

一张图片相当于16x16个单词： 用于大规模图像识别的TRANSFORMER

Alexey Dosovitskiy^{*,†}, Lucas Beyer^{*}, Alexander Kolesnikov^{*}, Dirk Weissenborn^{*},
Xiaohua Zhai^{*}, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby^{*,†}

^{*}equal technical contribution, [†]equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

虽然Transformer架构已经成为自然语言处理任务的事实标准，但其在计算机视觉中的应用仍然有限。在视觉中，注意力要么与卷积网络一起应用，要么用于替换卷积网络的某些组件，同时保持其整体结构不变。对cnn的这种依赖是不必要的，直接应用于图像块序列的纯transformer可以在图像分类任务上表现很好。当在大量数据上进行预训练并转移到多个中小型图像识别基准(ImageNet、CIFAR-100、VTAB等)时，与最先进的卷积网络相比，Vision Transformer (ViT)取得了出色的结果，同时所需的训练计算资源大大减少。¹

1 简介

基于自注意力的架构，特别是transformer (Vaswani et al., 2017)，已经成为自然语言处理(NLP)的首选模型。主流的方法是在大型文本语料库上进行预训练，然后在较小的特定任务数据集(Devlin et al., 2019)上进行微调。由于transformer的计算效率和可扩展性，训练具有超过100B参数的空前规模的模型成为可能(Brown et al., 2020; Lepikhin et al., 2020)。随着模型和数据集的增长，性能仍然没有饱和的迹象。

然而，在计算机视觉中，卷积架构仍然占主导地位(LeCun et al., 1989; Krizhevsky et al., 2012; He et al., 2016)。受NLP成功的启发，许多工作尝试将类似cnn的架构与自注意力结合(Wang et al., 2018; Carion et al., 2020)，一些工作完全取代卷积(Ramachandran et al., 2019; Wang et al., 2020a)。后者虽然理论上很有效，但由于使用了专门的注意力模式，尚未在现代硬件加速器上有效扩展。因此，在大规模图像识别中，经典的类似resnet的架构仍然是最先进的(Mahajan et al., 2018; Xie et al., 2020; Kolesnikov et al., 2020)。

受Transformer在NLP中缩放成功的启发，实验将标准Transformer直接应用于图像，并进行尽可能少的修改。为此，将图像分割为块，并提供这些块的线性嵌入序列作为Transformer的输入。在NLP应用中，图像块与标记(单词)的处理方式相同。以有监督的方式在图像分类上训练模型。

当在没有强正则化的中型数据集(如ImageNet)上进行训练时，这些模型产生的精度比同等大小的ResNets低几个百分点。这种看似令人沮丧的结果是可以预期的：transformer缺乏cnn固有的一些归纳偏差，如平移等变性和局部性，因此在数据量不足时，泛化效果不好。

然而，如果模型在更大的数据集(14M-300M图像)上训练，图像会发生变化。大规模训练胜过归纳偏差。当以足够的规模进行预训练并转移到具有较少数据点的任务时，Vision Transformer (ViT)取得了出色的结果。当在公共ImageNet-21k数据集或内部JFT-300M数据集上进行预训练时，ViT在多个图像识别基准上接近或超过了最先进水平。特别是，最好的模型在19个任务的VTAB套件上达到了88.55%在ImageNet上的精度，90.72%在ImageNet-ReaL上的精度，94.55%在CIFAR-100上的精度，77.63%。

¹微调代码和预训练模型可在https://github.com/google-research/vision_transformer

2 相关工作

transformer由Vaswani et al. (2017)提出用于机器翻译，并已成为许多NLP任务中的最先进方法。基于transformer的大型模型通常在大型语料库上进行预训练，然后针对手头的任务进行微调：BERT (Devlin et al., 2019)使用去噪自监督预训练任务，而GPT工作使用语言建模作为预训练任务(Radford et al., 2018; 2019; Brown et al., 2020)。

简单地将自注意力应用于图像，需要每个像素关注每个其他像素。由于像素数量为二次成本，这无法扩展到实际的输入大小。因此，为了将transformer应用于图像处理，过去曾尝试过几种近似方法。Parmar et al. (2018)仅将自注意力应用于每个查询像素的局部邻域，而不是全局。这种局部多头点积自注意力块可以完全取代卷积(Hu et al., 2019; Ramachandran et al., 2019; Zhao et al., 2020)。在另一种工作中，稀疏transformer (Child et al., 2019)采用可扩展的全局自注意力近似，以便适用于图像。扩展注意力的另一种方法是将其应用于不同大小的块(Weissenborn et al., 2019)，在极端情况下仅沿单个轴(Ho et al., 2019; Wang et al., 2020a)。许多这些专门的注意力架构在计算机视觉任务上表现出了有希望的结果，但需要复杂的工程才能在硬件加速器上有效实现。

与我们最相关的是Cordonnier et al. (2020)模型，它从输入图像中提取大小为 2×2 的块，并将全部自注意力应用于顶部。该模型与ViT非常相似，但本文工作进一步证明，大规模预训练使普通transformer与最先进的cnn竞争(甚至优于)。此外，Cordonnier et al. (2020)使用 2×2 像素的小patch大小，这使得模型仅适用于小分辨率图像，而我们也处理中分辨率图像。

人们对将卷积神经网络(CNN)与自注意力形式相结合也很感兴趣，例如通过增强用于图像分类的特征图(Bello et al., 2019)或通过使用自注意力进一步处理CNN的输出，例如用于目标检测(Hu et al., 2018; Carion et al., 2020)，视频处理(Wang et al., 2018; Sun et al., 2019)，图像分类(Wu et al., 2020)，无监督目标发现(Locatello et al., 2020)或统一文本视觉任务(Chen et al., 2020c; Lu et al., 2019; Li et al., 2019)。

另一个最近的相关模型是图像GPT (iGPT) (Chen et al., 2020a)，它在降低图像分辨率和颜色空间后将transformer应用于图像像素。该模型以无监督的方式作为生成模型进行训练，然后可以对产生的表示进行微调或线性探索分类性能，在ImageNet上实现72%的最大精度。

我们的工作增加了越来越多的论文集合，这些论文探索了比标准ImageNet数据集更大的尺度的图像识别。使用其他数据源可以在标准基准(Mahajan et al., 2018; Touvron et al., 2019; Xie et al., 2020)上取得最先进的结果。此外，Sun et al. (2017)研究了CNN性能如何随数据集大小而变化，Kolesnikov et al. (2020); Djolonga et al. (2020)从ImageNet-21k和JFT-300M等大规模数据集对CNN迁移学习进行了经验探索。我们也关注后两个数据集，但训练transformer，而不是之前工作中使用的基于resnet的模型。

3 方法

在模型设计中，我们尽可能地遵循原始Transformer (Vaswani et al., 2017)。这种故意简单设置的优势是可扩展的NLP Transformer架构及其高效实现，可以几乎开箱即用。

3.1 VISION TRANSFORMER (ViT)

该模型的概述见图1。标准转换器接收一个1维标记嵌入序列作为输入。为了处理2D图像，我们将图像 $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ 重塑为一个扁平的2D补丁序列 $\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$ ，其中 (H, W) 是原始图像的分辨率， C 是通道的数量， (P, P) 是每个图像补丁的分辨率， $N = HW/P^2$ 是结果补丁的数量，这也作为Transformer的有效输入序列长度。Transformer在其所有层中使用恒定的潜向量大小 D ，因此我们将补丁展平并通过可训练的线性投影映射到 D 维度(例如1)。我们将此投影的输出称为补丁嵌入。

类似于BERT的[`class`]标记，我们在嵌入式补丁序列($\mathbf{z}_0^0 = \mathbf{x}_{\text{class}}$)前添加一个可学习的嵌入，其在Transformer编码器输出处的状态(\mathbf{z}_L^0)用作图像表示(Eq. 4)。在预训练和微调期间，分类头都连接到 \mathbf{z}_L^0 。分类头由一个在预训练时具有一个隐藏层的MLP实现，在微调时由一个单一的线性层实现。

位置嵌入被添加到补丁嵌入中以保留位置信息。我们使用标准的可学习的1D位置嵌入，因为我们没有观察到使用更高级的2d感知位置嵌入带来的显著性能提升(附录D.4)。得到的嵌入向量序列作为编码器的输入。

