基于移动窗口的层级式 Vision Transformer

夏浩

摘要

本文提出了一种新的 Vision Transformer，叫做 Swin Transformer，可以作为计算机视觉的一个通用骨干网络。将自然语言处理领域的 Transformer 使用到计算机视觉领域有一些挑战，比如不同的图片之前尺寸的不一致，以及图片中的像素数要比句子中的单词数高几个量级。为了解决这些问题，本文提出了一种层级式的 Transformer，它在非重叠的局部窗口内计算注意力，相对于全局注意力机制计算复杂度大大降低，此外，为了计算相邻的非重叠局部窗口之间的注意力，本文提出了移动窗口机制（**S**hifted **win**dows）。本文在多个视觉任务上取得了 SOTA 的成绩，包括图像分类任务（87.3 top-1 accuracy on ImageNet-1K）和语义分割任务（53.5 mIoU on ADE20K val（+3.2））。

关键词：计算机视觉；Transformer；骨干网络；移动窗口

1 引言

在计算机视觉领域，卷积神经网络（CNN）长期处于统治地位。自从第一个现代深度卷积网络模型 AlexNet[1]在 ImageNet 分类竞赛中取得最高分数之后，CNN 架构如雨后春笋一般快速涌现，比如 VGGNet[2]使用了更小的卷积核来加速运算，GoogLeNet[3]通过增加网络宽度提升效果，ResNet[4]通过残差学习的思想解决梯度消失或者爆炸的问题，将 CNN 网络的深度提升至上千层。由于 CNN 具有局部性和平移不变性等良好的先验知识，将其作为视觉任务的骨干网络，可以取得良好的结果。

另一方面，自然语言处理领域（NLP）的骨干网络与计算机视觉领域有所不同，由于 Transformer[5]可以使用注意力机制来建模数据中的长期依赖关系[6]，同时可以并行加速计算，这使得 Transformer 在 NLP 领域逐渐处于统治地位。由于 Transformer 在自然语言处理领域的巨大成功，自然地，研究人员考虑将其使用到计算机视觉领域，比如 ViT[7]将其简易地迁移至图片分类任务。

本文进一步探索了 Transformer 在计算机视觉领域的能力，使其可以像 CNN 一样作为 CV 领域的骨干网络。我们提到 CV 领域与 NLP 领域的差异主要在于两处：第一，为了保障通用性，计算机视觉任务通常要求模型可扩展到多种尺寸的图片上[8]，也就是不同于 NLP 领域一句话中的词元个数通常是固定的[5]。第二，图片的分辨率相对于一段话中的单词个数往往要大几个量级，这就导致将 Transformer 应用于视觉领域需要付出巨大的计算复杂度，这使得 Transformer 很难应用到如语义分割等密集预测型视觉任务。为了克服这些问题，我们提出了一个通用视觉骨干网络，叫做 Swin Transformer，它可以构建分层特征图，并且计算复杂度与图像大小呈线性相关。如图 1(a) 所示，Swin Transformer 通过从小尺寸的图块开始，然后通过合并图块逐渐构建层次表达。有了这些分层特征图之后，Swin Transformer 模型可以方便地利用 FPN[8]或者 FCN[9]来完成目标检测或者图像分割等下游任务。

