Conv2Former: 一个简单的用于视觉识别的 Transformer 式卷积网络

侯淇彬, 陆承泽, 程明明, 和冯佳时

摘要—视觉变换器（Vision Transformers）由于其强大的全局信息编码能力，最近成为了视觉识别领域最受欢迎的网络架构。然而，处理高分辨率图像时的高计算成本限制了其在下游任务中的应用。在本文中，我们深入研究了自注意力的内部结构，并提出了一种简单的变换器风格的卷积神经网络用于视觉识别。通过比较近期的卷积神经网络和视觉变换器的设计原则，我们提出通过利用卷积调制操作来简化自注意力。我们展示了这种简单的方法可以更好地利用卷积层中嵌套的大核（$\geq 7 \times 7$），并且我们观察到，当逐渐增加核大小从 5×5 到 21×21 时，性能持续提高。我们构建了一组使用所提出的卷积调制的层次化卷积神经网络，称为 Conv2Former。我们的网络简单且易于理解。实验表明，我们的 Conv2Former 在所有 ImageNet 分类、COCO 目标检测和 ADE20k 语义分割任务中，超越了现有的流行卷积神经网络和视觉变换器，如 Swin Transformer 和 ConvNeXt。 我们的代码可在 https://github.com/HVision-NKU/Conv2Former 上获取。

Index Terms—卷积神经网络，视觉变换器（vision transformer），卷积调制，大核卷积

1 引言

在 2010 年代，视觉识别领域取得了巨大的进步，这主要归功于卷积神经网络（ConvNets），以 VGGNet [1]、Inception 系列 [2], [3], [4] 和 ResNet 系列 [5], [6], [7], [8] 等为代表。这些识别模型主要通过堆叠多个构建模块并采用金字塔网络架构来聚合具有大感受野的结果，但忽视了显式建模全局上下文信息的重要性。SENet 系列 [9], [10], [11] 突破了传统卷积神经网络的设计，将基于注意力的机制引入卷积神经网络中以捕获长距离依赖关系，取得了令人惊讶的良好性能。

自 2020 年以来，视觉变换器（ViTs）[12], [13], [14], [15], [16] 进一步推动了视觉识别模型的发展，并在 ImageNet 分类和下游任务上展示了比最先进的卷积神经网络 [17], [18] 更好的结果。这是因为与提供局部连接的卷积相比，变换器（Transformer）中的自注意力机制能够建模全局的成对依赖关系，提供了一种更有效的方式来编码空间信息，如 [19] 中所示。然而，在处理高分辨率图像时，自注意力造成的计算成本是相当高昂的。

最近，一项有趣的研究，名为 ConvNeXt [20]，揭示了通过简单地现代化标准 ResNet，并使用与变换器（Transformer）类似的设计和训练方法，卷积神经网络（ConvNets）的表现甚至可以超过一些流行的视觉变换器（ViTs） [14], [15]。

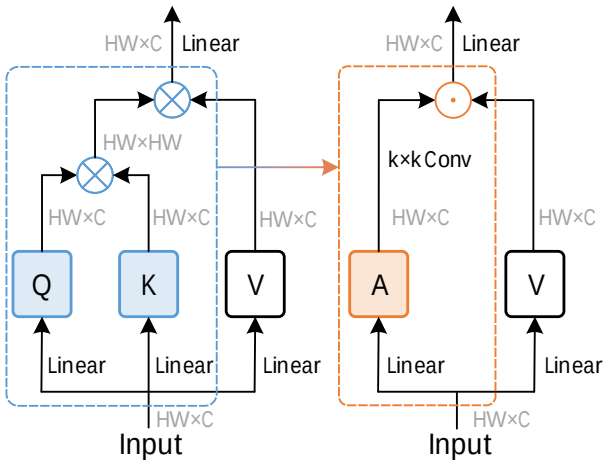

图 1. 自注意力机制与提出的卷积调制操作的比较。如图所示，我们不是通过查询（query）和键（key）之间的矩阵乘法来生成注意力矩阵，而是直接使用 $k \times k$ 深度卷积产生权重，通过哈达玛积（$\odot$：哈达玛积；$\otimes$：矩阵乘法）重新加权值。

RepLKNet [21] 也展示了利用大核卷积进行视觉识别的潜力。这些探索鼓励了许多研究人员重新思考卷积神经网络的设计，例如通过利用大核卷积 [22], [23]、高阶空间交互 [24] 或稀疏卷积核 [25] 等。到目前为止，如何更有效地利用卷积来构建强大的卷积神经网络架构仍然是计算机视觉中的一个热门研究话题。

在本文中，我们也对探索新方法以更好地利用空间卷积感兴趣。与 ConvNeXt 工作 [20] 不同，ConvNeXt 旨在调整训练配方或构建模块中空间卷积的位置，我们比较了 ViTs 和 ConvNets 用于编码空间信息的不同方式。如图 Fig. 1的左半

- 侯淇彬, 陆承泽和程明明隶属于中国天津南开大学计算机学院。*(andrewhoux@gmail.com, cmm@nankai.edu.cn)* 程明明是通讯作者。
- 冯佳时隶属于新加坡字节跳动。

- 该研究被国家自然科学基金 *(NO. 62225604, No. 62276145)*, 中央高校基本科研业务费专项资金 *(南开大学, 070-63223049)*, 青年精英科学家资助计划 *(No. YESS20210377)* 所支持。计算得到南开大学超级计算中心的支持 *(NKSC)*。

部分所示，自注意力通过所有其他位置的加权求和来计算每个像素的输出。这个过程也可以通过计算大核卷积的输出与值表示之间的哈达玛积来模拟，我们称之为卷积调制，如图 Fig. 1的右半部分所示。不同之处在于卷积核是静态的，而自注意力生成的注意力矩阵可以根据输入内容进行调整。我们的实验表明，使用卷积来生成权重矩阵也能产生很好的结果。

简单地将 ViTs 中的自注意力替换为我们提出的卷积调制操作，就得到了我们提出的网络，称为 Conv2Former。其背后的含义是我们旨在使用卷积来构建一个 Transformer 风格的卷积神经网络（ConvNet），在这个网络中，卷积特征被用作权重来调制值的表示。与具有自注意力的经典 ViTs 不同，我们的方法，像许多经典 ConvNets 一样，是完全卷积的，因此其计算量随着图像分辨率的提高而线性增加，而不是像变换器（Transformer）那样呈二次方增加。这使得我们的方法更适合于下游任务，如目标检测和高分辨率语义分割。

本文的另一个主要贡献是，我们展示了 Conv2Former 可以从更大的卷积核中获得更多的优势，比如 11×11 和 21×21。这与之前 ConvNets [20], [26] 的结论不同，他们表明使用标准深度卷积和大于 9×9 的核大小几乎不会带来性能提升，只会增加计算负担。我们的实验表明，当逐渐将卷积核大小从 5×5 增加到 21×21 时，可以获得一致的性能提升。我们还展示了我们使用 11×11 深度卷积的方法，其性能甚至超过了最近使用超大核卷积的工作 [21], [25]（例如，31×31），这反映了我们提出的空间编码方法的有效性。

我们在流行的视觉任务上评估了 Conv2Former ，包括 ImageNet 分类 [27]、COCO 目标检测/实例分割 [28]、以及 ADE20k 语义分割 [29]。为了验证 Conv2Former 在更大数据集上的能力，我们还在我们的模型上预训练了 ImageNet-22k 数据集，并在下游任务上评估性能。实验表明，Conv2Former 的表现优于流行的卷积神经网络，如 ConvNeXt [20] 和 EfficientNetV2 [18]。我们希望我们的工作能为未来的视觉效果识别模型提供有益的设计选择。

