sp}}, W^q_{\text{sp}}\} \in \mathbb{R}^{C\times C/r}$的映射，生成两个低维嵌入$K^{\text{td}}_{\text{sp}} \in \mathbb{R}^{H_hW_h\times C/r}$和$Q^{\text{td}}_{\text{sp}} \in \mathbb{R}^{k^2\times C/r}$，其中$r$是用于控制特征压缩比的非负比例因子。

$$K^{\text{td}}_{\text{sp}} = X_h W^k_{\text{sp}},\quad Q^{\text{td}}_{\text{sp}} = X'_l W^q_{\text{sp}}. \tag{6}$$

然后，通过计算$K^{\text{td}}_{\text{sp}}$和$Q^{\text{td}}_{\text{sp}}$的矩阵乘积得到跨层空间互相关关系，再依次通过线性投影$W^O_{\text{sp}} \in \mathbb{R}^{k^2\times 1}$，层归一化$LN(\cdot)$和 sigmoid 函数$\sigma(\cdot)$的处理，最终输出空间注意力图$S^{\text{td}}_{\text{sp}} \in \mathbb{R}^{H_hW_h\times 1}$。

$$S^{\text{td}}_{\text{sp}} = \sigma\left(LN\left(K^{\text{td}}_{\text{sp}} Q^{\text{td}^\top}_{\text{sp}} W^O_{\text{sp}}\right)\right). \tag{7}$$

在通道注意力的计算中，线性投影$W^q_{\text{ch}} \in \mathbb{R}^{C/r\times C}$首先将输入序列$X'_l$映射为一个低维嵌入$Q^{\text{td}}_{\text{ch}} \in \mathbb{R}^{C/r\times k^2}$。接着另一个线性投影$W^k_{\text{ch}} \in \mathbb{R}^{C\times C}$将输入序列$X'_h$映射为$K^{\text{td}}_{\text{ch}} \in \mathbb{R}^{C\times k^2}$。

$$K^{\text{td}}_{\text{ch}} = W^k_{\text{ch}} X'^{\top}_h,\quad Q^{\text{td}}_{\text{ch}} = W^q_{\text{ch}} X'^{\top}_l. \tag{8}$$

接下来，与空间注意力类似，通过计算$K^{\text{td}}_{\text{ch}}$和$Q^{\text{td}}_{\text{ch}}$的矩阵乘积得到跨层通道互相关关系，接着依次经过线性投影$W^O_{\text{ch}} \in \mathbb{R}^{C/r\times 1}$，层归一化$LN(\cdot)$和 sigmoid 激活函数$\sigma(\cdot)$，生成最后的通道注意力图$S^{\text{td}}_{\text{ch}} \in \mathbb{R}^{C\times 1}$

$$S^{\text{td}}_{\text{ch}} = \sigma\left(LN\left(K^{\text{td}}_{\text{ch}} Q^{\text{td}^\top}_{\text{ch}} W^O_{\text{ch}}\right)\right). \tag{9}$$

之后，学习到的空间注意力图S^{td}_{sp}与空间注意力图$S^{\text{td}}_{\text{ch}}$分别对高分辨率输入序列$X_h$进行重加权，并通过元素加法融合，生成融合特征$X_w \in \mathbb{R}^{H_hW_h\times C}$

$$X_w = (S^{\text{td}}_{\text{sp}} \odot X_h) \oplus (S^{\text{td}}_{\text{ch}} \odot X_h). \tag{10}$$

整个重加权过程作为残差函数运作，此设计使得 LCCM-TD 能够以端到端的方式进行训练

$$\widetilde{X_h} = X_w \oplus X_h. \tag{11}$$

需要说明的是，公式(10)中的两个加权操作分别对应列向和行向重加权，类似于 LSCM。产生的序列$\widetilde{X_h}$最终重构为与输入特征F_h维度相同的$\widetilde{F_h} \in \mathbb{R}^{C\times H_h\times W_h}$，为下一次的特征融合准备，如图 2 所示。

关于 LCCM-BU，其详细架构如图 4(c)所示。它与 LCCM-TD 的唯一差异在于在计算空间注意力时需要对特征分辨率进行两次下采样，以实现精确的重加权和恒等映射。

IV. 实验

为了评估我们提出的 DPNet，我们在三大高挑战性目标检测数据集上进行了详尽的实验：MS COCO[9]，Pascal VOC 2007[25]和 ImageNet[26]，涵盖了与当前实时目标检测网络的对比和消融实验。实验结果表明，我们提出的 DPNet 在检测精度和实现效率间达到了当前最优的平衡。

A.数据集与评估指标

1) *MS COCO*：MS COCO 数据集[9]是计算机视觉领域最流行的目标检测数据集。它包含了 80 个物体类别，涵盖了 11.8 万张训练图像，5000 张验证图像以及 2 万张测试图像。我们使用在训练集上训练出的 DPNet 开展了所有实验，在测试开发集上进行了与当前最优的模型的系统级性能对比，同时在验证集上开展一系列的消融实验。

2) *Pascal VOC 2007*：Pascal VOC 2007 数据集[25]相比于 MS COCO 数据集[9]规模较小，仅包含 20 个目标检测类别。类似于 Pelee[14]和 Tiny-DSOD[15]，DPNet 在 VOC 2007 与 VOC 2012 联合集上进行训练，共包含 16551 张图像，并在含 4952 张图像的 VOC 2007 测试集上进行评估。

3) *ImageNet*：ImageNet 数据集[26]被广泛用于大规模视觉识别挑战赛(LSVRC2017)，并自 2013 年起目标检测已成为其关键任务。该数据集包含 200 个物体类别，提供 28.8 万张训练图像、2 万张验证图像和 4 万张测试图像。我们在训练集上训练了 DPNet，并在验证集上完成了性能评估。