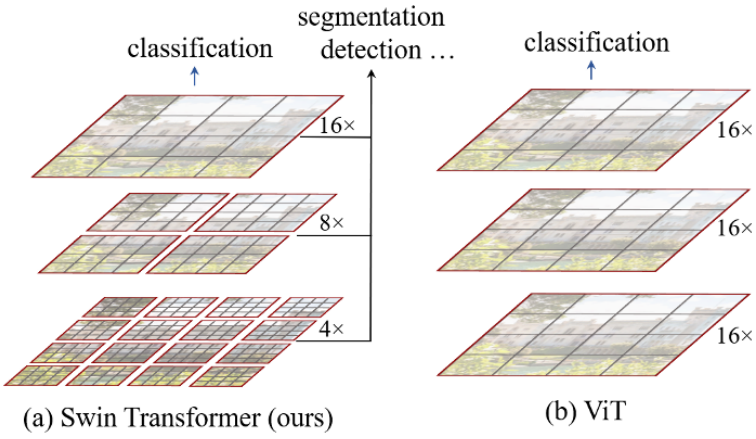

图 1: 基于局部自注意力的 Swin Transformer 和基于全局自注意力的 ViT 的对比图

Swin Transformer 的一个关键设计是连续的自注意力层之间的窗口移动，如图 2所示。移动窗口使得下一层的注意力可以融合上一层的不同窗口的图块，这显著增强了网络的建模能力。在移动窗口的实现中，我们通过掩码机制非常灵活地计算了需要的注意力，而避免了无用的注意力，这使我们的计算更加高效。

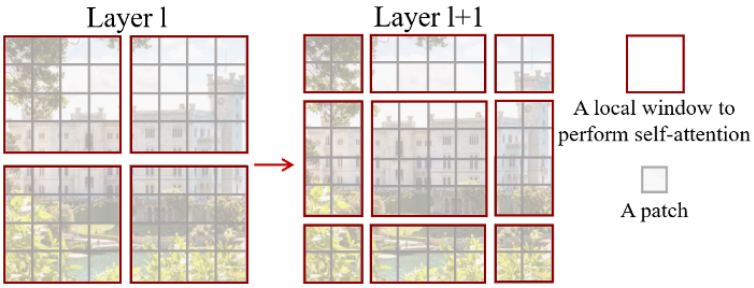

图 2: Swin Transformer 架构中的移动窗口方法

2 相关工作

2.1 CNN 及其变种

近些年来，CNN 被默认为计算机视觉的标准骨干网络。自 1998 年 Lecun 提出 LeNet[10]，CNN 逐渐进入人们的视野。但直到 2017 年 AlexNet[1]的提出，真正地让 CNN 成为了计算机视觉任务的主流。自此，网络架构如雨后春笋般涌现，研究者向着更深、更有效的卷积神经网络架构努力，提出了很多高效实用的 CNN 模型，比如 VGGNet[2]，GoogLeNet[3]，ResNet[4]，DenseNet[11]，HRNet[12]和 EfficientNet[13]等。除了这些架构上的进步，还有很多工作对单个卷积层进行了改进，例如可变形卷积[14]。虽然 CNN 及其变体仍然是计算机视觉应用的主要骨干架构，但我们很重视 Transformer 架构在计算机视觉和自然语言处理任务之间进行统一建模的强大潜力。

2.2 基于自注意力的骨干网络

由于受到自注意力层和 Transformer 架构在 NLP 领域的成功启发，一些工作使用自注意力层来代替流行的 ResNet 中的部分或全部空间卷积层[15-16]。在这些作品中，自注意力是在每个像素的局部窗口内计算的，这样可以加快优化，并且它们比对应的 ResNet 架构实现了更好的精度与速度之间的权衡。然而，它们昂贵的内存访问导致它们的实际延迟明显大于卷积网络[15]。我们建议在连续层之间移动窗口，而不是使用滑动窗口，这样可以在通用硬件中更有效地实现。

2.3 基于 Transformer 的视觉骨干网络

与我们的工作最接近的是 Vision Transformer (ViT)[7]，以及 ViT 的跟进工作[17-18]。ViT 的开创性工作直接将 Transformer 架构应用于非重叠的图像块上进行图像分类。与卷积网络相比，它在图像分类上实现了令人印象深刻的速度与准确性之间的权衡。虽然 ViT 需要大规模训练数据集才能表现良好，但 DeiT[17]引入了几种训练策略，允许 ViT 使用较小的 ImageNet-1K 数据集也能有效。ViT 模型在图像分类上的结果很好，但由于其计算复杂度是图像的平方倍，复杂度过高，不适合处理密集预测型任务。有一些工作通过直接上采样或反卷积将 ViT 模型应用于目标检测和语义分割的密集视觉任务，但性能相对较低[19-20]。我们的方法既高效又有效，在 COCO 对象检测和 ADE20K 语义分割上都达到了最先进的精度。

