arXiv:2103.14030v2 [cs.CV] 17 Aug 2021

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

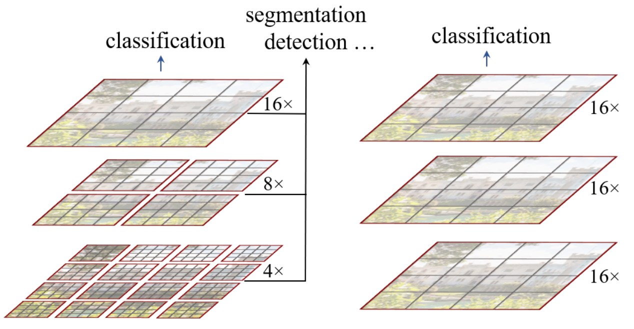
Ze Liu[†*] Yutong Lin[†*] Yue Cao[*] Han Hu[*‡] Yixuan Wei[†]
Zheng Zhang Stephen Lin Baining Guo
Microsoft Research Asia
{v-zeliu1,v-yutlin,yuecao,hanhu,v-yixwe,zhez,stevelin,bainguo}@microsoft.com

Abstract

This paper presents a new vision Transformer, called Swin Transformer, that capably serves as a general-purpose backbone for computer vision. Challenges in adapting Transformer from language to vision arise from differences between the two domains, such as large variations in the scale of visual entities and the high resolution of pixels in images compared to words in text. To address these differences, we propose a hierarchical Transformer whose representation is computed with ***S****hifted* ***win****dows. The shifted windowing scheme brings greater efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection. This hierarchical architecture has the flexibility to model at various scales and has linear computational complexity with respect to image size. These qualities of Swin Transformer make it compatible with a broad range of vision tasks, including image classification (87.3 top-1 accuracy on ImageNet-1K) and dense prediction tasks such as object detection (58.7 box AP and 51.1 mask AP on COCO test-dev) and semantic segmentation (53.5 mIoU on ADE20K val). Its performance surpasses the previous state-of-the-art by a large margin of +2.7 box AP and +2.6 mask AP on COCO, and +3.2 mIoU on ADE20K, demonstrating the potential of Transformer-based models as vision backbones. The hierarchical design and the shifted window approach also prove beneficial for all-MLP architectures. The code and models are publicly available at* `https://github.com/microsoft/Swin-Transformer`*.*

1. Introduction

Modeling in computer vision has long been dominated by convolutional neural networks (CNNs). Beginning with AlexNet [39] and its revolutionary performance on the ImageNet image classification challenge, CNN architectures have evolved to become increasingly powerful through greater scale [30, 76], more extensive connections [34], and more sophisticated forms of convolution [70, 18, 84]. With CNNs serving as backbone networks for a variety of vision tasks, these architectural advances have led to performance improvements that have broadly lifted the entire field.



(a) Swin Transformer (ours)

(b) ViT

Figure 1. (a) The proposed Swin Transformer builds hierarchical feature maps by merging image patches (shown in gray) in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window (shown in red). It can thus serve as a general-purpose backbone for both image classification and dense recognition tasks. (b) In contrast, previous vision Transformers [20] produce feature maps of a single low resolution and have quadratic computation complexity to input image size due to computation of self-attention globally.

On the other hand, the evolution of network architectures in natural language processing (NLP) has taken a different path, where the prevalent architecture today is instead the Transformer [64]. Designed for sequence modeling and transduction tasks, the Transformer is notable for its use of attention to model long-range dependencies in the data. Its tremendous success in the language domain has led researchers to investigate its adaptation to computer vision, where it has recently demonstrated promising results on certain tasks, specifically image classification [20] and joint vision-language modeling [47].

In this paper, we seek to expand the applicability of Transformer such that it can serve as a general-purpose

[*] Equal contribution. [†] Interns at MSRA. [‡] Contact person.

Swin Transformer：使用移位窗口的分层视觉Transformer

刘泽[†*] 林宇彤[†*] 曹越[*] 胡翰[*‡] 魏祎轩[†] 张峥 林史蒂芬 郭百宁
微软亚洲研究院

{v-zeliu1、v-yutlin、yuecao、hanhu、v-yixwe、zhez、stevelin、bainguo}@microsoft.com

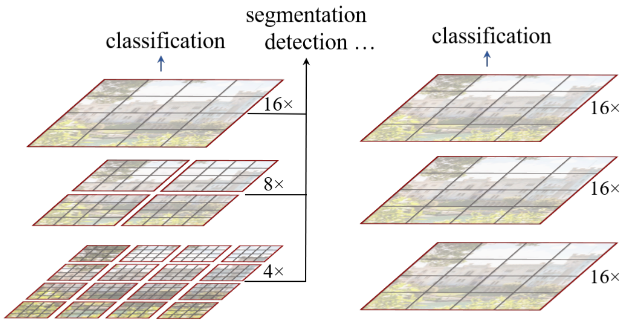
arXiv:2103.14030v2 [cs.CV] 17 Aug 2021

摘要

*This paper presents a new vision Transformer, called Swin Transformer, that capably serves as a general-purpose backbone for computer vision. Challenges in adapting Transformer from language to vision arise from differences between the two domains, such as large variations in the scale of visual entities and the high resolution of pixels in images compared to words in text. To address these differences, we propose a hierarchical Transformer whose representation is computed with **S**hifted **win**dows. The shifted windowing scheme brings greater efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection. This hierarchical architecture has the flexibility to model at various scales and has linear computational complexity with respect to image size. These qualities of Swin Transformer make it compatible with a broad range of vision tasks, including image classification (87.3 top-1 accuracy on ImageNet-1K) and dense prediction tasks such as object detection (58.7 box AP and 51.1 mask AP on COCO test-dev) and semantic segmentation (53.5 mIoU on ADE20K val). Its performance surpasses the previous state-of-the-art by a large margin of +2.7 box AP and +2.6 mask AP on COCO, and +3.2 mIoU on ADE20K, demonstrating the potential of Transformer-based models as vision backbones. The hierarchical design and the shifted window approach also prove beneficial for all-MLP architectures. The code and models are publicly available at* https://github.com/microsoft/Swin-Transformer.

1. 引言

计算机视觉中的建模长期以来一直由卷积神经网络（CNN）主导。从AlexNet [39]及其在ImageNet图像分类挑战中的革命性表现开始，CNN架构不断发展，通过更大的规模[30, 76]、更广泛的连接[34]以及更复杂的卷积形式[70, 18, 84]。随着CNN成为多种视觉任务的主干网络，这些架构上的进步带来了性能提升，从而广泛推动了整个领域的发展。



(a) Swin Transformer (ours)　(b) ViT

图1. (a) 提出的Swin Transformer通过合并深层图像块（灰色显示）构建分层特征图，且由于自注意力仅在各局部窗口（红色显示）内计算，其计算复杂度与输入图像尺寸呈线性关系。因此它可作为图像分类与密集识别任务的通用骨干网络。(b) 相比之下，先前的视觉Transformer[20]生成单一低分辨率特征图，且由于全局计算自注意力，其计算复杂度与输入图像尺寸呈平方关系。

另一方面，自然语言处理（NLP）领域的网络架构演进则走上了不同的道路，如今的主流架构是Transformer[64]。该架构专为序列建模与转换任务设计，其显著特点在于利用注意力机制来建模数据中的长程依赖关系。Transformer在语言领域的巨大成功促使研究者探索其在计算机视觉领域的适应性，近期已在特定任务——尤其是图像分类[20]与视觉-语言联合建模[47]——中展现出令人瞩目的成果。

在本文中，我们致力于拓展Transformer的适用范围，使其能够作为通用目的

[*] Equal contribution. [†] Interns at MSRA. [‡] Contact person.

backbone for computer vision, as it does for NLP and as CNNs do in vision. We observe that significant challenges in transferring its high performance in the language domain to the visual domain can be explained by differences between the two modalities. One of these differences involves scale. Unlike the word tokens that serve as the basic elements of processing in language Transformers, visual elements can vary substantially in scale, a problem that receives attention in tasks such as object detection [42, 53, 54]. In existing Transformer-based models [64, 20], tokens are all of a fixed scale, a property unsuitable for these vision applications. Another difference is the much higher resolution of pixels in images compared to words in passages of text. There exist many vision tasks such as semantic segmentation that require dense prediction at the pixel level, and this would be intractable for Transformer on high-resolution images, as the computational complexity of its self-attention is quadratic to image size. To overcome these issues, we propose a general-purpose Transformer backbone, called Swin Transformer, which constructs hierarchical feature maps and has linear computational complexity to image size. As illustrated in Figure 1(a), Swin Transformer constructs a hierarchical representation by starting from small-sized patches (outlined in gray) and gradually merging neighboring patches in deeper Transformer layers. With these hierarchical feature maps, the Swin Transformer model can conveniently leverage advanced techniques for dense prediction such as feature pyramid networks (FPN) [42] or U-Net [51]. The linear computational complexity is achieved by computing self-attention locally within non-overlapping windows that partition an image (outlined in red). The number of patches in each window is fixed, and thus the complexity becomes linear to image size. These merits make Swin Transformer suitable as a general-purpose backbone for various vision tasks, in contrast to previous Transformer based architectures [20] which produce feature maps of a single resolution and have quadratic complexity.

A key design element of Swin Transformer is its *shift* of the window partition between consecutive self-attention layers, as illustrated in Figure 2. The shifted windows bridge the windows of the preceding layer, providing connections among them that significantly enhance modeling power (see Table 4). This strategy is also efficient in regards to real-world latency: all *query* patches within a window share the same *key* set[1], which facilitates memory access in hardware. In contrast, earlier *sliding window* based self-attention approaches [33, 50] suffer from low latency on general hardware due to different *key* sets for different *query* pixels[2]. Our experiments show that the proposed *shifted window* approach has much lower latency than the *sliding window* method, yet is similar in modeling power (see Tables 5 and 6). The shifted window approach also proves beneficial for all-MLP architectures [61].

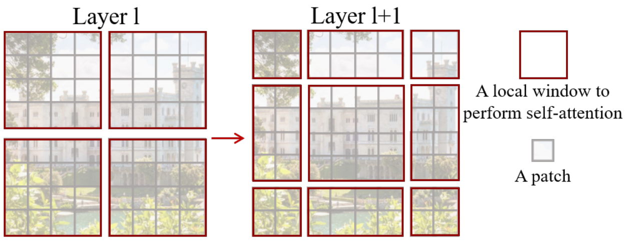


Figure 2. An illustration of the *shifted window* approach for computing self-attention in the proposed Swin Transformer architecture. In layer l (left), a regular window partitioning scheme is adopted, and self-attention is computed within each window. In the next layer $l + 1$ (right), the window partitioning is shifted, resulting in new windows. The self-attention computation in the new windows crosses the boundaries of the previous windows in layer l, providing connections among them.

The proposed Swin Transformer achieves strong performance on the recognition tasks of image classification, object detection and semantic segmentation. It outperforms the ViT / DeiT [20, 63] and ResNe(X)t models [30, 70] significantly with similar latency on the three tasks. Its 58.7 box AP and 51.1 mask AP on the COCO test-dev set surpass the previous state-of-the-art results by +2.7 box AP (Copy-paste [26] without external data) and +2.6 mask AP (DetectoRS [46]). On ADE20K semantic segmentation, it obtains 53.5 mIoU on the val set, an improvement of +3.2 mIoU over the previous state-of-the-art (SETR [81]). It also achieves a top-1 accuracy of 87.3% on ImageNet-1K image classification.

It is our belief that a unified architecture across computer vision and natural language processing could benefit both fields, since it would facilitate joint modeling of visual and textual signals and the modeling knowledge from both domains can be more deeply shared. We hope that Swin Transformer's strong performance on various vision problems can drive this belief deeper in the community and encourage unified modeling of vision and language signals.

2. Related Work

CNN and variants CNNs serve as the standard network model throughout computer vision. While the CNN has existed for several decades [40], it was not until the introduction of AlexNet [39] that the CNN took off and became mainstream. Since then, deeper and more effective convolutional neural architectures have been proposed to further propel the deep learning wave in computer vision, e.g., VGG [52], GoogleNet [57], ResNet [30], DenseNet [34],

[1] The *query* and *key* are projection vectors in a self-attention layer.

[2] While there are efficient methods to implement a sliding-window based convolution layer on general hardware, thanks to its shared kernel weights across a feature map, it is difficult for a sliding-window based self-attention layer to have efficient memory access in practice.

作为计算机视觉的骨干网络，正如它在自然语言处理中的作用以及卷积神经网络在视觉领域所做的那样。我们观察到，将其在语言领域的高性能迁移到视觉领域所面临的重大挑战，可以通过两种模态之间的差异来解释。其中一个差异涉及尺度问题。与作为语言Transformer基本处理单元的单词标记不同，视觉元素在尺度上可能存在显著差异，这一问题在目标检测等任务中备受关注[42, 53, 54]。在现有的基于Transformer的模型[64, 20]中，标记均采用固定尺度，这一特性并不适用于这些视觉应用。另一个差异是图像中像素的分辨率远高于文本段落中的单词。存在许多视觉任务（如语义分割）需要在像素级别进行密集预测，这对于高分辨率图像上的Transformer而言将是难以处理的，因为其自注意力机制的计算复杂度与图像尺寸呈平方关系。为克服这些问题，我们提出了一种通用型Transformer骨干网络——Swin Transformer，它构建了分层特征图，并具有与图像尺寸呈线性关系的计算复杂度。如图1(a)所示，Swin Transformer通过从小尺寸图像块（灰色轮廓）开始，并在更深的Transformer层中逐步合并相邻图像块，构建出分层表示。借助这些分层特征图，Swin Transformer模型可以便捷地利用先进技术进行密集预测，例如特征金字塔网络（FPN）[42]或U-Net [51]。线性计算复杂度是通过在划分图像的非重叠局部窗口（红色轮廓）内计算自注意力实现的。每个窗口中的图像块数量是固定的，因此计算复杂度与图像尺寸呈线性关系。这些优点使得Swin Transformer适合作为各种视觉任务的通用骨干网络，与此前基于Transformer的架构[20]形成对比——后者仅生成单一分辨率的特征图且具有平方级计算复杂度。

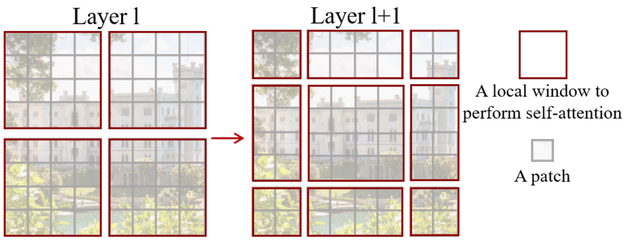


图2. 所提出的Swin Transformer架构中用于计算自注意力的*shifted window*方法示意图。在*l* (左侧)层中，采用常规窗口划分方案，并在每个窗口内计算自注意力。在下一层*l* +1（右侧）中，窗口划分发生偏移，从而形成新的窗口。新窗口中的自注意力计算跨越了*l*层中先前窗口的边界，从而在它们之间建立了连接。

Swin Transformer的一个关键设计元素是其在连续自注意力层之间采用窗口划分的*shift*，如图2所示。这种偏移窗口桥接了前一层的窗口，在窗口之间建立了连接，从而显著提升了建模能力（见表4）。该策略在实际延迟方面也表现出高效性：窗口内的所有*query*补丁共享相同的*key*集[1]，这有利于硬件中的内存访问。相比之下，早期基于*sliding window*的自注意力方法[33, 50]由于不同*query*像素[2]对应不同的*key*集，在通用硬件上存在延迟较高的问题。我们的实验表明，所提出的*shifted window* 该方法的延迟远低于*sliding window*方法，但在建模能力上相近（见表5和表6）。移位窗口方法也被证明对所有MLP架构[61]有益。

提出的Swin Transformer在图像分类、目标检测和语义分割等识别任务上表现出色。在三个任务中，它在保持相似延迟的同时，显著超越了ViT/DeiT [20, 63]和ResNe(X)t模型 [30, 70]。在COCO test-dev数据集上，其58.7的边界框AP和51.1的掩码AP分别以+2.7边界框AP（无外部数据的Copy-paste [26]）和+2.6掩码AP（DetectoRS [46]）的优势超越了先前的最优结果。在ADE20K语义分割任务中，它在验证集上获得了53.5的mIoU，较之前的最优方法（SETR [81]）提升了+3.2 mIoU。此外，在ImageNet-1K图像分类任务中，它实现了87.3%的top-1准确率。

我们相信，计算机视觉与自然语言处理的统一架构将惠及这两个领域，因为它能促进视觉与文本信号的联合建模，并使两个领域的建模知识得以更深入地共享。我们希望Swin Transformer在各种视觉问题上的强劲表现，能够推动这一理念在社区中进一步深化，并鼓励视觉与语言信号的统一建模。

2. 相关工作

CNN及其变体在整个计算机视觉领域作为标准网络模型。尽管CNN已经存在了几十年[40]，但直到AlexNet[39]的引入，CNN才真正起飞并成为主流。此后，为了进一步推动计算机视觉领域的深度学习浪潮，更深层、更有效的卷积神经架构被提出，例如VGG[52]、GoogleNet[57]、ResNet[30]、DenseNet[34]。

[1] The *query* and *key* are projection vectors in a self-attention layer.