4) *评估指标*：为了实现在 MS COCO 数据集[9]上实现与其他最佳实时检测器的公平对比，我们采用标准评估指标[5] ,[17], [60]，如 AP，AP_{50}，AP_{75}，AP_S，AP_M 和 AP_L。具体来说，AP 是在交并比(IoU，预测框与真实框重叠率)的值在 0.5 到 0.95 范围内(步长为 0.05)的平均精度，综合反映了检测器性能。AP_{50} 和 AP_{75} 分别对应 IoU 值为 0.5 和 0.75 时的检测精度，而 AP_S，AP_M 和 AP_L 则分别衡量边界框面积在 $(0, 32^2]$ 像素范围的小目标、$(32^2, 96^2)$ 像素范围的中目标以及 $[96^2, +\infty)$ 像素范围的大目标的检测性能。对于 Pascal VOC 2007[25]和 ImageNet 数据集[26]，我们仅报告了 AP_{50} 的结果，记作 mAP[13],[66]。此外，采用三项广泛使用的指标来衡量实现效率：每秒浮点运算数(FLOPs)，模型体积(参数量)，每秒帧数(FPS)。

B.实验细节

1) 训练设置：DPNet 在 MS COCO 数据集[9]上采用单块 RTX 2080Ti GPU 服务器进行训练，每批次处理 88 张图像。使用随机梯度下降算法[67]训练 300 轮，仅包含 5 轮预热。初始学习率设置为 1.5×10^{-2} 并采用余弦衰减策略[68]，权重衰减系数和动量因子分别设定为 5×10^{-4} 和 9×10^{-1}。我们还采用了半精度(FP16)[69]和指数移动平均机制[70]来降低显存消耗并加速训练收敛。在数据增强方面，我们没有采用复杂方法[63],[71]，只使用了最基本的 SSD 方法[4]。具体来说，我们首先对原始图像进行色彩干扰，接着执行扩展缩放与随机裁剪。之后将变换后的图像缩放调整为 320×320 大小，并实施随机翻转与归一化处理。在推理过程中，我们借鉴了 YOLOV3[5]和 Scaled-YOLOV4[17]的方法，将 DPNet 从 pytorch 框架转换为 TensorRT FP16 格式来加快检测速度。Pascal VOC 2007[25]和 ImageNet 数据集[26]的配置与 MS COCO 数据集[9]完全一致，唯一差异在于前两者的预测类别数量分别为 20 类和 200 类，而后者的预测类别为 80 类。本研究的代码已在 Github 开源：https://github.com/Huiminshii/DPNet.

2) 损失函数设置：如图 2 所示，在轻量化检测头中采用三种损失函数：分别是用于目标分类的交叉熵损失 $\mathcal{L}_{\text{cls}}$ 和用于 IoU 预测的损失 $\mathcal{L}_{\text{iou}}$，以及边界框回归的 IoU 损失 $\mathcal{L}_{\text{reg}}$。因此，总损失函数可表示为：

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \alpha \times \mathcal{L}_{\text{iou}} + \beta \times \mathcal{L}_{\text{reg}}. \tag{12}$$

与 YOLOX[69]相同，将两个非负参数 α 和 β 设置为 1 和 0.5。为了生成真实标注，我们采用 SimOTA 标签分配策略[69] ,[72]。

3) 挑选最佳基线模型：为了展现 DPNet 的优越性，我们选择了 20 种前沿的检测器来进行对比，涵盖高精度与轻量化两类。前者包含 SSD[4]，RetainNet[60]，YOLOs[5],[17]，ATSS[61]，Sparse R-CNN[62]，Swin transformer[38]，TopFormer[59]和 MobileFormer[31]；后者则包括微型 YOLO 系列[5],[17]，MobileNet-SSDLite[12]，Pelee[14]，Tiny-

表 III

与高精度和实时目标检测器在 MS COCO 测试开发集上的检测精度和实现效率的对比 “-”表示结果未公开；‘†’和‘‡’分别表示 DPNet 在 Image Net 1K 和 21K 数据集上进行了预训练；在所有轻量检测器中，最佳结果用粗体表示

模型	年份	主干网络	输入尺寸	FLOPs(G)	参数量(M)	AP(%)	AP_{50}(%)	AP_{75}(%)	FPS
TopFormer-RetinaNet	CVPR2022	TopFormer-Tiny	1333×800	160	10.5	27.1	-	-	-
YOLOV3	Arxiv2018	DarkNet-53	320×320	19.6	62.3	28.2	51.5	29.7	56
MobileFormer-RetinaNet	CVPR2022	MobileFormer	1333×800	161	14.4	34.2	53.4	36.0	-
RetainNet	ICCV2017	ResNet-50	1333×800	251	34.2	35.7	55.0	38.5	19
ATSS	CVPR2020	ResNet-50	1333×800	205	32	43.5	61.9	47.0	28
Sparse R-CNN	CVPR2021	ResNet-50	1333×800	109	53	44.5	63.5	48.2	21
Swin	ICCV2021	Swin-Tiny	1333×800	245	38.5	45.5	66.3	48.8	22
YOLOV4	CVPR2020	CSPDarkNet53	608×608	109	53	45.5	64.1	49.5	62
Tiny-YOLOV3	Arxiv2018	Tiny-DarkNet	416×416	2.78	8.7	16.0	33.1	-	368
YOLO-ReT	WACV2022	EfficientNet-B3	320×320	-	28.3	19.7	36.5	19.3	76
Tiny-YOLOV4	CVPR2021	CSPDarkNet53-Tiny	416×416	3.45	6.1	21.7	40.2	-	**371**
MobileNet-SSDLite	CVPR2018	MobileNet	320×320	1.30	-	22.1	-	-	80
Pelee	NeurIPS2018	PeleeNet	304×304	1.29	6.0	22.4	38.3	22.9	120
Tiny-DSOD	BMVC2018	DDB-Net	300×300	1.12	-	23.2	40.4	22.8	-
Mobilie-ViT-SSDLite	ICLR2022	Mobilie-ViT-XS	320×320	-	2.7	24.8	-	-	61
MobileDets	CVPR2021	IBN+Fused+Tucker	320×320	1.43	4.9	26.9	-	-	-
Mobilie-ViT-SSDLite	ICLR2022	Mobile-ViT-S	320×320	-	5.7	27.7	-	-	80
ThunderNet	ICCV2019	SNet-535	320×320	1.30	-	28.1	46.2	29.6	-
ParCNet-SSD	ECCV2022	ParCNet	320×320	-	5.2	28.5	46.5	30.1	107
Ours	-	DPNet	320×320	**1.04**(↓0.08)	**2.5**(↓0.2)	**29.6**(↑1.1)	**46.9**(↑0.4)	**31.3**(↑1.2)	164
Ours†	-	DPNet	320×320	**1.04**(↓0.08)	**2.5**(↓0.2)	**30.2**(↑1.7)	**47.5**(↑1.0)	**31.8**(↑1.7)	164
Ours‡	-	DPNet	320×320	**1.04**(↓0.28)	**2.5**(↓0.2)	**31.3**(↑2.8)	**48.7**(↑2.2)	**33.2**(↑3.1)	164

