PVT v2: 基于金字塔视觉 Transformer 改进的模型*

王文海 [1,2], 谢恩泽 [3], 李翔 [4], 范登平 [5], 宋凯涛 [4], 梁鼎 [6], 路通 [2], 罗平 [3], 邵岭 [5]
[1] 上海人工智能实验室 [2] 南京大学 [3] 香港大学 [4] 南京理工大学
[5] 起源人工智能研究院 [6] 商汤科技
wangwenhai@pjlab.org.cn

摘要

最近，*Transformer* 在计算机视觉领域取得了令人鼓舞的进展。在本研究中，我们在原始的金字塔视觉 *Transformer*（*PVT v1*）的基础上进行改进，添加了三个设计：*(1)* 线性复杂度的注意力层；*(2)* 重叠图像块嵌入；*(3)* 卷积前馈网络，提出了新的基线。通过这些改进，*PVT v2* 将 *PVT v1* 的计算复杂度降低到线性，显著改善了 *PVT v1* 在分类、检测和分割等基本视觉任务上的表现。值得注意的是，*PVT v2* 与 *Swin Transformer* 等最近的模型相比，取得了相当甚至更好的性能。我们希望这项工作能促进在计算机视觉领域最先进的 *Transformer* 研究。代码参见*https://github.com/whai362/PVT*。

1. 引言

目前，视觉 Transformer 的研究正聚焦于为下游视觉任务（如图像分类、目标检测、实例和语义分割等）而设计的主干网络 [8, 31, 34, 35, 23, 37, 10, 5]。迄今为止，这部分研究已取得一些令人鼓舞的成果。例如，Vision Transformer（ViT）[8] 首先证明，纯 Transformer 在图像分类任务中可达到最好的性能。Pyramid Vision Transformer （PVT v1） [34]则表明纯 Transformer 主干在检测和分割等密集预测任务方面的表现也可以超越 CNN [22, 42]。此后，Swin Transformer [23]、CoaT [37]、LeViT [10]、Twins [5]使用 Transformer 主干网络进一步提高了其在分类、检测和分割任务中的性能。

本文的工作旨在创建在 PVT v1 框架上更强更可行的基线。本文报告了三个设计改进，即：(1) 线性复杂度注意力层；(2) 重叠图像块嵌入；(3) 卷积前馈网络，这三个设计均正交于 PVT v1 框架，并且当与 PVT v1 结合使用时，可以带来更好的图像分类、目标检测、实例和语义分割性能。我们称改进后的框架为 PVT v2。其中，PVT v2-B5[1]在 ImageNet 上的 top-1 错误率为 83.8%，优于 Swin-B [23]和 Twins-SVT-L [5]，而且我们的模型具有更少的参数和 GFLOPs。此外，GFL [19]利用 PVT-B2 模型在 COCO `val2017` 数据集上的 AP 达到了 50.2，比使用 Swin-T [23] 模型高 2.6，比使用 ResNet50 [13] 模型高 5.7。本文希望这些改进的基线能为将来视觉 Transformer 的研究提供参考。

2. 相关工作

本文主要讨论与本工作相关的 Transformer 主干网络。ViT [8]将每个图像视为由固定长度的词元（图像块）组成的序列，并将它们输入到多个 Transformer 层中进行分类任务，这是首个证明在训练数据充足的情况下（如：ImageNet-22k [7]、JFT-300M），纯 Transformer 也能够在图像分类任务中达到最先进性能的工作。DeiT [31]进一步探索了一种数据高效的训练策略和一种用于 ViT 的蒸馏方法。

为提高图像分类的性能，最近的方法针对 ViT 进行了定制化改进。T2T ViT [38]逐步将重叠滑动窗口内的词元（tokens）连接成一个词元。TNT [11]利用内部和外部 Transformer 块分别生成像素和图像块的嵌入。CPVT [6]用条件位置编码替换了 ViT 中的固定大小位置嵌入，使其更容易处理任意分辨率的图像。

*本文为 PVT v2[33]论文中文翻译版
这篇论文已被 Computational Visual Media 接受。

[1]根据参数数量不同，PVT v2 有 B0 到 B5 6 种不同大小的变体。

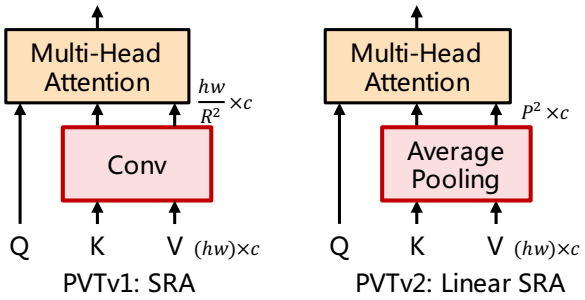

图 1: PVT v1 中的 SRA 与 PVT v2 中的线性 SRA 的比较。

CrossViT [2]通过双分支 Transformer 处理不同尺寸的图像块。LocalViT [20]将深度卷积引入到视觉 Transformer 中，以提高特征的局部连续性。

