AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

一幅图像胜过 16x16 个词: 用于大规模图像识别的变换器

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

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
* 等同技术贡献, † 等同指导
Google Research, Brain Team
谷歌研究，脑团队
{adosovitskiy, neilhoulsby}@google.com

{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

摘要

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train 1

尽管变换器架构已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结合使用，要么用于替代卷积网络的某些组件，同时保持其整体结构不变。我们表明，这种对卷积神经网络 (CNN) 的依赖并不是必要的，直接应用于图像块序列的纯变换器在图像分类任务中表现良好。当在大量数据上进行预训练并转移到多个中型或小型图像识别基准 (如 ImageNet、CIFAR-100、VTAB 等) 时，视觉变换器 (ViT) 在与最先进的卷积网络相比时取得了优秀的结果，同时训练所需的计算资源显著减少。

1 INTRODUCTION

1 引言

Self-attention-based architectures, in particular Transformers (Vaswani et al. 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al. 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters (Brown et al. 2020, Lepikhin et al. 2020). With the models and datasets growing, there is still no sign of saturating performance.

基于自注意力的架构，特别是变压器 (Vaswani et al. 2017)，已成为自然语言处理 (NLP) 中的首选模型。主流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上进行微调 (Devlin et al. 2019)。得益于变压器的计算效率和可扩展性，训练超过 100B 参数的前所未有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。

In computer vision, however, convolutional architectures remain dominant (LeCun et al., 1989, Krizhevsky et al. 2012, He et al. 2016). Inspired by NLP successes, multiple works try combining CNN-like architectures with self-attention (Wang et al. 2018; Carion et al. 2020), some replacing the

convolutions entirely (Ramachandran et al. 2019 | Wang et al. 2020a). The latter models, while theoretically efficient, have not yet been scaled effectively on modern hardware accelerators due to the use of specialized attention patterns. Therefore, in large-scale image recognition, classic ResNet-like architectures are still state of the art (Mahajan et al. 2018, Xie et al., 2020, Kolesnikov et al.,

然而，在计算机视觉领域，卷积架构仍然占主导地位 (LeCun et al., 1989, Krizhevsky et al. 2012, He et al. 2016)。受到 NLP 成功的启发，多个研究尝试将类似 CNN 的架构与自注意力结合 (Wang et al. 2018; Carion et al. 2020)，有些则完全替代了卷积 (Ramachandran et al. 2019 | Wang et al. 2020a)。尽管后者模型在理论上是高效的，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 ResNet 类架构仍然是最先进的技术 (Mahajan et al. 2018, Xie et al., 2020, Kolesnikov et al.)。2020).

Inspired by the Transformer scaling successes in NLP, we experiment with applying a standard Transformer directly to images, with the fewest possible modifications. To do so, we split an image into patches and provide the sequence of linear embeddings of these patches as an input to a Transformer. Image patches are treated the same way as tokens (words) in an NLP application. We train the model on image classification in supervised fashion.

受到 NLP 中变压器扩展成功的启发，我们尝试将标准变压器直接应用于图像，尽可能少地进行修改。为此，我们将图像分割成补丁，并将这些补丁的线性嵌入序列作为输入提供给变压器。图像补丁被视为 NLP 应用中与标记 (单词) 相同的方式。我们以监督方式在图像分类上训练模型。

When trained on mid-sized datasets such as ImageNet without strong regularization, these models yield modest accuracies of a few percentage points below ResNets of comparable size. This seemingly discouraging outcome may be expected: Transformers lack some of the inductive biases

当在中等规模的数据集 (如 ImageNet) 上进行训练时，如果没有强有力的正则化，这些模型的准确率仅比同等规模的 ResNet 低几个百分点。这一看似令人沮丧的结果是可以预期的: 变换器缺乏某些卷积神经网络固有的归纳偏置，例如平移等变性和局部性，因此在数据量不足时无法很好地泛化。

[1] Fine-tuning code and pre-trained models are available at https://github.com/ inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data.