DSOD[15]， MobileDets[30]， ThunderNet[13]，ParCNet-SSD[65]， YOLO-ReT[64]， Mobile-ViT-SSDLite[20]，PP-YOLO-Tiny[73]，YOLOX-Nano[69]以及 SCPNet[74]。若无特别说明，基线的结果均直接引用自原始文献。

C. 与先进检测器的比较

1) MS COCO 数据集上的实验结果：表 III 展示了与选定的最先进的检测器的定量对比结果，表明 DPNet 在检测精度和实现效率之间达到了最优的平衡。当从头开始训练时，DPNet 在 MS COCO 测试开发集上取得了 29.6%的 AP，模型大小仅为 2.5 M，计算量为 1.04 GFLOPs，帧率达到 164 FPS。DPNet 的检测精度 AP， AP_{50} 和 AP_{75} 均大幅超过其他所有基线模型(例如，分别比第二名的实时检测器 PaeCNet[65]高出 1.1%，0.4%，1.2%)。同时，DPNet 也具有最小的计算开销(例如比 Pelee[14]，MobileDets[30]和 Tiny-YOLOV4[17]分别约减少 0.3，0.4，2.4 GFLOPs)，并拥有最少的参数量(例如比 Mobile-ViT[20]，MobileDets[30]和 Pelee[14]分别少 0.2，2.4 和 3.5 M)。为了进一步提高检测精度，DPNet 的主干网络还在 ImageNet 1K 和 ImageNet 21K 数据集[26]上进行了预训练，相较于从头开始训练，分别带来了 0.6%和 1.7%的 AP 提升。

表 III 还展示了与部分准实时高精度检测器的对比结果。尽管这类复杂网络的检测精度比 DPNet 更高，但往往需要数十甚至数百 GFLOPs 的计算量和大量参数，这对于计算资源有限、存储空间受限的实际应用场景来说并不适用。值得特别关注的是，DPNet 的性能甚至优于模型体积更大的 YOLOV3[5]和 Top Former[59]。另一种同样采用双路径主干架构的检测器 MobileFormer[31]，其检测精度比 DPNet 高出 2.9%的 AP，然而其 GFLOPs 几乎是 DPNet 的 161 倍。

表 IV

与高精度和实时检测器在 Pascal VOC 2007 测试集上的检测精度和实现效率的对比 “-”表示结果未公开；‘†’和‘‡’分别表示 DPNet 在 Image Net 1K 和 21K 数据集上进行了预训练；在所有轻量检测器中，最佳结果用粗体表示

模型	年份	主干网络	输入尺寸	FLOPs(G)	参数量(M)	*mAP*(%)	FPS
SSD	ECCV2016	VGG-16	300×300	35.3	26.29	76.5	46
YOLOV2	CVPR2017	DarkNet-19	416×416	17.5	-	76.8	67
Tiny-YOLOV2	CVPR2017	Tiny-DarkNet	416×416	4.6	10.5	57.1	**232**
Tiny-YOLOV3	Arxiv2017	Tiny-DarkNet	416×416	2.8	8.7	58.4	210
PP-YOLO-Tiny	Arxiv2021	MobileNetV3	320×320	**0.4**	1.0	68.2	102
Pelee	NeurIPS2018	PeleeNet	304×304	1.2	6.0	70.9	125
Tiny-DSOD	BMVC2018	DDB-Net	300×300	1.1	-	72.1	105
SCPNet	JVCI2022	SCPNet	320×320	3.3	4.1	72.6	49
YOLO-ReT	WACV2022	EfficientNet-B3	320×320	-	28.3	72.9	76
YOLOX-Nano	Arxiv2021	Modified CSP	416×416	1.0	**0.9**	73.0	-
DSOD-Lite	PAMI2019	DSNet	300×300	-	10.4	76.7	26
ThunderNet	ICCV2019	SNet-535	320×320	1.3	-	78.6	214
Ours	-	DPNet	320×320	1.0(↑**0.6**)	2.5(↑**1.6**)	**79.2**(↑**0.6**)	196
Ours†	-	DPNet	320×320	1.0(↑**0.6**)	2.5(↑**1.6**)	**80.8**(↑**2.2**)	196
Ours‡	-	DPNet	320×320	1.0(↑**0.6**)	2.5(↑**1.6**)	**82.7**(↑**4.1**)	196