2 相关工作

2.1 卷积神经网络

早期视觉识别模型的成功主要归功于卷积神经网络（ConvNets）的发展，以 VGGNet [1] 和 GoogLeNet [2] 为代表。这些模型由于梯度消失问题，通常包含不到 20 层的卷积层。后来，ResNets [5] 的出现通过引入残差连接推进了传统卷积神经网络的发展，使得训练非常深的模型成为可能。Inceptions [3], [4] 和 ResNeXt [6] 进一步丰富了 ConvNets 的设计原则，并提出使用具有多个并行路径的专门滤波器卷积的构建模块。SENet [9] 及其后续工作 [10], [11] 旨在通过轻量级注意力模块改进 ConvNets，这些模块可以显式地建模通道间的相互依赖性。EfficientNets [17], [18] 和 MobileNetV3 [30] 利用神经架构搜索 [31] 来寻找高效的网络架构。最近，一些工作旨在展示引入大核卷积的优势 [20], [21], [22], [24], [25]。一个典型的例子是 VAN [22]，它利用标准深度卷积和扩张卷积来分解大核卷积。HorNet [24] 通过基于递归门控卷积明确构建高阶空间交互，进一步推进了 VAN。我们的 Conv2Former 与 VAN 和 HorNet 不同，我们的目标不是分解大核卷积，而是展示自注意力可以简化为卷积调制操作，这也带来了良好的识别性能。我们的工作也与 DWNet [32] 相关，后者也试图连接局部自注意力和深度卷积。与 DWNet 不同，我们的 Conv2Former 旨在通过深度卷积产生注意力权重，通过哈达玛积重新加权值，而 DWNet 用深度卷积替换了整个局部自注意力。 此外，还有一些工作利用不同的训练或优化方法或微调技术 [33], [34], [35] 来推进 EfficientNet 的发展。

2.2 视觉变换器（Vision Transformer）

Transformers，最初被设计用于自然语言处理任务 [36]，已被广泛应用于视觉识别。最典型的工作应该是视觉变换器（Vision Transformer，ViT）[12]，它展示了变换器（Transformer）在大规模图像分类数据处理中的巨大潜力。DeiT [13] 通过使用强大的数据增强方法和知识蒸馏改进了原始 ViT，并摆脱了 ViT 对大规模数据的依赖。受到卷积神经网络中金字塔架构成功的启发，一些工作 [14], [15], [37], [38] 设计了使用变换器（Transformer）的金字塔结构，以利用多尺度特征。一些工作 [39], [40], [41], [42], [43], [44] 提出将局部依赖引入 ViTs，在视觉识别中表现出色。此外，还有一些工作 [16], [45], [46], [47], [48] 探索了 ViTs 在视觉识别中的扩展能力。特别是，Yuan 等人 [16] 展示了两阶段 ViT 首次在 ImageNet 上超越了最先进的 CNN。

2.3 其他模型

近期的一些研究表明，将变换器（Transformers）和卷积混合使用 [19], [43], [49] 是开发更强大视觉识别模型的有前景的方式，特别是对于那些旨在高效网络设计的模型。一个典型的例子是 MobileViT [50]，它提供了一种有效融合卷积和变换器的方法。EfficientViT [51]、EdgeNeXt [52] 和 MobileFormer [53] 将卷积重新引入变换器（Transformer），并在图像分类和下游任务中展现出卓越的性能。此外，还有一些混合网络将不同的注意力机制引入卷积神经网络中，用于全局上下文编码 [54], [55], [56], [57], [58]。此外，设计类似 MLP（多层感知机）的架构也是视觉识别领域的热门研究课题 [59], [60], [61]。

我们的方法也与一些旨在提高 CNNs 或 ViTs 空间编码能力或效率的最新工作 [62], [63], [64], [65] 相关。RIFormer [62] 引入了重新参数化的思想，以减少 ViTs 中的令牌混合操作，提高推理效率。LITv2 [64] 在较低分辨率下使用自注意力编码部分空间信息，以提高运行速度。Hu 等人 [63] 通过实验研究了一些典型的空间令牌混合器，并分析了它们在多个视觉任

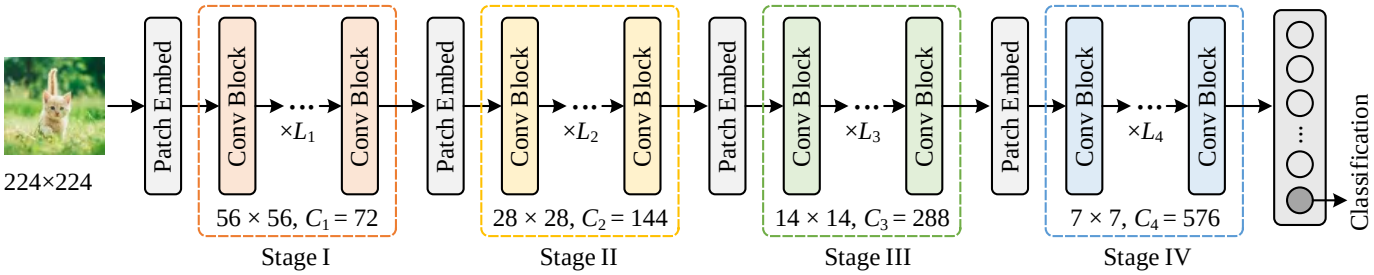

图 2. Conv2Former 的总体架构。像大多数以前的卷积神经网络和 Swin Transformer 一样，我们采用了具有四个阶段的金字塔结构。在每个阶段，使用了不同数量的卷积块。这个图展示了所提出的 Conv2Former-T 的设置，其中 $\{L_1, L_2, L_3, L_4\} = \{3, 3, 12, 3\}$。

表 1
提出的 Conv2Former 的简要配置。我们实现了 5 个变体，其参数数量分别为 15M、27M、50M、90M 和 199M。

Model	$\{C_1, C_2, C_3, C_4\}$	$\{L_1, L_2, L_3, L_4\}$
★ Conv2Former-N	$\{64, 128, 256, 512\}$	$\{2, 2, 8, 2\}$
★ Conv2Former-T	$\{72, 144, 288, 576\}$	$\{3, 3, 12, 3\}$
★ Conv2Former-S	$\{72, 144, 288, 576\}$	$\{4, 4, 32, 4\}$
★ Conv2Former-B	$\{96, 192, 384, 768\}$	$\{4, 4, 34, 4\}$
★ Conv2Former-L	$\{128, 256, 512, 1024\}$	$\{4, 4, 48, 4\}$

表 2
与三个流行的模型进行阶段比较。如最后一行所示，稍微调整卷积块的数量可以提高性能。

Model	Params.	FLOPs	Stage Conf.	Top-1 Acc.
ResNet-50 [5]	26M	4.0G	3-4-6-3	78.5%
Swin-T [14]	28M	4.5G	2-2-6-2	81.5%
ConvNeXt-T [20]	29M	4.5G	3-3-9-3	82.1%
★ Conv2Former-N	15M	2.2G	2-2-8-2	81.5%
★ Conv2Former-T	28M	4.4G	3-3-8-3	82.8%
★ Conv2Former-T	27M	4.4G	3-3-12-3	83.2%