为解决像目标检测、实例分割和语义分割等密集预测任务，也有一些方法 [34, 23, 35, 37, 10, 5]，将 CNN 中的金字塔结构引入到 Transformer 主干的设计中。PVT v1 是第一个金字塔结构的 Transformer，它给出了一个有四个阶段的分层 Transformer，这表明一个纯 Transformer 主干可以像 CNN 主干一样通用，并且可以在检测和分割任务上表现得更好。在此之后，一些改进 [23, 35, 37, 10, 5]增强了特征的局部连续性，去除了固定大小的位置嵌入。例如，Swin Transformer [23]用相对位置偏差替换固定大小的位置嵌入，并将自注意力机制限制在移动窗口内。CvT [35], CoaT [37]和 LeViT [10]在视觉 Transformer 中引入了类似卷积的操作。Twins [5]将局部注意力机制和全局注意力机制结合起来，以获得更强的特征表示。

3. 方法

3.1. PVT v1 中的限制

PVT v1 [34]主要有以下三个局限性：（1）与 ViT [8]类似，当处理高分辨率输入 (如：短边为 800 像素) 时，PVT v1 的计算复杂度相对较大；（2）PVT v1 [34]将图像视为一系列非重叠的图像块，这在一定程度上丢失了图像的局部连续性；（3）PVT v1 的位置编码是固定大小的，这对处理任意大小的图像不够灵活。这些局限性限制了 PVT v1 在处理视觉任务上的性能。

为解决这些问题，我们提出 PVT v2，通过三种设计对 PVT v1 进行改进，分别在第3.2、3.3和 3.4节列出。

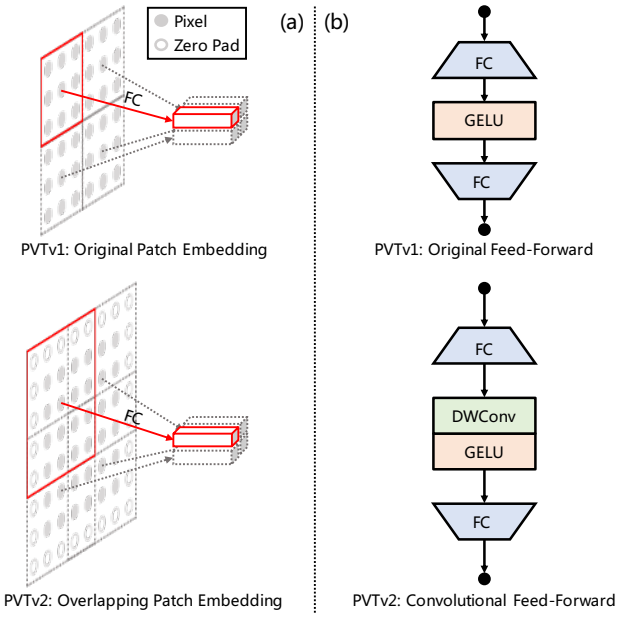

图 2: PVT v2 中的两个改进。（a）重叠图像块嵌入（b）卷积前馈网络。

3.2. 线性空间缩减注意力层

首先，为了减少由注意力操作引起的高计算成本，本方法提出了线性空间缩减注意力（SRA）层，如图 1 所示。不同于使用卷积进行空间缩减的 SRA [34]，线性 SRA 在进行注意力操作之前使用平均池化将空间维度（即：$h \times w$）缩减到一个固定大小（即：$P \times P$）。因此，线性 SRA 同卷积层一样享有线性的计算复杂度和内存开销。具体来说，给定大小为 $h \times w \times c$ 的输入，SRA 和线性 SRA 的复杂度为:

$$\Omega(\text{SRA}) = \frac{2h^2w^2c}{R^2} + hwc^2R^2, \tag{1}$$

$$\Omega(\text{Linear SRA}) = 2hwP^2c, \tag{2}$$

其中 R 是 SRA [34]的空间缩减比率，P 是线性 SRA 的池化大小，这里设置为 7。

3.3. 重叠图像块嵌入

其次，为了对局部连续信息进行建模，我们利用重叠图像块嵌入（Overlapping Patch Embedding）将图像分解成一系列的图像块。如图2（a）所示，本方法将图像块窗口放大，使相邻的窗口重叠一半的面积，并对特