[2] While there are efficient methods to implement a sliding-window based convolution layer on general hardware, thanks to its shared kernel weights across a feature map, it is difficult for a sliding-window based self-attention layer to have efficient memory access in practice.

HRNet [65], and EfficientNet [58]. In addition to these architectural advances, there has also been much work on improving individual convolution layers, such as depthwise convolution [70] and deformable convolution [18, 84]. While the CNN and its variants are still the primary backbone architectures for computer vision applications, we highlight the strong potential of Transformer-like architectures for unified modeling between vision and language. Our work achieves strong performance on several basic visual recognition tasks, and we hope it will contribute to a modeling shift.

Self-attention based backbone architectures Also inspired by the success of self-attention layers and Transformer architectures in the NLP field, some works employ self-attention layers to replace some or all of the spatial convolution layers in the popular ResNet [33, 50, 80]. In these works, the self-attention is computed within a local window of each pixel to expedite optimization [33], and they achieve slightly better accuracy/FLOPs trade-offs than the counterpart ResNet architecture. However, their costly memory access causes their actual latency to be significantly larger than that of the convolutional networks [33]. Instead of using sliding windows, we propose to *shift* windows between consecutive layers, which allows for a more efficient implementation in general hardware.

Self-attention/Transformers to complement CNNs Another line of work is to augment a standard CNN architecture with self-attention layers or Transformers. The self-attention layers can complement backbones [67, 7, 3, 71, 23, 74, 55] or head networks [32, 27] by providing the capability to encode distant dependencies or heterogeneous interactions. More recently, the encoder-decoder design in Transformer has been applied for the object detection and instance segmentation tasks [8, 13, 85, 56]. Our work explores the adaptation of Transformers for basic visual feature extraction and is complementary to these works.

Transformer based vision backbones Most related to our work is the Vision Transformer (ViT) [20] and its follow-ups [63, 72, 15, 28, 66]. The pioneering work of ViT directly applies a Transformer architecture on non-overlapping medium-sized image patches for image classification. It achieves an impressive speed-accuracy trade-off on image classification compared to convolutional networks. While ViT requires large-scale training datasets (i.e., JFT-300M) to perform well, DeiT [63] introduces several training strategies that allow ViT to also be effective using the smaller ImageNet-1K dataset. The results of ViT on image classification are encouraging, but its architecture is unsuitable for use as a general-purpose backbone network on dense vision tasks or when the input image resolution is high, due to its low-resolution feature maps and the quadratic increase in complexity with image size. There are a few works applying ViT models to the dense vision tasks of object detection and semantic segmentation by direct upsampling or deconvolution but with relatively lower performance [2, 81]. Concurrent to our work are some that modify the ViT architecture [72, 15, 28] for better image classification. Empirically, we find our Swin Transformer architecture to achieve the best speed-accuracy trade-off among these methods on image classification, even though our work focuses on general-purpose performance rather than specifically on classification. Another concurrent work [66] explores a similar line of thinking to build multi-resolution feature maps on Transformers. Its complexity is still quadratic to image size, while ours is linear and also operates locally which has proven beneficial in modeling the high correlation in visual signals [36, 25, 41]. Our approach is both efficient and effective, achieving state-of-the-art accuracy on both COCO object detection and ADE20K semantic segmentation.

3. Method

3.1. Overall Architecture

An overview of the Swin Transformer architecture is presented in Figure 3, which illustrates the tiny version (Swin-T). It first splits an input RGB image into non-overlapping patches by a patch splitting module, like ViT. Each patch is treated as a "token" and its feature is set as a concatenation of the raw pixel RGB values. In our implementation, we use a patch size of 4×4 and thus the feature dimension of each patch is $4 \times 4 \times 3 = 48$. A linear embedding layer is applied on this raw-valued feature to project it to an arbitrary dimension (denoted as C).

Several Transformer blocks with modified self-attention computation (*Swin Transformer blocks*) are applied on these patch tokens. The Transformer blocks maintain the number of tokens ($\frac{H}{4} \times \frac{W}{4}$), and together with the linear embedding are referred to as "Stage 1".

To produce a hierarchical representation, the number of tokens is reduced by patch merging layers as the network gets deeper. The first patch merging layer concatenates the features of each group of 2×2 neighboring patches, and applies a linear layer on the $4C$-dimensional concatenated features. This reduces the number of tokens by a multiple of $2 \times 2 = 4$ ($2\times$ downsampling of resolution), and the output dimension is set to $2C$. Swin Transformer blocks are applied afterwards for feature transformation, with the resolution kept at $\frac{H}{8} \times \frac{W}{8}$. This first block of patch merging and feature transformation is denoted as "Stage 2". The procedure is repeated twice, as "Stage 3" and "Stage 4", with output resolutions of $\frac{H}{16} \times \frac{W}{16}$ and $\frac{H}{32} \times \frac{W}{32}$, respectively. These stages jointly produce a hierarchical representation, HRNet [65]和EfficientNet [58]。除了这些架构上的进步，在改进单个卷积层方面也有许多工作，例如深度可分离卷积 [70] 和可变形卷积 [18, 84]。尽管CNN及其变体仍然是计算机视觉应用的主要骨干架构，但我们强调类Transformer架构在视觉与语言统一建模方面的巨大潜力。我们的工作在多个基础视觉识别任务上取得了强劲性能，我们希望这将有助于推动建模范式的转变。

基于自注意力的主干架构同样受到自注意力层和Transformer架构在NLP领域成功的启发，部分研究采用自注意力层替代主流ResNet[33,50,80]中的部分或全部空间卷积层。这些工作通过在像素的局部窗口内计算自注意力以加速优化[33]，相比对应的ResNet架构取得了略优的精度/FLOPs权衡。然而，其高昂的内存访问成本导致实际延迟显著高于卷积网络[33]。我们提出在连续层间采用{v*}窗口机制替代滑动窗口方案，这能在通用硬件上实现更高效的部署。

自注意力/Transformer对CNN的补充 另一研究方向是用自注意力层或Transformer增强标准CNN架构。自注意力层能够编码远距离依赖或异构交互，从而补充主干网络[67, 7, 3, 71, 23, 74, 55]或头部网络[32, 27]。最近，Transformer的编码器-解码器设计已被应用于目标检测和实例分割任务[8, 13, 85, 56]。我们的工作探索了Transformer在基础视觉特征提取中的适配，与这些研究形成互补。

基于Transformer的视觉骨干网络与我们的工作最相关的是Vision Transformer（ViT）[20]及其后续研究[63, 72, 15, 28, 66]。ViT的开创性工作直接将Transformer架构应用于非重叠的中等尺寸图像块进行图像分类。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-精度权衡。虽然ViT需要大规模训练数据集（即JFT-300M）才能表现良好，但DeiT[63]引入了多种训练策略，使ViT在较小的ImageNet-1K数据集上也能有效工作。ViT在图像分类上的成果令人鼓舞，但其架构不适合作为密集视觉任务或输入图像{v*}情况下的通用骨干网络。

分辨率较高，这得益于其低分辨率特征图以及复杂度随图像尺寸呈二次方增长的特性。已有少数研究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等密集视觉任务，但性能相对较低[2, 81]。与我们工作同期出现的一些研究通过改进ViT架构[72, 15, 28]以提升图像分类效果。实验表明，尽管我们的研究侧重于通用性能而非专门针对分类任务，但Swin Transformer架构在图像分类上实现了这些方法中最优的速度-精度平衡。另一项同期研究[66]探索了在Transformer上构建多分辨率特征图的类似思路，其复杂度仍与图像尺寸呈二次方关系，而我们的方法复杂度为线性且采用局部操作，这已被证明对建模视觉信号中的高相关性具有优势[36, 25, 41]。我们的方法兼具高效性与有效性，在COCO目标检测和ADE20K语义分割任务上均达到了最先进的精度水平。

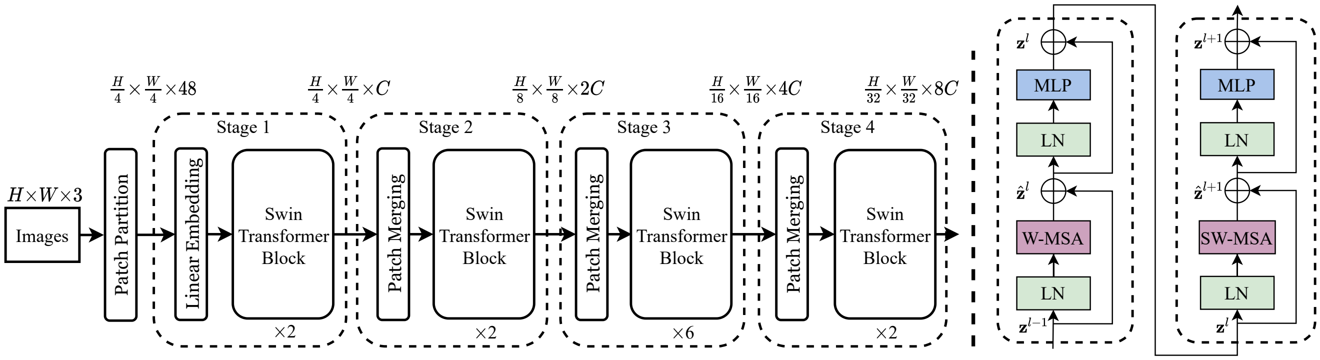
3. 方法

3.1. 整体架构

Swin Transformer架构的概览如图3所示，其中展示了其微小版本（Swin-T）。首先，通过一个图像块分割模块将输入的RGB图像分割成不重叠的块，类似于ViT。每个块被视为一个“令牌”，其特征设置为原始像素RGB值的拼接。在我们的实现中，我们使用4×4的块大小，因此每个块的特征维度为4×4×3=48。随后应用线性嵌入层将这些原始值特征投影到任意维度（记为C）。

在这些补丁标记上应用了几个具有改进自注意力计算（*Swin Transformer blocks*）的Transformer块。这些Transformer块保持标记数量不变（$\frac{H}{4} \times \frac{W}{4}$），并与线性嵌入层一起被称为“第一阶段”。

为了生成层次化表示，随着网络加深，通过补丁合并层减少令牌数量。第一层补丁合并层将每组2×2个相邻补丁的特征进行拼接，并对$4C$维拼接特征应用线性层。这使令牌数量减少至原来的2×2=4分之一（分辨率下采样2×倍），同时将输出维度设定为$2C$。随后应用Swin Transformer块进行特征变换，并保持分辨率在$\frac{H}{8} \times \frac{W}{8}$。这组补丁合并与特征变换的首个模块被称为“阶段2”。该过程重复两次，分别作为“阶段3”和“阶段4”，对应输出分辨率分别为$\frac{H}{16} \times \frac{W}{16}$和$\frac{H}{32} \times \frac{W}{32}$。这些阶段共同生成层次化表示，



(a) Architecture

(b) Two Successive Swin Transformer Blocks

Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

with the same feature map resolutions as those of typical convolutional networks, e.g., VGG [52] and ResNet [30]. As a result, the proposed architecture can conveniently replace the backbone networks in existing methods for various vision tasks.

Swin Transformer block Swin Transformer is built by replacing the standard multi-head self attention (MSA) module in a Transformer block by a module based on shifted windows (described in Section 3.2), with other layers kept the same. As illustrated in Figure 3(b), a Swin Transformer block consists of a shifted window based MSA module, followed by a 2-layer MLP with GELU nonlinearity in between. A LayerNorm (LN) layer is applied before each MSA module and each MLP, and a residual connection is applied after each module.

3.2. Shifted Window based Self-Attention

The standard Transformer architecture [64] and its adaptation for image classification [20] both conduct global self-attention, where the relationships between a token and all other tokens are computed. The global computation leads to quadratic complexity with respect to the number of tokens, making it unsuitable for many vision problems requiring an immense set of tokens for dense prediction or to represent a high-resolution image.

Self-attention in non-overlapped windows For efficient modeling, we propose to compute self-attention within local windows. The windows are arranged to evenly partition the image in a non-overlapping manner. Supposing each window contains $M \times M$ patches, the computational complexity of a global MSA module and a window based one on an image of $h \times w$ patches are[3]:

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

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

where the former is quadratic to patch number hw, and the latter is linear when M is fixed (set to 7 by default). Global self-attention computation is generally unaffordable for a large hw, while the window based self-attention is scalable.

Shifted window partitioning in successive blocks The window-based self-attention module lacks connections across windows, which limits its modeling power. To introduce cross-window connections while maintaining the efficient computation of non-overlapping windows, we propose a shifted window partitioning approach which alternates between two partitioning configurations in consecutive Swin Transformer blocks.