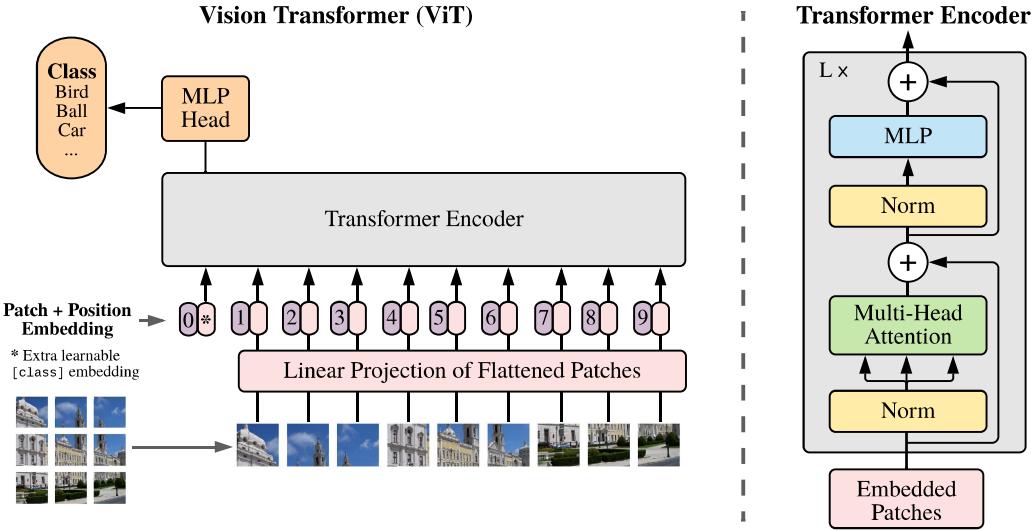


Figure 1: 模型概述。将图像分割为固定大小的块，对每个块进行线性嵌入，添加位置嵌入，并将得到的向量序列提供给标准的Transformer编码器。为了进行分类，我们使用标准的方法，即向序列中添加一个额外的可学习的“分类标记”。Transformer编码器的插图灵感来自Vaswani et al. (2017)。

Transformer编码器(Vaswani et al., 2017)由交替层的多头自注意力(见附录A)和MLP块(Eq. 2, 3)组成。在每个块之前应用Layernorm (LN)，在每个块之后应用残差连接(Wang et al., 2019; Baevski & Auli, 2019)。MLP包含两层，具有GELU非线性。

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \dots; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{\text{pos}}, \quad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D} \quad (1)$$

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \quad \ell = 1 \dots L \quad (2)$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \quad \ell = 1 \dots L \quad (3)$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \quad (4)$$

归纳偏差。Vision Transformer的图像特定归纳偏差比cnn小得多。在cnn中，局部性、二维邻域结构和平移等性被烘焙到整个模型的每一层中。在ViT中，只有MLP层是局部的和平移等变的，而自注意力层是全局的。二维邻域结构的使用非常谨慎：在模型开始时通过将图像切割成块，并在微调时调整不同分辨率图像的位置嵌入(如下所述)。除此之外，初始化时的位置嵌入不携带关于块的2D位置的信息，块之间的所有空间关系必须从头学习。

混合架构。作为原始图像块的替代方案，输入序列可以从CNN的特征图形成(LeCun et al., 1989)。在这个混合模型中，将块嵌入投影 \mathbf{E} (例如1)应用于从CNN特征图中提取的块。作为一个特例，patch的空间大小可以是 1×1 ，这意味着输入序列是通过简单地展平特征映射的空间维度并投影到Transformer维度来获得的。分类输入嵌入和位置嵌入按上述方式添加。

3.2 微调和更高的分辨率

通常，我们在大型数据集上预训练ViT，并对(较小的)下游任务进行微调。为此，我们删除了预训练的预测头，并附加了一个零初始化的 $D \times K$ 前馈层，其中 K 是下游类别的数量。与预训练相比，在更高的分辨率下进行微调通常是有益的(Touvron et al., 2019; Kolesnikov et al., 2020)。当输入更高分辨率的图像时，我们保持patch大小相同，这导致了更大的有效序列长度。Vision Transformer可以处理任意序列长度(直到内存限制)，但是，预训练的位置嵌入可能不再有意义。因此，根据预训练位置嵌入在原始图像中的位置，对其进行2D插值。请注意，这种分辨率调整和块提取是关于图像2D结构的归纳偏差被手动注入Vision Transformer的唯一点。

4 实验

评估了ResNet、Vision Transformer (ViT)和hybrid的表示学习能力。为了了解每个模型的数据需求，在不同规模的数据集上进行了预训练，并评估了许多基准任务。在考虑预训练模

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Vision Transformer模型变体的详细信息。

型的计算成本时，ViT的表现非常好，以较低的预训练成本在大多数识别基准上达到了最先进的水平。最后，使用自监督进行了一个小型实验，并表明自监督ViT具有未来的希望。

4.1 设置

数据集。为了探索模型的可扩展性，我们使用具有1k类和1.3M图像的ILSVRC-2012 ImageNet数据集(我们在下文中将其称为ImageNet)，它的超集ImageNet-21k具有21k类和14M张图像(Deng et al., 2009)，JFT (Sun et al., 2017)具有18k类和303M高分辨率图像。我们对预训练数据集进行重复数据消除，对下游任务的测试集进行重复Kolesnikov et al. (2020)。将在这些数据集上训练的模型迁移到几个基准任务：ImageNet在原始验证标签和清理后的真实标签(Beyer et al., 2020)，CIFAR-10/100 (Krizhevsky, 2009)，牛津-iiit宠物(Parkhi et al., 2012)，和牛津花-102 (Nilsback & Zisserman, 2008)。对于这些数据集，预处理如下Kolesnikov et al. (2020)。

我们还在19个任务的VTAB分类套件(Zhai et al., 2019b)上进行了评估。VTAB评估了不同任务的低数据传输，每个任务使用1 000个训练示例。任务分为以下三组。自然——像上面的任务，宠物，CIFAR等。专业，医学和卫星图像，还有结构化——需要几何理解的任务，如本地化。

模型变体。我们基于BERT (Devlin et al., 2019)使用的配置ViT，如表1所示。‘Base’和‘Large’模型直接来自BERT，我们添加了更大的‘Huge’模型。在接下来的内容中，我们使用简短的符号来表示模型大小和输入补丁大小：例如，ViT -L/16表示具有 16×16 输入补丁大小的“大”变体。请注意，Transformer的序列长度与patch大小的平方成反比，因此具有较小patch大小的模型在计算上更昂贵。

对于基线cnn，我们使用ResNet (He et al., 2016)，但将批归一化层(Ioffe & Szegedy, 2015)替换为组归一化层(Wu & He, 2018)，并使用标准化卷积(Qiao et al., 2019)。这些修改改进了传输(Kolesnikov et al., 2020)，我们表示修改后的模型“ResNet (BiT)”。对于混合模型，我们将中间特征图输入ViT，其patch大小为一个“像素”。我们也可以尝试不同的序列长度(i)取常规ResNet50或的第4阶段的输出(ii)删除阶段4，将相同的层数放在阶段3中(保持总层数)，并获得这个扩展阶段3的输出。选项(ii)导致4倍长的序列长度，以及更昂贵的ViT模型。

训练和微调。我们训练了包括ResNets在内的所有模型，使用Adam (Kingma & Ba, 2015)与 $\beta_1 = 0.9$, $\beta_2 = 0.999$ ，批量大小为4096，并应用0.1的高权重衰减，我们发现这对所有模型的迁移很有用(附录D.1表明，与常见实践相比，Adam在我们的设置中比ResNets的SGD工作得稍好)。我们使用线性学习率预热和衰减，详见附录B.1。对于微调，我们使用带有动量的SGD，批量大小为512，对于所有模型，请参见附录B.1.1。对于表2中的ImageNet结果，我们在更高的分辨率上进行了微调：ViT为512 -L/16, 518为ViT-H/14，还使用Polyak & Juditsky (1992)与0.9999(Ramachandran et al., 2019; Wang et al., 2020b)因子进行平均。

指标。通过少样本或微调精度报告下游数据集的结果。微调精度捕捉了每个模型在各自数据集上进行微调后的性能。通过解决正则化最小二乘回归问题来获得少样本精度，该问题将训练图像子集的(冻结)表示映射到 $\{-1, 1\}^K$ 目标向量。这个公式使我们能够以闭式形式得到精确的解。虽然主要关注微调性能，但有时也使用线性少样本精度来进行快速的实时评估，其中微调成本太高。

4.2 与最新技术的比较

首先将最大的模型ViT-H/14和ViT-L/16与文献中最先进的cnn进行了比较。第一个比较点是Big Transfer (BiT) (Kolesnikov et al., 2020)，它使用大型ResNets进行监督迁移学习。第二个是吵闹的学生(Xie et al., 2020)，这是一个大型的EfficientNet，使用ImageNet和JFT-300M上

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.55 ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReAL	90.72 ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	99.50 ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	—
CIFAR-100	94.55 ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	—
Oxford-IIIT Pets	97.56 ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	—
Oxford Flowers-102	99.68 ± 0.02	99.74 ± 0.00	99.61 ± 0.02	99.63 ± 0.03	—
VTAB (19 tasks)	77.63 ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	—
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: 在流行的图像分类基准上与最新技术进行比较。报告了精度的均值和标准偏差，在三次微调运行中平均。Vision Transformer在JFT-300M数据集上预训练的模型在所有数据集上的表现都优于基于resnet的基线，同时预训练所需的计算资源大大减少。ViT在较小的公共ImageNet-21k数据集上进行的预训练也表现良好。*略有改善88.5%结果报告在Touvron et al. (2020)。