	Output Size	Layer Name	Pyramid Vision Transformer v2						
			B0	B1	B2	B2-Li	B3	B4	B5
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	Overlapping Patch Embedding	$S_1 = 4$						
			$C_1 = 32$	$C_1 = 64$					
		Transformer Encoder	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 2$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 2$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 3$	$P_1 = 7$ $N_1 = 1$ $E_1 = 8$ $L_1 = 3$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 3$	$R_1 = 8$ $N_1 = 1$ $E_1 = 8$ $L_1 = 3$	$R_1 = 8$ $N_1 = 1$ $E_1 = 4$ $L_1 = 3$
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	Overlapping Patch Embedding	$S_2 = 2$						
			$C_2 = 64$	$C_2 = 128$					
		Transformer Encoder	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 2$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 2$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 3$	$P_2 = 7$ $N_2 = 2$ $E_2 = 8$ $L_2 = 3$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 3$	$R_2 = 4$ $N_2 = 2$ $E_2 = 8$ $L_2 = 8$	$R_2 = 4$ $N_2 = 2$ $E_2 = 4$ $L_2 = 6$
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	Overlapping Patch Embedding	$S_3 = 2$						
			$C_3 = 160$	$C_3 = 320$					
		Transformer Encoder	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 2$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 2$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 6$	$P_3 = 7$ $N_3 = 5$ $E_3 = 4$ $L_3 = 6$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 18$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 27$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 40$
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	Overlapping Patch Embedding	$S_4 = 2$						
			$C_4 = 256$	$C_4 = 512$					
		Transformer Encoder	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 2$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 2$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$P_4 = 7$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$

表 1: PVT v2 系列设置的细节。“-L”表示带有线性 SRA 的 PVT v2。

征图进行零填充以保持分辨率。在该设计中，我们使用零填充的卷积来实现重叠图像块嵌入。具体来说，给定一个大小为 $h \times w \times c$ 的输入，采用卷积的方式，步长为 S，卷积核大小为 $2S-1$，填充大小为 $S-1$，卷积核大小为 c'，则输出大小为 $\frac{h}{S} \times \frac{w}{S} \times C'$。

3.4. 卷积前馈

第三，受先前研究的启发 [17, 6, 20]，我们移除了固定大小的位置编码 [8]，并在 PVT 中引入了零填充位置编码。如图2（b）所示，在前馈网络中的第一个全连接层（FC）和 GELU [15]激活函数之间，我们添加了一个填充大小为 1 的 3×3 深度卷积 [16]。

3.5. PVT v2 序列的细节

我们通过改变超参数将 PVT v2 从 B0 扩展到 B5，调整的超参数如下所列：

- S_i：第 i 阶段中重叠图像块嵌入的步长；
- C_i：第 i 阶段输出通道数；
- L_i：第 i 阶段编码器层的数量；
- R_i：第 i 阶段 SRA 的减少比率；
- P_i：第 i 阶段线性 SRA 的自适应平均池化大小；
- N_i：第 i 阶段高效的自注意力头数 i；
- E_i：第 i 阶段前馈层 [32]的扩张比率；

表1 展示了 PVT v2 系列的详细信息。本工作的设计遵循了 ResNet [14]的原则：（1）随着层数增加，通道维度增加，空间分辨率减小；（2）在第三阶段分配了大部分的计算成本。

3.6. PVT v2 的优势

结合这些改进，PVT v2 能够：（1）获得更多图像和特征图中的局部连续性；（2）更灵活地处理可变分辨率的输入；（3）享受与 CNN 相同的线性复杂度。

4. 实验

4.1. 图像分类

设置：图像分类实验是在 ImageNet-1K[27] 数据集上进行的，该数据集包含了来自 1,000 个类别的 128 万张训练图像和 50,000 张验证图像。为了公平比较，所有的模型都在训练集上进行训练并报告了在验证集上的 top-1 错误率。我们遵循 DeiT [31]中的设置，并使用随机裁剪、随机水平翻转 [29]、标签平滑正则化 [30]、mixup [39]和随机擦除 [41]来进行数据增强。在训练期间，我们使用动量为 0.9、批量大小为 128、权重衰减为 5×10^{-2} 的 AdamW [25]来优化模型。本实验的初始学习率设置为 1×10^{-3}，并按照余弦退火策略 [24]递减。所有模型都是在 8 个 V100 GPU 上从零开始训练 300