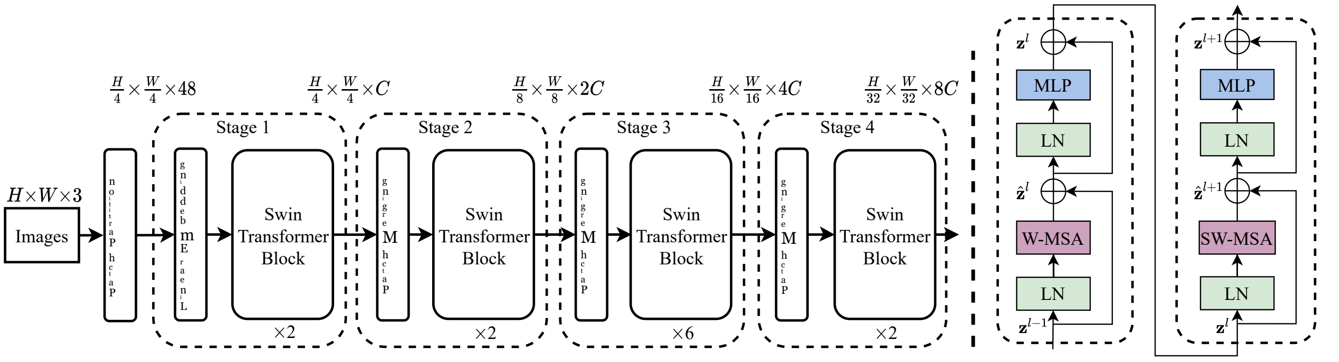
As illustrated in Figure 2, the first module uses a regular window partitioning strategy which starts from the top-left pixel, and the 8×8 feature map is evenly partitioned into 2×2 windows of size 4×4 ($M = 4$). Then, the next module adopts a windowing configuration that is shifted from that of the preceding layer, by displacing the windows by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ pixels from the regularly partitioned windows.

With the shifted window partitioning approach, consecutive Swin Transformer blocks are computed as

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

where $\hat{\mathbf{z}}^l$ and $\mathbf{z}^l$ denote the output features of the (S)W-MSA module and the MLP module for block l, respectively;

[3] We omit SoftMax computation in determining complexity.



(a) Architecture

(b) Two Successive Swin Transformer Blocks

图 3. (a) Swin Transformer（Swin-T）的架构；(b) 两个连续的 Swin Transformer 模块（符号表示与公式(3)一致）。W-MSA 和 SW-MSA 分别是采用规则窗口配置和移位窗口配置的多头自注意力模块。

具有与典型卷积网络（例如VGG [52] 和 ResNet [30]）相同的特征图分辨率。因此，所提出的架构可以方便地替代现有方法中的骨干网络，适用于各种视觉任务。

Swin Transformer 模块 Swin Transformer 通过将 Transformer 模块中的标准多头自注意力（MSA）模块替换为基于移位窗口的模块（详见第 3.2 节）构建而成，其余层保持不变。如图 3(b) 所示，一个 Swin Transformer 模块包含一个基于移位窗口的 MSA 模块，其后是一个 2 层 MLP，中间采用 GELU 非线性激活函数。每个 MSA 模块和 MLP 前均应用 LayerNorm（LN）层，每个模块后均采用残差连接。

3.2. 基于移位窗口的自注意力机制

标准Transformer架构[64]及其在图像分类任务中的适配版本[20]均采用全局自注意力机制，即计算每个标记与所有其他标记之间的关系。这种全局计算方式导致计算复杂度随标记数量呈二次方增长，使其难以适用于许多视觉任务——这些任务需要海量标记进行密集预测或表示高分辨率图像。

非重叠窗口中的自注意力机制 为了高效建模，我们提出在局部窗口内计算自注意力。这些窗口以非重叠方式均匀划分图像。假设每个窗口包含 $M \times M$ 个图像块，全局多头自注意力模块与基于窗口的模块的计算复杂度分别为在$h \times w$的图像上，补丁是[3]:

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

前者与补丁数量hw呈二次关系，而后者在M固定时（默认设为7）呈线性关系。对于较大的hw，全局自注意力计算通常难以承受，而基于窗口的自注意力则具有良好的可扩展性。

连续块中的移位窗口划分　基于窗口的自注意力模块缺乏跨窗口的连接，这限制了其建模能力。为了在保持非重叠窗口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划分方法，该方法在连续的Swin Transformer块中交替使用两种划分配置。

如图2所示，第一个模块采用常规窗口划分策略，从左上角像素开始，将8×8特征图均匀划分为2×2个尺寸为4×4的窗口。随后，下一模块采用相对于前一层窗口配置进行偏移的窗口设置，通过将窗口位移（$\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor$) 从规则划分的窗口中提取的像素。

通过移动窗口划分方法，连续的Swin Transformer块计算方式为

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

其中$\hat{\mathbf{z}}^l$和$\mathbf{z}^l$分别表示块l的(S)W-MSA模块和MLP模块的输出特征；

[3] We omit SoftMax computation in determining complexity.

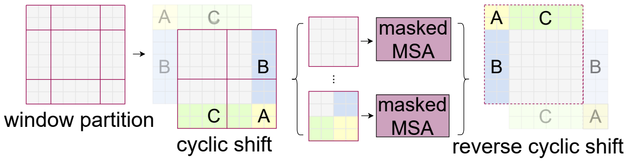


Figure 4. Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

W-MSA and SW-MSA denote window based multi-head self-attention using regular and shifted window partitioning configurations, respectively.

The shifted window partitioning approach introduces connections between neighboring non-overlapping windows in the previous layer and is found to be effective in image classification, object detection, and semantic segmentation, as shown in Table 4.

Efficient batch computation for shifted configuration An issue with shifted window partitioning is that it will result in more windows, from $\lceil \frac{h}{M} \rceil \times \lceil \frac{w}{M} \rceil$ to $(\lceil \frac{h}{M} \rceil + 1) \times (\lceil \frac{w}{M} \rceil + 1)$ in the shifted configuration, and some of the windows will be smaller than $M \times M$[4]. A naive solution is to pad the smaller windows to a size of $M \times M$ and mask out the padded values when computing attention. When the number of windows in regular partitioning is small, e.g. 2×2, the increased computation with this naive solution is considerable ($2 \times 2 \rightarrow 3 \times 3$, which is 2.25 times greater). Here, we propose a *more efficient batch computation approach* by cyclic-shifting toward the top-left direction, as illustrated in Figure 4. After this shift, a batched window may be composed of several sub-windows that are not adjacent in the feature map, so a masking mechanism is employed to limit self-attention computation to within each sub-window. With the cyclic-shift, the number of batched windows remains the same as that of regular window partitioning, and thus is also efficient. The low latency of this approach is shown in Table 5.

Relative position bias In computing self-attention, we follow [49, 1, 32, 33] by including a relative position bias $B \in \mathbb{R}^{M^2 \times M^2}$ to each head in computing similarity:

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V, \quad (4)$$

where $Q, K, V \in \mathbb{R}^{M^2 \times d}$ are the *query*, *key* and *value* matrices; d is the *query*/*key* dimension, and M^2 is the number of patches in a window. Since the relative position along each axis lies in the range $[-M+1, M-1]$, we parameterize a smaller-sized bias matrix $\hat{B} \in \mathbb{R}^{(2M-1)\times(2M-1)}$, and values in B are taken from $\hat{B}$.

We observe significant improvements over counterparts without this bias term or that use absolute position embedding, as shown in Table 4. Further adding absolute position embedding to the input as in [20] drops performance slightly, thus it is not adopted in our implementation.

The learnt relative position bias in pre-training can be also used to initialize a model for fine-tuning with a different window size through bi-cubic interpolation [20, 63].

3.3. Architecture Variants

We build our base model, called Swin-B, to have of model size and computation complexity similar to ViT-B/DeiT-B. We also introduce Swin-T, Swin-S and Swin-L, which are versions of about $0.25\times$, $0.5\times$ and $2\times$ the model size and computational complexity, respectively. Note that the complexity of Swin-T and Swin-S are similar to those of ResNet-50 (DeiT-S) and ResNet-101, respectively. The window size is set to $M = 7$ by default. The query dimension of each head is $d = 32$, and the expansion layer of each MLP is $\alpha = 4$, for all experiments. The architecture hyper-parameters of these model variants are:

- Swin-T: $C = 96$, layer numbers = $\{2, 2, 6, 2\}$
- Swin-S: $C = 96$, layer numbers =$\{2, 2, 18, 2\}$
- Swin-B: $C = 128$, layer numbers =$\{2, 2, 18, 2\}$
- Swin-L: $C = 192$, layer numbers =$\{2, 2, 18, 2\}$

where C is the channel number of the hidden layers in the first stage. The model size, theoretical computational complexity (FLOPs), and throughput of the model variants for ImageNet image classification are listed in Table 1.

4. Experiments

We conduct experiments on ImageNet-1K image classification [19], COCO object detection [43], and ADE20K semantic segmentation [83]. In the following, we first compare the proposed Swin Transformer architecture with the previous state-of-the-arts on the three tasks. Then, we ablate the important design elements of Swin Transformer.

4.1. Image Classification on ImageNet-1K

Settings For image classification, we benchmark the proposed Swin Transformer on ImageNet-1K [19], which contains 1.28M training images and 50K validation images from 1,000 classes. The top-1 accuracy on a single crop is reported. We consider two training settings:

- *Regular ImageNet-1K training*. This setting mostly follows [63]. We employ an AdamW [37] optimizer for 300 epochs using a cosine decay learning rate scheduler and 20 epochs of linear warm-up. A batch size of 1024, an initial learning rate of 0.001, and a

[4] To make the window size (M, M) divisible by the feature map size of (h, w), bottom-right padding is employed on the feature map if needed.

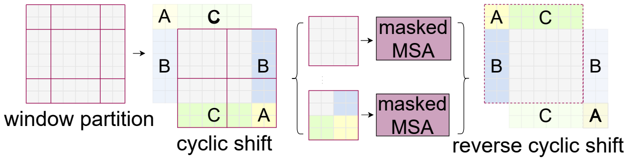


图4. 移位窗口划分中自注意力高效批量计算方法的示意图。

W-MSA和SW-MSA分别表示采用规则窗口划分和偏移窗口划分配置的基于窗口的多头自注意力机制。

移位窗口分区方法在相邻非重叠窗口之间引入了连接，如表4所示，该方法在图像分类、物体检测和语义分割任务中被证明是有效的。

移位配置的高效批量计算　移位窗口划分的一个问题是，它会导致窗口数量增加，从 $\lceil \frac{h}{M} \rceil \times \lceil \frac{w}{M} \rceil$ 到 $(\lceil \frac{h}{M} \rceil + 1) \times (\lceil \frac{w}{M} \rceil+1)$，并且部分窗口的尺寸会小于 $M \times M$[4]。一种简单的解决方案是将较小的窗口填充至 $M \times M$ 大小，并在计算注意力时屏蔽填充值。当常规划分的窗口数量较少时（例如 2×2），这种简单方案带来的计算量增加相当显著（$2 \times 2 \rightarrow 3 \times 3$，即增加了 2.25 倍）。为此，我们提出一种 *more efficient batch computation ap- proach* 的循环移位方法，如图 4 所示。经过移位后，一个批处理窗口可能由特征图中不相邻的多个子窗口组成，因此需要采用掩码机制将自注意力计算限制在各子窗口内。通过循环移位，批处理窗口的数量与常规窗口划分保持一致，从而保持了计算效率。该方法在表 5 中展示了较低的延迟。

相对位置偏置 在计算自注意力时，我们遵循[49, 1, 32, 33]的方法，在计算相似度时为每个注意力头加入相对位置偏置 $B \in \mathbb{R}^{M^2 \times M^2}$：

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V, \quad (4)$$

其中 $Q, K, V \in \mathbb{R}^{M^2 \times d}$ 是 *query*、*key* 和 *value* 矩阵；d 是 *query*/*key* 维度，M^2 是窗口中的补丁数量。由于每个轴上的相对位置范围在 $[-M+1, M-1]$ 内，我们将较小尺寸的偏置矩阵 $\hat{B} \in \mathbb{R}^{(2M-1)\times(2M-1)}$ 参数化，B 中的值取自 $\hat{B}$。

我们观察到，与没有这个偏置项或使用绝对位置嵌入的对应方法相比，性能有显著提升，如表4所示。进一步像文献[20]那样在输入中添加绝对位置嵌入会导致性能略有下降，因此我们的实现中没有采用这种方法。

预训练中学到的相对位置偏置也可以通过双三次插值[20, 63]来初始化具有不同窗口大小的微调模型。

3.3. 架构变体

我们构建的基础模型名为Swin-B，其模型规模和计算复杂度与ViT-B/DeiT-B相近。我们还推出了Swin-T、Swin-S和Swin-L版本，它们的模型规模和计算复杂度分别约为ViT-B/DeiT-B的0.25×、0.5×和2×倍。需要注意的是，Swin-T和Swin-S的复杂度分别与ResNet-50（DeiT-S）和ResNet-101相当。默认窗口大小设为 M =7。在所有实验中，每个注意力头的查询维度为 d =32，每个MLP扩展层的维度扩展倍数为α =4。这些模型变体的架构超参数如下：

- Swin-T：$C = 96$，层数 = {2, 2, 6, 2}
- Swin-S：$C = 96$，层数 ={2, 2, 18, 2}
- Swin-B：$C = 128$，层数 ={2, 2, 18, 2}
- Swin-L：$C = 192$，层数 ={2, 2, 18, 2}

其中C是第一级隐藏层的通道数。模型变体在ImageNet图像分类中的模型大小、理论计算复杂度（FLOPs）和吞吐量如表1所示。

4. 实验

我们在ImageNet-1K图像分类[19]、COCO目标检测[43]和ADE20K语义分割[83]上进行了实验。接下来，我们首先将提出的Swin Transformer架构与先前在这三项任务上的最先进技术进行比较。随后，我们对Swin Transformer的关键设计要素进行消融研究。

4.1. ImageNet-1K 图像分类

在图像分类方面，我们在ImageNet-1K [19]上对提出的Swin Transformer进行了基准测试，该数据集包含来自1,000个类别的128万张训练图像和5万张验证图像。报告了单次裁剪的top-1准确率。我们考虑了两种训练设置：

- *Regular ImageNet-1K training*该设置主要遵循[63]。我们采用AdamW[37]优化器进行300个周期的训练，使用余弦衰减学习率调度器，并包含20个周期的线性预热。批处理大小为1024，初始学习率为0.001，且

[4] To make the window size (M, M) divisible by the feature map size of (h, w), bottom-right padding is employed on the feature map if needed.

weight decay of 0.05 are used. We include most of the augmentation and regularization strategies of [63] in training, except for repeated augmentation [31] and EMA [45], which do not enhance performance. Note that this is contrary to [63] where repeated augmentation is crucial to stabilize the training of ViT.

- *Pre-training on ImageNet-22K and fine-tuning on ImageNet-1K.* We also pre-train on the larger ImageNet-22K dataset, which contains 14.2 million images and 22K classes. We employ an AdamW optimizer for 90 epochs using a linear decay learning rate scheduler with a 5-epoch linear warm-up. A batch size of 4096, an initial learning rate of 0.001, and a weight decay of 0.01 are used. In ImageNet-1K fine-tuning, we train the models for 30 epochs with a batch size of 1024, a constant learning rate of 10^{-5}, and a weight decay of 10^{-8}.

Results with regular ImageNet-1K training Table 1(a) presents comparisons to other backbones, including both Transformer-based and ConvNet-based, using regular ImageNet-1K training.