务上的性能。SMT [65] 是我们方法的同时期工作，它展示了混合使用不同核大小的卷积有助于视觉识别。

3 模型设计

在这部分中，我们描述了我们提出的 Conv2Former 的架构，并提供了一些在模型设计和模块层调整方面的有用建议。

3.1 结构

总体结构。整体架构已在图 2中展示。类似于 ConvNeXt [20] 和 Swin Transformer 网络 [14]，我们的 Conv2Former 也采用了金字塔架构。总共有四个阶段，每个阶段具有不同的特征图分辨率。在两个连续阶段之间，使用块嵌入层（patch embedding）来降低分辨率，这通常是一个步长为 2 的 2×2 卷积。不同阶段具有不同数量的卷积块。我们构建了五个 Conv2Former 变体，即 Conv2Former-N、Conv2Former-T、Conv2Former-S、Conv2Former-B、Conv2Former-L。详情总结在表 1中。

阶段配置。当可学习的参数数量固定时，如何安排网络的宽度和深度对模型性能有影响 [17], [35]。原始的 ResNet-50 将每个阶段的块数设置为 $(3, 4, 6, 3)$。ConvNeXt-T 将块数更改为 $(3, 3, 9, 3)$，遵循 Swin-T 中使用的原则，并为更大的模型使用 $1:1:9:1$ 的阶段计算比率。不同地，我们稍微调整了比率，如表 1所示。我们观察到对于小型模型（参数少于 30M），更深的网络表现更好。可以在表 2中找到四种不同小型模型的简要比较。

3.2 卷积调制模块

我们在每个阶段使用的卷积块与 Transformers 有类似的结构，主要包含用于空间编码的自注意力层和用于通道混合的多层感知机（MLP）。不同之处在于，我们用一个简单的卷积调制层替换了自注意力层。

自注意力。对于长度为 N 的输入令牌序列 $\mathbf{X}$，自注意力机制首先使用线性层生成键 $\mathbf{K}$、查询 $\mathbf{Q}$ 和值 $\mathbf{V}$，其中 $\mathbf{X}, \mathbf{K}, \mathbf{Q}, \mathbf{V} \in \mathbb{R}^{N \times C}$，$N = H \times W$，$C$ 是通道数，H 和 W 是输入的空间尺寸。输出是基于相似度得分 $\mathbf{A}$ 的值的加权平均，

$$\text{Attention}(\mathbf{X}) = \mathbf{AV}, \tag{1}$$

其中 $\mathbf{A}$ 测量每一对输入令牌之间的关系，可以写作，

$$\mathbf{A} = \text{Softmax}(\mathbf{QK}^\top). \tag{2}$$

注意，为了简化，我们省略了缩放因子。尽管自注意力在编码空间信息方面具有高效率，相似度得分矩阵 $\mathbf{A}$ 的形状为 $\mathbb{R}^{N \times N}$，使得自注意力的计算复杂度随着序列长度 N 的增加而呈二次方增长。

卷积调制。在我们的卷积调制层中，不是通过方程 2 计算相似度得分矩阵 $\mathbf{A}$，我们通过用卷积特征调制值 $\mathbf{V}$ 来简化自注意力。具体来说，给定输入令牌 $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$，我们使用具有核大小 $k \times k$ 的简单深度卷积和哈达玛积（Hadamard product）来计算输出 $\mathbf{Z}$ 如下：

$$\mathbf{Z} = \mathbf{V} \odot \text{depthwiseConv}(\mathbf{X})$$

这里，depthwiseConv($\mathbf{X}$) 是对输入 $\mathbf{X}$ 应用深度卷积，得到与 $\mathbf{V}$ 相同尺寸的特征图，然后使用哈达玛积（逐元素相乘）与 $\mathbf{V}$ 结合，得到最终的输出 $\mathbf{Z}$。这种方法减少了计算复杂度，同时保持了特征的有效编码：

$$\mathbf{Z} = \mathbf{A} \odot \mathbf{V}, \tag{3}$$

$$\mathbf{A} = \mathrm{DConv}_{k\times k}(\mathbf{W}_1\mathbf{X}), \tag{4}$$

$$\mathbf{V} = \mathbf{W}_2\mathbf{X}, \tag{5}$$

其中 $\odot$ 表示哈达玛积，$\mathbf{W}_1$ 和 $\mathbf{W}_2$ 是两个线性层的权重矩阵，$\mathrm{DConv}_{k\times k}$ 表示核大小为 $k \times k$ 的深度卷积。

上述卷积调制操作使得每个空间位置 (h, w) 能够与以 (h, w) 为中心的 $k \times k$ 方形区域内的所有像素相关联。通道之间的信息交互可以通过线性层实现。每个空间位置的输出是方格区域内所有像素的加权和。

优势。与自注意力相比，我们的方法利用卷积来建立关系，这在处理高分辨率图像时比自注意力更加内存高效。与经典的残差块 [5], [20] 相比，我们的方法由于调制操作，也能够适应输入内容。

3.3 微设计

大于 7×7 的核。如何有效利用空间卷积对于卷积神经网络（ConvNet）的设计至关重要。自从 VGGNet [1] 和 ResNets [5], [6] 以来，3×3 卷积已成为构建卷积神经网络（ConvNets）的标准选择。后来，深度可分离卷积 [66] 的出现改变了这种状况。ConvNeXt 表明，将卷积神经网络（ConvNets）的核大小从 3×3 扩大到 7×7 可以提高分类性能。然而，没有重新参数化的情况下，进一步增加核大小几乎不会带来性能提升，反而会增加计算负担 [21], [67]。

我们认为，ConvNeXt 从大于 7×7 的核大小中获益甚微的原因在于使用空间卷积的方式。对于 Conv2Former，我们观察到，随着核大小从 5×5 增加到 21×21，性能持续提升。这一现象不仅在 Conv2Former-T（从 82.8% 提升到 83.4%）中发生，而且在拥有超过 80M 参数的 Conv2Former-B 中也成立（从 84.1% 提升到 84.5%）。考虑到模型效率，我们默认将核大小设置为 11×11。

权重策略。如图 1所示，我们考虑将深度卷积的输出作为权重，以调制线性投影后的特征。值得注意的是，我们在哈达玛积之前不使用激活函数或归一化层（例如，Sigmoid 或 L_p 归一化）。这是获得良好性能的一个关键因素。例如，如 SENet [9] 中所做的那样添加 Sigmoid 函数会将性能降低超过 0.5%。

我们想强调的是，FocalNet [69] 采用了与我们相似的权重策略，但其动机不同。FocalNet 旨在通过 3×3 深度卷积和全局平均池化来提取多级特征，以实现层次化上下文聚合。不同地，我们尝试通过利用简单的大核卷积来简化自注意力操作，并研究一种有效的方式，使大核卷积在卷积神经网络中发挥作用。我们的方法比 FocalNet 简单得多，实验表明 Conv2Former 比 FocalNet 有优势。

归一化和激活。对于归一化层，我们遵循原始的 ViT 和 ConvNeXt，采用层归一化 [71] 而不是广泛使用的批量归一化 [72]。对于激活层，我们使用 GELU [73]。我们发现，层归一化和 GELU 的组合为我们的 Conv2Former 带来了 0.1%-0.2% 的性能提升。

表 3
在 ImageNet [27] 上的 Top-1 准确率结果比较。与之前的流行 Transformer 和 ConvNets 相比，我们的 Conv2Former 在不同模型大小的网络变体上取得了令人惊讶的好结果。