[1] 微调代码和预训练模型可在 https://github.com/ 获取。

However, the picture changes if the models are trained on larger datasets (14M-300M images). We find that large scale training trumps inductive bias. Our Vision Transformer (ViT) attains excellent results when pre-trained at sufficient scale and transferred to tasks with fewer datapoints. When pre-trained on the public ImageNet-21k dataset or the in-house JFT-300M dataset, ViT approaches or beats state of the art on multiple image recognition benchmarks. In particular, the best model reaches the accuracy of 88.55% on ImageNet, 90.72% on ImageNet-ReaL, 94.55% on CIFAR-100, and 77.63% on the VTAB suite of 19 tasks.

然而，如果在更大规模的数据集 (1400 万至 3 亿张图像) 上进行训练，情况就会有所不同。我们发现大规模训练优于归纳偏置。当在足够规模上进行预训练并转移到数据点较少的任务时，我们的视觉变换器 (ViT) 取得了优异的结果。当在公共的 ImageNet-21k 数据集或内部的 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的水平。特别是，最佳模型在 ImageNet 上达到了 88.55% 的准确率，在 ImageNet-ReaL 上达到了 90.72%，在 CIFAR-100 上达到了 94.55%，以及在 VTAB 的 19 个任务上达到了 77.63% 。

2 RELATED WORK

2 相关工作

Transformers were proposed by Vaswani et al. (2017) for machine translation, and have since become the state of the art method in many NLP tasks. Large Transformer-based models are often pre-trained on large corpora and then fine-tuned for the task at hand: BERT (Devlin et al. 2019) uses a denoising self-supervised pre-training task, while the GPT line of work uses language modeling as its pre-training task (Radford et al., 2018, 2019, Brown et al., 2020).

变换器是由 Vaswani 等人 (2017) 提出用于机器翻译的，随后在许多自然语言处理任务中成为了最先进的方法。大型基于变换器的模型通常在大型语料库上进行预训练，然后针对当前任务进行微

google-research/vision_transformer
google-research/vision_transformer

调:BERT(Devlin 等人 2019) 使用去噪自监督预训练任务，而 GPT 系列则使用语言建模作为其预训练任务 (Radford 等人，2018，2019，Brown 等人，2020)。

Naive application of self-attention to images would require that each pixel attends to every other pixel. With quadratic cost in the number of pixels, this does not scale to realistic input sizes. Thus, to apply Transformers in the context of image processing, several approximations have been tried in the past. Parmar et al. (2018) applied the self-attention only in local neighborhoods for each query pixel instead of globally. Such local multi-head dot-product self attention blocks can completely replace convolutions (Hu et al., 2019; Ramachandran et al., 2019; Zhao et al., 2020). In a different line of work, Sparse Transformers (Child et al. 2019) employ scalable approximations to global self-attention in order to be applicable to images. An alternative way to scale attention is to apply it in blocks of varying sizes (Weissenborn et al. 2019), in the extreme case only along individual axes (Ho et al. 2019 Wang et al. 2020a). Many of these specialized attention architectures demonstrate promising results on computer vision tasks, but require complex engineering to be implemented efficiently on hardware accelerators.

对图像的自注意力的天真应用将要求每个像素关注其他每个像素。由于像素数量的平方成本，这在现实输入大小上无法扩展。因此，为了在图像处理的背景下应用变换器，过去尝试了几种近似方法。Parmar 等人 (2018) 仅在每个查询像素的局部邻域中应用自注意力，而不是全局应用。这种局部多头点积自注意力块可以完全替代卷积 (Hu 等，2019；Ramachandran 等，2019；Zhao 等，2020)。在另一项工作中，稀疏变换器 (Child 等，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在不同大小的块中应用它 (Weissenborn 等，2019)，在极端情况下仅沿单个轴应用 (Ho 等，2019；Wang 等，2020a)。许多这些专门的注意力架构在计算机视觉任务上展示了有希望的结果，但需要复杂的工程才能在硬件加速器上高效实现。

Most related to ours is the model of Cordonnier et al. (2020), which extracts patches of size 2×2 from the input image and applies full self-attention on top. This model is very similar to ViT, but our work goes further to demonstrate that large scale pre-training makes vanilla transformers competitive with (or even better than) state-of-the-art CNNs. Moreover, Cordonnier et al. (2020) use a small patch size of 2×2 pixels, which makes the model applicable only to small-resolution images, while we handle medium-resolution images as well.