Compared to the previous state-of-the-art Transformer-based architecture, i.e. DeiT [63], Swin Transformers noticeably surpass the counterpart DeiT architectures with similar complexities: +1.5% for Swin-T (81.3%) over DeiT-S (79.8%) using 224^2 input, and +1.5%/1.4% for Swin-B (83.3%/84.5%) over DeiT-B (81.8%/83.1%) using 224^2/384^2 input, respectively.

Compared with the state-of-the-art ConvNets, i.e. RegNet [48] and EfficientNet [58], the Swin Transformer achieves a slightly better speed-accuracy trade-off. Noting that while RegNet [48] and EfficientNet [58] are obtained via a thorough architecture search, the proposed Swin Transformer is adapted from the standard Transformer and has strong potential for further improvement.

Results with ImageNet-22K pre-training We also pre-train the larger-capacity Swin-B and Swin-L on ImageNet-22K. Results fine-tuned on ImageNet-1K image classification are shown in Table 1(b). For Swin-B, the ImageNet-22K pre-training brings 1.8%∼1.9% gains over training on ImageNet-1K from scratch. Compared with the previous best results for ImageNet-22K pre-training, our models achieve significantly better speed-accuracy trade-offs: Swin-B obtains 86.4% top-1 accuracy, which is 2.4% higher than that of ViT with similar inference throughput (84.7 vs. 85.9 images/sec) and slightly lower FLOPs (47.0G vs. 55.4G). The larger Swin-L model achieves 87.3% top-1 accuracy, +0.9% better than that of the Swin-B model.

(a) Regular ImageNet-1K trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [48]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [48]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [48]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [58]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [58]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [58]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [58]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [58]	600^2	66M	37.0G	55.1	84.3
ViT-B/16 [20]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [20]	384^2	307M	190.7G	27.3	76.5
DeiT-S [63]	224^2	22M	4.6G	940.4	79.8
DeiT-B [63]	224^2	86M	17.5G	292.3	81.8
DeiT-B [63]	384^2	86M	55.4G	85.9	83.1
Swin-T	224^2	29M	4.5G	755.2	81.3
Swin-S	224^2	50M	8.7G	436.9	83.0
Swin-B	224^2	88M	15.4G	278.1	83.5
Swin-B	384^2	88M	47.0G	84.7	84.5

(b) ImageNet-22K pre-trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [38]	384^2	388M	204.6G	-	84.4
R-152x4 [38]	480^2	937M	840.5G	-	85.4
ViT-B/16 [20]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [20]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.4
Swin-L	384^2	197M	103.9G	42.1	87.3

Table 1. Comparison of different backbones on ImageNet-1K classification. Throughput is measured using the GitHub repository of [68] and a V100 GPU, following [63].

4.2. Object Detection on COCO

Settings Object detection and instance segmentation experiments are conducted on COCO 2017, which contains 118K training, 5K validation and 20K test-dev images. An ablation study is performed using the validation set, and a system-level comparison is reported on test-dev. For the ablation study, we consider four typical object detection frameworks: Cascade Mask R-CNN [29, 6], ATSS [79], RepPoints v2 [12], and Sparse RCNN [56] in mmdetection [10]. For these four frameworks, we utilize the same settings: multi-scale training [8, 56] (resizing the input such that the shorter side is between 480 and 800 while the longer side is at most 1333), AdamW [44] optimizer (initial learning rate of 0.0001, weight decay of 0.05, and batch size of 16), and 3x schedule (36 epochs). For system-level comparison, we adopt an improved HTC [9] (denoted as HTC++) with instaboost [22], stronger multi-scale training [7], 6x schedule (72 epochs), soft-NMS [5], and ImageNet-22K pre-trained model as initialization.

We compare our Swin Transformer to standard Con-

使用了0.05的权重衰减。在训练中，我们采用了[63]中的大部分增强和正则化策略，但未包含重复增强[31]和指数移动平均[45]，因为它们并未提升性能。需要注意的是，这与[63]相反——在[63]中，重复增强对于稳定ViT的训练至关重要。

- *Pre-training on ImageNet-22K and fine-tuning on ImageNet-1K*我们还在更大的ImageNet-22K数据集上进行了预训练，该数据集包含1420万张图像和22K个类别。我们采用AdamW优化器训练90个周期，使用带5周期线性预热的学习率线性衰减调度器。设置批量大小为4096，初始学习率为0.001，权重衰减为0.01。在ImageNet-1K微调阶段，我们以1024的批量大小训练模型30个周期，采用恒定学习率10^{-5}和权重衰减10^{-8}。

使用常规ImageNet-1K训练的结果 表1(a)展示了与其他骨干网络的比较，包括基于Transformer和基于ConvNet的架构，均采用常规ImageNet-1K训练。

与之前最先进的基于Transformer的架构（即DeiT [63]）相比，Swin Transformers在复杂度相近的情况下显著超越了对应的DeiT架构：使用224^2输入时，Swin-T（81.3%）比DeiT-S（79.8%）高出+1.5%；分别使用224^2/384^2输入时，Swin-B（83.3%/84.5%）比DeiT-B（81.8%/83.1%）高出+1.5%/1.4%。

与最先进的卷积网络（即RegNet [48]和EfficientNet [58]）相比，Swin Transformer实现了略优的速度-精度权衡。值得注意的是，尽管RegNet [48]和EfficientNet [58]是通过详尽的架构搜索获得的，而提出的Swin Transformer改编自标准Transformer架构，具备强大的进一步改进潜力。

使用ImageNet-22K预训练的结果 我们还在ImageNet-22K上预训练了更大容量的Swin-B和Swin-L模型。在ImageNet-1K图像分类任务上微调的结果如表1(b)所示。对于Swin-B模型，ImageNet-22K预训练相比从零开始在ImageNet-1K上训练带来了1.8%~1.9%的性能提升。与此前ImageNet-22K预训练的最佳结果相比，我们的模型实现了显著更优的速度-精度权衡：Swin-B获得了86.4%的top-1准确率，比具有相似推理吞吐量（84.7 vs. 85.9 图像/秒）且FLOPs略低（47.0G vs. 55.4G）的ViT模型高出2.4%。更大的Swin-L模型达到了87.3%的top-1准确率，+比Swin-B模型提升了0.9%。

(a) Regular ImageNet-1K trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [48]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [48]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [48]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [58]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [58]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [58]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [58]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [58]	600^2	66M	37.0G	55.1	84.3
ViT-B/16 [20]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [20]	384^2	307M	190.7G	27.3	76.5
DeiT-S [63]	224^2	22M	4.6G	940.4	79.8
DeiT-B [63]	224^2	86M	17.5G	292.3	81.8
DeiT-B [63]	384^2	86M	55.4G	85.9	83.1
Swin-T	224^2	29M	4.5G	755.2	81.3
Swin-S	224^2	50M	8.7G	436.9	83.0
Swin-B	224^2	88M	15.4G	278.1	83.5
Swin-B	384^2	88M	47.0G	84.7	84.5

(b) ImageNet-22K pre-trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [38]	384^2	388M	204.6G	-	84.4
R-152x4 [38]	480^2	937M	840.5G	-	85.4
ViT-B/16 [20]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [20]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.4
Swin-L	384^2	197M	103.9G	42.1	87.3

表1. 不同主干网络在ImageNet-1K分类任务上的比较。吞吐量测试遵循[63]的方法，使用[68]的GitHub仓库及V100 GPU进行测量。

4.2. COCO数据集上的目标检测

设置 目标检测与实例分割实验在COCO 2017数据集上进行，该数据集包含118K训练图像、5K验证图像和20K测试开发图像。消融研究使用验证集进行，系统级比较则在测试开发集上报告。对于消融研究，我们考虑了四种典型的目标检测框架：mmdetection [10] 中的 Cascade Mask R-CNN [29, 6]、ATSS [79]、RepPoints v2 [12] 和 Sparse RCNN [56]。针对这四种框架，我们采用相同的设置：多尺度训练 [8, 56]（调整输入尺寸，使短边在480至800之间，长边最大为1333）、AdamW [44] 优化器（初始学习率0.0001，权重衰减0.05，批量大小16）以及3倍训练计划（36个周期）。对于系统级比较，我们采用改进的HTC [9]（记为HTC++），结合了instaboost [22]、更强的多尺度训练 [7]、6倍训练计划（72个周期）、soft-NMS [5]，并以ImageNet-22K预训练模型进行初始化。

我们将我们的Swin Transformer与标准Con-

(a) Various frameworks

Method	Backbone	AP^{box}	$\text{AP}^{\text{box}}_{50}$	$\text{AP}^{\text{box}}_{75}$	#param.	FLOPs	FPS
Cascade	R-50	46.3	64.3	50.5	82M	739G	18.0
Mask R-CNN	Swin-T	**50.5**	**69.3**	**54.9**	86M	745G	15.3
ATSS	R-50	43.5	61.9	47.0	32M	205G	28.3
	Swin-T	**47.2**	**66.5**	**51.3**	36M	215G	22.3
RepPointsV2	R-50	46.5	64.6	50.3	42M	274G	13.6
	Swin-T	**50.0**	**68.5**	**54.2**	45M	283G	12.0
Sparse	R-50	44.5	63.4	48.2	106M	166G	21.0
R-CNN	Swin-T	**47.9**	**67.3**	**52.3**	110M	172G	18.4

(b) Various backbones w. Cascade Mask R-CNN

	AP^{box}	$\text{AP}^{\text{box}}_{50}$	$\text{AP}^{\text{box}}_{75}$	AP^{mask}	$\text{AP}^{\text{mask}}_{50}$	$\text{AP}^{\text{mask}}_{75}$	param	FLOPs	FPS
DeiT-S†	48.0	67.2	51.7	41.4	64.2	44.3	80M	889G	10.4
R50	46.3	64.3	50.5	40.1	61.7	43.4	82M	739G	18.0
Swin-T	**50.5**	**69.3**	**54.9**	**43.7**	**66.6**	**47.1**	86M	745G	15.3
X101-32	48.1	66.5	52.4	41.6	63.9	45.2	101M	819G	12.8
Swin-S	**51.8**	**70.4**	**56.3**	**44.7**	**67.9**	**48.5**	107M	838G	12.0
X101-64	48.3	66.4	52.3	41.7	64.0	45.1	140M	972G	10.4
Swin-B	**51.9**	**70.9**	**56.5**	**45.0**	**68.4**	**48.7**	145M	982G	11.6

(c) System-level Comparison

Method	mini-val		test-dev		#param.	FLOPs
	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}		
RepPointsV2* [12]	-	-	52.1	-	-	-
GCNet* [7]	51.8	44.7	52.3	45.4	-	1041G
RelationNet++* [13]	-	-	52.7	-	-	-
SpineNet-190 [21]	52.6	-	52.8	-	164M	1885G
ResNeSt-200* [78]	52.5	-	53.3	47.1	-	-
EfficientDet-D7 [59]	54.4	-	55.1	-	77M	410G
DetectoRS* [46]	-	-	55.7	48.5	-	-
YOLOv4 P7* [4]	-	-	55.8	-	-	-
Copy-paste [26]	55.9	47.2	56.0	47.4	185M	1440G
X101-64 (HTC++)	52.3	46.0	-	-	155M	1033G
Swin-B (HTC++)	56.4	49.1	-	-	160M	1043G
Swin-L (HTC++)	57.1	49.5	57.7	50.2	284M	1470G
Swin-L (HTC++)*	**58.0**	**50.4**	**58.7**	**51.1**	284M	-

Table 2. Results on COCO object detection and instance segmentation. †denotes that additional decovolution layers are used to produce hierarchical feature maps. * indicates multi-scale testing.

vNets, i.e. ResNe(X)t, and previous Transformer networks, e.g. DeiT. The comparisons are conducted by changing only the backbones with other settings unchanged. Note that while Swin Transformer and ResNe(X)t are directly applicable to all the above frameworks because of their hierarchical feature maps, DeiT only produces a single resolution of feature maps and cannot be directly applied. For fair comparison, we follow [81] to construct hierarchical feature maps for DeiT using deconvolution layers.

Comparison to ResNe(X)t Table 2(a) lists the results of Swin-T and ResNet-50 on the four object detection frameworks. Our Swin-T architecture brings consistent +3.4∼4.2 box AP gains over ResNet-50, with slightly larger model size, FLOPs and latency.

ADE20K		val	test	#param.	FLOPs	FPS
Method	Backbone	mIoU	score			
DANet [23]	ResNet-101	45.2	-	69M	1119G	15.2
DLab.v3+ [11]	ResNet-101	44.1	-	63M	1021G	16.0
ACNet [24]	ResNet-101	45.9	38.5	-		
DNL [71]	ResNet-101	46.0	56.2	69M	1249G	14.8
OCRNet [73]	ResNet-101	45.3	56.0	56M	923G	19.3
UperNet [69]	ResNet-101	44.9	-	86M	1029G	20.1
OCRNet [73]	HRNet-w48	45.7	-	71M	664G	12.5
DLab.v3+ [11]	ResNeSt-101	46.9	55.1	66M	1051G	11.9
DLab.v3+ [11]	ResNeSt-200	48.4	-	88M	1381G	8.1
SETR [81]	T-Large‡	50.3	61.7	308M	-	-
UperNet	DeiT-S†	44.0	-	52M	1099G	16.2
UperNet	Swin-T	46.1	-	60M	945G	18.5
UperNet	Swin-S	49.3	-	81M	1038G	15.2
UperNet	Swin-B‡	51.6	-	121M	1841G	8.7
UperNet	Swin-L‡	**53.5**	**62.8**	234M	3230G	6.2

Table 3. Results of semantic segmentation on the ADE20K val and test set. † indicates additional deconvolution layers are used to produce hierarchical feature maps. ‡ indicates that the model is pre-trained on ImageNet-22K.

Table 2(b) compares Swin Transformer and ResNe(X)t under different model capacity using Cascade Mask R-CNN. Swin Transformer achieves a high detection accuracy of 51.9 box AP and 45.0 mask AP, which are significant gains of +3.6 box AP and +3.3 mask AP over ResNeXt101-64x4d, which has similar model size, FLOPs and latency. On a higher baseline of 52.3 box AP and 46.0 mask AP using an improved HTC framework, the gains by Swin Transformer are also high, at +4.1 box AP and +3.1 mask AP (see Table 2(c)). Regarding inference speed, while ResNe(X)t is built by highly optimized Cudnn functions, our architecture is implemented with built-in PyTorch functions that are not all well-optimized. A thorough kernel optimization is beyond the scope of this paper.

Comparison to DeiT The performance of DeiT-S using the Cascade Mask R-CNN framework is shown in Table 2(b). The results of Swin-T are +2.5 box AP and +2.3 mask AP higher than DeiT-S with similar model size (86M vs. 80M) and significantly higher inference speed (15.3 FPS vs. 10.4 FPS). The lower inference speed of DeiT is mainly due to its quadratic complexity to input image size.

Comparison to previous state-of-the-art Table 2(c) compares our best results with those of previous state-of-the-art models. Our best model achieves 58.7 box AP and 51.1 mask AP on COCO test-dev, surpassing the previous best results by +2.7 box AP (Copy-paste [26] without external data) and +2.6 mask AP (DetectoRS [46]).