Method	#Param (M)	GFLOPs	Top-1 Acc (%)
PVTv2-B0 (ours)	**3.4**	**0.6**	**70.5**
ResNet18 [14]	11.7	1.8	69.8
DeiT-Tiny/16 [31]	**5.7**	**1.3**	72.2
PVTv1-Tiny [34]	13.2	1.9	75.1
PVTv2-B1 (ours)	13.1	2.1	**78.7**
ResNet50 [14]	25.6	4.1	76.1
ResNeXt50-32x4d [36]	25.0	4.3	77.6
RegNetY-4G [26]	21.0	4.0	80.0
DeiT-Small/16 [31]	22.1	4.6	79.9
T2T-ViT$_t$-14 [38]	22.0	6.1	80.7
PVTv1-Small [34]	24.5	3.8	79.8
TNT-S [11]	23.8	5.2	81.3
Swin-T [23]	29.0	4.5	81.3
CvT-13 [35]	**20.0**	4.5	81.6
CoaT-Lite Small [37]	**20.0**	4.0	81.9
Twins-SVT-S [5]	24.0	**2.8**	81.7
PVTv2-B2-Li (ours)	22.6	3.9	**82.1**
PVTv2-B2 (ours)	25.4	4.0	82.0
ResNet101 [14]	44.7	7.9	77.4
ResNeXt101-32x4d [36]	44.2	8.0	78.8
RegNetY-8G [26]	39.0	8.0	81.7
T2T-ViT$_t$-19 [38]	39.0	9.8	81.4
PVTv1-Medium [34]	44.2	6.7	81.2
CvT-21 [35]	**32.0**	7.1	82.5
PVTv2-B3 (ours)	45.2	**6.9**	**83.2**
ResNet152 [14]	60.2	11.6	78.3
T2T-ViT$_t$-24 [38]	64.0	15.0	82.2
PVTv1-Large [34]	61.4	9.8	81.7
TNT-B [11]	66.0	14.1	82.8
Swin-S [23]	**50.0**	8.7	83.0
Twins-SVT-B [5]	56.0	**8.3**	83.2
PVTv2-B4 (ours)	62.6	10.1	**83.6**
ResNeXt101-64x4d [36]	83.5	15.6	79.6
RegNetY-16G [26]	84.0	16.0	82.9
ViT-Base/16 [8]	86.6	17.6	81.8
DeiT-Base/16 [31]	86.6	17.6	81.8
Swin-B [23]	88.0	15.4	83.3
Twins-SVT-L [5]	99.2	14.8	83.7
PVTv2-B5 (ours)	**82.0**	**11.8**	**83.8**

表 2: ImageNet 验证集上的图像分类性能。“#Param” 指的是参数数量。“GFLOPs” 是在输入尺度为 224×224 下计算的。“*” 表示该方法在其原始论文的训练策略下的性能。“-Li” 表示带有线性 SRA 的 PVT v2。

轮。为了进行基准对比，本文在验证集上使用了中心裁剪，裁剪了一个 224×224 的图像块来评估分类的精度。**结果：**从表2 中我们看出，PVT v2 是在 ImageNet-1K 上进行分类任务的最先进方法。PVT v2 的 GFLOPs 和参数与 PVT 相似，但图像分类准确率大幅提升。例如，PVT v2-B1 的准确率比 PVT v1-Tiny 高 3.6%，PVT v2-B4 比 PVT-Large 高 1.9%。与其他最近的模型相比，PVT v2 系列在准确性和模型大小方面也具有显著优势。例如，PVT v2-B5 在 ImageNet 上的 top-1 准确率为 83.8%，比 Swin Transformer [23] 和 Twins [5]高 0.5%，而参数和 GFLOPS 却更少。

4.2. 目标检测

设置：本实验是在具有挑战性的 COCO [22]基准测试数据集上进行的。所有模型都在 COCO `train2017`（11.8 万张图像）上训练，并在 `val2017`（5,000 张图像）上评估。本文在主流的目标检测器（即 RetinaNet [21]、Mask R-CNN [12]、Cascade Mask R-CNN [1]、ATSS [40]、GFL [19]和 Sparse R-CNN [28]）上验证 PVT v2 骨干网络的有效性。在训练之前，我们使用在 ImageNet 上预训练的权重来初始化主干网络，并使用 Xavier [9]方法来初始化新增的层。本实验在 8 个 V100 GPU 上以批量大小为 16 来进行训练模型，并使用初始学习率为 1×10^{-4} 的 AdamW [25]进行优化。按照惯例 [21, 12, 3]，本文采用 $1\times$ 或 $3\times$ 的训练计划（即 12 或 36 轮）对所有检测模型进行训练。训练图像的短边为 800 像素，长边不超过 1333 像素。当使用 $3\times$ 训练策略时，本文将输入图像的短边随机地调整在 $[640, 800]$ 这个范围内。在测试阶段，输入图像的短边固定为 800 像素。

结果：如表 3 所示，在模型大小相似的条件下，PVT v2 在单阶段（one-stage）目标检测器和双阶段（two-stage）目标检测器上的表现均明显优于 PVT v1。例如，PVT v2-B4 在 RetinaNet [21] 上的 AP 为 46.1，在 Mask R-CNN [12]的 AP 为 47.5，分别比 PVT v1 模型高出 3.5 和 4.6。我们在图3 中展示了 PVT v2 在 COCO [22] `val2017` 数据集上的一些定性的目标检测和实例分割结果，这也显示了本模型的良好性能。