与我们的工作最相关的是 Cordonnier 等人 (2020) 的模型，该模型从输入图像中提取大小为 2×2 的补丁，并在其上应用全自注意力。该模型与 ViT 非常相似，但我们的工作进一步证明大规模预训练使得普通变换器在竞争中与 (甚至优于) 最先进的 CNNs 相媲美。此外，Cordonnier 等人 (2020) 使用了小补丁大小 2×2 像素，这使得该模型仅适用于小分辨率图像，而我们也处理中等分辨率图像。

There has also been a lot of interest in combining convolutional neural networks (CNNs) with forms of self-attention, e.g. by augmenting feature maps for image classification (Bello et al. 2019) or by further processing the output of a CNN using self-attention, e.g. for object detection (Hu et al. 2018 Carion et al. 2020), video processing (Wang et al. 2018, Sun et al. 2019), image classification (Wu et al. 2020), unsupervised object discovery (Locatello et al. 2020), or unified text-vision tasks (Chen et al., 2020c; Lu et al., 2019; Li et al., 2019).

结合卷积神经网络 (CNN) 与自注意力形式的兴趣也在增加，例如通过增强图像分类的特征图 (Bello et al. 2019) 或通过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) 或统一的文本-视觉任务 (Chen et al. 2020c；Lu et al. 2019；Li et al. 2019)。

Another recent related model is image GPT (iGPT) (Chen et al. 2020a), which applies Transformers to image pixels after reducing image resolution and color space. The model is trained in an unsupervised fashion as a generative model, and the resulting representation can then be fine-tuned or probed linearly for classification performance, achieving a maximal accuracy of 72% on ImageNet.

另一个相关的近期模型是图像 GPT(iGPT)(Chen et al. 2020a)，该模型在降低图像分辨率和色彩空间后，将变换器应用于图像像素。该模型作为生成模型以无监督的方式进行训练，得到的表示可以进一步微调或线性探测以提高分类性能，在 ImageNet 上达到最大准确率为 72% 。

Our work adds to the increasing collection of papers that explore image recognition at larger scales than the standard ImageNet dataset. The use of additional data sources allows to achieve state-of-the-art results on standard benchmarks (Mahajan et al. 2018) Touvron et al. 2019 Xie et al. 2020). Moreover, Sun et al. (2017) study how CNN performance scales with dataset size, and Kolesnikov et al. (2020); Djolonga et al. (2020) perform an empirical exploration of CNN transfer learning from large scale datasets such as ImageNet-21k and JFT-300M. We focus on these two latter datasets as well, but train Transformers instead of ResNet-based models used in prior works.

我们的工作增加了越来越多的论文，这些论文探讨了比标准 ImageNet 数据集更大规模的图像识别。使用额外的数据源可以在标准基准上实现最先进的结果 (Mahajan 等，2018；Touvron 等，2019；Xie 等，2020)。此外，Sun 等 (2017) 研究了 CNN 性能如何随着数据集规模的变化而变化，而 Kolesnikov 等

(2020)；Djolonga 等 (2020) 则对从大规模数据集 (如 ImageNet-21k 和 JFT-300M) 进行 CNN 迁移学习进行了实证探索。我们也关注这两个后者的数据集，但训练的是 Transformers，而不是之前工作中使用的基于 ResNet 的模型。