4.3. Semantic Segmentation on ADE20K

Settings ADE20K [83] is a widely-used semantic segmentation dataset, covering a broad range of 150 semantic

(a) Various frameworks

Method	Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	#param.	FLOPs	FPS
Cascade Mask R-CNN	R-50	46.3	64.3	50.5	82M	739G	18.0
	Swin-T	**50.5**	**69.3**	**54.9**	86M	745G	15.3
ATSS	R-50	43.5	61.9	47.0	32M	205G	28.3
	Swin-T	**47.2**	**66.5**	**51.3**	36M	215G	22.3
RepPointsV2	R-50	46.5	64.6	50.3	42M	274G	13.6
	Swin-T	**50.0**	**68.5**	**54.2**	45M	283G	12.0
Sparse R-CNN	R-50	44.5	63.4	48.2	106M	166G	21.0
	Swin-T	**47.9**	**67.3**	**52.3**	110M	172G	18.4

(b) Various backbones w. Cascade Mask R-CNN

	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	param	FLOPs	FPS
DeiT-S†	48.0	67.2	51.7	41.4	64.2	44.3	80M	889G	10.4
R50	46.3	64.3	50.5	40.1	61.7	43.4	82M	739G	18.0
Swin-T	**50.5**	**69.3**	**54.9**	**43.7**	**66.6**	**47.1**	86M	745G	15.3
X101-32	48.1	66.5	52.4	41.6	63.9	45.2	101M	819G	12.8
Swin-S	**51.8**	**70.4**	**56.3**	**44.7**	**67.9**	**48.5**	107M	838G	12.0
X101-64	48.3	66.4	52.3	41.7	64.0	45.1	140M	972G	10.4
Swin-B	**51.9**	**70.9**	**56.5**	**45.0**	**68.4**	**48.7**	145M	982G	11.6

(c) System-level Comparison

Method	mini-val		test-dev		#param.	FLOPs
	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}		
RepPointsV2* [12]	-	-	52.1	-	-	-
GCNet* [7]	51.8	44.7	52.3	45.4	-	1041G
RelationNet++* [13]	-	-	52.7	-	-	-
SpineNet-190 [21]	52.6	-	52.8	-	164M	1885G
ResNeSt-200* [78]	52.5	-	53.3	47.1	-	-
EfficientDet-D7 [59]	54.4	-	55.1	-	77M	410G
DetectoRS* [46]	-	-	55.7	48.5	-	-
YOLOv4 P7* [4]	-	-	55.8	-	-	-
Copy-paste [26]	55.9	47.2	56.0	47.4	185M	1440G
X101-64 (HTC++)	52.3	46.0	-	-	155M	1033G
Swin-B (HTC++)	56.4	49.1	-	-	160M	1043G
Swin-L (HTC++)	57.1	49.5	57.7	50.2	284M	1470G
Swin-L (HTC++)*	**58.0**	**50.4**	**58.7**	**51.1**	284M	-

表2. COCO目标检测与实例分割结果。†表示使用额外的反卷积层生成分层特征图。*表示多尺度测试。

vNets，即ResNe(X)t，以及先前的Transformer网络，例如DeiT。这些比较仅通过更换骨干网络进行，其他设置保持不变。需要注意的是，由于Swin Transformer和ResNe(X)t具有分层特征图，可直接适用于上述所有框架；而DeiT仅生成单一分辨率的特征图，无法直接应用。为公平比较，我们遵循[81]的方法，使用反卷积层为DeiT构建分层特征图。

与ResNe(X)t的对比 表2(a)列出了Swin-T和ResNet-50在四种目标检测框架上的结果。我们的Swin-T架构相比ResNet-50带来了持续的+3.4~4.2边界框AP提升，同时模型大小、FLOPs和延迟仅有小幅增加。

表2(b) 对比了 Swin Transformer 与 ResNe(X)t

ADE20K Method	Backbone	val mIoU	test score	#param.	FLOPs	FPS
DANet [23]	ResNet-101	45.2	-	69M	1119G	15.2
DLab.v3+ [11]	ResNet-101	44.1	-	63M	1021G	16.0
ACNet [24]	ResNet-101	45.9	38.5	-		
DNL [71]	ResNet-101	46.0	56.2	69M	1249G	14.8
OCRNet [73]	ResNet-101	45.3	56.0	56M	923G	19.3
UperNet [69]	ResNet-101	44.9	-	86M	1029G	20.1
OCRNet [73]	HRNet-w48	45.7	-	71M	664G	12.5
DLab.v3+ [11]	ResNeSt-101	46.9	55.1	66M	1051G	11.9
DLab.v3+ [11]	ResNeSt-200	48.4	-	88M	1381G	8.1
SETR [81]	T-Large‡	50.3	61.7	308M	-	-
UperNet	DeiT-S†	44.0	-	52M	1099G	16.2
UperNet	Swin-T	46.1	-	60M	945G	18.5
UperNet	Swin-S	49.3	-	81M	1038G	15.2
UperNet	Swin-B‡	51.6	-	121M	1841G	8.7
UperNet	Swin-L‡	**53.5**	**62.8**	234M	3230G	6.2

表3. ADE20K验证集和测试集上的语义分割结果。†表示使用了额外的反卷积层来生成层次特征图。‡表示该模型在ImageNet-22K上进行了预训练。

在不同模型容量下使用级联掩码R-CNN时，Swin Transformer实现了51.9边界框AP和45.0掩码AP的高检测精度，相较于具有相似模型大小、FLOPs和延迟的ResNeXt101-64x4d，分别显著提升了+3.6边界框AP和+3.3掩码AP。在使用改进的HTC框架达到52.3边界框AP和46.0掩码AP的更高基线时，Swin Transformer带来的提升依然显著，分别达到+4.1边界框AP和+3.1掩码AP（见表2(c)）。在推理速度方面，尽管ResNe(X)t基于高度优化的Cudnn函数构建，我们的架构采用PyTorch内置函数实现，其中部分函数尚未充分优化。彻底的核优化已超出本文研究范围。

与DeiT的对比 DeiT-S在Cascade Mask R-CNN框架下的性能展示于表2(b)。在模型规模相近（86M vs. 80M）的情况下，Swin-T的结果比DeiT-S高出+2.5个框AP和+2.3个掩码AP，且推理速度显著更快（15.3 FPS vs. 10.4 FPS）。DeiT较低的推理速度主要源于其对输入图像尺寸的二次计算复杂度。

与先前最先进技术的比较 表2(c)将我们的最佳结果与先前最先进模型的结果进行了比较。我们的最佳模型在COCO test-dev上实现了58.7的边界框AP和51.1的掩码AP，分别以+2.7边界框AP（无外部数据的复制-粘贴[26]）和+2.6掩码AP（DetectoRS[46]）超越了先前的最佳结果。

4.3. 在ADE20K数据集上的语义分割

设置 ADE20K [83] 是一个广泛使用的语义分割数据集，涵盖了150种语义类别的广泛范围。

	ImageNet		COCO		ADE20k
	top-1	top-5	AP^{box}	AP^{mask}	mIoU
w/o shifting	80.2	95.1	47.7	41.5	43.3
shifted windows	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**
no pos.	80.1	94.9	49.2	42.6	43.8
abs. pos.	80.5	95.2	49.0	42.4	43.2
abs.+rel. pos.	81.3	95.6	50.2	43.4	44.0
rel. pos. w/o app.	79.3	94.7	48.2	41.9	44.1
rel. pos.	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**

Table 4. Ablation study on the *shifted windows* approach and different position embedding methods on three benchmarks, using the Swin-T architecture. w/o shifting: all self-attention modules adopt regular window partitioning, without *shifting*; abs. pos.: absolute position embedding term of ViT; rel. pos.: the default settings with an additional relative position bias term (see Eq. (4)); app.: the first scaled dot-product term in Eq. (4).

method	MSA in a stage (ms)				Arch. (FPS)		
	S1	S2	S3	S4	T	S	B
sliding window (naive)	122.5	38.3	12.1	7.6	183	109	77
sliding window (kernel)	7.6	4.7	2.7	1.8	488	283	187
Performer [14]	4.8	2.8	1.8	1.5	638	370	241
window (w/o shifting)	2.8	1.7	1.2	0.9	770	444	280
shifted window (padding)	3.3	2.3	1.9	2.2	670	371	236
shifted window (cyclic)	3.0	1.9	1.3	1.0	755	437	278

Table 5. Real speed of different self-attention computation methods and implementations on a V100 GPU.

categories. It has 25K images in total, with 20K for training, 2K for validation, and another 3K for testing. We utilize UperNet [69] in mmseg [16] as our base framework for its high efficiency. More details are presented in the Appendix.

Results Table 3 lists the mIoU, model size (#param), FLOPs and FPS for different method/backbone pairs. From these results, it can be seen that Swin-S is +5.3 mIoU higher (49.3 vs. 44.0) than DeiT-S with similar computation cost. It is also +4.4 mIoU higher than ResNet-101, and +2.4 mIoU higher than ResNeSt-101 [78]. Our Swin-L model with ImageNet-22K pre-training achieves 53.5 mIoU on the val set, surpassing the previous best model by +3.2 mIoU (50.3 mIoU by SETR [81] which has a larger model size).

4.4. Ablation Study

In this section, we ablate important design elements in the proposed Swin Transformer, using ImageNet-1K image classification, Cascade Mask R-CNN on COCO object detection, and UperNet on ADE20K semantic segmentation.

Shifted windows Ablations of the *shifted window* approach on the three tasks are reported in Table 4. Swin-T with the shifted window partitioning outperforms the counterpart built on a single window partitioning at each stage by +1.1% top-1 accuracy on ImageNet-1K, +2.8 box AP/+2.2 mask AP on COCO, and +2.8 mIoU on ADE20K. The results indicate the effectiveness of using shifted windows to build connections among windows in the preceding layers. The latency overhead by *shifted window* is also small, as shown in Table 5.

Relative position bias Table 4 shows comparisons of different position embedding approaches. Swin-T with relative position bias yields +1.2%/+0.8% top-1 accuracy on ImageNet-1K, +1.3/+1.5 box AP and +1.1/+1.3 mask AP on COCO, and +2.3/+2.9 mIoU on ADE20K in relation to those without position encoding and with absolute position embedding, respectively, indicating the effectiveness of the relative position bias. Also note that while the inclusion of absolute position embedding improves image classification accuracy (+0.4%), it harms object detection and semantic segmentation (-0.2 box/mask AP on COCO and -0.6 mIoU on ADE20K).

While the recent ViT/DeiT models abandon translation invariance in image classification even though it has long been shown to be crucial for visual modeling, we find that inductive bias that encourages certain translation invariance is still preferable for general-purpose visual modeling, particularly for the dense prediction tasks of object detection and semantic segmentation.

Different self-attention methods The real speed of different self-attention computation methods and implementations are compared in Table 5. Our cyclic implementation is more hardware efficient than naive padding, particularly for deeper stages. Overall, it brings a 13%, 18% and 18% speed-up on Swin-T, Swin-S and Swin-B, respectively.

The self-attention modules built on the proposed *shifted window* approach are 40.8×/2.5×, 20.2×/2.5×, 9.3×/2.1×, and 7.6×/1.8× more efficient than those of *sliding windows* in naive/kernel implementations on four network stages, respectively. Overall, the Swin Transformer architectures built on *shifted windows* are 4.1/1.5, 4.0/1.5, 3.6/1.5 times faster than variants built on *sliding windows* for Swin-T, Swin-S, and Swin-B, respectively. Table 6 compares their accuracy on the three tasks, showing that they are similarly accurate in visual modeling.

Compared to Performer [14], which is one of the fastest Transformer architectures (see [60]), the proposed *shifted window* based self-attention computation and the overall Swin Transformer architectures are slightly faster (see Table 5), while achieving +2.3% top-1 accuracy compared to Performer on ImageNet-1K using Swin-T (see Table 6).

5. Conclusion

This paper presents Swin Transformer, a new vision Transformer which produces a hierarchical feature repre-

	ImageNet		COCO		ADE20k
	top-1	top-5	AP^{box}	AP^{mask}	mIoU
w/o shifting	80.2	95.1	47.7	41.5	43.3
shifted windows	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**
no pos.	80.1	94.9	49.2	42.6	43.8
abs. pos.	80.5	95.2	49.0	42.4	43.2
abs.+rel. pos.	81.3	95.6	50.2	43.4	44.0
rel. pos. w/o app.	79.3	94.7	48.2	41.9	44.1
rel. pos.	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**

表4. 在三个基准测试上对*shifted windows*方法及不同位置嵌入方法进行的消融研究，采用Swin-T架构。w/o shifting：所有自注意力模块采用常规窗口划分，不使用*shifting*；abs. pos. ：ViT的绝对位置嵌入项；rel. pos.：默认设置，包含额外的相对位置偏置项（见公式(4)）；app.：公式(4)中的第一个缩放点积项。

该数据集包含25K张图像，其中20K用于训练，2K用于验证，另外3K用于测试。我们采用mmseg[16]中的UperNet[69]作为基础框架，因其高效性。更多细节详见附录。

结果 表3列出了不同方法/骨干网络组合的mIoU、模型大小（#参数）、FLOPs和FPS。从这些结果可以看出，在计算成本相近的情况下，Swin-S比DeiT-S的mIoU高出+5.3（49.3对比44.0）。同时，它比ResNet-101高出+4.4 mIoU，比ResNeSt-101高出+2.4 mIoU。我们使用ImageNet-22K预训练的Swin-L模型在验证集上达到了53.5 mIoU，以+3.2 mIoU的优势超越了此前的最佳模型（SETR的50.3 mIoU，且其模型规模更大）。

4.4. 消融研究

在本节中，我们通过ImageNet-1K图像分类、COCO目标检测上的Cascade Mask R-CNN以及ADE20K语义分割上的UperNet，对所提出的Swin Transformer中的重要设计要素进行了消融实验。

在三个任务上对*shifted window*方法的消融研究结果如表4所示。采用移位窗口分区的Swin-T模型在ImageNet-1K上以+1.1%的top-1准确率、在COCO上以+2.8边界框AP/+2.2掩码AP、在ADE20K上以+2.8 mIoU的表现，超越了各阶段采用单一窗口分区的对应模型。这些结果表明，使用移位窗口在前一层窗口间建立连接是有效的。如表5所示，*shifted window*带来的延迟开销也较小。

相对位置偏置 表4展示了不同位置嵌入方法的比较。采用相对位置偏置的Swin-T在ImageNet-1K上实现了+1.2%/+0.8%的top-1准确率，+1.3/+1.5的边界框AP以及+1.1/+1.3的掩码AP。

method	MSA in a stage (ms)				Arch. (FPS)		
	S1	S2	S3	S4	T	S	B
sliding window (naive)	122.5	38.3	12.1	7.6	183	109	77
sliding window (kernel)	7.6	4.7	2.7	1.8	488	283	187
Performer [14]	4.8	2.8	1.8	1.5	638	370	241
window (w/o shifting)	2.8	1.7	1.2	0.9	770	444	280
shifted window (padding)	3.3	2.3	1.9	2.2	670	371	236
shifted window (cyclic)	3.0	1.9	1.3	1.0	755	437	278