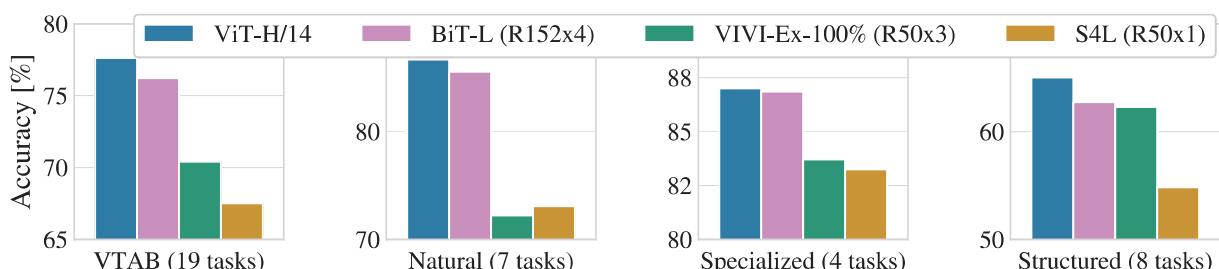


Figure 2: VTAB在自然、专门化和结构化任务组中的性能分解。

的半监督学习进行训练，并删除了标签。目前，Noisy Student在ImageNet上是最先进的，在这里报告的其他数据集上是BiT-L。所有模型都是在TPUv3硬件上训练的，我们报告了预训练每个模型所需的TPUv3核天数，即用于训练的TPUv3核数量(每个芯片2个)乘以以天为单位的训练时间。

表2显示了结果。在JFT-300M上预训练的更小的ViT-L/16模型在所有任务上的表现都优于BiT-L(在同一数据集上预训练)，同时训练所需的计算资源也大大减少。更大的模型ViT-H/14进一步提高了性能，特别是在更具挑战性的数据集上——ImageNet、CIFAR-100和VTAB套件。有趣的是，与之前的技术相比，该模型预训练所需的计算量仍然少得多。预训练效率不仅会受到架构选择的影响，还会受到其他参数的影响，如训练计划、优化器、权重衰减等。我们在4.4节中提供了不同架构的性能与计算的对照研究。最后，在公共ImageNet-21k数据集上预训练的ViT-L/16模型在大多数数据集上也表现良好，同时预训练所需的资源更少：它可以在大约30天内使用具有8核的标准云TPUv3进行训练。

图2将VTAB任务分解为各自的组，并在此基准上与以前的SOTA方法进行比较：BiT，VIVI——一个在ImageNet和Youtube上共同训练的ResNet (Tschanne et al., 2020)，S4L——监督加半监督学习ImageNet (Zhai et al., 2019a)。ViT -H/14在自然和结构化任务上的性能优于BiT-R152x4和其他方法。在专门化方面，前两种模型的性能相近。

4.3 预训练数据要求

Vision Transformer在大型JFT-300M数据集上进行预训练时表现良好。由于视觉的归纳偏差比ResNets更少，那么数据集大小有多重要？我们进行了两个系列的实验。

首先，在越来越大的数据集上预训练ViT模型：ImageNet、ImageNet-21k和JFT-300M。为了提高在较小数据集上的性能，优化了三个基本正则化参数——权重衰减、dropout和标签平滑。图3显示了微调到ImageNet后的结果(其他数据集的结果显示在表5中)²。当在最小数据集上进行预训练时，ImageNet，ViT -大型模型与ViT -基础模型相比表现欠佳，尽管(适度)正

²请注意，ImageNet预训练模型也进行了微调，但还是在ImageNet上。这是因为微调期间分辨率的增加提高了性能。

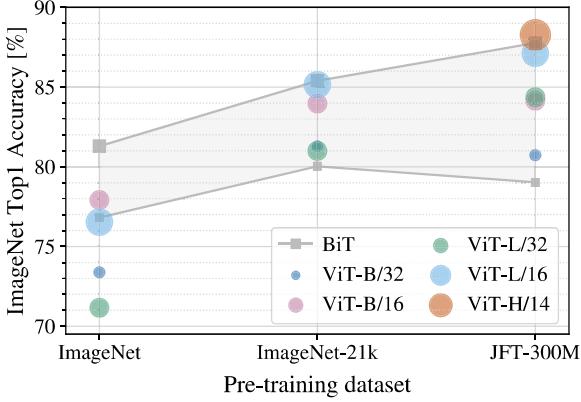


Figure 3: 转至ImageNet。虽然在小数据集上预训练时，大型ViT模型的表现比BiT ResNets(阴影区域)差，但在更大的数据集上预训练时，它们会大出风头。类似地，随着数据集的增长，较大的ViT变体超过较小的变体。

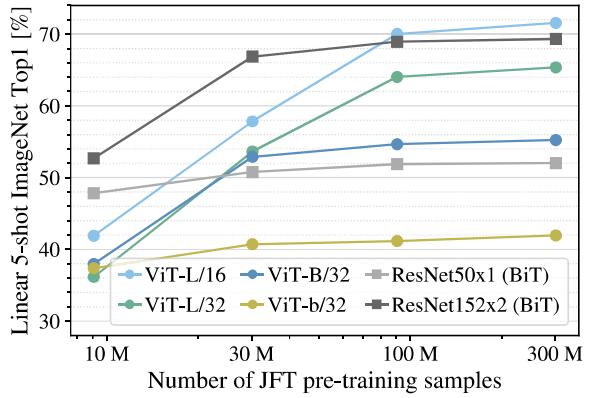


Figure 4: ImageNet与预训练大小的线性少样本评估。ResNets在较小的预训练数据集上表现更好，但比ViT更快地停滞不前，而在较大的预训练数据集上表现更好。ViT-b为ViT-b，所有隐藏尺寸减半。

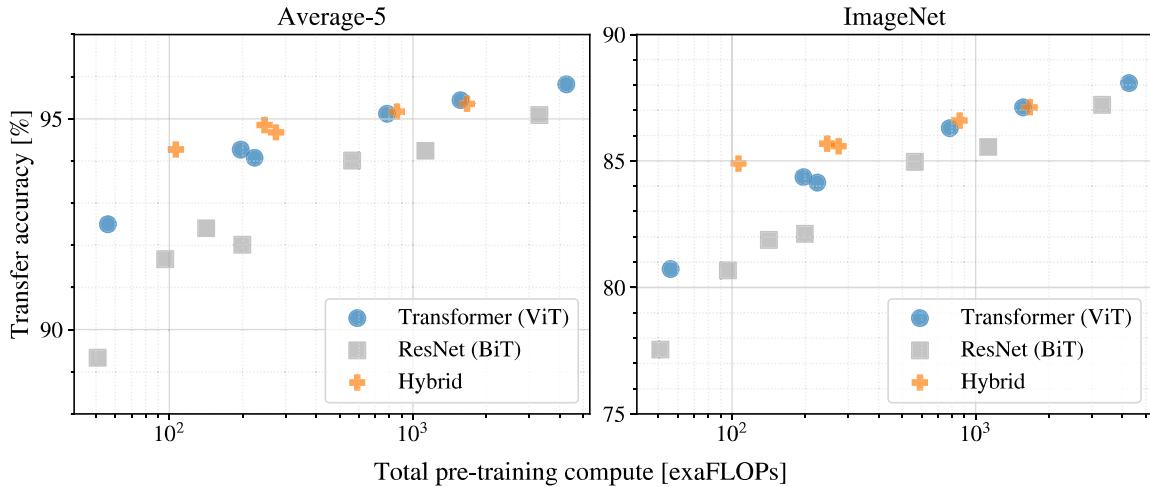


Figure 5: 不同架构的性能与预训练计算的比较:Vision Transformer s、ResNets和hybrid。Vision Transformer s通常在相同的计算预算下优于ResNets。混合模型在较小的模型尺寸上改进了纯transformer，但在较大的模型上差距消失了。

则化。通过ImageNet-21k预训练，它们的性能类似。只有JFT-300M，我们才能看到更大型号的全部好处。图3还显示了不同大小的位模型所跨越的性能区域。BiT cnn在ImageNet上的表现优于ViT，但随着数据集更大，ViT超越了。

在9M、30M和90M的随机子集以及完整的JFT-300M数据集上训练模型。我们不对较小的子集进行额外的正则化，并对所有设置使用相同的超参数。这样，我们可以评估模型的内在属性，而不是正则化的影响。但是，我们确实使用了早期停止，并报告了在训练期间达到的最佳验证精度。为了节省计算量，我们报告了少样本线性精度，而不是全微调精度。图4包含结果。Vision Transformer s在较小的数据集上比ResNets更容易过拟合，具有相当的计算成本。例如，ViT-B/32比ResNet50稍快；它在900万子集上的表现差得多，但在9000万以上子集上的表现更好。对于ResNet152x2和ViT-L/16也是如此。这个结果加强了卷积归纳偏差对较小的数据集有用的直觉，但对于较大的数据集，直接从数据中学习相关的模式就足够了，甚至是有益的。

总的来说，ImageNet上的少样本结果(图4)以及VTAB上的低数据量结果(表2)似乎对非常低的数据传输有希望。对ViT的少样本属性的进一步分析是未来令人兴奋的工作方向。