为了公平比较 PVT v2 和 Swin Transformer [23]的性能，本实验保持所有设置不变，包括 ImageNet-1K 的预训练权值和 COCO 数据集上微调的策略。在 Cascade R-CNN [1]、ATSS [40], GFL [19]和 Sparse R-CNN [28]这四种最先进的目标检测器上评估 Swin Transformer 和 PVT v2 的性能。本文观察到 PVT v2 在所有检测器上的 AP 均明显优于 Swin Transformer，这显示出其更好的特征表示能力。例如，在 ATSS 上，PVT v2 与 Swin-T 的参数和浮点运算数相似，但是 PVT v2 的 AP 达到了 49.9，比 Swin-T 高 2.7。同时，我们的 PVT v2-Li 能够大幅减少计算量，只需牺牲少许性能，就可将 GFLOPs 从 258 减少到 194。

Backbone	RetinaNet 1×							Mask R-CNN 1×						
	#P (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	#P (M)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
PVTv2-B0	**13.0**	**37.2**	**57.2**	**39.5**	**23.1**	**40.4**	**49.7**	**23.5**	**38.2**	**60.5**	**40.7**	**36.2**	**57.8**	**38.6**
ResNet18 [14]	**21.3**	31.8	49.6	33.6	16.3	34.3	43.2	**31.2**	34.0	54.0	36.7	31.2	51.0	32.7
PVTv1-Tiny [34]	23.0	36.7	56.9	38.9	22.6	38.8	50.0	32.9	36.7	59.2	39.3	35.1	56.7	37.3
PVTv2-B1 (ours)	23.8	**41.2**	**61.9**	**43.9**	**25.4**	**44.5**	**54.3**	33.7	**41.8**	**64.3**	**45.9**	**38.8**	**61.2**	**41.6**
ResNet50 [14]	37.7	36.3	55.3	38.6	19.3	40.0	48.8	44.2	38.0	58.6	41.4	34.4	55.1	36.7
PVTv1-Small [34]	34.2	40.4	61.3	43.0	25.0	42.9	55.7	44.1	40.4	62.9	43.8	37.8	60.1	40.3
PVTv2-B2-Li (ours)	**32.3**	43.6	64.7	46.8	28.3	47.6	57.4	**42.2**	44.1	66.3	48.4	40.5	63.2	43.6
PVTv2-B2 (ours)	35.1	**44.6**	**65.6**	**47.6**	**27.4**	**48.8**	**58.6**	45.0	**45.3**	**67.1**	**49.6**	**41.2**	**64.2**	**44.4**
ResNet101 [14]	56.7	38.5	57.8	41.2	21.4	42.6	51.1	63.2	40.4	61.1	44.2	36.4	57.7	38.8
ResNeXt101-32x4d [36]	56.4	39.9	59.6	42.7	22.3	44.2	52.5	**62.8**	41.9	62.5	45.9	37.5	59.4	40.2
PVTv1-Medium [34]	**53.9**	41.9	63.1	44.3	25.0	44.9	57.6	63.9	42.0	64.4	45.6	39.0	61.6	42.1
PVTv2-B3 (ours)	55.0	**45.9**	**66.8**	**49.3**	**28.6**	**49.8**	**61.4**	64.9	**47.0**	**68.1**	**51.7**	**42.5**	**65.7**	**45.7**
PVTv1-Large [34]	**71.1**	42.6	63.7	45.4	25.8	46.0	58.4	**81.0**	42.9	65.0	46.6	39.5	61.9	42.5
PVTv2-B4 (ours)	72.3	**46.1**	**66.9**	**49.2**	**28.4**	**50.0**	**62.2**	82.2	**47.5**	**68.7**	**52.0**	**42.7**	**66.1**	**46.1**
ResNeXt101-64x4d [36]	95.5	41.0	60.9	44.0	23.9	45.2	54.0	101.9	42.8	63.8	47.3	38.4	60.6	41.3
PVTv2-B5 (ours)	**91.7**	**46.2**	**67.1**	**49.5**	**28.5**	**50.0**	**62.5**	**101.6**	**47.4**	**68.6**	**51.9**	**42.5**	**65.7**	**46.0**

表 3: 在 COCO 的 `val2017` 数据集上进行目标检测和实例分割 `val2017`。“#P”表示参数数量。AP^b 和 AP^m 分别表示 bounding box AP 和 mask AP。“-Li”表示带有线性 SRA 的 PVT v2

Backbone	Method	AP^b	AP^b_{50}	AP^b_{75}	#P (M)	GFLOPs
ResNet50 [14]	Cascade Mask R-CNN [1]	46.3	64.3	50.5	82	739
Swin-T [23]		50.5	69.3	54.9	86	745
PVTv2-B2-Li (ours)		50.9	69.5	55.2	**80**	**725**
PVTv2-B2 (ours)		**51.1**	**69.8**	**55.3**	83	788
ResNet50 [14]	ATSS [40]	43.5	61.9	47.0	32	205
Swin-T [23]		47.2	66.5	51.3	36	215
PVTv2-B2-Li (ours)		48.9	68.1	53.4	**30**	**194**
PVTv2-B2 (ours)		**49.9**	**69.1**	**54.1**	33	258
ResNet50 [14]	GFL [19]	44.5	63.0	48.3	32	208
Swin-T [23]		47.6	66.8	51.7	36	215
PVTv2-B2-Li (ours)		49.2	68.2	53.7	**30**	**197**
PVTv2-B2 (ours)		**50.2**	**69.4**	**54.7**	33	261
ResNet50 [14]	Sparse R-CNN [28]	44.5	63.4	48.2	106	166
Swin-T [23]		47.9	67.3	52.3	110	172
PVTv2-B2-Li (ours)		48.9	68.3	53.4	**104**	**151**
PVTv2-B2 (ours)		**50.1**	**69.5**	**54.9**	107	215