表5. 不同自注意力计算方法和实现在V100 GPU上的实际速度。

在COCO数据集上，相对于未使用位置编码和采用绝对位置嵌入的模型，相对位置偏置分别带来了+2.3/+2.9 mIoU的性能提升；在ADE20K数据集上同样观察到其有效性。同时需注意，虽然加入绝对位置嵌入提升了图像分类准确率（+0.4%），但它对目标检测和语义分割任务产生了负面影响（在COCO上导致-0.2 box/mask AP，在ADE20K上导致-0.6 mIoU下降）。

尽管最近的ViT/DeiT模型在图像分类中放弃了平移不变性——尽管长期以来平移不变性被证明对视觉建模至关重要，但我们发现，鼓励一定平移不变性的归纳偏置对于通用视觉建模仍然更可取，特别是在目标检测和语义分割这类密集预测任务中。

不同的自注意力方法 不同自注意力计算方法和实现的实际速度对比见表5。我们的循环实现比简单填充更硬件高效，尤其在深层阶段。总体而言，它在Swin-T、Swin-S和Swin-B上分别带来了13%、18%和18%的速度提升。

基于所提出的*shifted window*方法构建的自注意力模块，在四个网络阶段的朴素/内核实现中，分别比*sliding windows*的效率高出40.8×/2.5×、20.2×/2.5×、9.3×/2.1×和7.6×/1.8×。总体而言，基于*shifted windows*构建的Swin Transformer架构，在Swin-T、Swin-S和Swin-B上分别比基于*sliding windows*构建的变体快4.1/1.5、4.0/1.5、3.6/1.5倍。表6比较了它们在三个任务上的准确性，表明它们在视觉建模中具有相似的精度。

与最快的Transformer架构之一Performer[14]相比（参见[60]），所提出的基于*shiftedwindow*的自注意力计算及整体Swin Transformer架构在速度上略有优势（见表5），同时在使用Swin-T的ImageNet-1K数据集上达到了比Performer高+2.3%的top-1准确率（见表6）。

5. 结论

本文提出了一种新的视觉Transformer——Swin Transformer，它能生成层次化的特征表示。

		ImageNet		COCO		ADE20k
	Backbone	top-1	top-5	AP^{box}	AP^{mask}	mIoU
sliding window	Swin-T	81.4	95.6	50.2	43.5	45.8
Performer [14]	Swin-T	79.0	94.2	-	-	-
shifted window	Swin-T	81.3	95.6	50.5	43.7	46.1

Table 6. Accuracy of Swin Transformer using different methods for self-attention computation on three benchmarks.

sentation and has linear computational complexity with respect to input image size. Swin Transformer achieves the state-of-the-art performance on COCO object detection and ADE20K semantic segmentation, significantly surpassing previous best methods. We hope that Swin Transformer's strong performance on various vision problems will encourage unified modeling of vision and language signals.

As a key element of Swin Transformer, the *shifted window* based self-attention is shown to be effective and efficient on vision problems, and we look forward to investigating its use in natural language processing as well.

Acknowledgement

We thank many colleagues at Microsoft for their help, in particular, Li Dong and Furu Wei for useful discussions; Bin Xiao, Lu Yuan and Lei Zhang for help on datasets.

A1. Detailed Architectures

The detailed architecture specifications are shown in Table 7, where an input image size of 224×224 is assumed for all architectures. "Concat $n \times n$" indicates a concatenation of $n \times n$ neighboring features in a patch. This operation results in a downsampling of the feature map by a rate of n. "96-d" denotes a linear layer with an output dimension of 96. "win. sz. 7×7" indicates a multi-head self-attention module with window size of 7×7.

A2. Detailed Experimental Settings

A2.1. Image classification on ImageNet-1K

The image classification is performed by applying a global average pooling layer on the output feature map of the last stage, followed by a linear classifier. We find this strategy to be as accurate as using an additional `class` token as in ViT [20] and DeiT [63]. In evaluation, the top-1 accuracy using a single crop is reported.

Regular ImageNet-1K training The training settings mostly follow [63]. For all model variants, we adopt a default input image resolution of 224^2. For other resolutions such as 384^2, we fine-tune the models trained at 224^2 resolution, instead of training from scratch, to reduce GPU consumption.

When training from scratch with a 224^2 input, we employ an AdamW [37] optimizer for 300 epochs using a cosine decay learning rate scheduler with 20 epochs of linear warm-up. A batch size of 1024, an initial learning rate of 0.001, a weight decay of 0.05, and gradient clipping with a max norm of 1 are used. We include most of the augmentation and regularization strategies of [63] in training, including RandAugment [17], Mixup [77], Cutmix [75], random erasing [82] and stochastic depth [35], but not repeated augmentation [31] and Exponential Moving Average (EMA) [45] which do not enhance performance. Note that this is contrary to [63] where repeated augmentation is crucial to stabilize the training of ViT. An increasing degree of stochastic depth augmentation is employed for larger models, i.e. $0.2, 0.3, 0.5$ for Swin-T, Swin-S, and Swin-B, respectively.

For fine-tuning on input with larger resolution, we employ an adamW [37] optimizer for 30 epochs with a constant learning rate of 10^{-5}, weight decay of 10^{-8}, and the same data augmentation and regularizations as the first stage except for setting the stochastic depth ratio to 0.1.

ImageNet-22K pre-training We also pre-train on the larger ImageNet-22K dataset, which contains 14.2 million images and 22K classes. The training is done in two stages. For the first stage with 224^2 input, we employ an AdamW optimizer for 90 epochs using a linear decay learning rate scheduler with a 5-epoch linear warm-up. A batch size of 4096, an initial learning rate of 0.001, and a weight decay of 0.01 are used. In the second stage of ImageNet-1K fine-tuning with 224^2/384^2 input, we train the models for 30 epochs with a batch size of 1024, a constant learning rate of 10^{-5}, and a weight decay of 10^{-8}.

A2.2. Object detection on COCO

For an ablation study, we consider four typical object detection frameworks: Cascade Mask R-CNN [29, 6], ATSS [79], RepPoints v2 [12], and Sparse RCNN [56] in mmdetection [10]. For these four frameworks, we utilize the same settings: multi-scale training [8, 56] (resizing the input such that the shorter side is between 480 and 800 while the longer side is at most 1333), AdamW [44] optimizer (initial learning rate of 0.0001, weight decay of 0.05, and batch size of 16), and 3x schedule (36 epochs with the learning rate decayed by 10× at epochs 27 and 33).

For system-level comparison, we adopt an improved HTC [9] (denoted as HTC++) with instaboost [22], stronger multi-scale training [7] (resizing the input such that the shorter side is between 400 and 1400 while the longer side is at most 1600), 6x schedule (72 epochs with the learning rate decayed at epochs 63 and 69 by a factor of 0.1), soft-NMS [5], and an extra global self-attention layer appended at the output of last stage and ImageNet-22K pre-trained

		ImageNet		COCO		ADE20k
	Backbone	top-1	top-5	AP^{box}	AP^{mask}	mIoU
sliding window	Swin-T	81.4	95.6	50.2	43.5	45.8
Performer [14]	Swin-T	79.0	94.2	-	-	-
shifted window	Swin-T	81.3	95.6	50.5	43.7	46.1

表6. 在三个基准测试中，使用不同自注意力计算方法的Swin Transformer准确率。

Swin Transformer 具有线性计算复杂度，其计算量与输入图像大小呈线性关系。在COCO目标检测和ADE20K语义分割任务上，Swin Transformer实现了最先进的性能，显著超越了以往的最佳方法。我们希望Swin Transformer在各种视觉问题上的强大表现能够促进视觉与语言信号的统一建模。

作为Swin Transformer的关键要素，基于*shifted window*的自注意力机制已被证明在视觉问题上高效且有效，我们期待在自然语言处理领域进一步探索其应用。

致谢

我们感谢微软的许多同事给予的帮助，特别是李东和韦福荣的有益讨论；感谢肖斌、袁路和张磊在数据集方面的帮助。

A1. 详细架构

详细架构规格如表7所示，其中所有架构均假设输入图像尺寸为224×224。“Concat $n \times n$”表示对补丁中$n \times n$个相邻特征进行拼接。该操作会使特征图以n的比率下采样。“96-d”表示输出维度为96的线性层。“win. sz. 7 × 7”表示窗口大小为7 × 7的多头自注意力模块。

A2. 详细实验设置

A2.1. 在ImageNet-1K数据集上的图像分类

图像分类通过在最后阶段的输出特征图上应用全局平均池化层，随后连接线性分类器来完成。我们发现这一策略与ViT[20]和DeiT[63]中采用额外分类标记的方法同样精确。在评估中，报告的是使用单次裁剪得到的top-1准确率。

常规ImageNet-1K训练　训练设置主要遵循[63]。对于所有模型变体，我们默认采用224^2的输入图像分辨率。对于其他分辨率（如384^2），我们会对224^2分辨率下训练的模型进行微调，而非从头开始训练，以降低GPU消耗。

在使用224^2输入从头开始训练时，我们采用AdamW [37]优化器进行300个周期的训练，使用余弦衰减学习率调度器，并包含20个周期的线性预热。采用1024的批次大小、0.001的初始学习率、0.05的权重衰减，以及最大范数为1的梯度裁剪。我们在训练中包含了[63]的大部分增强和正则化策略，包括RandAugment [17]、Mixup [77]、Cutmix [75]、随机擦除 [82]和随机深度 [35]，但未使用重复增强 [31]和指数移动平均（EMA）[45]，因为它们并未提升性能。请注意，这与[63]相反，在[63]中重复增强对于稳定ViT的训练至关重要。对于更大的模型，我们采用了递增的随机深度增强程度，即Swin-T、Swin-S和Swin-B分别使用0.2、0.3,和0.5。

对于在更高分辨率输入上进行微调，我们采用adamW [37]优化器进行30个周期的训练，保持恒定学习率为10^{-5}，权重衰减为10^{-8}，数据增强和正则化策略与第一阶段相同，但将随机深度比率设置为0.1。

ImageNet-22K预训练　我们还在规模更大的ImageNet-22K数据集上进行了预训练，该数据集包含1420万张图像和22K个类别。训练分为两个阶段进行。在首个使用224^2输入尺寸的阶段中，我们采用AdamW优化器训练90个周期，学习率调度器采用线性衰减策略并包含5个周期的线性预热。批次大小设置为4096，初始学习率为0.001，权重衰减为0.01。在第二阶段使用224^2/384^2输入尺寸的ImageNet-1K微调中，我们以1024的批次大小训练模型30个周期，保持恒定学习率10^{-5}，权重衰减为10^{-8}。

A2.2. COCO数据集上的目标检测

在消融研究中，我们基于mmdetection[10]考虑了四种典型目标检测框架：Cascade Mask R-CNN[29, 6]、ATSS[79]、RepPoints v2[12]和Sparse RCNN[56]。针对这四种框架，我们采用统一设置：多尺度训练[8, 56]（将输入图像短边缩放至480-800像素之间，长边不超过1333像素）、AdamW[44]优化器（初始学习率0.0001，权重衰减0.05，批量大小16）以及3倍训练计划（共36个训练周期，在第27和33周期将学习率衰减10×）。

在系统级比较中，我们采用改进的HTC [9]（记为HTC++），其结合了instaboost [22]、更强的多尺度训练[7]（将输入图像短边调整至400到1400之间，长边不超过1600）、6倍训练计划（共72个训练周期，学习率在第63和69周期时衰减为原值的0.1倍）、软性非极大值抑制 [5]，并在最后阶段的输出层后添加了一个额外的全局自注意力层，同时使用ImageNet-22K预训练模型。

	downsp. rate (output size)	Swin-T	Swin-S	Swin-B	Swin-L
stage 1	4× (56×56)	concat 4×4, 96-d, LN	concat 4×4, 96-d, LN	concat 4×4, 128-d, LN	concat 4×4, 192-d, LN
		$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 96, head 3}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 96, head 3}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 128, head 4}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 192, head 6}\end{bmatrix} \times 2$
stage 2	8× (28×28)	concat 2×2, 192-d , LN	concat 2×2, 192-d , LN	concat 2×2, 256-d , LN	concat 2×2, 384-d , LN
		$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 192, head 6}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 192, head 6}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 256, head 8}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 384, head 12}\end{bmatrix} \times 2$
stage 3	16× (14×14)	concat 2×2, 384-d , LN	concat 2×2, 384-d , LN	concat 2×2, 512-d , LN	concat 2×2, 768-d , LN
		$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 384, head 12}\end{bmatrix} \times 6$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 384, head 12}\end{bmatrix} \times 18$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 512, head 16}\end{bmatrix} \times 18$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 768, head 24}\end{bmatrix} \times 18$
stage 4	32× (7×7)	concat 2×2, 768-d , LN	concat 2×2, 768-d , LN	concat 2×2, 1024-d , LN	concat 2×2, 1536-d , LN
		$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 768, head 24}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 768, head 24}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 1024, head 32}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 1536, head 48}\end{bmatrix} \times 2$

Table 7. Detailed architecture specifications.

model as initialization. We adopt stochastic depth with ratio of 0.2 for all Swin Transformer models.

A2.3. Semantic segmentation on ADE20K

ADE20K [83] is a widely-used semantic segmentation dataset, covering a broad range of 150 semantic categories. It has 25K images in total, with 20K for training, 2K for validation, and another 3K for testing. We utilize UperNet [69] in mmsegmentation [16] as our base framework for its high efficiency.

In training, we employ the AdamW [44] optimizer with an initial learning rate of 6×10^{-5}, a weight decay of 0.01, a scheduler that uses linear learning rate decay, and a linear warmup of 1,500 iterations. Models are trained on 8 GPUs with 2 images per GPU for 160K iterations. For augmentations, we adopt the default setting in mmsegmentation of random horizontal flipping, random re-scaling within ratio range [0.5, 2.0] and random photometric distortion. Stochastic depth with ratio of 0.2 is applied for all Swin Transformer models. Swin-T, Swin-S are trained on the standard setting as the previous approaches with an input of 512×512. Swin-B and Swin-L with ‡ indicate that these two models are pre-trained on ImageNet-22K, and trained with the input of 640×640.

In inference, a multi-scale test using resolutions that are [0.5, 0.75, 1.0, 1.25, 1.5, 1.75]× of that in training is employed. When reporting test scores, both the training images and validation images are used for training, following common practice [71].

A3. More Experiments

A3.1. Image classification with different input size

Table 8 lists the performance of Swin Transformers with different input image sizes from 224^2 to 384^2. In general, a larger input resolution leads to better top-1 accuracy but with slower inference speed.

	Swin-T		Swin-S		Swin-B	
input size	top-1 acc	throughput (image / s)	top-1 acc	throughput (image / s)	top-1 acc	throughput (image / s)
224^2	81.3	755.2	83.0	436.9	83.3	278.1
256^2	81.6	580.9	83.4	336.7	83.7	208.1
320^2	82.1	342.0	83.7	198.2	84.0	132.0
384^2	82.2	219.5	83.9	127.6	84.5	84.7

Table 8. Swin Transformers with different input image size on ImageNet-1K classification.