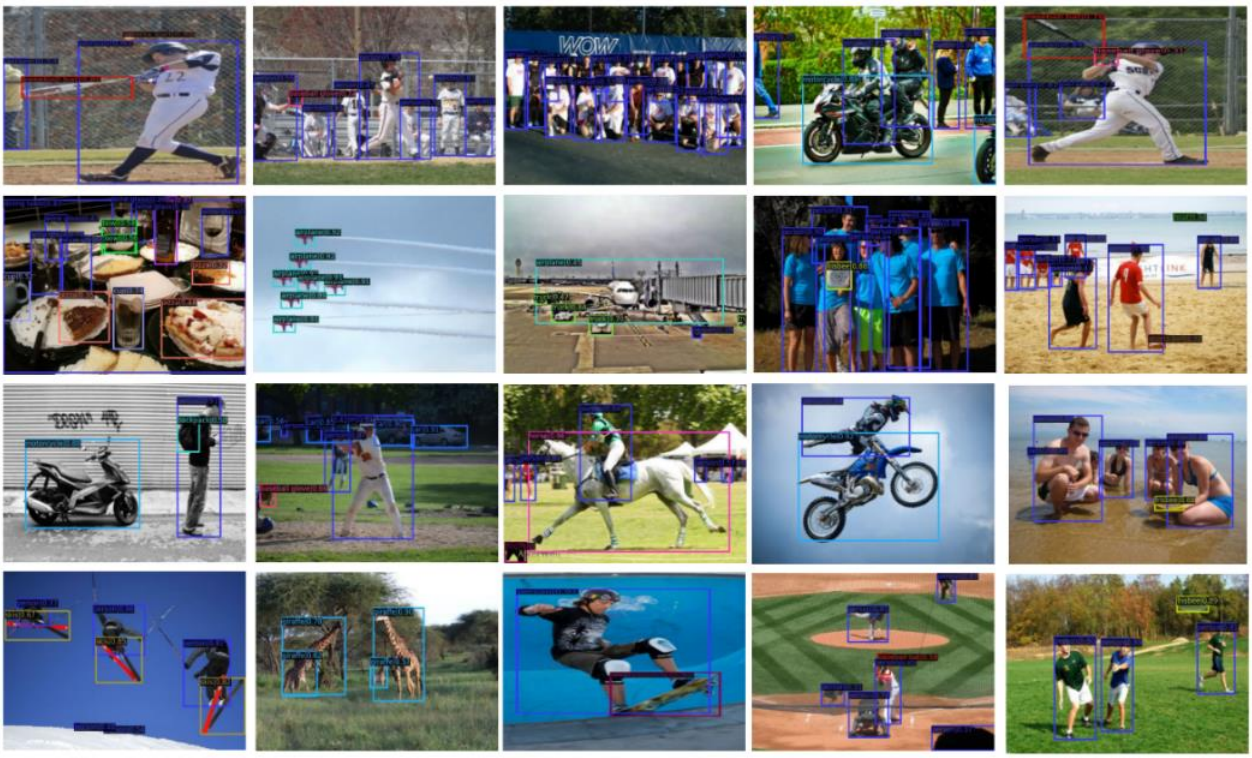

图 5 一些在 MS COCO 测试开发集上的定性检测结果的可视化示例 为了清晰起见，预测的边界框和对应标签也叠加在检测目标上(彩色视图最佳)

图 5 展示了 MS COCO 测试开发集上一些可视化示例的定性检测结果。为了便于可视化，检测到的目标上叠加显示了预测的边界框及其对应标签。结果表明，DPNet 不仅能够对不同尺度的目标进行准确分类，还能为所有目标生成精确的边界框。例如，第 2 个和第 3 个示例中的球员以及第 10 个和第 11 个示例中的人，尽管目标密集或高度重叠，DPNet 依然能够准确检测到他们。此外，DPNet 可

图 6 在 Pascal VOC 2007 测试集上的定性检测结果的可视化示例 为了清晰起见，预测的边界框和对应标签也叠加在检测目标上(彩色视图最佳)

表 V

与实时目标检测器在 Image Net 验证集上的检测精度和效率对比

模型	FLOPs(G)	参数量(M)	$mAP(\%)$	FPS
Tiny-YOLOV4	2.04	6.1	34.7	**220**
MobileDets	1.43	4.9	38.5	103
Mobile-ViT-SSDLite	1.70	2.7	39.3	61
ThunderNet	1.30	4.5	39.8	143
Ours	**1.04**(↓0.26)	**2.5**(↓0.2)	**41.6**(↑1.8)	164

以应对目标尺度的变化，如第 8 个示例中的飞机与卡车以及第 17 个示例中的长颈鹿。DPNet 在准确检测微小目标方面同样表现出色，如第 7 个示例中的飞机以及第 1 个、第 5 个、第 10 个和第 19 个示例中的小球。该数据集上的所有实验结果表明，DPNet 能够学习强大的特征表达能力，以捕获空间和通道维度的依赖关系与交互作用，在极低的计算开销下实现了卓越的检测性能。