Model	#Params	FLOPs	Image Size	Top-1 Acc.
VAN-B1 [22]	14M	2.5G	224×224	81.1%
★ Conv2Former-N	15M	2.2G	224×224	81.5%
ResNet50-d [5], [68]	26M	4.3G	224×224	79.5%
SwinT-T [14]	28M	4.5G	224×224	81.5%
DWNet-T [32]	26M	4.4G	224×224	81.8%
ConvNeXt-T [20]	29M	4.5G	224×224	82.1%
VAN-B2 [22]	27M	5.0G	224×224	82.8%
★ Conv2Former-T	27M	4.4G	224×224	83.2%
SwinT-S [14]	50M	8.7G	224×224	83.0%
ConvNeXt-S [20]	50M	8.7G	224×224	83.1%
VAN-B3 [22]	45M	9.0G	224×224	83.9%
NFNet-F0 [35]	72M	12.4G	256×256	83.6%
★ Conv2Former-S	50M	8.7G	224×224	84.1%
DeiT-B [13]	86M	17.5G	224×224	81.8%
DWNet-B [32]	80M	14.3G	224×224	83.4%
RepLKNet-31B [21]	79M	15.3G	224×224	83.5%
SwinT-B [14]	88M	15.4G	224×224	83.5%
ConvNeXt-B [20]	89M	15.4G	224×224	83.8%
FocalNet-B [69]	89M	15.4G	224×224	83.9%
SLak-B [25]	95M	17.1G	224×224	84.0%
MOAT-2 [70]	73M	17.2G	224×224	84.2%
EffNet-B7 [51]	66M	37.0G	600×600	84.3%
★ Conv2Former-B	90M	15.9G	224×224	84.4%

4 实验

4.1 实验设置

数据集。我们评估了提出的 Conv2Former 在广泛使用的数据集 ImageNet-1k [27] 上的分类性能，该数据集包含约 120 万张训练图像和 1,000 个不同类别。我们在总共有 5 万张图像的验证集上报告结果。像一些其他流行的模型 [14], [20] 一样，我们还使用大规模的 ImageNet-22k 数据集对提出的 Conv2Former 进行预训练，以测试其扩展能力，该数据集包

表 4
在 ImageNet 数据集上，使用 ImageNet-22k 数据集进行预训练的 Top-1 准确率结果。我们可以观察到与 ConvNeXt 相比的持续改进。我们的 Conv2Former-L 也比 EfficientNetV2-XL 和 CoAtNet-3 表现得更好。

Model	#Params	FLOPs	Image Size	Top-1 Acc.
ConvNeXt-S [20]	50M	8.7G	224×224	84.6%
★ Conv2Former-S	50M	8.7G	224×224	84.9%
SwinT-B [14]	88M	15.4G	224×224	85.2%
ConvNeXt-B [20]	89M	15.4G	224×224	85.8%
★ Conv2Former-B	90M	15.9G	224×224	86.2%
SwinT-B [14]	88M	47.0G	384×384	86.4%
ConvNeXt-B [20]	89M	45.1G	384×384	86.8%
★ Conv2Former-B	90M	46.7G	384×384	87.0%
EffNet-V2-XL [18]	208M	94.0G	480×480	87.3%
SwinT-L [14]	197M	34.5G	224×224	86.3%
ConvNeXt-L [20]	198M	34.4G	224×224	86.6%
★ Conv2Former-L	199M	36.0G	224×224	87.0%
SwinT-L [14]	197M	104G	384×384	87.3%
ConvNeXt-L [20]	198M	101G	384×384	87.5%
CoAtNet-3 [43]	168M	107G	384×384	87.6%
★ Conv2Former-L	199M	105.9G	384×384	87.7%

含约 1400 万张图像和 21,841 个类别。预训练后，我们使用 ImageNet-1k 数据集进行微调，并在 ImageNet-1k 验证集上报告结果。

训练设置。我们基于 `PyTorch` [74] 实现了我们的模型。在训练期间，我们使用 AdamW 优化器 [75] 并采用线性学习率缩放策略 $lr = \mathrm{LR}_{\mathrm{base}} \times \mathrm{batch\_size}/1024$。初始学习率 $\mathrm{LR}_{\mathrm{base}}$ 设置为 0.001，权重衰减率设置为 5×10^{-2}，如之前工作 [20] 中建议的。在 ImageNet 上的整个实验中，我们随机裁剪图像大小为 224×224，并采用一些常见的数据增强方法，例如 MixUp [76] 和 CutMix [77]。我们还使用了随机深度 [78]，随机擦除 [79]，标签平滑 [3]，RandAug [80]，以及初始值为 1×10^{-6} 的层尺度 [46]。我们训练所有模型 300 个周期。对于 ImageNet-22k 上的实验，我们首先在这个数据集上预训练我们的模型 90 个周期，然后按照 ConvNeXt [20] 的做法，在 ImageNet-1k 上进行 30 个周期的微调。

4.2 与其他方法的比较

我们将我们的 Conv2Former 与一些流行的网络架构进行比较，包括 Swin Transformer [14]、ResNet [5]、ConvNeXt [20]、NFNet [35]、DeiT [13]、DWNet [32]、FocalNet [69]、VAN [22]、SLak [25]、EfficientNets [17], [18]、CoAtNet [43]、RepLKNet [21], 和 MOAT [70]。请注意，其中一些是 CNN 和 Transformer 的混合模型。

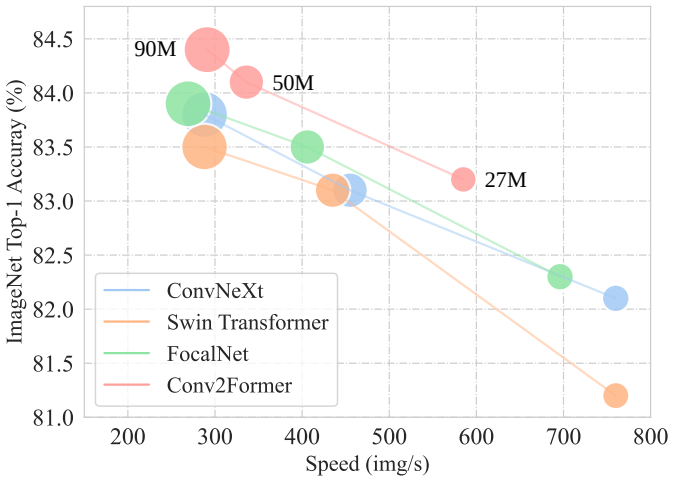

图 3. ImageNet 分类准确率与推理速度的对比。我们可以看到，我们提出的 Conv2Former 在两者之间取得了最佳的平衡。推理速度是在 NVIDIA V100 GPU 上测试的。

ImageNet-1k。我们首先在 ImageNet-1k 数据集上训练我们的 Conv2Former，并在表 3中展示结果。对于小型模型（小于 30M 参数），我们的 Conv2Former 与 ConvNeXt-T 和 SwinT-T 相比，性能分别提高了 1.1% 和 1.7%。即使是我们的 Conv2Former-N，拥有 15M 参数和 2.2G FLOPs，也与拥有 28M 参数和 4.5G FLOPs 的 SwinT-T 表现相当。对于基础模型，性能提升有所减少，但与 ConvNeXt-B 和 SwinT-B 相比，仍然分别提高了 0.6% 和 0.9%。与其它流行模型相比，我们的 Conv2Former 在类似模型大小的情况下也表现更好。特别是，我们的 Conv2Former 在小型和基础型模型上都取得了比 DWNet [32] 更好的结果，后者也尝试将局部自注意力和深度卷积连接起来。值得注意的是，我们的 Conv2Former-B 甚至比 EfficientNet-B7 表现得更好（84.4% 对比 84.3%），其计算量是我们的两倍（37G 对比 15G）。