Backbone	Optimizer	AP^{box}	$\text{AP}^{\text{box}}_{50}$	$\text{AP}^{\text{box}}_{75}$	AP^{mask}	$\text{AP}^{\text{mask}}_{50}$	$\text{AP}^{\text{mask}}_{75}$
R50	SGD	45.0	62.9	48.8	38.5	59.9	41.4
	AdamW	46.3	64.3	50.5	40.1	61.7	43.4
X101-32x4d	SGD	47.8	65.9	51.9	40.4	62.9	43.5
	AdamW	48.1	66.5	52.4	41.6	63.9	45.2
X101-64x4d	SGD	48.8	66.9	53.0	41.4	63.9	44.7
	AdamW	48.3	66.4	52.3	41.7	64.0	45.1

Table 9. Comparison of the SGD and AdamW optimizers for ResNe(X)t backbones on COCO object detection using the Cascade Mask R-CNN framework.

A3.2. Different Optimizers for ResNe(X)t on COCO

Table 9 compares the AdamW and SGD optimizers of the ResNe(X)t backbones on COCO object detection. The Cascade Mask R-CNN framework is used in this comparison. While SGD is used as a default optimizer for Cascade Mask R-CNN framework, we generally observe improved accuracy by replacing it with an AdamW optimizer, particularly for smaller backbones. We thus use AdamW for ResNe(X)t backbones when compared to the proposed Swin Transformer architectures.

A3.3. Swin MLP-Mixer

We apply the proposed hierarchical design and the shifted window approach to the MLP-Mixer architectures [61], referred to as Swin-Mixer. Table 10 shows the performance of Swin-Mixer compared to the original MLP-Mixer architectures MLP-Mixer [61] and a follow-up ap-

	downsp. rate (output size)	Swin-T	Swin-S	Swin-B	Swin-L
stage 1	4× (56×56)	concat 4×4, 96-d, LN	concat 4×4, 96-d, LN	concat 4×4, 128-d, LN	concat 4×4, 192-d, LN
		$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 96, head 3}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 96, head 3}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 128, head 4}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 192, head 6}\end{bmatrix} \times 2$
stage 2	8× (28×28)	concat 2×2, 192-d , LN	concat 2×2, 192-d , LN	concat 2×2, 256-d , LN	concat 2×2, 384-d , LN
		$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 192, head 6}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 192, head 6}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 256, head 8}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 384, head 12}\end{bmatrix} \times 2$
stage 3	16× (14×14)	concat 2×2, 384-d , LN	concat 2×2, 384-d , LN	concat 2×2, 512-d , LN	concat 2×2, 768-d , LN
		$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 384, head 12}\end{bmatrix} \times 6$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 384, head 12}\end{bmatrix} \times 18$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 512, head 16}\end{bmatrix} \times 18$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 768, head 24}\end{bmatrix} \times 18$
stage 4	32× (7×7)	concat 2×2, 768-d , LN	concat 2×2, 768-d , LN	concat 2×2, 1024-d , LN	concat 2×2, 1536-d , LN
		$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 768, head 24}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 768, head 24}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 1024, head 32}\end{bmatrix} \times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7, \\ \text{dim 1536, head 48}\end{bmatrix} \times 2$

表7. 详细架构规格说明。

模型作为初始化。我们对所有Swin Transformer模型采用随机深度，比例为0.2。

A2.3. 在ADE20K数据集上的语义分割

ADE20K [83] 是一个广泛使用的语义分割数据集，涵盖150个广泛的语义类别。该数据集共包含25,000张图像，其中20,000张用于训练，2,000张用于验证，另有3,000张用于测试。我们采用mmsegmentation [16]中的UperNet [69]作为基础框架，因其高效性。

在训练过程中，我们采用AdamW [44]优化器，初始学习率为6×10，权重衰减为0.01，并使用线性学习率衰减调度器及1500次迭代的线性预热。模型在8张GPU上进行训练，每张GPU处理2张图像，共训练16万次迭代。数据增强方面，我们采用mmsegmentation的默认设置，包括随机水平翻转、在[0.5, 2.0]比例范围内随机缩放以及随机光度失真。所有Swin Transformer模型均采用比例为0.2的随机深度策略。Swin-T和Swin-S采用与先前方法相同的标准设置，输入尺寸为512×512。标注‡的Swin-B和Swin-L表示这两个模型在ImageNet-22K上进行了预训练，并以640×640的输入尺寸进行训练。

在推理过程中，采用了多尺度测试，使用的分辨率是训练时的[0.5, 0.75, 1.0, 1.25, 1.5, 1.75]×倍。在报告测试分数时，遵循常见做法[71]，同时使用训练图像和验证图像进行训练。

A3. 更多实验

A3.1. 不同输入尺寸下的图像分类

表8列出了Swin Transformer在不同输入图像尺寸（从224^2到384^2）下的性能。总体而言，更大的输入分辨率会带来更高的top-1准确率，但推理速度会变慢。

	Swin-T		Swin-S		Swin-B	
input size	top-1 acc	throughput (image / s)	top-1 acc	throughput (image / s)	top-1 acc	throughput (image / s)
224^2	81.3	755.2	83.0	436.9	83.3	278.1
256^2	81.6	580.9	83.4	336.7	83.7	208.1
320^2	82.1	342.0	83.7	198.2	84.0	132.0
384^2	82.2	219.5	83.9	127.6	84.5	84.7

表8. 不同输入图像尺寸的Swin Transformers在ImageNet-1K分类任务上的表现。

Backbone	Optimizer	AP^{box}	$\text{AP}^{\text{box}}_{50}$	$\text{AP}^{\text{box}}_{75}$	AP^{mask}	$\text{AP}^{\text{mask}}_{50}$	$\text{AP}^{\text{mask}}_{75}$
R50	SGD	45.0	62.9	48.8	38.5	59.9	41.4
	AdamW	46.3	64.3	50.5	40.1	61.7	43.4
X101-32x4d	SGD	47.8	65.9	51.9	40.4	62.9	43.5
	AdamW	48.1	66.5	52.4	41.6	63.9	45.2
X101-64x4d	SGD	48.8	66.9	53.0	41.4	63.9	44.7
	AdamW	48.3	66.4	52.3	41.7	64.0	45.1

表9. 在COCO目标检测任务中，使用Cascade Mask R-CNN框架，对ResNe(X)t骨干网络采用SGD与AdamW优化器的比较。

A3.2. COCO数据集上ResNe(X)t的不同优化器

表9比较了ResNe(X)t主干网络在COCO目标检测任务上使用AdamW与SGD优化器的效果。此项比较采用Cascade Mask R-CNN框架进行。虽然Cascade Mask R-CNN框架默认使用SGD优化器，但我们发现将其替换为AdamW优化器通常能提升精度，对于较小规模的主干网络尤为明显。因此，在与提出的Swin Transformer架构进行对比时，我们对ResNe(X)t主干网络均采用AdamW优化器。

A3.3. Swin MLP-Mixer

我们将提出的分层设计和移位窗口方法应用于MLP-Mixer架构[61]，称之为Swin-Mixer。表10展示了Swin-Mixer与原始MLP-Mixer架构[61]及其后续改进方案的性能对比。

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
MLP-Mixer-B/16 [61]	224^2	59M	12.7G	-	76.4
ResMLP-S24 [62]	224^2	30M	6.0G	715	79.4
ResMLP-B24 [62]	224^2	116M	23.0G	231	81.0
Swin-T/D24 (Transformer)	256^2	28M	5.9G	563	81.6
Swin-Mixer-T/D24	256^2	20M	4.0G	807	79.4
Swin-Mixer-T/D12	256^2	21M	4.0G	792	79.6
Swin-Mixer-T/D6	256^2	23M	4.0G	766	79.7
Swin-Mixer-B/D24 (no shift)	224^2	61M	10.4G	409	80.3
Swin-Mixer-B/D24	224^2	61M	10.4G	409	81.3

Table 10. Performance of Swin MLP-Mixer on ImageNet-1K classification. D indictes the number of channels per head. Throughput is measured using the GitHub repository of [68] and a V100 GPU, following [63].

proach, ResMLP [61]. Swin-Mixer performs significantly better than MLP-Mixer (81.3% vs. 76.4%) using slightly smaller computation budget (10.4G vs. 12.7G). It also has better speed accuracy trade-off compared to ResMLP [62]. These results indicate the proposed hierarchical design and the shifted window approach are generalizable.

References

[1] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning*, pages 642–652. PMLR, 2020. 5

[2] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020. 3

[3] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks, 2020. 3

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 7

[5] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms – improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 6, 9

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018. 6, 9

[7] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 3, 6, 7, 9

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 3, 6, 9

[9] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019. 6, 9

[10] 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. 6, 9

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 7

[12] Yihong Chen, Zheng Zhang, Yue Cao, Liwei Wang, Stephen Lin, and Han Hu. Reppoints v2: Verification meets regression for object detection. In *NeurIPS*, 2020. 6, 7, 9

[13] Cheng Chi, Fangyun Wei, and Han Hu. Relationnet++: Bridging visual representations for object detection via transformer decoder. In *NeurIPS*, 2020. 3, 7

[14] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. 8, 9

[15] Xiangxiang Chu, Bo Zhang, Zhi Tian, Xiaolin Wei, and Huaxia Xia. Do we really need explicit position encodings for vision transformers? *arXiv preprint arXiv:2102.10882*, 2021. 3

[16] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020. 8, 10

[17] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 9

[18] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 764–773, 2017. 1, 3

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
MLP-Mixer-B/16 [61]	224^2	59M	12.7G	-	76.4
ResMLP-S24 [62]	224^2	30M	6.0G	715	79.4
ResMLP-B24 [62]	224^2	116M	23.0G	231	81.0
Swin-T/D24 (Transformer)	256^2	28M	5.9G	563	81.6
Swin-Mixer-T/D24	256^2	20M	4.0G	807	79.4
Swin-Mixer-T/D12	256^2	21M	4.0G	792	79.6
Swin-Mixer-T/D6	256^2	23M	4.0G	766	79.7
Swin-Mixer-B/D24 (no shift)	224^2	61M	10.4G	409	80.3
Swin-Mixer-B/D24	224^2	61M	10.4G	409	81.3

表10. Swin MLP-Mixer在ImageNet-1K分类任务上的性能。*D* 表示每个注意力头对应的通道数。吞吐量测试遵循[63]的方法，使用[68]的GitHub仓库及V100 GPU进行测量。

方法，ResMLP [61]。Swin-Mixer 在略小的计算预算下（10.4G vs. 12.7G）表现显著优于 MLP-Mixer（81.3% vs. 76.4%）。与 ResMLP [62] 相比，它也具有更好的速度-精度权衡。这些结果表明，所提出的分层设计和移位窗口方法具有可推广性。

参考文献

[1] 鲍航波、董力、韦福如、王文辉、杨楠、刘小东、王宇、高剑峰、朴松浩、周明等。Unilmv2：用于统一语言模型预训练的伪掩码语言模型。载于*International Conference on Machine Learning*，第642–652页。PMLR，2020年。5[2] Josh Beal、Eric Kim、Eric Tzeng、Dong Huk Park、Andrew Zhai、Dmitry Kislyuk。迈向基于Transformer的目标检测。*arXiv preprint arXiv:2012.09958*，2020年。3[3] Irwan Bello、Barret Zoph、Ashish Vaswani、Jonathon Shlens、Quoc V. Le。注意力增强卷积网络，2020年。3[4] Alexey Bochkovskiy、王建尧、廖弘源。Yolov4：目标检测的最佳速度与精度。*arXiv preprint arXiv:2004.10934*，2020年。7[5] Navaneeth Bodla、Bharat Singh、Rama Chellappa、Larry S. Davis。Soft-nms——用一行代码改进目标检测。载于*Proceedings of the IEEE International Conference on Computer Vision (ICCV)*，2017年10月。6, 9[6] 蔡兆伟、Nuno Vasconcelos。Cascade R-CNN：深入高质量目标检测。载于*Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni-tion*，第6154–6162页，2018年。6, 9[7] 曹越、徐佳瑞、林达华、韦芳云、胡瀚。GCNet：非局部网络与挤压激励网络及其超越。载于*Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*，2019年10月。3, 6, 7, 9[8] Nicolas Carion、Francisco Massa、Gabriel Synnaeve、Nicolas Usunier、Alexander Kirillov、Sergey Zagoruyko。端到结束基于Transformer的目标检测。于*European Conference on Computer Vision*，第213–229页。Springer，2020年。3, 6, 9 [9] 陈凯，庞江淼，王嘉琪，熊宇，李晓晓，孙舒阳，冯万森，刘子威，史建平，欧阳万里等。用于实例分割的混合任务级联。于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第4974–4983页，2019年。6, 9 [10] 陈凯，王嘉琪，庞江淼，曹宇航，熊宇，李晓晓，孙舒阳，冯万森，刘子威，徐佳瑞等。MMDetection：OpenMMLab检测工具箱与基准测试。*arXiv preprint arXiv:1906.07155*，2019年。6, 9 [11] 陈良杰，朱宇坤，George Papandreou，Florian Schroff，Hartwig Adam。用于语义图像分割的带空洞可分离卷积的编码器-解码器。于*Proceedings of the European conference on computer vision (ECCV)*，第801–818页，2018年。7 [12] 陈奕宏，张政，曹越，王立伟，林斯蒂芬，胡翰。RepPoints v2：目标检测中验证与回归的结合。于*NeurIPS*，2020年。6, 7, 9 [13] 池成，魏方耘，胡翰。RelationNet++：通过Transformer解码器桥接目标检测的视觉表示。于*NeurIPS*，2020年。3, 7 [14] Krzysztof Marcin Choromanski，Valerii Likhosherstov，David Dohan，Xingyou Song，Andreea Gane，Tamas Sarlos，Peter Hawkins，Jared Quincy Davis，Afroz Mohiuddin，Lukasz Kaiser，David Benjamin Belanger，Lucy J Colwell，Adrian Weller。重新思考注意力机制：Performers。于*International Conference on Learning Representations*，2021年。8, 9 [15] 初祥祥，张博，田智，魏晓林，夏华霞。视觉Transformer真的需要显式位置编码吗？*arXiv preprint arXiv:2102.10882*，2021年。3 [16] MMSegmentation贡献者。MMSegmentation：OpenMMLab语义分割工具箱与基准测试。https://github.com/open-mmlab/mmsegmentation，2020年。8, 10 [17] Ekin D Cubuk，Barret Zoph，Jonathon Shlens，Quoc V Le。RandAugment：具有缩减搜索空间的实用自动数据增强。于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*，第702–703页，2020年。9 [18] 戴继峰，齐浩志，熊宇闻，李毅，张国栋，胡翰，魏亦宸。可变形卷积网络。于*Proceedings of the IEEE International Conference on Computer Vision*，第764–773页，2017年。1, 3 [19] 邓嘉，董伟，Richard Socher，李立佳，李凯，李飞飞。ImageNet：一个大规模分层图像数据库。于*2009 IEEE conference on computer vision and pattern recognition*，第248–255页。IEEE，2009年。5 [20] Alexey Dosovitskiy，Lucas Beyer，Alexander Kolesnikov，Dirk Weissenborn，翟晓华，Thomas Unterthiner，Mostafa Dehghani，Matthias Minderer，Georg Heigold，Sylvain Gelly，Jakob Uszkoreit，Neil Houlsby。一幅图像即

worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 4, 5, 6, 9

[21] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V Le, and Xiaodan Song. Spinenet: Learning scale-permuted backbone for recognition and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11592–11601, 2020. 7