4.4 尺度研究

通过评估JFT-300M的传输性能，对不同模型进行了受控缩放研究。在这种情况下，数据大小不会成为模型性能的瓶颈，并对每个模型的性能与预训练成本进行了评估。

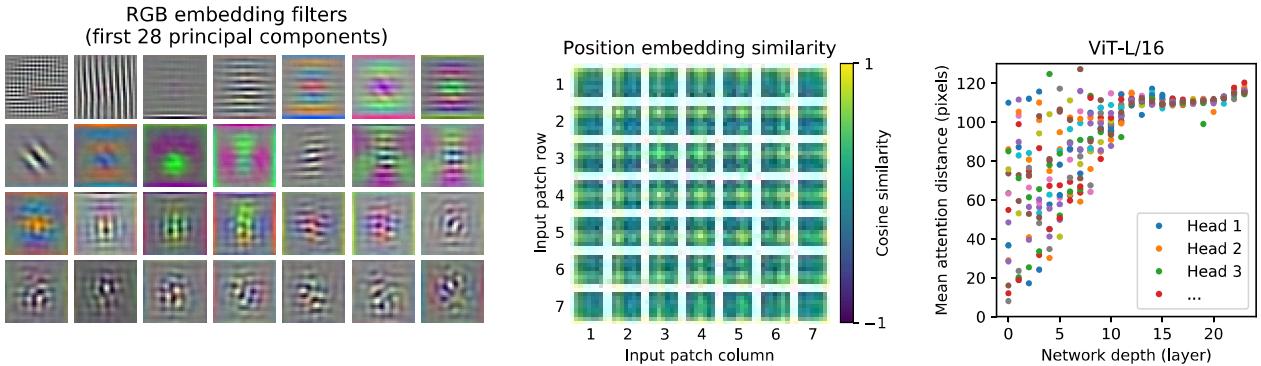


Figure 7: 左:初始线性嵌入RGB值ViT-L/32的滤波器。中心:ViT-L/32的位置嵌入相似度。Tiles显示了patch与所指示的行和列的位置嵌入与所有其他patch的位置嵌入之间的余弦相似性。右:按人头和网络深度划分的参与区域大小。每个点显示了一层中16个头部中的一个图像的平均注意距离。详情请参见附录D.7。

模型集包括: 7个ResNets, R50x1, R50x2 R101x1, R152x1, R152x2, 预训练了7个epoch, 加上R152x2和R200x3预训练了14个epoch; 6 Vision Transformer s, ViT-B/32, B/16, L/32, L/16, 预训练7个epoch, 加上L/16和H/14预训练14个epoch; 和5个混合模型, R50+ ViT-B/32, B/16, L/32, L/16预训练了7个epoch, 加上R50+ ViT-L/16预训练了14个epoch(对于混合模型, 模型名称末尾的数字不是代表补丁大小, 而是代表ResNet骨干中的总下采样率)。

图5包含传输性能与总预训练计算的比较(有关计算成本的详细信息, 请参见附录D.5)。每种模型的详细结果见附录中的表6。可以观察到一些模式。首先, Vision Transformer s在性能/计算权衡方面主导了ResNets。ViT使用大约 $2 - 4 \times$ 较少的计算来达到相同的性能(平均超过5个数据集)。其次, 在较小的计算预算下, 混合动力车的性能略优于ViT, 但对于较大的型号, 差异消失了。这个结果有点令人惊讶, 因为人们可能期望卷积局部特征处理在任何大小上都可以帮助ViT。第三, Vision Transformer s在尝试的范围内似乎没有饱和, 这激励了未来的扩展努力。

4.5 检查VISION TRANSFORMER

为了开始理解Vision Transformer如何处理图像数据, 我们分析其内部表示。Vision Transformer的第一层将平坦的补丁线性地投射到低维空间(例如1)。图7(左)展示了学到了位置嵌入滤波器的顶部主成分。分量类似于合理的基函数, 用于每个块内精细结构的低维表示。

在投影之后, 将学习到的位置嵌入添加到块表示中。图7(中心)表明该模型学习以位置嵌入的相似性对图像内的距离进行编码, 即越近的块往往具有越相似的位置嵌入。此外, 出现了行-列结构;同一行/列中的补丁具有相似的嵌入。最后, 对于较大的网格, 正弦谐振有时很明显(附录D)。位置嵌入学习表示2D图像拓扑, 这解释了为什么手工制作的2D感知嵌入变体不会产生改进(附录D.4)。

Self-attention允许ViT在最低层中整合整个图像的信息。我们研究了网络在多大程度上利用了这种能力。具体来说, 我们根据注意力权重计算图像空间中整合信息的平均距离(图7, 右)。这种“注意力距离”类似于cnn中的感受野大小。发现一些头部关注的是已经位于最底层的大部分图像, 表明该模型确实使用了全局整合信息的能力。其他注意力头在低层中始终具有较小的注意力距离。这种高度局部化的注意力在Transformer之前应用ResNet的混合模型中不太明显(图7, 右), 这表明它可能与cnn中的早期卷积层具有类似的功能。此外, 注意距离随着网络深度的增加而增加。全局而言, 我们发现该模型关注与分类语义相关的图像区域(图6)。

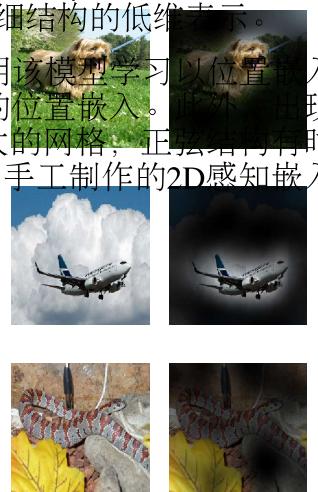


Figure 6: 从输出标记到输入空间的代表性注意示例。详情请参见附录D.7。

4.6 自我监督

transformer在NLP任务上表现出令人印象深刻的性能。然而，它们的成功不仅源于其出色的可扩展性，还源于大规模的自监督预训练(Devlin et al., 2019; Radford et al., 2018)。对用于自监督的掩码块预测进行了初步探索，模拟了BERT中使用的掩码语言建模任务。通过自监督预训练，我们较小的ViT-B/16模型在ImageNet上达到了79.9%的精度，比从头开始训练显著提高了2%，但仍然落后于监督预训练4%。附录B.1.2包含更多细节。本文将对比性预训练的探索(Chen et al., 2020b; He et al., 2020; Bachman et al., 2019; Hénaff et al., 2020)留给未来的工作。

5 结论

我们已经探索了transformer在图像识别中的直接应用。与之前在计算机视觉中使用自注意力的工作不同，除了初始块提取步骤外，没有将图像特定的归纳偏差引入到架构中。相反，我们将图像解释为一系列补丁，并通过NLP中使用的标准Transformer编码器进行处理。当与大型数据集的预训练相结合时，这种简单但可扩展的策略效果惊人。因此，Vision Transformer在许多图像分类数据集上匹配或超过了最先进的技术，同时预训练成本相对较低。

虽然这些初步结果令人鼓舞，但仍存在许多挑战。一种是将ViT应用于其他计算机视觉任务，如检测和分割。我们的结果，加上Carion et al. (2020)上的结果，表明了这种方法的前景。另一个挑战是继续探索自监督预训练方法。初步实验表明，自监督预训练的效果有所改善，但在自监督和大规模监督预训练之间仍有很大的差距。最后，ViT的进一步扩展可能会提高性能。

致谢

这项工作在柏林、Zürich和阿姆斯特丹进行。我们感谢谷歌的许多同事的帮助，特别是Andreas Steiner在基础设施和代码的开源发布方面提供了至关重要的帮助;Joan Puigcerver和Maxim Neumann帮助建立大规模的培训基础设施;Dmitry Lepikhin、Aravindh Mahendran、Daniel Keysers、Mario Lučić、Noam Shazeer、Ashish Vaswani和Colin Raffel进行了有用的讨论。

REFERENCES

- Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *ACL*, 2020.
- Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019.
- Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *ICLR*, 2019.
- I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens. Attention augmented convolutional networks. In *ICCV*, 2019.
- Lucas Beyer, Olivier J. Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv*, 2020.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv*, 2020.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

Mark Chen, Alec Radford, Rewon Child, Jeff Wu, and Heewoo Jun. Generative pretraining from pixels. In *ICML*, 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020b.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Universal Image-TExt Representation Learning. In *ECCV*, 2020c.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv*, 2019.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *ICLR*, 2020.

J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D’Amour, Dan Moldovan, Sylvan Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On robustness and transferability of convolutional neural networks. *arXiv*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv*, 2019.

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, 2018.

Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *ICCV*, 2019.

Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, and Thomas S. Huang. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2020.

Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (BiT): General visual representation learning. In *ECCV*, 2020.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv*, 2020.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language. In *Arxiv*, 2019.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *arXiv*, 2020.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pre-training Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*. 2019.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pre-training. In *ECCV*, 2018.

M. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018.

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992. doi: 10.1137/0330046. URL <https://doi.org/10.1137/0330046>.

Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Weight standardization. *arXiv preprint arXiv:1903.10520*, 2019.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. *Technical Report*, 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *Technical Report*, 2019.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *NeurIPS*, 2019.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019.

Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Herve Jegou. Fixing the train-test resolution discrepancy. In *NeurIPS*. 2019.

Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Herve Jegou. Fixing the train-test resolution discrepancy: Fixefficientnet. *arXiv preprint arXiv:2003.08237*, 2020.

Michael Tschannen, Josip Djolonga, Marvin Ritter, Aravindh Mahendran, Neil Houlsby, Sylvain Gelly, and Mario Lucic. Self-supervised learning of video-induced visual invariances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *ECCV*, 2020a.

Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. *arXiv preprint arXiv:2003.07853*, 2020b.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In *ACL*, 2019.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *ICLR*, 2019.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arxiv*, 2020.

Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer.
S⁴L: Self-Supervised Semi-Supervised Learning. In *ICCV*, 2019a.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019b.

Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, 2020.

Models	Dataset	Epochs	Base LR	LR decay	Weight decay	Dropout
ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
ViT-L/32	JFT-300M	7	$6 \cdot 10^{-4}$	linear	0.1	0.0
ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	linear	0.1	0.0
ViT-H/14	JFT-300M	14	$3 \cdot 10^{-4}$	linear	0.1	0.0
R50x{1,2}	JFT-300M	7	10^{-3}	linear	0.1	0.0
R101x1	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
R152x{1,2}	JFT-300M	7	$6 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-L/32	JFT-300M	7	$2 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	linear	0.1	0.0
ViT-B/{16,32}	ImageNet-21k	90	10^{-3}	linear	0.03	0.1
ViT-L/{16,32}	ImageNet-21k	30/90	10^{-3}	linear	0.03	0.1
ViT-*	ImageNet	300	$3 \cdot 10^{-3}$	cosine	0.3	0.1

Table 3: 用于训练的超参数。所有模型的批量大小为4096，学习率热身步骤为10k。对于ImageNet，我们发现在全局范数上额外应用梯度裁剪是有益的。训练分辨率为224。

附录

A 多头自注意力

标准 \mathbf{qkv} self-attention (SA, Vaswani et al. (2017))是神经架构的流行构建块。对于输入序列 $\mathbf{z} \in \mathbb{R}^{N \times D}$ 中的每个元素，我们计算序列中所有值 \mathbf{v} 的加权和。注意力权重 A_{ij} 基于序列的两个元素及其各自的查询 \mathbf{q}^i 和键 \mathbf{k}^j 表示之间的成对相似性。

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z} \mathbf{U}_{qkv} \quad \mathbf{U}_{qkv} \in \mathbb{R}^{D \times 3D_h}, \quad (5)$$

$$A = \text{softmax} \left(\mathbf{q} \mathbf{k}^\top / \sqrt{D_h} \right) \quad A \in \mathbb{R}^{N \times N}, \quad (6)$$

$$\text{SA}(\mathbf{z}) = A \mathbf{v}. \quad (7)$$

多头自注意力(MSA)是SA的扩展，在其中我们并行运行 k 自注意力操作，称为“头”，并将它们的串联输出进行投影。为了在更改 k 时保持计算和参数数量不变， D_h (例如5)通常设置为 D/k 。

$$\text{MSA}(\mathbf{z}) = [\text{SA}_1(z); \text{SA}_2(z); \dots; \text{SA}_k(z)] \mathbf{U}_{msa} \quad \mathbf{U}_{msa} \in \mathbb{R}^{k \cdot D_h \times D} \quad (8)$$

B 实验细节

B.1 培训

表3总结了我们针对不同模型的训练设置。我们发现在ImageNet上从头开始训练模型时，强正则化是关键。当使用Dropout时，除 \mathbf{qkv} 投影外，在每个密集层之后应用，并在添加位置-到补丁嵌入后直接应用。混合模型使用与ViT对应的精确设置进行训练。最后，所有的训练都在分辨率224上进行。

B.1.1 微调

我们使用动量为0.9的SGD对所有ViT模型进行微调。我们对学习率运行一个网格搜索，学习率范围见表4。为此，我们使用训练集的小分段(10%用于宠物和花卉，2%用于CIFAR, 1%ImageNet)作为开发集，并在剩余数据上进行训练。为了最终结果，我们在整个训练集上进行训练，并在各自的测试数据上进行评估。对于微调ResNets和混合模型，我们使用完全相同的设置，唯一的例外是ImageNet，我们将另一个值0.06添加到学习率扫描。此外，对于ResNets，我们还运行Kolesnikov et al. (2020)的设置，并在这次运行和扫描中选择最佳结果。最后，如果没有特别提到，所有微调实验都在384分辨率下运行(在与训练不同的分辨率下运行微调是常见的做法(Kolesnikov et al., 2020))。

Dataset	Steps	Base LR
ImageNet	20 000	{0.003, 0.01, 0.03, 0.06}
CIFAR100	10 000	{0.001, 0.003, 0.01, 0.03}
CIFAR10	10 000	{0.001, 0.003, 0.01, 0.03}
Oxford-IIIT Pets	500	{0.001, 0.003, 0.01, 0.03}
Oxford Flowers-102	500	{0.001, 0.003, 0.01, 0.03}
VTAB (19 tasks)	2 500	0.01

Table 4: 用于微调的超参数。所有模型都使用余弦学习率衰减、批量大小为512、无权重衰减和全局范数为1的梯度裁剪进行微调。如果没有特别提到，微调分辨率是384。

当将ViT模型迁移到另一个数据集时，我们删除整个头部(两个线性层)，并将其替换为一个单一的、零初始化的线性层，输出目标数据集所需的类的数量。我们发现这比简单地重新初始化最后一层更健壮。

对于VTAB，我们遵循Kolesnikov et al. (2020)中的协议，并对所有任务使用相同的超参数设置。我们使用0.01的学习率并训练2500步骤(Tab. 4)。我们通过对两个学习率和两个调度进行小扫描来选择这个设置，并在200个示例验证集上选择VTAB得分最高的设置。我们遵循Kolesnikov et al. (2020)中使用的预处理，只是不使用特定于任务的输入分辨率。相反，我们发现Vision Transformer从所有任务的高分辨率(384×384)中获益最多。

B.1.2 自我监督

采用掩码块预测目标进行初步的自监督实验。为了做到这一点，我们通过用一个可学习的[mask]嵌入(80%)替换它们的嵌入，一个随机的其他补丁嵌入(10%)或只是保持它们原样(10%)来破坏50%的补丁嵌入。这个设置与Devlin et al. (2019)用于language的设置非常相似。最后，使用每个损坏块各自的块表示预测每个损坏块的3位平均颜色(即总共512种颜色)。

在JFT上训练了1M步(约14个epoch)的自监督模型，批处理大小为4096。我们使用Adam，其基础学习率为 $2 \cdot 10^{-4}$ ，热身步骤为10k步，余弦学习率衰减。作为预训练的预测目标，我们尝试了以下设置：1)仅预测平均的3bit颜色(即1次预测512种颜色)，2)并行预测 4×4 缩小版 16×16 patch的3bit颜色(即16次预测512种颜色)，3)使用L2对整个patch进行回归(即在3个RGB通道上进行256次回归)。令人惊讶的是，我们发现它们都运行得很好，尽管L2稍差。我们只报告了选项1)的最终结果，因为它显示了最佳的少样本性能。我们还实验了Devlin et al. (2019)使用的15%的腐敗率，但在我们的少样本指标上的结果也稍差。

最后，我们想指出的是，为了在ImageNet分类上获得类似的性能提升，我们对掩码块预测的实例化不需要如此大量的预训练，也不需要像JFT这样的大型数据集。也就是说，我们观察到在100k预训练步骤后，下游性能的收益递减，并且在ImageNet上进行预训练时，看到了类似的收益。

C 其他结果

我们报告了与论文中给出的数字相对应的详细结果。表5对应于论文中的图3，显示了在不断增大的数据集上预训练的不同ViT模型的迁移性能：ImageNet、ImageNet-21k和JFT-300M。表6对应于论文中的图5，显示了ViT、ResNet和不同大小的混合模型的迁移性能，以及它们预训练的估计计算成本。

D 附加分析

D.1 RESNETS的SGD和ADAM

ResNets通常使用SGD进行训练，我们使用Adam作为优化器是非常非常规的。这里我们展示了激发这种选择的实验。将在JFT上预训练的两个resnet——50x1和152x2——的微调性能与SGD和Adam进行了比较。对于SGD，我们使用Kolesnikov et al. (2020)推荐的超参数。结果如表7所示。Adam预训练在

		ViT-B/16	ViT-B/32	ViT-L/16	ViT-L/32	ViT-H/14
ImageNet	CIFAR-10	98.13	97.77	97.86	97.94	-
	CIFAR-100	87.13	86.31	86.35	87.07	-
	ImageNet	77.91	73.38	76.53	71.16	-
	ImageNet ReaL	83.57	79.56	82.19	77.83	-
	Oxford Flowers-102	89.49	85.43	89.66	86.36	-
	Oxford-IIIT-Pets	93.81	92.04	93.64	91.35	-
ImageNet-21k	CIFAR-10	98.95	98.79	99.16	99.13	99.27
	CIFAR-100	91.67	91.97	93.44	93.04	93.82
	ImageNet	83.97	81.28	85.15	80.99	85.13
	ImageNet ReaL	88.35	86.63	88.40	85.65	88.70
	Oxford Flowers-102	99.38	99.11	99.61	99.19	99.51
	Oxford-IIIT-Pets	94.43	93.02	94.73	93.09	94.82
JFT-300M	CIFAR-10	99.00	98.61	99.38	99.19	99.50
	CIFAR-100	91.87	90.49	94.04	92.52	94.55
	ImageNet	84.15	80.73	87.12	84.37	88.04
	ImageNet ReaL	88.85	86.27	89.99	88.28	90.33
	Oxford Flowers-102	99.56	99.27	99.56	99.45	99.68
	Oxford-IIIT-Pets	95.80	93.40	97.11	95.83	97.56