表 4: 与 Swin Transformer 在目标检测上进行比较。“AP^b”表示 bounding box AP。“#P”指的是参数数量。“GFLOPs" 是在输入尺度为 1280 × 800 下计算的。“-Li”表示带有线性 SRA 的 PVT v2。

Backbone	Semantic FPN		
	#Param (M)	GFLOPs	mIoU (%)
PVTv2-B0 (ours)	**7.6**	**25.0**	**37.2**
ResNet18 [14]	**15.5**	**32.2**	32.9
PVTv1-Tiny [34]	17.0	33.2	35.7
PVTv2-B1 (ours)	17.8	34.2	**42.5**
ResNet50 [14]	28.5	45.6	36.7
PVTv1-Small [34]	28.2	44.5	39.8
PVTv2-B2-Li (ours)	**26.3**	**41.0**	45.1
PVTv2-B2 (ours)	29.1	45.8	**45.2**
ResNet101 [14]	47.5	65.1	38.8
ResNeXt101-32x4d [36]	**47.1**	64.7	39.7
PVTv1-Medium [34]	48.0	**61.0**	41.6
PVTv2-B3 (ours)	49.0	62.4	**47.3**
PVTv1-Large [34]	**65.1**	**79.6**	42.1
PVTv2-B4 (ours)	66.3	81.3	**47.9**
ResNeXt101-64x4d [36]	86.4	103.9	40.2
PVTv2-B5 (ours)	**85.7**	**91.1**	**48.7**

表 5: 不同骨干网络在 ADE20K 验证集上的语义分割性能。“GFLOPs”是在输入尺度为 512 × 512 下计算的。“-Li”表示带有线性 SRA 的 PVT v2。

4.3. 语义分割

设置：同 PVT v1 [34]一样，我们选择 ADE20K [42] 数据集来对语义分割的性能进行基准测试。为了公平比较，本实验在 Semantic FPN [18]的基础上对 PVT v2 主干网络进行评估。在训练阶段，我们使用在 ImageNet [7]数据集上预训练的权重来初始化主干网络，对于其他新添加的层，使用 Xavier [9]上预训练的权重来初始化。我们使用初始学习率为 1e-4 的 AdamW [25]来优化模型。按照惯例 [18, 4]，本实验在 4 个 V100 GPU 上对模型进行 40k 次迭代训练，其中批量大小为 16。本实验的学习率按照幂为 0.9 的多项式衰减策略进行衰减。为了适合训练，我们将图片随机裁剪为 512 × 512 像素，在测试时，将图片较短的一边调整为 512 像素。

结果：如表5 所示，当使用 Semantic FPN [18]进行语义分割时，PVT v2 比 PVT v1 [34]和其他模型表现更好。例如，在参数数量和 GFLOPs 几乎相同的情况下，PVT v2-B1/B2/B3/B4 的 mIoU 至少比 PVT v1-