ImageNet-22k。我们在大型的 ImageNet-22k 数据集上对 Conv2Former 进行预训练，然后在 ImageNet-1k 数据集上进行微调。这个实验可以反映我们的 Conv2Former 在数据扩展能力上的表现。对于所有实验，我们遵循 [20] 中使用的设置来训练和微调模型。结果已在表 4中列出。与 ConvNeXt 的不同变体相比，我们的 Conv2Former 在模型大小相似时都表现得更好。此外，我们可以看到，在更高分辨率（384×384）上进行微调时，我们的 Conv2Former-L 比 CoAtNet 这样的混合模型获得了更好的结果。我们的 Conv2Former-L 实现了最好的结果，达到了 87.7%。

计时。 在这里，我们比较了我们的 Conv2Former 与三个流行的模型——ConvNeXt、Swin Transformer 和 FocalNet——在推理速度方面的表现。我们在图 3中展示了所有四种方法的准确率-速度曲线。我们可以看到，我们的 Conv2Former 在 ImageNet 上实现了分类准确率和推理速度之间的最佳权衡。

有效感受野分析。 为了展示为什么我们提出的 Conv2Former

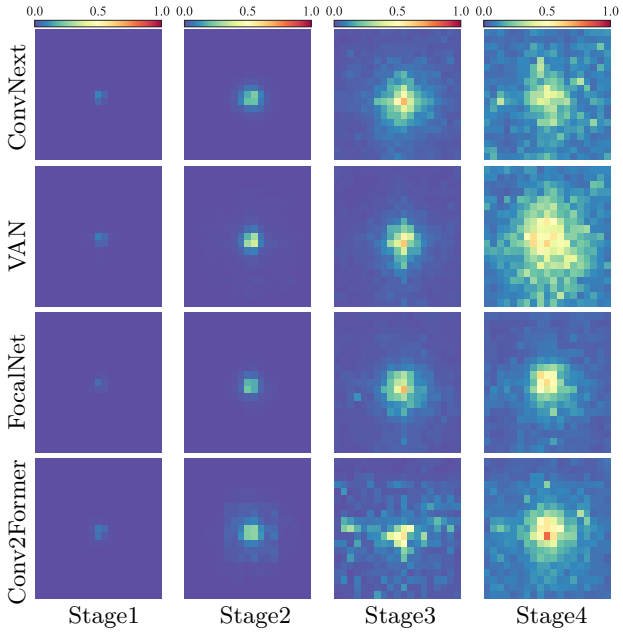

图 4. 有效的感受野可视化，针对四种不同的基于 CNN 的方法，展示了四个阶段。

表 5
对四种基于 CNN 的方法的有效感受野进行统计分析。“平均值” 表示在不同阈值（从 0.5 到 0.9，步长为 0.05）下的平均结果。无论是 “边长” 还是 “面积比”，我们的 Conv2Former 都取得了最佳结果。

Model	Side Length			Area Ratio (%)		
	0.5	0.75	Mean	0.5	0.75	Mean
ConvNeXt-T	70.0	102.5	96.6	14.8	30.4	27.9
VAN-B1	62.0	115.5	104.8	10.9	31.5	28.5
FocalNet-T	59.0	100.0	94.2	10.0	27.0	25.2
Conv2Former-T	75.0	113.5	**106.7**	15.4	33.4	**30.5**

比最近的最先进的基于 CNN 的方法效果更好，我们分析了不同方法产生的有效感受野（ERFs），如表 5所示。我们使用两个阈值 0.5 和 0.75 来分别计算边长和面积比。此外，我们还计算了在 0.5 到 0.9 不同阈值下，步长为 0.05 的平均结果。我们可以看到，我们的 Conv2Former 在边长方面的表现优于 FocalNet、VAN 和 ConvNeXt。对于面积比，我们的 Conv2Former 取得了最好的结果。

如图 4所示，我们还尝试可视化四个模型不同阶段的有效感受野（ERFs）。可以清楚地看到，在第二和第三阶段，我们的 Conv2Former 也可以比其他三种方法拥有更大的 ERFs。我们认为，这主要是因为所提出的卷积调制块能够以更合适的方式编码空间信息。它使得我们的 Conv2Former 的中间层也能够捕获到大的 ERFs。因此，我们的 Conv2Former，即使采用简单的网络架构，也能产生最先进的结果。

与其他大核方法的优势。使用大核卷积是帮助 CNN 建立长距离关系的一种直接方式。然而，直接在现有的基于 CNN 的架构中使用大核卷积（大于 7×7）会使识别模型难以优化 [20], [26]。最近，有一些工作旨在开发新技术，以在 CNN 中激发大核卷积的利用。在表 6中，我们展示了最近具有不同核大小的 state-of-the-art ConvNets 的结果。我们可以看到，没有任何其他训练技术，如重新参数化或使用稀疏权重，我们的 Conv2Former 即使使用 7×7 的核大小，在基础模型设置下也已经比其他方法表现更好。使用更大的核大小 11×11 可以获得更好的性能提升。这些结果反映了我们卷积调制块的优势。

表 6
与最近不同核大小的最先进的 ConvNets 的比较。我们可以看到，没有任何其他训练技术，如重新参数化或使用稀疏权重，我们的 Conv2Former 使用 11×11 的核大小取得了最佳结果。这些实验表明，我们的卷积调制操作可以更有效地编码空间信息。

Model	Kernel size	#Params	FLOPs	Acc.
RepLKNet-31B [21]	31×31	79M	15.3G	83.5%
ConvNeXt-B [20]	7×7	89M	15.4G	83.8%
SLaK-B [25]	51×51	95M	17.1G	84.0%
★ Conv2Former-B	7×7	89M	15.6G	84.2%
★ Conv2Former-B	11×11	90M	15.9G	84.4%

表 7
在使用我们卷积调制块中不同融合策略时的性能比较。所有结果均基于 Conv2Former-T。我们可以看到，使用简单的哈达玛积（Hadamard product）可以获得最佳结果。

Weighting Strategy	Top-1 Acc.
Element-wise sum	82.7%
Adding a Sigmoid function after $\mathbf{A}$	82.3%
Adding an L_1 normalization after $\mathbf{A}$	82.8%
Linearly normalizing the values of $\mathbf{A}$ to $(0, 1]$	82.2%
★ Hadamard product	83.2%

4.3 方法分析

在这一部分，我们对提出的卷积调制操作进行了一系列方法分析。

核大小。ConvNeXt 的工作 [20] 表明，当深度卷积的核大小超过 7×7 时，性能没有提升。在这里，我们研究了使用更大的核大小时模型性能会如何变化。我们为深度卷积选择了 6 种不同的核，即 $\{5\times5, 7\times7, 9\times9, 11\times11, 15\times15, 21\times21\}$，并展示了基于两种模型变体 Conv2Former-T 和 Conv2Former-B 的结果。结果可以在图 5(a) 中找到。性能增益似乎在核大小增加到 21×21 时趋于饱和。这个结果与 ConvNeXt 得出的结论大相径庭，ConvNeXt 认为使用大于 7×7 的核不会带来明显的性能提升。这表明，如公式 Eqn. 3中所制定的，将卷积特征作为权重使用，可以比传统方法 [5], [20] 更有效地利用大核。