[22] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 682–691, 2019. 6, 9

[23] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 3, 7

[24] Jun Fu, Jing Liu, Yuhang Wang, Yong Li, Yongjun Bao, Jinhui Tang, and Hanqing Lu. Adaptive context network for scene parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6748–6757, 2019. 7

[25] Kunihiko Fukushima. Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, 20(3):121–136, 1975. 3

[26] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. *arXiv preprint arXiv:2012.07177*, 2020. 2, 7

[27] Jiayuan Gu, Han Hu, Liwei Wang, Yichen Wei, and Jifeng Dai. Learning region features for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[28] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021. 3

[29] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 6, 9

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2, 4

[31] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138, 2020. 6, 9

[32] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 3, 5

[33] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3464–3473, October 2019. 2, 3, 5

[34] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1, 2

[35] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. 9

[36] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962. 3

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5, 9

[38] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6(2):8, 2019. 6

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 2

[40] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[41] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999. 3

[42] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[43] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6, 9, 10

[45] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 6, 9

[46] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. *arXiv preprint arXiv:2006.02334*, 2020. 2, 7

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1

价值16x16个词：用于大规模图像识别的Transformer。在*International Conference on Learning Representa- tions*，2021年。1, 2, 3, 4, 5, 6, 9 [21] 杜先知、林宗毅、金鹏冲、Golnaz Ghiasi、谭明星、崔寅、Quoc V Le、宋晓丹。SpineNet：学习用于识别与定位的尺度置换骨干网络。在 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第11592–11601页，2020年。7 [22] 方浩舒、孙建华、王润中、荀明浩、李永禄、卢策武。InstaBoost：通过概率图引导的复制粘贴提升实例分割。在 *Proceedings of the IEEE/CVF International Conference on Computer Vision*，第682–691页，2019年。6, 9 [23] 傅俊、刘晶、田海杰、李勇、包永军、方志伟、卢汉卿。用于场景分割的双注意力网络。在 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*，第3146–3154页，2019年。3, 7 [24] 傅俊、刘晶、王宇航、李勇、包永军、唐金辉、卢汉卿。用于场景解析的自适应上下文网络。在 *Proceedings of the IEEE/CVF International Conference on Computer Vision*，第6748–6757页，2019年。7 [25] 福岛邦彦。认知机：一种自组织的多层神经网络。*Biological cybernetics*，20(3):121–136，1975年。3 [26] Golnaz Ghiasi、崔寅、Aravind Srinivas、钱瑞、林宗毅、Ekin D Cubuk、Quoc V Le、Barret Zoph。简单的复制粘贴是一种强大的实例分割数据增强方法。*arXiv preprint arXiv:2012.07177*，2020年。2, 7 [27] 顾家园、胡涵、王立伟、魏亦宸、代季峰。学习用于目标检测的区域特征。在*Proceedings of the European Conference on Computer Vision (ECCV)*，2018年。3 [28] 韩凯、肖安、吴恩华、郭建元、徐春景、王云鹤。Transformer中的Transformer。*arXiv preprint arXiv:2103.00112*，2021年。3 [29] 何恺明、Georgia Gkioxari、Piotr Dollár、Ross Girshick。Mask R-CNN。在 *Proceedings of the IEEE international conference on computer vision*，第2961–2969页，2017年。6, 9 [30] 何恺明、张祥雨、任少卿、孙剑。用于图像识别的深度残差学习。在*Proceedings of the IEEE conference on computer vision and pattern recognition*，第770–778页，2016年。1, 2, 4 [31] Elad Hoffer、Tal Ben-Nun、Itay Hubara、Niv Giladi、Torsten Hoefler、Daniel Soudry。增强你的批次：通过实例重复改进泛化能力。在*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第8129–8138页，2020年。6, 9 [32] 胡涵、顾家园、张政、代季峰、魏亦宸。用于目标检测的关系网络。在 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*，第3588–3597页，2018年。3, 5 [33] 胡涵、张政、谢振达、林史蒂芬。用于图像识别的局部关系网络。在 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*，第3464–3473页，2019年10月。2, 3, 5 [34] 高黄、庄刘、劳伦斯·范德马滕和基利安·Q温伯格。密集连接卷积网络。于 *Proceedings of the IEEE conference on computer vision and pattern recognition*，第4700–4708页，2017年。1, 2 [35] 高黄、余孙、庄刘、丹尼尔·塞德拉和基利安·Q温伯格。具有随机深度的深度网络。于 *European conference on computer vision*，第646–661页。施普林格，2016年。9 [36] 戴维·H休伯尔和托尔斯滕·N维泽尔。猫视觉皮层中的感受野、双眼交互及功能架构。*The Journal of physiology*，160(1):106–154，1962年。3 [37] 迪德里克·P金马和吉米·巴。Adam：一种随机优化方法。*arXiv preprint arXiv:1412.6980*，2014年。5, 9 [38] 亚历山大·科列斯尼科夫、卢卡斯·拜尔、翟晓华、琼·普伊格塞尔韦、杰西卡·扬、西尔万·盖利和尼尔·霍尔斯比。大迁移（BiT）：通用视觉表示学习。*arXiv preprint arXiv:1912.11370*，6(2):8，2019年。6 [39] 亚历克斯·克里热夫斯基、伊利亚·苏茨克弗和杰弗里·E辛顿。使用深度卷积神经网络进行ImageNet分类。于 *Advances in neural information processing sys- tems*，第1097–1105页，2012年。1, 2 [40] 扬·勒昆、Léon Bottou、约书亚·本吉奥、帕特里克·哈夫纳等。基于梯度的学习在文档识别中的应用。*Proceedings of the IEEE*，86(11):2278–2324，1998年。2 [41] 扬·勒昆、帕特里克·哈夫纳、Léon Bottou和约书亚·本吉奥。使用基于梯度的学习进行物体识别。于 *Shape, contour and grouping in computer vision*，第319–345页。施普林格，1999年。3 [42] 林腾毅、皮奥特·多拉尔、罗斯·吉尔希克、何恺明、巴拉特·哈里哈兰和谢尔盖·贝隆吉。用于物体检测的特征金字塔网络。于 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*，2017年7月。2 [43] 林腾毅、迈克尔·梅尔、谢尔盖·贝隆吉、詹姆斯·海斯、彼得罗·佩罗纳、德瓦·拉马南、皮奥特·Dollár和C·劳伦斯·齐特尼克。Microsoft COCO：上下文中的常见物体。于 *European conference on computer vision*，第740–755页。施普林格，2014年。5 [44] 伊利亚·洛什奇洛夫和弗兰克·胡特。解耦权重衰减正则化。于 *International Conference on Learning Representations*，2019年。6, 9, 10 [45] 鲍里斯·T波利亚克和阿纳托利·B朱迪茨基。通过平均加速随机逼近。*SIAM journal on control and optimization*，30(4):838–855，1992年。6, 9 [46] 乔思远、陈亮杰和艾伦·尤伊尔。Detectors：使用递归特征金字塔和可切换空洞卷积检测物体。*arXiv preprint arXiv:2006.02334*，2020年。2, 7 [47] 亚历克·拉德福德、金钟旭、克里斯·哈拉西、阿迪蒂亚·拉梅什、加布里埃尔·戈、桑迪尼·阿加瓦尔、吉里什·萨斯特里、阿曼达·阿斯克尔、帕梅拉·米什金、杰克·克拉克、格雷琴·克鲁格和伊利亚·苏茨克弗。从自然语言监督中学习可迁移的视觉模型，2021年。1

[48] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020. 6

[49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 5

[50] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2, 3

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[52] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, May 2015. 2, 4

[53] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3578–3587, 2018. 2

[54] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2

[55] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021. 3

[56] 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. *arXiv preprint arXiv:2011.12450*, 2020. 3, 6, 9

[57] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[58] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 3, 6

[59] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 7

[60] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021. 8

[61] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021. 2, 10, 11

[62] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training, 2021. 11

[63] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 2, 3, 5, 6, 9, 11

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 1, 2, 4

[65] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 3

[66] 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. 3

[67] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, 2018. 3

[68] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 6, 11

[69] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 7, 8, 10

[70] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017. 1, 2, 3

[71] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, 2020. 3, 7, 10

[72] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, 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. 3

[73] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In

[48] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. 设计网络设计空间。于 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 页 10428–10436, 2020. 6[49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 探索统一文本到文本Transformer的迁移学习极限。*Journal of Machine Learn- ing Research*, 21(140):1–67, 2020. 5[50] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. 视觉模型中的独立自注意力。于 *Advances in Neural Information Processing Systems*, 卷 32. Curran Associates, Inc., 2019. 2, 3[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: 用于生物医学图像分割的卷积网络。于 *International Conference on Medical image computing and computer-assisted intervention*, 页 234–241. Springer, 2015. 2[52] K. Simonyan and A. Zisserman. 用于大规模图像识别的极深卷积网络。于 *International Conference on Learning Representations*, 2015年5月. 2, 4[53] Bharat Singh and Larry S Davis. 目标检测SNIP中尺度不变性分析。于 *Proceedings of the IEEE conference on computer vision and pattern recogni- tion*, 页 3578–3587, 2018. 2[54] Bharat Singh, Mahyar Najibi, and Larry S Davis. SNIPER: 高效多尺度训练。于 *Advances in Neural Infor- mation Processing Systems*, 卷 31. Curran Associates, Inc., 2018. 2[55] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. 用于视觉识别的瓶颈Transformer。*arXiv preprint arXiv:2101.11605*, 2021. 3[56] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, 等. Sparse R-CNN: 使用可学习建议框的端到端目标检测。*arXiv preprint arXiv:2011.12450*, 2020. 3, 6, 9[57] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 深入卷积网络。于 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 页 1–9, 2015. 2[58] Mingxing Tan and Quoc Le. EfficientNet: 重新思考卷积神经网络的模型缩放。于 *International Conference on Machine Learning*, 页 6105–6114. PMLR, 2019. 3, 6[59] Mingxing Tan, Ruoming Pang, and Quoc V Le. EfficientDet: 可扩展且高效的目标检测。于 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 页 10781–10790, 2020. 7[60] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long Range Arena: 一个基准高效Transformer的标记。在*International Conference on Learning Representations*，2021年。8 [61] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, 和 Alexey Dosovitskiy。MLP-Mixer：一种用于视觉的全MLP架构，2021年。2, 10, 11 [62] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, 和 Hervé Jégou。ResMLP：用于图像分类的前馈网络，具有数据高效训练，2021年。11 [63] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, 和 Hervé Jégou。训练数据高效的图像Transformer和通过注意力的蒸馏。*arXiv preprint arXiv:2012.12877*，2020年。2, 3, 5, 6, 9, 11 [64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, 和 Illia Polosukhin。注意力就是您所需要的全部。在*Advances in Neural Information Processing Systems*，第5998–6008页，2017年。1, 2, 4 [65] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, 等。用于视觉识别的深度高分辨率表示学习。*IEEE transactions on pattern analysis and machine intelligence*，2020年。3 [66] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, 和 Ling Shao。金字塔视觉Transformer：一种用于密集预测的无卷积多功能骨干网络。*arXiv preprint arXiv:2102.12122*，2021年。3 [67] Xiaolong Wang, Ross Girshick, Abhinav Gupta, 和 Kaiming He。非局部神经网络。在*IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*，2018年。3 [68] Ross Wightman。PyTorch图像模型。https://github.com/rwightman/pytorch-image-models，2019年。6, 11 [69] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, 和 Jian Sun。用于场景理解的统一感知解析。在*Proceedings of the European Conference on Com- puter Vision (ECCV)*，第418–434页，2018年。7, 8, 10 [70] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, 和 Kaiming He。用于深度神经网络的聚合残差变换。在*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*，第1492–1500页，2017年。1, 2, 3 [71] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, 和 Han Hu。解耦的非局部神经网络。在*Proceedings of the European conference on computer vision (ECCV)*，2020年。3, 7, 10 [72] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, 和 Shuicheng Yan。Tokens-to-Token ViT：在ImageNet上从头开始训练视觉Transformer。*arXiv preprint arXiv:2101.11986*，2021年。3 [73] Yuhui Yuan, Xilin Chen, 和 Jingdong Wang。用于语义分割的对象上下文表示。在

16th European Conference Computer Vision (ECCV 2020), August 2020. 7

[74] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 3

[75] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 9

[76] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 1

[77] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 9

[78] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 7, 8

[79] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020. 6, 9

[80] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10076–10085, 2020. 3

[81] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020. 2, 3, 7, 8

[82] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020. 9

[83] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision*, 2018. 5, 7, 10

[84] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 1, 3

[85] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 3

16th European Conference Computer Vision (ECCV 2020)，2020年8月。7 [74] 袁宇辉和王井东。Ocnet：用于场景解析的对象上下文网络。*arXiv preprint arXiv:1809.00916*，2018年。3 [75] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe和Youngjoon Yoo。Cutmix：一种训练具有可定位特征的强分类器的正则化策略。在*Proceedings of the IEEE/CVF International Conference on Computer Vision*中，第6023–6032页，2019年。9 [76] Sergey Zagoruyko和Nikos Komodakis。宽残差网络。在*BMVC*中，2016年。1 [77] 张弘毅、Moustapha Cisse、Yann N Dauphin和David Lopez-Paz。mixup：超越经验风险最小化。*arXiv preprint arXiv:1710.09412*，2017年。9 [78] 张航、吴崇若、张忠岳、朱毅、张植、林海滨、孙悦、何彤、Jonas Mueller、R Manmatha等。Resnest：拆分注意力网络。*arXiv preprint arXiv:2004.08955*，2020年。7, 8 [79] 张世峰、迟成、姚永强、雷震和李Stan Z。通过自适应训练样本选择弥合基于锚点与无锚点检测之间的差距。在*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*中，第9759–9768页，2020年。6, 9 [80] 赵恒双、贾家亚和Vladlen Koltun。探索自注意力机制用于图像识别。在*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*中，第10076–10085页，2020年。3 [81] 郑思晓、卢嘉晨、赵恒双、朱霞天、罗泽坤、王亚彪、傅彦伟、冯建峰、项涛、Philip HS Torr等。从序列到序列的视角重新思考语义分割：基于Transformer的方法。*arXiv preprint arXiv:2012.15840*，2020年。2, 3, 7, 8 [82] 钟准、郑亮、康国亮、李少子和杨毅。随机擦除数据增强。在*Proceedings of the AAAI Conference on Artificial Intelligence*中，第34卷，第13001–13008页，2020年。9 [83] 周博磊、赵航、Xavier Puig、肖特特、Sanja Fidler、Adela Barriuso和Antonio Torralba。通过ADE20K数据集理解场景语义。*International Journal on Computer Vision*，2018年。5, 7, 10 [84] 朱希洲、胡涵、林世峰和戴继峰。可变形卷积网络v2：更可变形，更好结果。在*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*中，第9308–9316页，2019年。1, 3 [85] 朱希洲、苏伟杰、卢乐威、李斌、王晓刚和戴继峰。可变形{DETR}：用于端到端目标检测的可变形Transformer。在*International Confer- ence on Learning Representations*中，2021年。3