2) Pascal VOC 2007 数据集上的实验结果：在表 IV 中所展示的 Pascal VOC 2007 测试集[25]的量化结果中，双路径主干网络先在 MS COCO 数据集[9]上进行了预训练，然后在 Pascal VOC 2007 和 2012 数据集上进行微调。DPNet 依然实现了最佳的平衡，获得了 79.2%的 mAP，同时模型大小仅为 2.5 M，计算量仅为 1.0 GFLOPs，显著优于其他最先进的实时检测器。例如，与第二名的 ThunderNet[13]相比，后者采用了包含数百层的复杂主干网络，而 DPNet 仅需 1.0 GFLOPs，计算更简便且实现了 0.6%的 mAP 提升。为进一步提升性能，我们还使用了 ImageNet 1K 和 21K 数据集[26]对双路径主干网络进行了预训练。在相同的模型大小和计算量下，mAP 平均提升了 3.15%。在帧率方面，DPNet 的运行速度快于大多数基线模型，如 Pelee[14]，Tiny-DSOD[15]和 DSOD-Lite[66]。尽管 DPNet 的运行速度略低于 ThunderNet[13]和 Tiny-YOLOV3[5]，但我们仍然实现了 196FPS 的实时检测速度，能够满足实际应用中的实时性要求。在所有基线模型中，虽然 PP-YOYO-Tiny[73] 和 YOLOX-Nano[69]分别节省了约 60%的计算量和 64%的参数量，但它们的检测效果较差，mAP 分别下降了 11.0%和 6.2%。需要注意的是，DPNet 在 Pascal VOC 2007 数据集上的运行速度略快于 MS COCO 数据集[9]，这主要是由于 Pascal VOC 2007 的分类和检测类别较少。这主要是由于 Pascal VOC 2007 数据集的分类和检测类别较少。图 6 展示了在 Pascal VOC 2007 测试集上一些目标检测结果的可视化示例。图所示，DPNet 依然获得了视觉上令

表 VI

主干网络中不同组件贡献的消融实验(红色数值是相对于基线方法的提升，参数量和 FLOPs 是在输入分辨率为 320×320 的情况下计算的)

LRP	HRP	Bi-FM	参数量(M)	FLOPs(G)	AP(%)	AP_{50}(%)	AP_{75}(%)	AP_L(%)	AP_M(%)	AP_S(%)
✓	×	×	1.82	0.87	24.0	38.2	24.6	39.2	23.8	8.2
✓	✓	×	1.97(↑**0.15**)	1.11(↑**0.24**)	26.0(↑**2.0**)	41.3(↑**3.1**)	26.6(↑**2.0**)	41.2(↑**2.0**)	26.1(↑**2.3**)	9.8(↑**1.6**)
✓	✓	✓	2.16(↑**0.34**)	1.17(↑**0.30**)	**27.0**(↑**3.0**)	**42.3**(↑**4.1**)	**28.1**(↑**3.5**)	**42.6**(↑**3.4**)	**28.0**(↑**4.2**)	**10.1**(↑**1.9**)

人满意的检测结果，能够很好地应对目标外观、方向和尺度上的视觉差异，这与图 5 中展示的检测结果保持一致。

3) ImageNet 数据集上的实验结果：本节展示了在 ImageNet 数据集[26]上的性能对比结果。由于大多数最新的实时检测方法都是在 Pascal VOC 2007[25]和 MS COCO 数据集[9]上进行评估的，我们复现了一些最先进的方法，并将输入尺寸调整为 320×320 以确保比较的公平。相关结果在表 V 中展示。与表 III 和表 IV 的结果一致，与排名第二的 ThundeNet[13]相比，DPNet 在 mAP 上大幅提高了 1.8%。即便如此，DPNet 仍然拥有最小的计算量和模型参数，并且在实时运行速度上位列第二。

D.消融实验

为了解析 DPNet 的内在机制，本节报告了一系列消融实验的结果。

1) 主干网络组成部分的消融实验：在保持颈部网络和检测头固定的情况下，表 VI 列出了针对主干网络中不同组件的消融实验结果，用以量化各部分的贡献。其中，首先仅引入 LRP 来构建基线模型，随后逐步加入 HRP 和 Bi-FM。实验结果表明，每一项组件的引入均能提升检测性能，模型大小和计算量仅有小幅增加。所有组件中，HRP 带来的提升最为显著(例如在AP，AP_{50}和AP_{75}上分别提升了 2.0%，3.1%和 2.0%)，充分体现了双路径架构设计的优势。此外，双路径主干网络还分别提升了 3.4%的AP_L，4.2%的AP_M和 1.9%的AP_S，这主要得益于 HRP 尽可能地保留了目标的细节，尤其对中小型目标效果显著。图 7 展示了依次引入各组件后的部分可视化结果。可以观察到，当各组件被依次添加后，检测结果与真实标签越来越接近，这与表 VI 中展示的定量结果一致。值得注意的是，在第 3 个示例中，尽管鲜花在真实标签中未被标注未盆栽植物，但我们的方法依然能够正确检测并识别

图 7 一些定性检测结果的可视化示例用于评估每个组件的贡献 从左到右分别为输入图像对应的标注图像，LRP 和 LRP+HRP 的预测结果 为了清晰起见，预测的边界框和对应标签叠加在检测目标上(彩色视图最佳)

为盆栽植物。

2) 不同轻量级主干网络的消融实验：不同的主干网络表征大规模视觉数据的能力存在差异。为了深入分析 DPNet 的特性，我们在固定颈部网络和检测头的情况下，分别将 DPNet 的主干网络依次替换为 ResNet-18[7]，MobileNetV2[12]，ShuffleNetV2[10]和 Tiny-DarkNet[5]。如表 VII 所示，ShuffleNetV2[10]具有最小的模型体积和计算量，但其检测精度垫底，这可能是由于其网络容量极为有限。而 DPNet 的主干网络在模型体积缩小 5 倍，运行速度提升 3.5 倍的情况下，但在检测性能上依然优于 ResNet-18[7]。这主要得益于嵌入式 LSCM 模块能以极低的计算成本高效捕捉元素级的特征交互。

3) LCSM 的消融实验：目前已经有许多注意力模块被用于捕捉全局上下文信息。因此，本部分对提出的 LCSM 模块与最先进的注意力模块进行对比。具体来说，基线模型为完整的 DPNet，但去除了主干网络中所有 ASU 单元中的 LSCM 模块，随后将 LSCM 和其他注意力模块分别插入到图 3(a)所示的相同位置进行实验。对比结果如表 VIII 所示。实验表明，LSCM 不仅优于仅关注通道注意力的模块(如 SE[45]、ECA[46]和 GC[48])，也超过了同时融合空间和通道注意力的模块(如 CBAM[49]、SK[75]和 SA[76])。和基线模型相比，模型大小和计算量仅有小幅增加，这表明 LSCM 是轻量且高