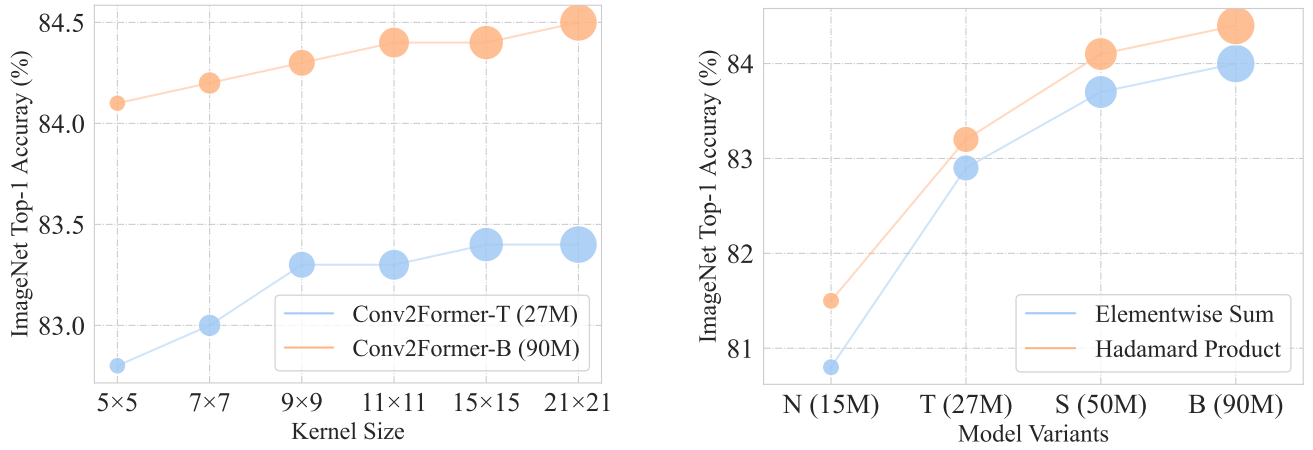

图 5. 消融实验。对于 Conv2Former-T 和 Conv2Former-B，我们可以看到，当核大小从 5×5 增加到 21×21 时，性能一致提高。当将哈达玛积（Hadamard product）替换为逐元素求和操作时，我们 Conv2Former 的所有四种变体的性能都有所下降。

哈达玛积优于求和。如图 1所示，我们使用深度卷积提取的卷积特征通过哈达玛积操作来调制右侧线性分支的权重。在我们的实验中，我们还尝试利用逐元素求和来融合两个分支。图 5(b) 展示了在不同模型大小下我们 Conv2Former 的比较结果。哈达玛积的表现优于逐元素求和，这表明卷积调制在编码空间信息方面比求和更有效。我们还可以观察到，小型模型从哈达玛积中获益更多。

权重策略。除了前述的两种融合策略外，我们还尝试使用其他方式来融合特征图，包括在 $\mathbf{A}$ 之后添加 Sigmoid 函数，对 $\mathbf{A}$ 应用 L_1 归一化，以及将 $\mathbf{A}$ 的值线性归一化到 $(0, 1]$。结果总结在表 7中。我们可以看到，哈达玛积比其他所有操作都带来了更好的结果。更有趣的是，当使用 Sigmoid 函数或线性归一化到 $(0, 1]$ 来调整 $\mathbf{A}$ 的值为正值时，性能下降得更多。这与传统的注意力机制不同，如 SE [9] 和 CA [11] 在重新加权前利用 Sigmoid 函数。我们将这个问题留作未来的研究。

4.4 可视化分析

特征可视化。为了进一步展示我们提出的方法相对于最近的最先进模型（如 FocalNet 和 ConvNeXt）的有效性，我们使用 Grad-CAM [81] 来可视化不同模型产生的特征图。可视化结果在图 6中展示。我们可以看到，与其它三个模型相比，我们的 Conv2Former 能更准确地定位目标对象。特别是，对于形状细长的对象（见上图两行），我们的 Conv2Former 也能精确捕捉到它们。这使得我们的 Conv2Former 在识别方面比其他模型更优。

4.5 在等向性模型到 ViT 上的结果

与采用层次化架构的经典 CNN 不同，原始的 ViT [12], [13] 由于沉重的自注意力层，采用了一种简单的架构，该架构包含一个补丁嵌入层和一系列具有相同序列长度的 Transformers。

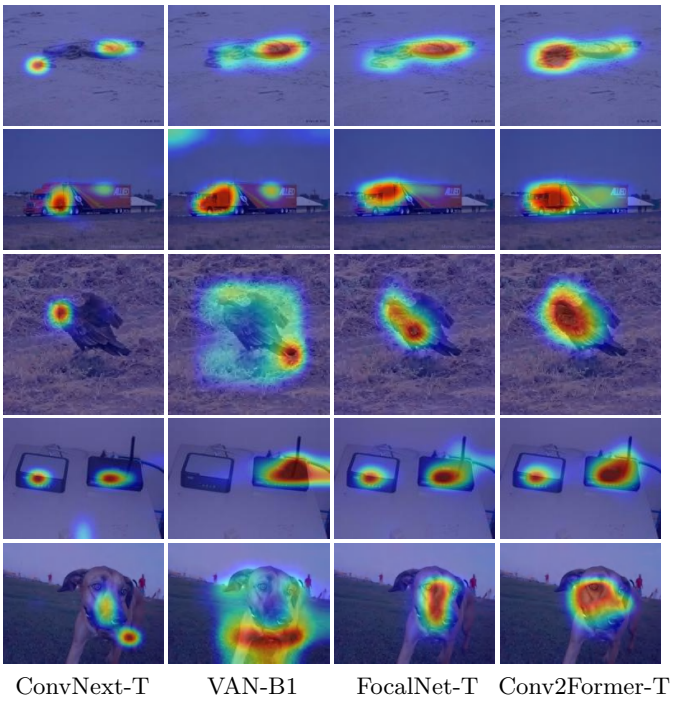

图 6. 四种方法的特征可视化。我们采用广泛使用的 Grad-CAM [81] 作为我们的可视化工具。我们可以看到，与其它三种基于 CNN 的方法相比，我们的方法能更准确地定位目标对象。

这种简单的架构已在最近的 Transformers 研究工作中被广泛使用。在这里，我们跟随 ConvNeXt [20]，也尝试研究在 ViT 风格的架构设置下 Conv2Former 的性能。与 ConvNeXt 类似，我们将 Conv2Former-IS 和 Conv2Former-IB 的块数设置为 18，并调整通道数以匹配模型大小。我们使用了两种版本的补丁嵌入模块：一种是具有 16 步长的 16×16 卷积，另一种是如 [45] 中所做的三个卷积。

Tab. 8 展示了结果。我们将 DeiT-S 和 DeiT-B 模型作为基线。为了简洁，我们在模型名称中添加字母 “I”，表示相应的

表 8
我们的等向性 Conv2Former、ConvNeXt 和 ViT 之间的比较。"3 Convs" 意味着我们像 [43], [45], [48] 中所做的那样，在网络的开始处使用三个卷积层进行补丁嵌入。对于小型和基础型模型，我们的 Conv2Former 在参数和计算量与其他方法相当的情况下取得了更好的结果。

Model	Patch Embed	#Params	FLOPs	Top-1 Acc.
DeiT-S	1 Conv	22M	4.6G	79.8%
ConvNeXt-IS	1 Conv	22M	4.3G	79.7%
★ Conv2Former-IS	1 Conv	23M	4.3G	81.2%
★ Conv2Former-IS	3 Convs	23M	4.5G	82.0%
DeiT-B	1 Conv	87M	17.6G	81.8%
ConvNeXt-IB	1 Conv	87M	16.9G	82.0%
★ Conv2Former-IB	1 Conv	86M	16.5G	82.7%
★ Conv2Former-IB	3 Convs	87M	17.3G	83.0%