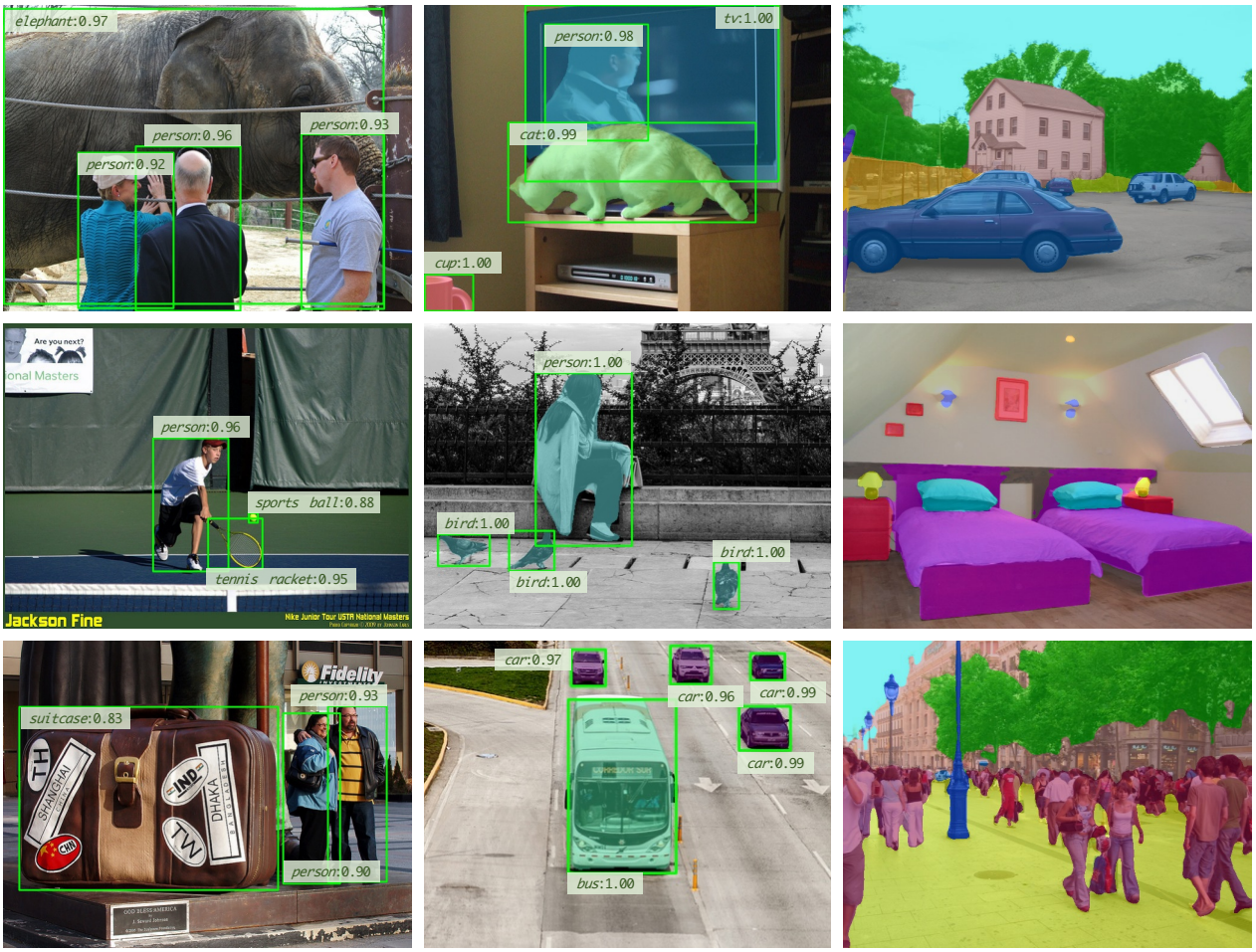

图 3: 在 COCO `val2017` 上进行目标检测和实例分割以及在 ADE20K 上进行语义分割的定性结果 [22, 42]。结果（从左到右）分别由基于 PVT v2-B2 的 RetinaNet [21]、Mask R-CNN [12] 和 Semantic FPN [18] 生成。

Tiny/Small/Medium/Large 高出 5.3%。

此外，尽管 PVT-Large 的 GFLOPs 比 ResNeXt101-64x4d 低 12%，但 mIoU 比后者高 8.5（48.7 *vs.* 40.2）。最后，在图3 中，本文还展示了在 ADE20K [42]上的一些定性语义分割结果。这些结果表明，PVT v2 主干网络通过改进的设计可以更好的提取语义分割特征。

4.4. 消融实验

4.4.1 模型分析

如表6 中 PVT v2 的消融实验结果所示，本文提出的三种设计能在性能、参数数量或计算开销方面改进模型。**重叠图像块嵌入（OPE）是重要的**。如表6 所示，对比

#	Setting	Top-1 Acc (%)	RetinaNet 1x		
			#P (M)	GFLOPs	AP
1	PVTv1-Small [34]	79.8	34.2	285.8	40.4
2	+ OPE	81.1	34.9	288.6	42.2
3	++ CFFN (PVTv2-B2)	82.0	35.1	290.7	**44.6**
4	+++ LSRA (PVTv2-B2-Li)	**82.1**	**32.3**	**227.4**	43.6

表 6: PVT v2 的消融实验。"OPE"、"CFFN" 和"LSRA" 分别代表重叠图像块嵌入、卷积前馈网络和线性 SRA。

#1 和 #2，本文发现采用 OPE 的模型相较于使用原始图像块嵌入的模型 (PE) [8]，在 ImageNet 上的 top-1 准确率更高（81.1% *vs.* 79.8%），在 COCO 上的 AP 更高（42.2% *vs.* 40.4%）。因此，OPE 是有效的，它可以通过重叠滑动窗口对图像和特征图的局部连续性进行建模。