表 VII

使用不同轻量主干网络的消融实验(红色数值是相对于第二名 ResNet-18 的提升)

主干网络	参数量(M)	FLOPs(G)	AP(%)	AP_{50}(%)	AP_{75}(%)
Tiny-	7.33	1.67	18.1	32.6	17.8
ShuNetV2	**1.57**	**0.78**	22.5	37.1	23.4
MobNetV2	2.16	1.17	26.3	41.6	27.3
ResNet-18	11.98	4.25	28.4	44.6	30.0
DPNet	2.42	1.20	**28.7**(↑**0.3**)	**44.8**(↑**0.2**)	**30.2**(↑**0.2**)

表 VIII

LSCM 和其他先进注意力模块的消融实验和对比(红色数值是相对于基线的提升)

主干网络	参数量(M)	FLOPs(G)	AP(%)	AP_{50}(%)	AP_{75}(%)
基线方法	2.16	1.17	27.0	42.3	28.1
SE	2.23	**1.18**	27.3	42.7	28.4
ECA	**2.17**	**1.18**	27.6	43.1	28.5
CBAM	2.23	**1.18**	27.4	42.6	28.2
SK	2.47	1.24	27.8	43.1	28.9
SA	**2.17**	**1.18**	27.7	43.4	28.5
GC	2.41	1.19	28.1	43.2	29.3
LSCM	2.42	1.20	**28.7**(↑**1.7**)	**44.8**(↑**2.5**)	**30.2**(↑**2.1**)

效的模块，显著提升了 1.7%的AP，2.5%的AP_{50}和 2.1%的AP_{75}。此外，尤为值得关注的是，LSCM 与其他注意力模块在参数量和计算量上几乎相同，却能获得更为精确的检测结果。

4) LCCM 的消融实验：由于颈部网络在实时目标检测中起着关键作用，本节在保持主干网络和检测头不变的前提下，评估了所引入的 LCCM 模块的效果。为了深入分析 LCCM，我们首先仅考虑 LCCM-TD，随后逐步加入 LCCM-BU。实验结果如表 IX 所示，并与当前广泛使用的 FPN 进行了对比。当仅采用 LCCM-TD 时，相较于 AugFPN[77]，其性能略有下降(AP 和AP_{75}均下降 0.2%)，这表明仅采用自上而下的特征融合方式难以获得理想的结果。然而，当同时引入 LCCM-TD 和 LCCM-BU 后，DPNet 的性能较 AugFPN[77]有了大幅提升。此外，结合了 LCCM-TD 和 LCCM-BU 的 LCCM 以最小的模型体积和计算量实现了最佳的检测性能。具体而言，采用了 LCCM 的完整 DPNet 仅有 2.42 M 的参数量和 1.04 GFLOPs 的计算量，但在 AP 上相较于 FPN[53]和 BFP[78]分别提升了 1.4%

表 IX

LCCM 和其他先进 FPN 的消融实验和对比(红色数值是相对于第二名 ASF 的提升)

算法	参数量	FLOPs	AP(%)	AP_{50}(%)	AP_{75}(%)
FPN	2.79M	1.20G	28.7	44.8	30.2
BFP	2.86M	1.22G	29.0	45.0	30.4
PAFPN	3.38M	1.36G	29.1	45.3	30.4
ASF	3.04M	1.24G	29.3	45.5	30.5
LCCM-TD	2.39M	1.01G	29.1(↓**0.2**)	45.5(↑**0.0**)	30.3(↓**0.2**)
LCCM-TD-BU	**2.42M**	**1.04G**	**30.1**(↑**0.8**)	**46.0**(↑**0.5**)	**30.9**(↑**0.4**)

表 X

DPNet 在不同视觉任务上适应性的消融实验 包括图像分类、目标检测和语义分割(使用 Top-1 准确率、AP 和 Mask AP 评估)

视觉任务	数据集	输入尺寸	FLOPs	参数量	性能	FPS
图像分类	ImageNet	320×320	1.36G	3.2M	76.7%	144
目标检测	MS COCO	320×320	1.04G	2.5M	29.6%	164
语义分割	MS COCO	320×320	1.13G	2.7M	31.4%	151

和 1.1%。值得注意的是，PAFPN[56]同样采用了自下而上和自上而下的双向特征融合路径，但 LCCM 仍以极低的计算开销在AP 、AP_{50}和AP_{75}上全面超越了它。

5) DPNet 通用性的消融实验：本部分评估了 DPNet 在分类、检测和分割等不同视觉任务中的泛化能力。对于分类任务，我们在 ImageNet 数据集[26]上进行了实验，具体做法是将检测头替换为全连接层(FCL)来预测图像标签。我们还在 MS COCO 数据集[9]上评估了 DPNet 在实例分割任务中的表现，具体做法是为每个检测头输入增加一个分支用于掩码估计。DPNet 在各项任务中的性能与实现效率如表 X 所示。分类任务需要更大的模型体积和计算量，且推理速度较慢，这可能是因为添加的全连接层引入了大量的参数和计算。相反，对于实例分割任务，参数量和计算量仅有小幅增加。

6) 不同输入尺寸的消融实验：为了进一步验证我们提出方法的优势，我们在 Pascal VOC[25]和 MS COCO 数据集[9]上，分别以 224×224、300×300、320×320 和 416×416 的输入尺寸进行了消融实验。在 Pascal VOC 数据集[25]上，我们选择 DSOD-Lite[66]和 ThundeNet[13]作为基线模型；在 MS COCO 数据集[9]上，则与 Mobile-ViT[20]、ThunderNet[13]和 ParCNet[65]进行对比。图 8 的对