模型使用与原始 ViT 相同的等向性架构。我们可以看到，对于大约有 2200 万参数的小型模型，我们的 Conv2Former-IS 的表现远远好于 DeiT-S 和 ConvNeXt-IS。性能提升约为 1.5%。当模型大小增加到 8000 万 + 时，我们的 Conv2Former-IB 达到了 82.7% 的 Top-1 准确率，这也比 ConvNeXt-IB 高出 0.7%，比 DeiT-B 高出 0.9%。此外，使用三个卷积层进行补丁嵌入可以进一步改善结果。

4.6 在下游任务上的结果

在这一部分，我们评估了我们的方法在两个下游任务上的表现，包括在 COCO [28] 上的目标检测和 ADE20k [29] 上的语义分割。

COCO 上的结果。按照之前的作品 [14], [20]，我们使用两个流行的目标检测器，Mask R-CNN [82] 和 Cascade Mask R-CNN [83] 进行实验，并报告目标检测和实例分割的结果。对于训练，我们遵循 ConvNeXt [20] 中使用的实验设置，包括多尺度训练，AdamW 优化器与 3× 学习计划，GIoU 损失 [84] 等。读者可以参考 [20], [85] 了解更多详细的实验设置。我们使用 MMDetection 工具箱 [86] 来运行所有的目标检测实验。

结果可以在表 9中找到。对于小型模型，我们的 Conv2Former-T 在使用 Mask R-CNN 框架进行目标检测时，比 SwinT-T 和 ConvNeXt-T 实现了大约 2% 的 AP 提升。对于实例分割，性能提升也超过 1%。当使用 Cascade Mask R-CNN 框架时，我们可以看到比 SwinT-T 和 ConvNeXt-T 超过 1% 的性能提升。当扩大模型规模时，改进也是明显的。

ADE20k 上的结果。遵循 [14], [20]，我们使用训练集训练模型，并在验证集上报告结果。对于小型、基础型模型，我们随机裁剪图像至 512×512，对于大型模型，我们将图像裁剪至 640×640。我们使用 UperNet [87] 作为我们的解码器。

结果总结在表 10中。对于不同规模的模型，我们的 Conv2Former 都能超越 Swin Transformer 和 ConvNeXt。值得注意的是，在小型规模上与 ConvNeXt 相比有 1.3% 的 mIoU 提升，而在基础规模上提升为 1.1%。当我们进一步增加模型规模时，我们的 Conv2Former-L 结合 UperNet 达到了 54.3% 的 mIoU 得分，这也明显优于 Swin-L 和 ConvNeXt-L。

表 9
使用 Mask R-CNN [82] 和 Cascade Mask R-CNN [83] 在 COCO [28] 上的目标检测和实例分割结果。我们使用 ImageNet-1k 预训练的骨干网络。

Model	FLOPs	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Mask R-CNN [82] 3× schedule							
SwinT-T	267G	46.0	68.1	50.3	41.6	65.1	44.9
ConvNeXt-T	262G	46.2	67.9	50.8	41.7	65.0	44.9
★ Conv2Former-T	255G	48.0	69.5	52.7	43.0	66.8	46.1
Cascade Mask R-CNN [83] 3× schedule							
SwinT-T	743G	50.4	69.2	54.7	43.7	66.6	47.3
ConvNeXt-T	741G	50.4	69.1	54.8	43.7	66.5	47.3
SLaK-T	841G	51.3	70.0	55.7	44.3	67.2	48.1
★ Conv2Former-T	734G	51.4	69.8	55.9	44.5	67.4	48.3
SwinT-S	833G	51.9	70.7	56.3	45.0	68.2	48.8
ConvNeXt-S	827G	51.9	70.8	56.5	45.0	68.4	49.1
★ Conv2Former-S	823G	52.8	71.4	57.3	45.7	69.0	49.8
SwinT-B	975G	51.9	70.5	56.4	45.0	68.1	48.9
ConvNeXt-B	964G	52.7	71.3	57.5	45.6	69.0	49.8
★ Conv2Former-B	968G	52.8	71.1	57.2	45.6	68.7	49.3

表 10
与 Swin-T 和 ConvNeXt 在 ADE20k [29] 上的比较。我们使用 UperNet [87] 作为解码器。在所有模型规模上，我们的 Conv2Former 都取得了最佳结果。

Model	Crop Size	#Params	FLOPs	mIoU (%)
ImageNet-1K pre-trained				
SwinT-T	512^2	59M	946G	45.8
ConvNeXt-T	512^2	59M	940G	46.7
★ Conv2Former-T	512^2	55M	931G	48.0
SwinT-S	512^2	80M	1039G	49.5
ConvNeXt-S	512^2	81M	1024G	49.6
★ Conv2Former-S	512^2	78M	1021G	50.3
SwinT-B	512^2	120M	1189G	49.7
ConvNeXt-B	512^2	121M	1166G	49.9
★ Conv2Former-B	512^2	119M	1171G	51.0
ImageNet-22K pre-trained				
SwinT-L	640^2	232M	2479G	53.5
ConvNeXt-L	640^2	233M	2453G	53.7
★ Conv2Former-L	640^2	230M	2483G	54.3

5 结论与讨论

本文介绍了 Conv2Former，这是一种新的用于视觉识别的卷积网络架构。我们 Conv2Former 的核心是卷积调制操作，它通过仅使用卷积和哈达玛积来简化自注意力机制。我们展示了我们的卷积调制操作是一种更有效的方式，以利用大核卷积。我们在 ImageNet 分类、目标检测和语义分割中的实验也表明，我们提出的 Conv2Former 比以往的基于 CNN 的模型和大多数基于 Transformer 的模型表现更好。我们相信，提高 ConvNets 的性能还有很大的空间，我们希望我们的方法能为未来的 ConvNets 研究提供洞见。

参考文献

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1–9.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2818–2826.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI Conf. on Artif. Intel.*, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[6] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1492–1500.

[7] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, 2021.

[8] D. Zhou, Q. Hou, Y. Chen, J. Feng, and S. Yan, "Rethinking bottleneck structure for efficient mobile network design," in *Eur. Conf. Comput. Vis.* Springer, 2020, pp. 680–697.

[9] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7132–7141.

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

[11] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 13 713–13 722.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Int. Conf. Learn. Represent.*, 2020.

[13] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020.

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Int. Conf. Comput. Vis.*, 2021.

[15] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Int. Conf. Comput. Vis.*, 2021.

[16] L. Yuan, Q. Hou, Z. Jiang, J. Feng, and S. Yan, "Volo: Vision outlooker for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

[17] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 6105–6114.

[18] ——, "Efficientnetv2: Smaller models and faster training," in *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 10 096–10 106.

[19] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 16 519–16 529.

[20] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[21] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31x31: Revisiting large kernel design in cnns," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 11 963–11 975.

[22] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *arXiv preprint arXiv:2202.09741*, 2022.

[23] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, and S.-M. Hu, "Segnext: Rethinking convolutional attention design for semantic segmentation," in *Adv. Neural Inform. Process. Syst.*, 2022.

[24] Y. Rao, W. Zhao, Y. Tang, J. Zhou, S.-N. Lim, and J. Lu, "Hornet: Efficient high-order spatial interactions with recursive gated convolutions," *arXiv preprint arXiv:2207.14284*, 2022.