3 本文方法

3.1 总体架构

Swin Transformer 的总体架构如图 3所示，首先通过一个划分模块（Patch Partition）将输入图像划分为大小为 4*4 的块，每一个块被视作一个词元（token），然后使用一个线性嵌入层（Linear Embedding）将该词元投射到任意维度（记作 C）。

然后将词元输入 Swin Transformer Block，用来提取所需特征，输出维度保持不变。Swin Transformer Block 和线性嵌入层合并称作第一阶段。

为了得到多尺度的、层次的特征表示，随着网络变深，词元被合并模块（Patch Merging）合并，将分离的词元融合在一起。第一个合并模块将特征图从维度 $\frac{H}{4} \times \frac{W}{4} \times C$ 合并为 $\frac{H}{8} \times \frac{W}{8} \times 2C$，然后经过一个 Swin Transformer Block，组成第二阶段。同样地，再经过两个相同的由合并模块和 Swin Transformer Block 模块组成的第三、第四阶段，这便得到了我们需要的层次化特征表示，这些特征表示和 VGG、ResNet 等典型的 CNN 网络得到的特征图相似。有了这些特征表示，之后便可以方便地用于各种视觉任务的骨干网络。

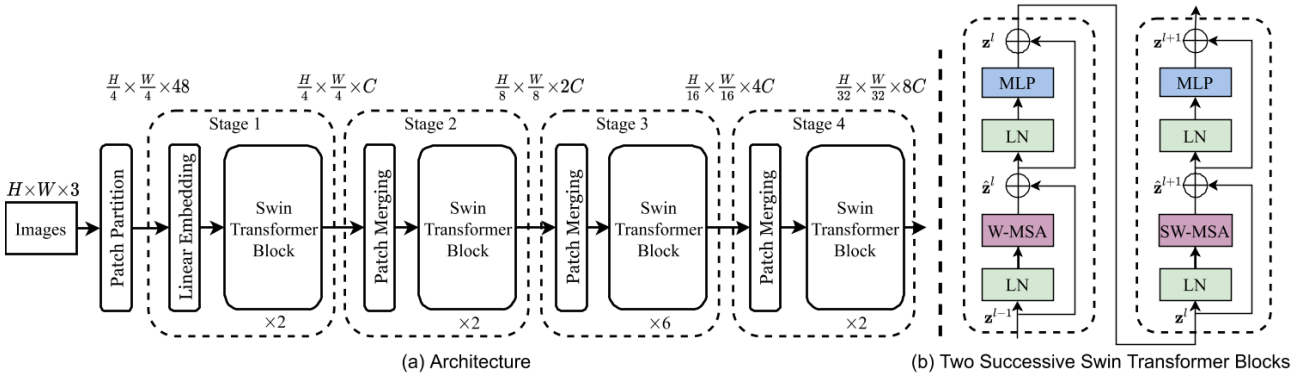

图 3: (a):Swin Transformer 架构;(b): 连续的 Swin Transformer Blocks

3.2 Patch Embedding

PatchEmbedding 模块包括 patch partition 层和 Linear Embedding 层。

我们想将大小为 [3,224,224] 的图片划分为 56x56 个大小为 [3,4,4] 的块，每个 patch 作为 Transformer 的一个 token。卷积中，卷积核大小为 [4,4]，它便对应着图片的一个 patch，如图 4，将左上角卷积核所在区域三个通道映射到卷积后的 56x56 特征图的左上角的一个点，通道数为 48，也即卷积后的结果，