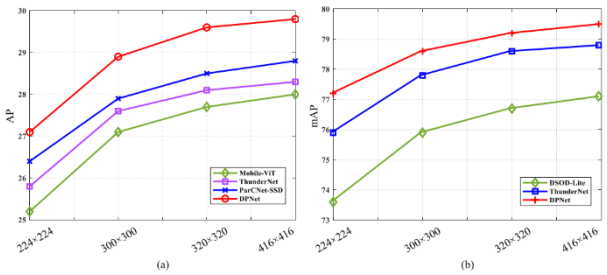

图 8 在(a) MS COCO 和(b) Pascal VOC 2007 数据集上性能随不同输入大小的变化

表 XI

池化核大小k的消融实验

池化核大小k	参数量(M)	FLOPs(G)	AP(%)	AP_{50}(%)	AP_{75}(%)
3	**2.78**	**1.13**	28.5	44.3	29.9
5	2.79	1.20	**28.8**	**44.8**	**30.2**
7	2.79	1.61	28.6	44.6	30.1
9	2.79	1.80	28.6	44.4	30.0

表 XII

比例因子r的消融实验

比例因子r	参数量(M)	FLOPs(G)	AP(%)	AP_{50}(%)	AP_{75}(%)
1	4.01	1.43	**29.3**	**45.1**	**30.2**
2	3.31	1.30	28.8	45.0	29.9
4	2.96	1.23	28.7	44.7	30.1
8	2.79	1.20	29.0	44.8	**30.2**
16	**2.70**	**1.19**	28.5	44.7	30.0

比结果显示：无论输入分辨率如何变化，DPNet 始终保持着最优的检测性能。同时还可以观察到，随着输入尺寸的增加，检测性能也持续提升，这表明更大的输入尺寸能够带来更优的性能表现。

E. 参数设置分析

1) LSCM 中池化核大小k的影响：池化核大小k决定参与互相关计算的元素数量，显著影响着 LSCM 的计算效率。当k的值以步长 2 从 3 增至 9 时，结果如表 XI 所示。随着池化核大小k的增大，参与互相关计算的特征元素数目也随之增加，导致计算量几乎增加了两倍，但模型参数量无显著变化。当k值为 5 时，AP 达到最佳的 28.8%，因此在 DPNet 中将其设为默认参数。

2) LSCM 中压缩比r的影响：除了池化核大小k之外，压缩比r也是影响 LSCM 容量和运行速度的重要参数。因此，我们通过调整r的取值进行实验，结果如表 XII 所示。当$r=1$时，LSCM 退化为类似于自注意力[23]的密集注意力机制，此时检测的 AP 值达到最高，但模型参数量和开销也最大；随着r的增大，模型参数量和计算量逐渐降低，但 AP 在 r=8 时达到峰值，因此在 DPNet 中将其设为默认参数。

V. 结论与未来展望

本文提出了一种用于实时检测的双路径轻量级网络 DPNet。采用了双路径设计的主干网络既能捕捉高层语义信息，同时又能很好地保留低层细节。此外，这两条并行路径并非相互独立，特征的交互增强了双路径之间的信息交流。为了提升 DPNet 的表征能力，我们在主干网络中设计了轻量级注意力模块 LSCM，能够以极低的计算开销捕获全局交互信息。我们还在颈部网络将 LSCM 扩展为 LCCM，充分挖掘不同分辨率邻近尺度特征的关联依赖。我们在 MS COCO、Pascal VOC 2007 和 ImageNet 三个主流目标检测数据集上对所提出的方法进行了评估，实验结果表明，DPNet 在检测精度与实现效率之间达到了最优的平衡。

未来，我们计划从两个方向继续优化 DPNet。其一，如表 III 所示，DPNet 与高精度检测器之间仍然存在显著性能差距，需进一步改进和提升我们的模型；其二，在保证实时检测性能优势的同时，我们认为 DPNet 还可以应用到其他视觉任务，如图像分类[7]、语义分割[51],[52]和显著性目标检测[79],[80]。

参考文献

[1] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards realtime object detection with region proposal networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99

[2] Z. Wu, J. Wen, Y. Xu, J. Yang, X. Li, and D. Zhang, “Enhanced spatial feature learning for

weakly supervised object detection," *IEEE Trans. Neural Netw. Learn. Syst*., pp. 1–12, 2022.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[4] W. Liu et al., "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis*., Oct. 2016, pp. 21–37.

[5] J. Redmon and A. Farhadi, "YOLOv3: An i ncremental improvement,"2018, *arXiv:1804.02 767*.

[6] K. Shih, C. Chiu, J. Lin, and Y. Bu, "Real-time object detection with reduced region proposal network via multi-feature concatenation," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 31, no. 6, pp. 2164–2173, Jun. 2020.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–12.

[9] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis*., 2014, pp. 740–755.

[10] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.

[11] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*., Jun. 2018, pp. 4510–4520.

[13] Z. Qin et al., "ThunderNet: Towards real-time generic object detection on mobile devices," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6718–6727.

[14] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Proc. Int. Conf. Neural Inf. Process. Syst*., 2018, pp. 1967–1976.

[15] Y. Li, J. Li, W. Lin, and J. Li, "Tiny-DSOD: Lightweight object detection for resource-restricted usages," in *Proc. BMVC*, 2018, pp. 6718–6727.

[16] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[17] C.-Y. Wang, A. Bochkovskiy, and H. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13024–13033.

[18] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., Jun. 2018, pp. 6848–6856.

[19] X. Li et al., "Semantic flow for fast and accurate scene parsing," in *Proc. Eur. Conf. Comput. Vis*., Aug. 2020, pp. 775–793.

[20] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," in *Proc. ICLR*, 2022, pp. 1–12.

[21] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—Improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1743–1751.

[22] X. Ding, X. Zhang, Y. Zhang, J. Han, G. Ding, and J. Sun, "Scaling up your kernels to 31x31: Revisiting large kernel design in CNNs," 2022, *arXiv:2203.06717*.