Table 5: 在ImageNet、ImageNet-21k或JFT300M上预训练时，Vision Transformer在各种数据集上的准确率(以%为单位)排名第一。这些值对应于正文中的图3。模型在384分辨率进行微调。请注意，ImageNet结果的计算不需要额外的技术(Polyak平均和512分辨率图像)，用于实现表2中的结果。

name	Epochs	ImageNet	ImageNet ReaL	CIFAR-10	CIFAR-100	Pets	Flowers	exaFLOPs
ViT-B/32	7	80.73	86.27	98.61	90.49	93.40	99.27	55
ViT-B/16	7	84.15	88.85	99.00	91.87	95.80	99.56	224
ViT-L/32	7	84.37	88.28	99.19	92.52	95.83	99.45	196
ViT-L/16	7	86.30	89.43	99.38	93.46	96.81	99.66	783
ViT-L/16	14	87.12	89.99	99.38	94.04	97.11	99.56	1567
ViT-H/14	14	88.08	90.36	99.50	94.71	97.11	99.71	4262
ResNet50x1	7	77.54	84.56	97.67	86.07	91.11	94.26	50
ResNet50x2	7	82.12	87.94	98.29	89.20	93.43	97.02	199
ResNet101x1	7	80.67	87.07	98.48	89.17	94.08	95.95	96
ResNet152x1	7	81.88	87.96	98.82	90.22	94.17	96.94	141
ResNet152x2	7	84.97	89.69	99.06	92.05	95.37	98.62	563
ResNet152x2	14	85.56	89.89	99.24	91.92	95.75	98.75	1126
ResNet200x3	14	87.22	90.15	99.34	93.53	96.32	99.04	3306
R50x1+ViT-B/32	7	84.90	89.15	99.01	92.24	95.75	99.46	106
R50x1+ViT-B/16	7	85.58	89.65	99.14	92.63	96.65	99.40	274
R50x1+ViT-L/32	7	85.68	89.04	99.24	92.93	96.97	99.43	246
R50x1+ViT-L/16	7	86.60	89.72	99.18	93.64	97.03	99.40	859
R50x1+ViT-L/16	14	87.12	89.76	99.31	93.89	97.36	99.11	1668

Table 6: 详细的模型缩放实验结果。这些与主论文中的图5相对应。在几个数据集上显示了迁移精度，以及预训练计算(以exaFLOPs为单位)。

Dataset	ResNet50		ResNet152x2	
	Adam	SGD	Adam	SGD
ImageNet	77.54	78.24	84.97	84.37
CIFAR10	97.67	97.46	99.06	99.07
CIFAR100	86.07	85.17	92.05	91.06
Oxford-IIIT Pets	91.11	91.00	95.37	94.79
Oxford Flowers-102	94.26	92.06	98.62	99.32
Average	89.33	88.79	94.01	93.72

Table 7: 微调使用Adam和SGD预训练的ResNet模型。

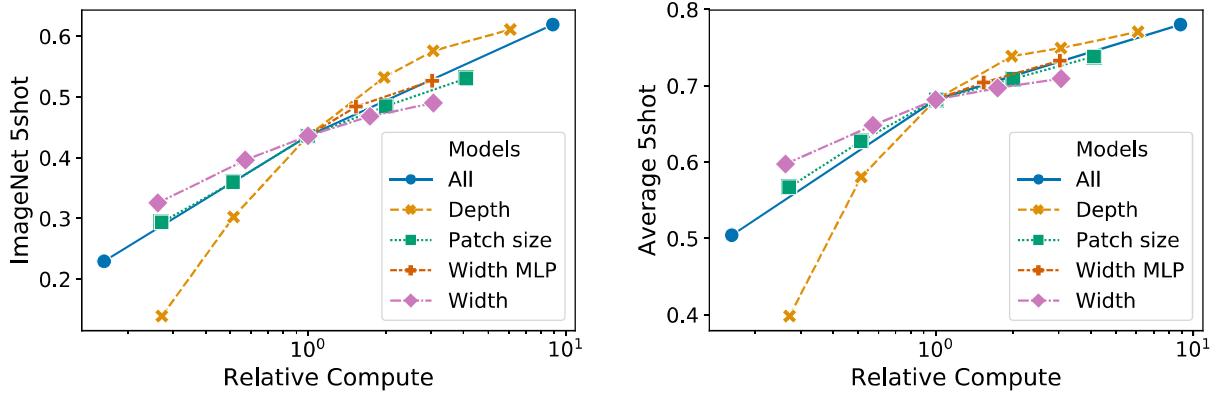


Figure 8: 缩放不同的模型尺寸Vision Transformer。

大多数数据集上和平均上都优于SGD预训练。这证明了选择Adam作为在JFT上预训练ResNets的优化器是合理的。请注意，绝对数字低于Kolesnikov et al. (2020)报告的数字，因为我们只对7时代进行预训练，而不是30。

D.2 变压器形状

我们对Transformer架构的不同维度进行了扩展，以找出最适合扩展到非常大的模型的架构。图8显示了不同配置在ImageNet上的5倍性能。所有配置都基于ViT模型，8层， $D = 1024$, $D_{MLP} = 2048$ 和patch大小32，所有线的交点。我们可以看到，扩展深度可以带来最大的改进，直到64层都是清晰可见的。然而，在16层之后，已经可以看到收益递减。有趣的是，缩放网络的宽度似乎可以导致最小的变化。在不引入参数的情况下，减少patch大小从而增加有效序列长度显示了令人惊讶的鲁棒性改进。这些发现表明，与参数数量相比，计算可能是更好的性能预测器，如果有的话，扩展应该强调深度而不是宽度。总而言之，按比例缩放所有维度会带来鲁棒的改进。

D.3 头部类型和类令牌

为了尽可能地保持与原始Transformer模型接近，我们使用了一个额外的[*class*] token，该token被用作图像表示。然后，该令牌的输出通过小型多层感知器(MLP)转换为类预测，在单个隐藏层中tanh作为非线性。

这种设计继承自用于文本的Transformer模型，我们在全文中都使用它。最初尝试只使用图像块嵌入，全局平均池化(GAP)它们，然后是线性分类器——就像ResNet的最终特征图一样——表现非常糟糕。然而，我们发现这既不是由于额外的token，也不是由于GAP操作。相反，性能上的差异可以通过不同的学习率来充分解释，参见图9。

D.4 位置嵌入

我们使用位置嵌入对编码空间信息的不同方法进行了消融。我们尝试了以下情况：

- 不提供位置信息：将输入视为补丁包。

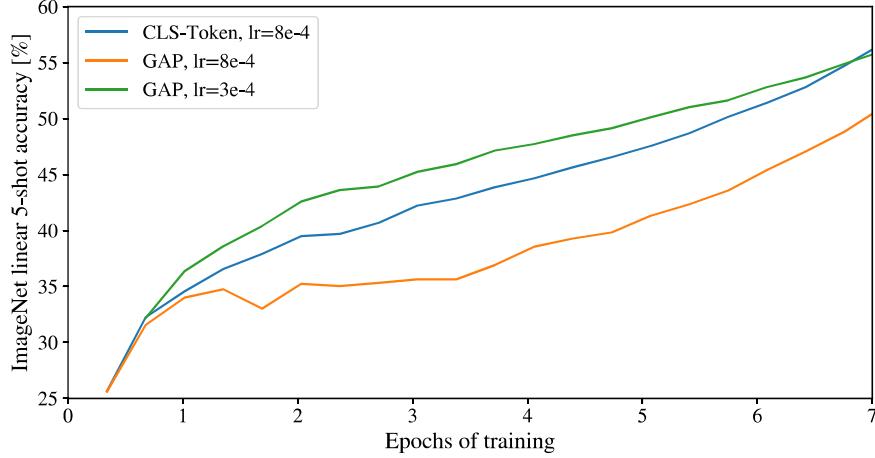


Figure 9: 类标记和全局平均池化分类器的比较。两者的效果相似，但需要不同的学习率。

Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

Table 8: 在ImageNet 5-shot线上评估了ViT-B/16模型对位置嵌入的消融研究结果。

- **1维位置嵌入:**将输入视为栅格顺序的一系列补丁(在本文其他所有实验中都是默认的)。
- **二维位置嵌入:**将输入视为二维块的网格。在这种情况下，学习了两组嵌入，每组都针对其中一个轴， X -embedding和 Y -embedding，每组大小为 $D/2$ 。然后，基于输入中路径上的坐标，我们连接 X 和 Y 嵌入以获得该补丁的最终位置嵌入。
- **相对位置嵌入:**考虑块之间的相对距离来编码空间信息，而不是它们的绝对位置。为此，我们使用1维相对注意力，其中定义了所有可能补丁对的相对距离。因此，对于每个给定的对(一个作为查询，另一个作为注意力机制中的键/值)，我们有一个偏移量 $p_q - p_k$ ，其中每个偏移量与嵌入相关联。然后，我们简单地运行额外关注，其中我们使用原始查询(查询的内容)，但使用相对位置嵌入作为键。然后，我们使用相对注意力的logits作为偏差项，并在应用softmax之前将其添加到主要注意力(基于内容的注意力)的logits中。