每个点在深度方向上组织成为一个词元。

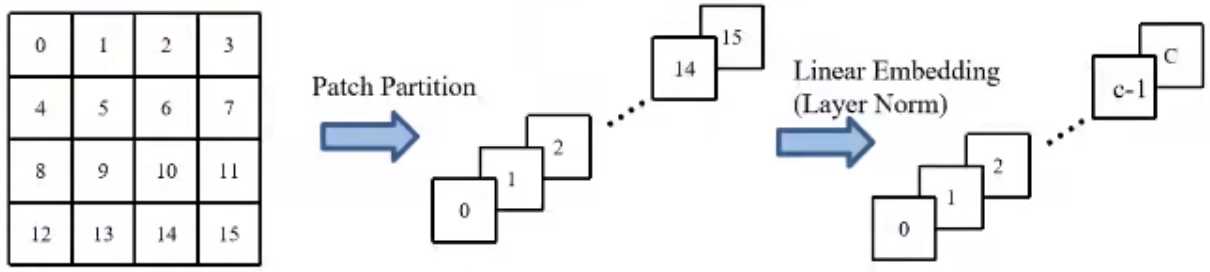

图 4: PatchEmbedding 操作示意图

3.3 非重叠窗口中的自注意力

标准的 Transformer 架构[5]及其对应用于图像分类领域的 ViT[7]都进行全局自注意力，需要计算一个词元与所有其他词元之间的关系。全局计算导致计算复杂度与词元数量成二次相关，使其不适用于许多需要大量词元进行密集预测或表示高分辨率图像的视觉问题。

为了模型的高效性，我们建议在局部窗口内计算自注意力。将一张图片不重叠地均匀分割开来得到一些窗口。假设每个窗口包括 $M \times M$ 个块，那么全局多头自注意力模块（MSA）和在拥有 $h \times w$ 个块的窗口中计算多头自注意力模块（W-MSA）的计算复杂度如下：

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C, \tag{1}$$

$$\Omega(\text{W-MSA}) = 4hwC^2 + 2M^2hwC, \tag{2}$$

由上式可见，前者是 hw 的二次方，后者是 hw 的一次方乘以固定的常数 M（文中默认设置为 7），所以基于窗口的 MSA 的计算复杂度是线性的，这将大大减少处理视觉任务中的计算复杂度。

3.4 移动窗口

基于窗口的自注意力模块缺乏跨窗口的连接，这限制了它的建模能力。为了在保持非重叠窗口的高效计算的同时引入跨窗口连接，我们提出了一种移位窗口分区方法，该方法在连续的 Swin Transformer 块中的两个分区配置之间交替。

如图 2所示，左边的模块的方法是从左上角的像素开始均匀划分，将 8×8 特征图均匀划分为 2×2 大小为 $4 \times 4 (M = 4)$ 的窗口。下一个模块采用从前一层的窗口结果进行移动，通过将窗口向右下角移动 $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ 个像素。

使用移动窗口的分区方法时，需要保证在初始划分的基础之上，所以二者必须先后形成整体，连续的 Swin Transformer 块被计算为：

$$\begin{aligned}
\hat{\mathbf{z}}^l &= \text{W-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1}, \\
\mathbf{z}^l &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^l\right)\right) + \hat{\mathbf{z}}^l, \\
\hat{\mathbf{z}}^{l+1} &= \text{SW-MSA}\left(\text{LN}\left(\mathbf{z}^l\right)\right) + \mathbf{z}^l, \\
\mathbf{z}^{l+1} &= \text{MLP}\left(\text{LN}\left(\hat{\mathbf{z}}^{l+1}\right)\right) + \hat{\mathbf{z}}^{l+1},
\end{aligned} \tag{3}$$

$\hat{\mathbf{z}}^l$ 和 $\mathbf{z}^l$ 分别代表第 l 块的 (S)W-MSA 模块和 MLP 模块的输出特征；W-MSA 和 SW-MSA 分别表示使用常规和移位窗口分区配置的基于窗口的多头自注意力。

移位窗口分区方法在上一层中引入了相邻非重叠窗口之间的连接，增强了特征学习能力，并且被

证明在图像分类、对象检测和语义分割方面是有效的。

3.5 Patch Merging

patch merging 是用来将相邻的小块合成一个大块，这样就可以看到前面四个小块的内容，可以大幅度增加感受野，有效抓住了多尺度的特征。PatchMerging 模块具体的作用是下采样，下采样之后高宽减半，通道数翻倍；具体方法为如图 5所示：假设特征矩阵高宽 4*4，以 2x2 大小为一个窗口，将窗口中相同位置上的像素取出，得到四个矩阵，然后在深度方向上拼接，然后在通道方向上进行 LayerNorm 的处理，然后使用全连接层 (1x1 卷积) 将通道数减半。