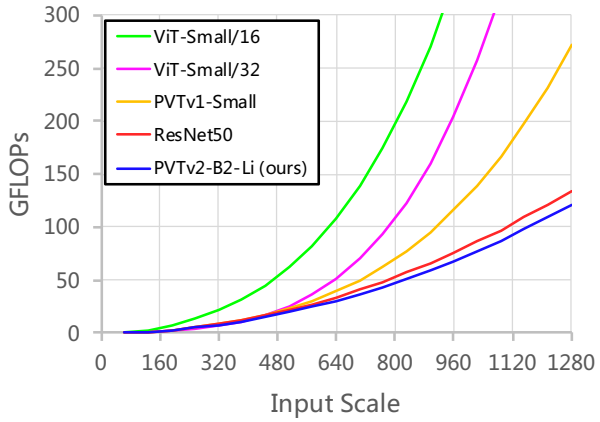

图 4: 不同输入尺度下模型的 GFLOPs。 GFLOPs 的增长率：ViT-Small/16 [8]>ViT-Small/32 [8]>PVT v1-Small [34]>ResNet50 [14]>PVT v2-B2-Li （我们的）。

卷积前馈网络（CFFN）很重要。与原始的前馈网络（FFN） [8]相比，CFFN 使用一个零填充卷积层来捕获输入张量的局部连续性。此外，由于 OPE 和 CFFN 中零填充引入的位置信息，我们可以移除 PVT v1 中使用的固定大小的位置嵌入，使得模型能够灵活处理不同分辨率的输入。如表6 中的 #2 和 #3 所示，CFFN 在 ImageNet 上的 top-1 准确率提升了 0.9%（82.0% *vs.* 81.1%），在 COCO 上的 AP 提升了 2.4，这表明了其有效性。

线性 SRA（LSRA）有助于构建更好的模型。如表6 中的 #3 和 #4 所示，相较于 SRA [34]，LSRA 将模型的计算开销（GFLOPs）显著地减少了 22%，并仍能在 ImageNet 上保持了相当的 top-1 准确率（82.1% *vs.* 82.0%），并且在 COCO 上的 AP 只下降了 1（43.6 *vs.* 44.6）。这些结果表明了 LSRA 的低计算成本和良好效果。

4.4.2 计算开销分析

如图4 所示，随着输入尺度的增加，本文提出的 PVT v2-B2-Li 的 GFLOPs 增长速率越来越低于 PVT v1-Small [34]，并且与 ResNet-50 [13]的增长速率相似。这个结果证明了我们的 PVT v2-Li 成功地解决了由注意力层引起的高计算开销问题。

5. 结论

本文研究了金字塔视觉 Transformer（PVT v1）的局限性，并通过三项设计进行了改进，这三项设计分别是重叠图像块嵌入、卷积前馈网络和线性空间缩减注意力层。通过广泛的实验（如图像分类、目标检测和语义分割等）证明，本文提出的 PVT v2 在相同参数数量下比其前身 PVT v1 和其他最先进的基于 Transformer 的主干网络更强大。我们希望这项工作能促进在计算机视觉领域最先进的 Transformer 的研究。

参考文献

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018. 4, 5

[2] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *arXiv preprint arXiv:2103.14899*, 2021. 2

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 4

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017. 5

[5] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *arXiv preprint arXiv:2104.13840*, 2021. 1, 2, 4

[6] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 1, 3

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2009. 1, 5

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *Proc. Int. Conf. Learn. Representations*, 2021. 1, 2, 3, 4, 6, 7

[9] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. Int. Conf. Artificial Intell. & Stat.*, 2010. 4, 5

[10] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2021. 1, 2

[11] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021. 1, 4

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017. 4, 6

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2015. 1, 7

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016. 3, 4, 5, 7

[15] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3

[16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[17] Md. Amirul Islam, Sen Jia, and Neil D. B. Bruce. How much position information do convolutional neural networks encode? In *Proc. Int. Conf. Learn. Representations*, 2020. 3

[18] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019. 5, 6

[19] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In *Proc. Advances in Neural Inf. Process. Syst.*, 2020. 1, 4, 5

[20] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 2, 3

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017. 4, 6

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, 2014. 1, 4, 6

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1, 2, 4, 5

[24] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *Proc. Int. Conf. Learn. Representations*, 2017. 3

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. Int. Conf. Learn. Representations*, 2019. 3, 4, 5

[26] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020. 4

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 2015. 3

[28] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2021. 4, 5

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015. 3

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016. 3

[31] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proc. Int. Conf. Mach. Learn.*, 2021. 1, 3, 4

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Advances in Neural Inf. Process. Syst.*, 2017. 3

[33] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021. 1

[34] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 1, 2, 4, 5, 6, 7

[35] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 1, 2, 4

[36] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017. 4, 5

[37] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. *arXiv preprint arXiv:2104.06399*, 2021. 1, 2, 4

[38] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 1, 4

[39] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proc. Int. Conf. Learn. Representations*, 2018. 3

[40] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020. 4, 5

[41] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proc. AAAI Conf. Artificial Intell.*, 2020. 3

[42] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017. 1, 5, 6