除了不同的编码空间信息的方法，我们还尝试了不同的方法来将这些信息合并到我们的模型中。对于一维和二维位置嵌入，我们尝试了三种不同的情况:(1)在输入模型的主干之后和将输入输入到Transformer编码器之前，将位置嵌入添加到输入中(在本文所有其他实验中都是默认的);(2)在每一层的开始学习并向输入添加位置嵌入;(3)在每层的开始添加一个学习到的位置嵌入到输入中(在层之间共享)。

表8总结了ViT-B/16模型上的消融研究结果。正如我们所看到的，虽然没有位置嵌入的模型和有位置嵌入的模型之间的性能有很大的差距，但编码位置信息的不同方法之间几乎没有区别。我们推测，由于Transformer编码器是在块级输入上操作，而不是像素级，因此如何编码空间信息的差异不那么重要。更准确地说，在块级输入中，空间维度比原始像素级输入小得多，例如 14×14 而不是 224×224 ，对于这些不同的位置编码策略来说，学习在这种分辨率中表示空间关系同样容易。即便如此，网络学习到的位置嵌入相似性的具体模式取决于训练超参数(图10)。

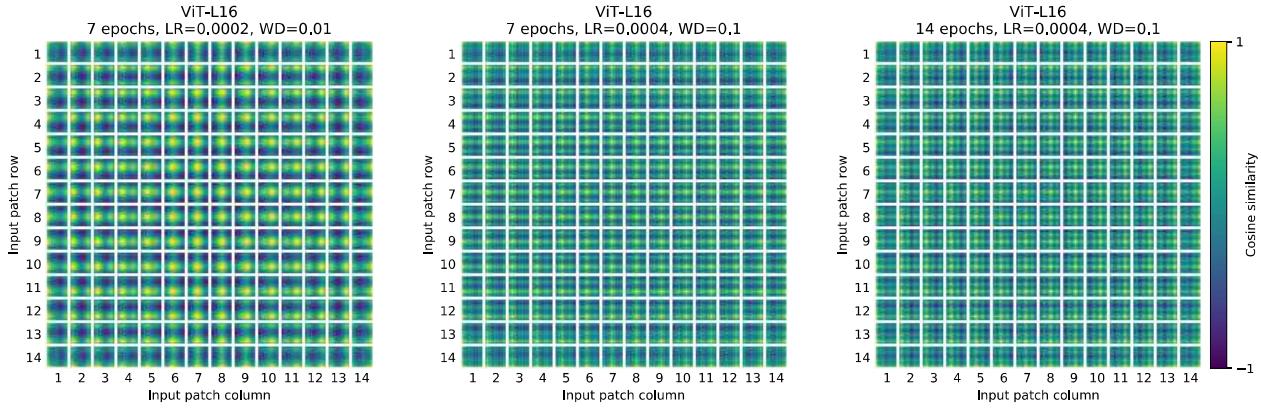


Figure 10: 用不同超参数训练的模型的位置嵌入。

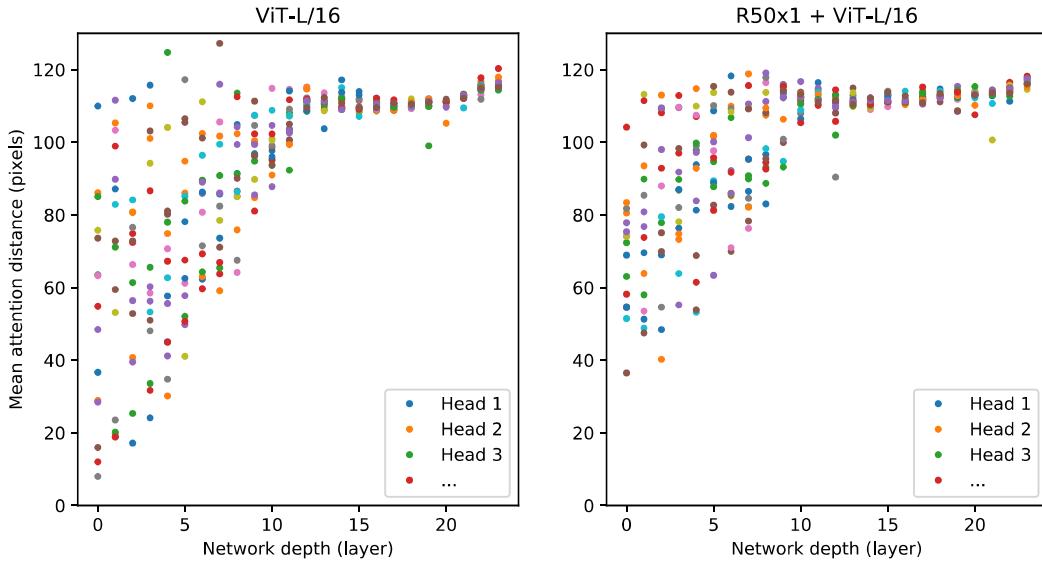


Figure 11: 由头部和网络深度确定的参与区域的大小。将查询像素与所有其他像素之间的距离用注意力权重加权平均，计算128幅示例图像的注意力距离。每个点显示了一层中16个头部中的一个图像的平均注意距离。图像宽度为224像素。

D.5 经验计算成本

我们还对硬件上体系统结构的真实速度感兴趣，由于车道宽度和缓存大小等细节，理论计算并不总是能很好地预测。为此，在TPUv3加速器上对感兴趣的主要模型进行推理速度的计时；推断速度和反向传播速度之间的差异是一个恒定的模型无关因素。

图12(左)显示了一个核每秒可以处理不同大小的图像。每一个点都是指在不同批处理大小范围内测量的峰值性能。可以看到，ViT与图像大小的理论双二次缩放仅在最大分辨率下刚刚开始发生在最大的模型上。

另一个有趣的数量是每个模型可以装入核心的最大批量大小，越大越适合扩展到大型数据集。图12(右)显示了同一组模型的数量。这表明，大型ViT模型在内存效率方面比ResNet模型具有明显的优势。

D.6 轴向注意

轴向注意力(Huang et al., 2020; Ho et al., 2019)是一种简单而有效的技术，用于在组织为多维张量的大型输入上运行自注意力。轴向注意力的一般思想是执行多个注意力操作，每个操作沿着输入张量的一个轴进行，而不是将一维注意力应用于输入的平坦版本。在轴向注意力中，每个注意都沿着特定的轴混合信息，同时保持沿其他轴的信息独立。沿着这条思路，Wang et al. (2020b)提出了AxialResNet模型，其中ResNet50中所有具有内核大小 3×3 的

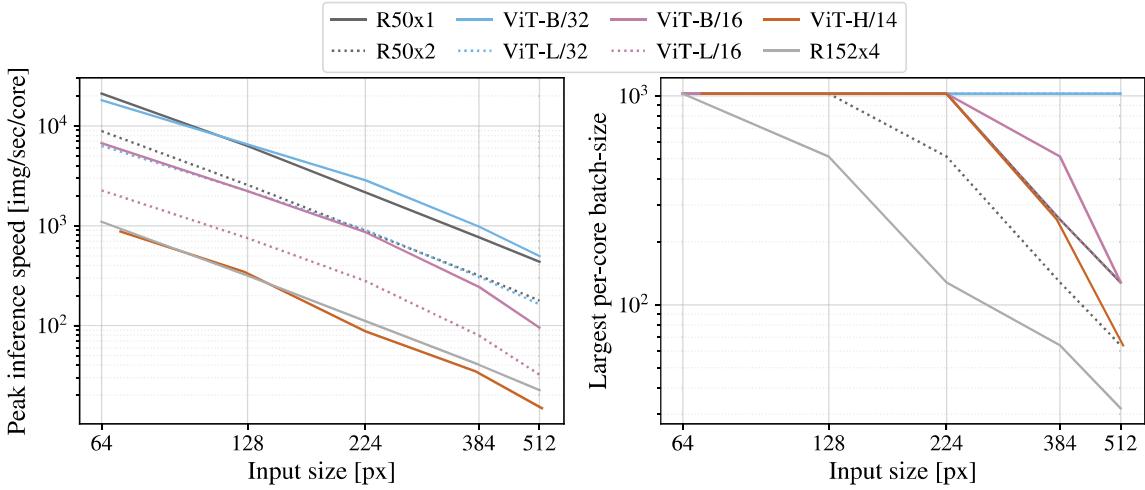


Figure 12: 左:跨输入大小的各种体系结构的实际时钟时间。ViT模型的速度与类似的ResNets相当。右:最大的每核批处理大小，适用于具有不同输入大小的架构的设备。ViT模型的内存效率显然更高。

卷积都被轴向自注意力(即行和列注意力)取代，并通过相对位置编码增强。我们已经实现了AxialResNet作为基线模型。³

此外，我们修改了ViT，以二维形状处理输入，而不是一维块序列，并合并了轴向Transformer块，其中不是自注意力然后MLP，而是行自注意力加上MLP，列自注意力加上MLP。