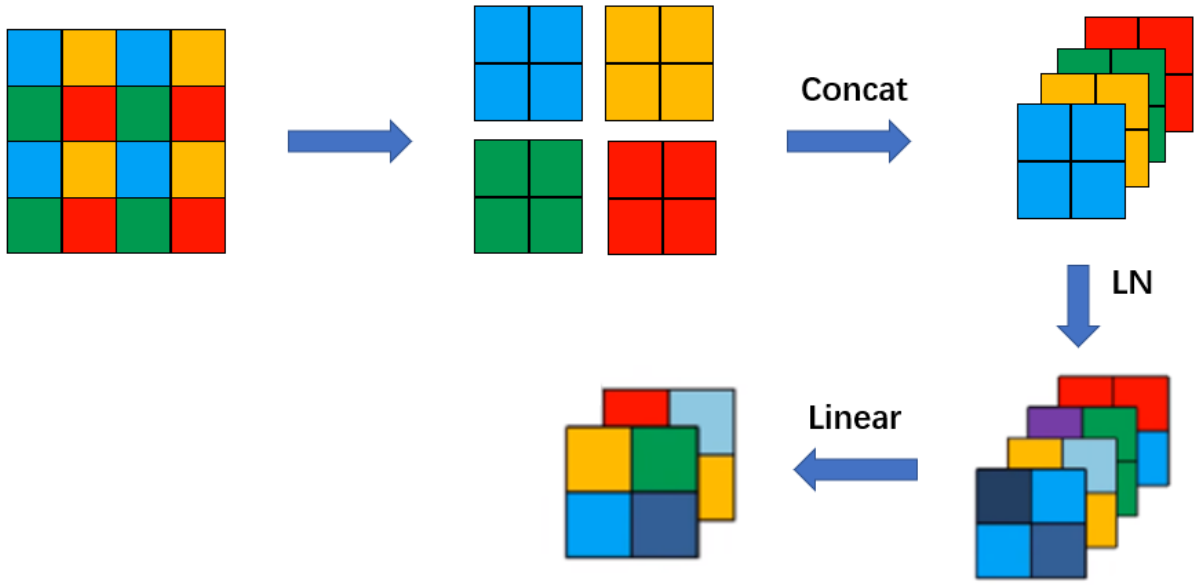

图 5: PatchMerging 操作示意图

4 复现细节

4.1 与已有开源代码对比

本文通过将 Swin Transformer 分类模型更改为语义分割模型。对于输入的图片，我们可以通过一系列 Patch Embedding、Patch Merging 和 Swin Transformer Block 模块得到具有多尺度的特征表示，所以本文的重点在于得到较好的特征表示，有了特征信息之后，输入给一个分类头便可进行图像分类了，同理，输入到一个分割头便可进行图像分割。

我在图像分类的代码基础上进行了修改，将分类头更换成了全卷积神经网络分割头，得到了基于 Swin Transformer 的语义分割模型，然后在 ADE-20k 数据集上获得了不错的结果。