[23] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF*

Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7794–7803.

[24] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[25] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis*., vol. 88, pp. 303–338, Jun. 2010.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., Jun. 2009, pp. 248–255.

[27] Z. Wang, J. Lu, Z. Wu, and J. Zhou, "Learning efficient binarized object detectors with information compression," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 44, no. 6, pp. 3082–3095, Jun. 2022.

[28] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst*., vol. 28, 2015, pp. 1135–1143.

[29] X. Dai et al., "General instance distillation for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7842–7851.

[30] Y. Xiong et al., "MobileDets: Searching for object detection architectures for mobile accelerators," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*., Jun. 2021, pp. 3825–3834.

[31] Y. Chen et al., "Mobile-Former: Bridging MobileNet and transformer,"in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5260–5269.

[32] Y. Li et al., "MicroNet: Improving image recognition with extremely low FLOPs," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 458–467.

[33] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., Jun. 2017, pp. 4700–4708.

[34] X. Jin et al., "A lightweight encoder–decoder path for deep residual networks," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 33, no. 2, pp. 866–878, Feb. 2022.

[35] Y. Tian et al., "Learning lightweight dynamic kernels with attention inside via local–global context fusion," *IEEE Trans. Neural Netw. Learn. Syst*., 2022.

[36] X. Chang, H. Pan, W. Sun, and H. Gao, "YolTrack: Multitask learning based real-time multiobject tracking and segmentation for autonomous vehicles," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 32, no. 12,pp. 5323–5333, Dec. 2021.

[37] C. Yeh et al., "Lightweight deep neural network for joint learning of underwater object detection and color conversion," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 33, no. 11, pp. 6129–6143, Nov. 2022.

[38] Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[39] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[40] Z. Wu, Z. Liu, J. Lin, Y. Lin, and S. Han, "Lite transformer with longshort range attention," in *Proc. ICLR*, 2020, pp. 1–12.

[41] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity,"*J. Mach. Learn. Res*., vol. 23, no. 120, pp. 1–39, 2022.

[42] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," 2022, *arXiv:2205.14135*.

[43] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc.*

Eur. Conf. Comput. Vis., Sep. 2018, pp. 325–341.

[44] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "BiSeNet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3051–3068, Nov. 2021.

[45] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*., Jun. 2018, pp. 7132–7141.

[46] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*. (CVPR), Jun. 2020, pp. 11531–11539.

[47] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., Jun. 2019, pp. 3024–3033.

[48] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1971–1980.

[49] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis*., Sep. 2018, pp. 3–19.

[50] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst*., vol. 31, 2018, pp. 9423–9433.

[51] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[52] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proc. CVPR*, Oct. 2019, pp. 593–602.

[53] T. Y. Lin, P. Dollàr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., Jul. 2017, pp. 2117–2125.

[54] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*., Jun. 2018, pp. 8759–8768.

[55] Q. Tang, J. Li, Z. Shi, and Y. Hu, "Lightdet: A lightweight and accurate object detection network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2243–2247.

[56] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[57] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in P*roc. Int. Conf. Mach. Learn*., 2015, pp. 448–456.

[58] R. Prajit, B. Zoph, and V. L. Quoc, "Swish: A self-gated activation function," 2017, *arXiv:1710.059417*.

[59] W. Zhang et al., "TopFormer: Token pyramid transformer for mobile semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12083–12093.

[60] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis*., Jul. 2017, pp. 2980–2988.

[61] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9759–9768.

[62] P. Sun et al., "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*

(CVPR), Jun. 2021, pp. 14454–14463.

[63] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *2020, arXiv:2004.10934*.

[64] P. Ganesh, Y. Chen, Y. Yang, D. Chen, and M. Winslett, "YOLO-ReT: Towards high accuracy real-time object detection on edge GPUs," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1311–1321.

[65] H. Zhang, W. Hu, and X. Wang, "ParC-Net: Position aware circular convolution with merits from ConvNets and transformer," in *Proc. Eur. Conf. Comput. Vis*., 2022, pp. 613–630.

[66] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "Object detection from scratch with deep supervision," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 42, no. 2, pp. 398–412, Feb. 2020.

[67] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist*., 2010, pp. 177–186.

[68] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.

[69] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," *2021, arXiv:2107.08430*.

[70] P. Micikevicius et al., "Mixed precision training," 2017, arXiv:1710.03740.

[71] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.

[72] Z. Ge, S. Liu, Z. Li, O. Yoshie, and J. Sun, "OTA: Optimal transport assignment for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 303–312.

[73] X. Huang et al., "PP-YOLOv2: A practical object detector," 2021, *arXiv:2104.10419*.

[74] X. Zhong, M. Wang, W. Liu, J. Yuan, and W. Huang, "SCPNet: Selfconstrained parallelism network for keypoint-based lightweight object detection," *J. Vis. Commun. Image Represent*., vol. 90, Feb. 2022, Art. no. 103719.

[75] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[76] Q.-L. Zhang and Y.-B. Yang, "SA-Net: Shuffle attention for deep convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2235–2239.

[77] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "AugFPN: Improving multi-scale feature learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12595–12604.

[78] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 821–830.

[79] Y. Ji, H. Zhang, Z. Jie, L. Ma, and Q. M. J. Wu, "CASNet: A crossattention Siamese network for video salient object detection," *IEEE Trans. Neural Netw. Learn. Syst*., vol. 32, no. 6, pp. 2676–2690, Jul. 2021.

[80] Y. Liu, M. Cheng, X. Zhang, G. Nie, and M. Wang, "DNA: Deeply supervised nonlinear aggregation for salient object detection," *IEEE Trans. Cybern*., vol. 52, no. 7, pp. 6131–6142, Jul. 2022.