[25] S. Liu, T. Chen, X. Chen, X. Chen, Q. Xiao, B. Wu, M. Pechenizkiy, D. Mocanu, and Z. Wang, "More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity," *arXiv preprint arXiv:2207.03620*, 2022.

[26] M. Tan and Q. V. Le, "Mixconv: Mixed depthwise convolutional kernels," *arXiv preprint arXiv:1907.09595*, 2019.

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

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Eur. Conf. Comput. Vis.* Springer, 2014, pp. 740–755.

[29] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 633–641.

[30] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.

[31] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[32] Q. Han, Z. Fan, Q. Dai, L. Sun, M.-M. Cheng, J. Liu, and J. Wang, "On the connection between local atten-

tion and dynamic depth-wise convolution," *arXiv preprint arXiv:2106.04263*, 2021.

[33] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu *et al.*, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," *Adv. Neural Inform. Process. Syst.*, vol. 32, pp. 103–112, 2019.

[34] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 819–828.

[35] A. Brock, S. De, S. L. Smith, and K. Simonyan, "High-performance large-scale image recognition without normalization," *arXiv preprint arXiv:2102.06171*, 2021.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inform. Process. Syst.*, 2017, pp. 5998–6008.

[37] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," *arXiv preprint arXiv:2103.16302*, 2021.

[38] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," *arXiv preprint arXiv:2107.00641*, 2021.

[39] C.-F. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," *arXiv preprint arXiv:2103.14899*, 2021.

[40] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *arXiv preprint arXiv:2103.00112*, 2021.

[41] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, "Scaling local self-attention for parameter efficient visual backbones," *arXiv preprint arXiv:2103.12731*, 2021.

[42] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "Cmt: Convolutional neural networks meet vision transformers," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 12 175–12 185.

[43] Z. Dai, H. Liu, Q. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.

[44] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Int. Conf. Comput. Vis.*, October 2021, pp. 558–567.

[45] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, and J. Feng, "Deepvit: Towards deeper vision transformer," *arXiv preprint arXiv:2103.11886*, 2021.

[46] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 32–42.

[47] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 12 009–12 019.

[48] Z.-H. Jiang, Q. Hou, L. Yuan, D. Zhou, Y. Shi, X. Jin, A. Wang, and J. Feng, "All tokens matter: Token labeling for training better vision transformers," *Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 18 590–18 602, 2021.

[49] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-scale conv-attentional image transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 9981–9990.

[50] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[51] H. Cai, C. Gan, and S. Han, "Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition," *arXiv preprint arXiv:2205.14756*, 2022.

[52] M. Maaz, A. Shaker, H. Cholakkal, S. Khan, S. W. Zamir, R. M. Anwer, and F. S. Khan, "Edgenext: Efficiently amalgamated cnn-transformer architecture for mobile vision applications," *arXiv preprint arXiv:2206.10589*, 2022.

[53] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu, "Mobile-former: Bridging mobilenet and transformer," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 5270–5279.

[54] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *arXiv preprint arXiv:1906.05909*, 2019.

[55] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Global context networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[56] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7794–7803.

[57] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3588–3597.

[58] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," *Adv. Neural Inform. Process. Syst.*, vol. 31, 2018.

[59] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.

[60] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek *et al.*, "Resmlp: Feedforward networks for image classification with data-efficient training," *arXiv preprint arXiv:2105.03404*, 2021.

[61] Q. Hou, Z. Jiang, L. Yuan, M.-M. Cheng, S. Yan, and J. Feng, "Vision permutator: A permutable mlp-like architecture for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

[62] J. Wang, S. Zhang, Y. Liu, T. Wu, Y. Yang, X. Liu, K. Chen, P. Luo, and D. Lin, "Riformer: Keep your vision backbone effective but removing token mixer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 443–14 452.

[63] X. Hu, M. Shi, W. Wang, S. Wu, L. Xing, W. Wang, X. Zhu, L. Lu, J. Zhou, X. Wang, Y. Qiao, and J. Dai, "Demystify transformers & convolutions in modern image deep networks," *arXiv preprint arXiv:2211.05781*, 2022.

[64] Z. Pan, J. Cai, and B. Zhuang, "Fast vision transformers with hilo attention," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14 541–14 554, 2022.

[65] W. Lin, Z. Wu, J. Chen, J. Huang, and L. Jin, "Scale-aware modulation meet transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6015–6026.

[66] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1251–1258.

[67] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 13 733–13 742.

[68] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 558–567.

[69] J. Yang, C. Li, and J. Gao, "Focal modulation networks," *arXiv preprint arXiv:2203.11926*, 2022.

[70] C. Yang, S. Qiao, Q. Yu, X. Yuan, Y. Zhu, A. Yuille, H. Adam, and L.-C. Chen, "Moat: Alternating mobile convolution and attention brings strong vision models," *arXiv preprint arXiv:2210.01820*, 2022.

[71] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

[72] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*. PMLR, 2015, pp. 448–456.

[73] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[74] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inform. Process. Syst.*, 2019, pp. 8026–8037.

[75] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

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

[77] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Int. Conf. Comput. Vis.*, 2019, pp. 6023–6032.

[78] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Eur. Conf. Comput. Vis.* Springer, 2016, pp. 646–661.

[79] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *AAAI Conf. on Artif. Intel.*, 2020, pp. 13 001–13 008.

[80] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2020, pp. 702–703.

[81] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

[82] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Int. Conf. Comput. Vis.*, Oct 2017.

[83] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[84] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 658–666.

[85] H. Bao, L. Dong, S. Piao, and F. Wei, "BEit: BERT pre-training of image transformers," in *Int. Conf. Learn. Represent.*, 2022.

[86] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[87] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Eur. Conf. Comput. Vis.*, 2018, pp. 418–434.

侯淇彬 在南开大学计算机科学与工程学院获得了博士学位。之后，他在新加坡国立大学担任研究员。现在，他是南开大学计算机科学与工程学院的副教授。他在包括 T-PAMI、CVPR、ICCV、NeurIPS 等顶级会议/期刊上发表了 20 多篇论文。他的研究兴趣包括深度学习、图像处理和计算机视觉。

陆承泽 目前是南开大学计算机科学学院的硕士研究生，导师为程明明教授。在此之前，他于 2020 年在西安电子科技大学获得了工学学士学位。他的研究兴趣包括深度学习和计算机视觉。

程明明 于 2012 年在清华大学获得博士学位。之后，他在牛津大学与 Philip Torr 教授共事，担任了两年的研究员。他现在在南开大学担任教授，并领导媒体计算实验室。他的研究兴趣包括计算机图形学、计算机视觉和图像处理。他获得的研究奖项包括 ACM 中国新星奖、IBM 全球 SUR 奖、CCF-Intel 青年教师研究计划等。

冯佳时 目前在字节跳动担任研究科学家。在加入字节跳动之前，他是新加坡国立大学电气与计算机工程系的助理教授，以及加州大学伯克利分校 EECS 系和 ICSI 的博士后研究员。他于 2014 年在新加坡国立大学获得博士学位。他的研究领域包括深度学习及其在计算机视觉中的应用。他最近的研究兴趣集中在深度学习模型、表示学习和 3D 视觉上。他曾获得 ACM MM 2012 最佳技术演示奖、TASK-CV ICCV 2015 最佳论文奖、ACM MM 2018 最佳学生论文奖。他还是 2018 年麻省理工学院技术评论 "35 岁以下创新者" 亚洲区的获奖者。他曾担任 NeurIPS、ICML、CVPR、ICLR、WACV 的区域主席，并担任 ICMR 2017 的项目主席。