4.2 实验环境搭建

操作系统：Linux version 5.4.0-135-generic

发行版信息：Ubuntu 18.04.2 LTS

Python 3.7.15

PyTorch 1.8.0

5 实验结果分析

本部分对实验所得结果进行分析，详细对实验内容进行说明，实验结果进行描述并分析。

5.1 图像分类任务

首先我使用官方提供的源码在一个网络上找到的花朵分类数据集上进行了实验，数据集分为 5 类：菊花、蒲公英、玫瑰花、向日葵和郁金香，数据集总共包括 3670 张图片，取出其中的 80% 作为训练集，剩下的 20% 作为验证集，由图 6所示，在经过 5 轮训练之后，损失值降到了 0.161，分类准确度也达到了 94.7%，是一个很不错的分类结果。

```
[train epoch 0] loss: 0.441, acc: 0.836: 100%|██████████| 368/368 [00:26<00:00, 14.06it/s]
[valid epoch 0] loss: 0.150, acc: 0.953: 100%|██████████| 92/92 [00:01<00:00, 48.69it/s]
[train epoch 1] loss: 0.256, acc: 0.906: 100%|██████████| 368/368 [00:26<00:00, 13.92it/s]
[valid epoch 1] loss: 0.157, acc: 0.947: 100%|██████████| 92/92 [00:01<00:00, 48.06it/s]
[train epoch 2] loss: 0.230, acc: 0.919: 100%|██████████| 368/368 [00:25<00:00, 14.20it/s]
[valid epoch 2] loss: 0.197, acc: 0.933: 100%|██████████| 92/92 [00:02<00:00, 44.56it/s]
[train epoch 3] loss: 0.201, acc: 0.929: 100%|██████████| 368/368 [00:25<00:00, 14.28it/s]
[valid epoch 3] loss: 0.206, acc: 0.937: 100%|██████████| 92/92 [00:01<00:00, 46.20it/s]
[train epoch 4] loss: 0.184, acc: 0.934: 100%|██████████| 368/368 [00:26<00:00, 14.01it/s]
[valid epoch 4] loss: 0.161, acc: 0.947: 100%|██████████| 92/92 [00:01<00:00, 46.22it/s]
```

图 6: 花分类实验结果

5.2 语义分割任务

在对源码进行修改之后，可将其良好地迁移到语义分割任务上。本文在 ADE20k 数据集上进行训练，得到的结果如图 7所示：

Scope	mIoU	mAcc	aAcc
global	39.29	57.74	76.75

图 7: ADE20k 分割数据集结果

一些测试结果如图 8所示。

图 8: 分割结果展示

6 总结与展望

本文在深入理解了 ViT、Swin Transformer 等论文的基础之上，仔细对官方提供的代码进行了研读，对深度学习领域代码的整体框架有了更进一步的理解，看懂之后我将其运用到花朵分类的数据集上完成了图像分类任务，取得了满意的结果。此外，我修改代码，将其很好地运用到了图像语义分割任务上，在 ADE20k 数据集上也取得了不错的成绩。

总之，此次大作业的设置让我对一篇论文的产生过程，从想法的产生、设计代码和实验、书写论文等整个框架熟悉起来，虽然技术上改进不多，但说实话，对于一个刚入门的小白，去尝试改善大佬的工作是令人闻而丧胆的，所以骨架上我没做修改，但对源码做了认真的研读，然后对于骨架得到的图像特征，对后续的分类、分割任务进行了应用和修改。

但是，我深刻见到了我在于代码方面的薄弱性，日后需持续填补。对于研究课题，既然选择了一条路，便坚定地做下去吧，尽管一步一个难点。

参考文献

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[2] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv:1409.1556, 2014.

[3] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

[4] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[5] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[6] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[7] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[8] LIN T Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.

[9] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.

[10] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[11] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.

[12] WANG J, SUN K, CHENG T, et al. Deep high-resolution representation learning for visual recognition [J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 43(10): 3349-3364.

[13] TAN M, LE Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]//International conference on machine learning. 2019: 6105-6114.

[14] DAI J, QI H, XIONG Y, et al. Deformable convolutional networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 764-773.

[15] HU H, ZHANG Z, XIE Z, et al. Local relation networks for image recognition[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 3464-3473.

[16] RAMACHANDRAN P, PARMAR N, VASWANI A, et al. Stand-alone self-attention in vision models [J]. Advances in Neural Information Processing Systems, 2019, 32.

[17] TOUVRON H, CORD M, DOUZE M, et al. Training data-efficient image transformers & distillation through attention[C]//International Conference on Machine Learning. 2021: 10347-10357.

[18] YUAN L, CHEN Y, WANG T, et al. Tokens-to-token vit: Training vision transformers from scratch on imagenet[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 558-567.

[19] BEAL J, KIM E, TZENG E, et al. Toward transformer-based object detection[J]. arXiv preprint arXiv:2012.09958, 2020.

[20] ZHENG S, LU J, ZHAO H, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 6881-6890.