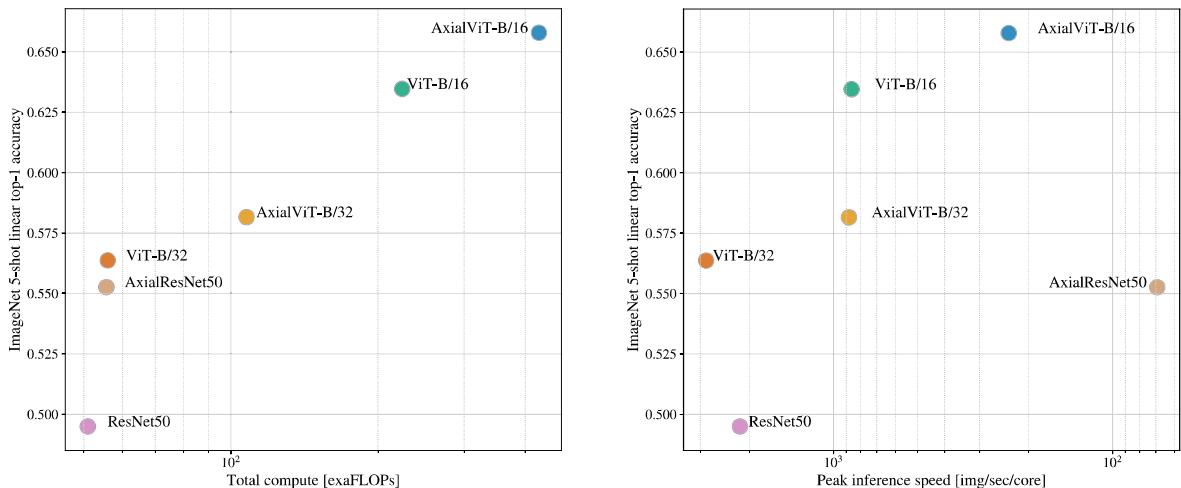


Figure 13: 基于轴注意力的模型的性能，在ImageNet 5-shot linear上的top-1精度，与它们的速度，在FLOPs数量(左)和推理时间(左)方面。

图13，在JFT数据集上进行预训练时，展示了axis ResNet、axis - ViT- b /32和axis - ViT- b /16在ImageNet 5shot linear上的性能与预训练计算的性能，包括FLOPs数量和推理时间(每秒示例)。我们可以看到，两者都有在性能方面，axis - ViT- b /32和axis - ViT- b /16比它们的对应产品ViT- b做得更好，但它是以更多的计算为代价的。这是因为在axis - ViT模型中，具有全局自注意力的每个Transformer块被两个轴向Transformer块取代，一个具有行自注意力，一个具有列自注意力，尽管在axis情况下，自注意力操作的序列长度较小，但每个axis - ViT块都有一个额外的MLP。对于AxialResNet，尽管它在精度/计算权衡方面看起来很合理(图13，左)，但在tpu上的朴素实现非常慢(图13，右)。

³我们的实现基于<https://github.com/csrhddlam/axial-deeplab>中的开源PyTorch实现。在我们的实验中，我们重现了在(Wang et al., 2020b)上报告的准确性分数，然而，我们的实现，类似于开源实现，在tpu上非常慢。因此，我们无法将其用于广泛的大规模实验。这些可以通过精心优化的实现来解锁。

D.7 注意距离

为了理解ViT如何使用自注意力来整合整个图像的信息，我们分析了不同层的注意力权重所跨越的平均距离(图11)。这种“注意力距离”类似于cnn中的感受野大小。在较低层次的头部之间，平均注意力距离有很大的变化，一些头部负责图像的大部分，而另一些头部负责查询位置或附近的小区域。随着深度的增加，所有头部的注意力距离也会增加。在网络的后半部分，大多数头部广泛参与迭代。

D.8 注意力图

为了计算注意力从输出标记到输入空间的映射(图6和14)，我们使用注意力Rollout (Abnar & Zuidema, 2020)。简单地说，我们在所有头部上平均ViT-L/16的注意力权重，然后递归地乘以所有层的权重矩阵。这解释了注意力在所有层中跨token的混合。

D.9 OBJECTNET结果

按照Kolesnikov et al. (2020)中的评估设置，在ObjectNet基准上评估了旗舰的ViT-H/14模型，获得了82.1%的top-5准确率和61.7%的top-1准确率。

D.10 VTAB分解

表9显示了在每个VTAB-1k任务上获得的分数。

Table 9: VTAB-1k跨任务性能分解。

	Caltech101	CIFAR-100	DTD	Flowers102	Pets	Sun397	SVHN	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	dSpr-Loc	dSpr-Ori	KITTI-Dist	sNORB-Azim	sNORB-Elev	Mean
ViT-H/14 (JFT)	95.3	85.5	75.2	99.7	97.2	65.0	88.9	83.3	96.7	91.4	76.6	91.7	63.8	53.1	79.4	63.3	84.5	33.2	51.2	77.6
ViT-L/16 (JFT)	95.4	81.9	74.3	99.7	96.7	63.5	87.4	83.6	96.5	89.7	77.1	86.4	63.1	49.7	74.5	60.5	82.2	36.2	51.1	76.3
ViT-L/16 (I21k)	90.8	84.1	74.1	99.3	92.7	61.0	80.9	82.5	95.6	85.2	75.3	70.3	56.1	41.9	74.7	64.9	79.9	30.5	41.7	72.7

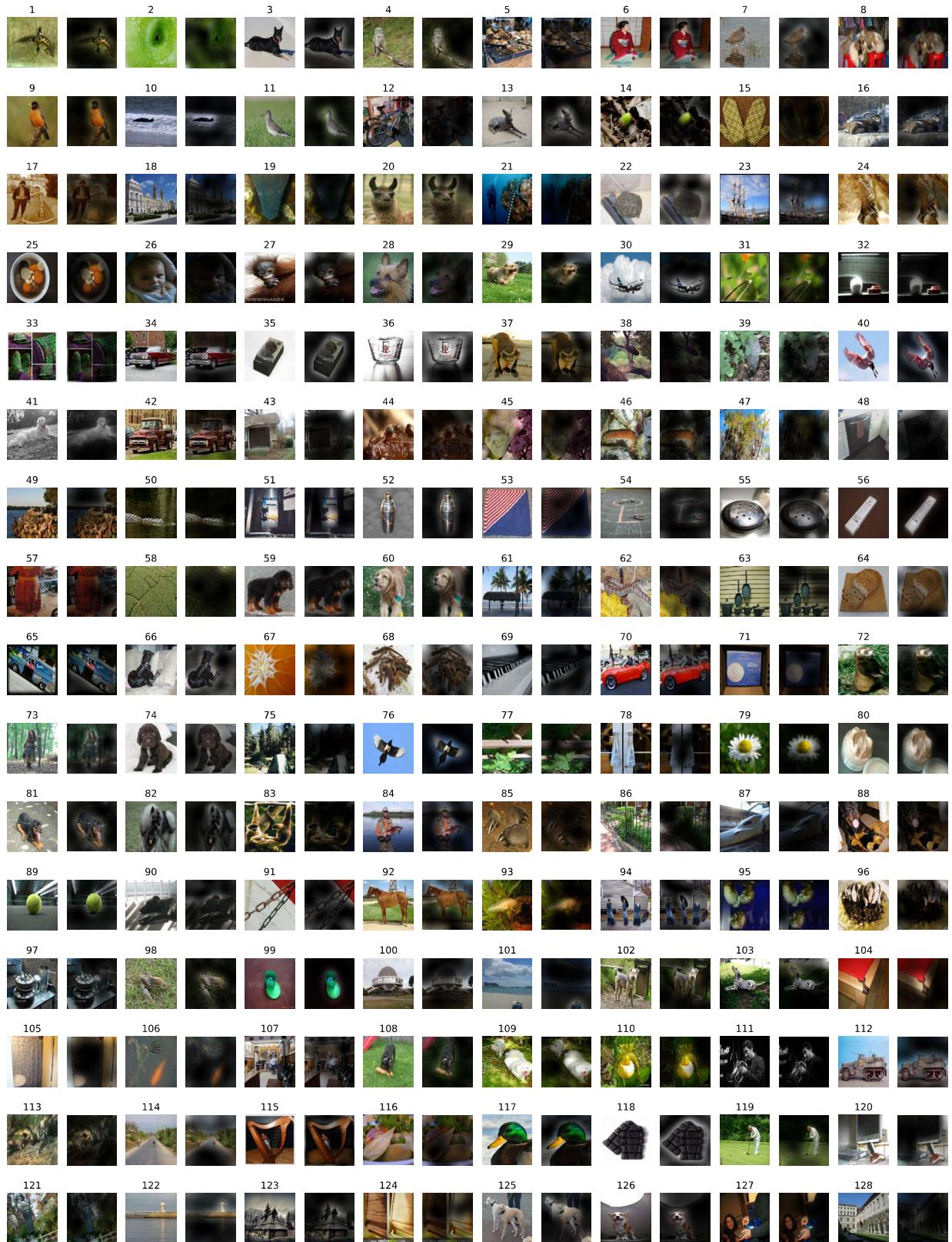


Figure 14: 注意力图的进一步示例如图6(随机选择)所示。