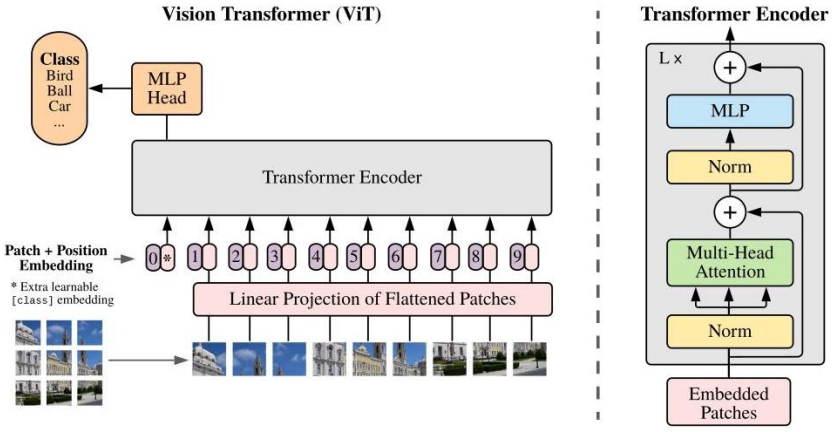


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

图 1: 模型概述。我们将图像分割成固定大小的补丁，线性嵌入每个补丁，添加位置嵌入，并将得到的向量序列输入到标准 Transformer 编码器中。为了进行分类，我们使用标准方法向序列中添加一个额外的可学习 "分类标记"。Transformer 编码器的插图灵感来自 Vaswani 等 (2017)。

3 METHOD

3 方法

In model design we follow the original Transformer (Vaswani et al. 2017) as closely as possible. An advantage of this intentionally simple setup is that scalable NLP Transformer architectures - and their efficient implementations - can be used almost out of the box.

在模型设计中，我们尽可能紧密地遵循原始 Transformer(Vaswani 等，2017)。这种故意简单的设置的一个优势是，可扩展的 NLP Transformer 架构及其高效实现几乎可以开箱即用。

3.1 VISION TRANSFORMER (VIT)

3.1 视觉变换器 (VIT)

An overview of the model is depicted in Figure 1. The standard Transformer receives as input a 1D sequence of token embeddings. To handle 2D images, we reshape the image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ into a sequence of flattened 2D patches $\mathbf{x}_p \in \mathbb{R}^{N \times \left(P^2 \cdot C\right)}$, where(H, W)is the resolution of the original image, C is the number of channels,(P, P)is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches, which also serves as the effective input sequence length for the Transformer. The Transformer uses constant latent vector size D through all of its layers, so we flatten the patches and map to D dimensions with a trainable linear projection (Eq. 1). We refer to the output of this projection as the patch embeddings.

模型的概述如图 1 所示。标准的 Transformer 接收一维的令牌嵌入序列作为输入。为了处理二维图像，我们将图像 $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ 重新形状为一系列展平的 2D 补丁 $\mathbf{x}_p \in \mathbb{R}^{N \times \left(P^2 \cdot C\right)}$ ，其中 (H, W) 是原始图像的分辨率，C 是通道数，(P, P) 是每个图像补丁的分辨率，而 $N = HW/P^2$ 是生成的补丁数量，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在向量大小 D ，因此我们将补丁展平并映射到 D 维度，使用可训练的线性投影 (公式 1)。我们将该投影的输出称为补丁嵌入。

Similar to BERT's [class] token, we prepend a learnable embedding to the sequence of embedded patches $\left(\mathbf{z}_0^0 = \mathbf{x}_{\text{class}}\right)$, whose state at the output of the Transformer encoder $\left(\mathbf{z}_L^0\right)$ serves as the image representation $\mathbf{y}$ (Eq. 4). Both during pre-training and fine-tuning, a classification head is attached to $\mathbf{z}_L^0$. The classification head is implemented by a MLP with one hidden layer at pre-training time and by a single linear layer at fine-tuning time.

类似于 BERT 的 [class] 令牌，我们在嵌入补丁的序列 $\left(\mathbf{z}_0^0 = \mathbf{x}_{\text{class}}\right)$ 前添加一个可学习的嵌入，其在 Transformer 编码器输出时的状态 $\left(\mathbf{z}_L^0\right)$ 作为图像表示 $\mathbf{y}$ (公式 4)。在预训练和微调期间，分类头附加到 $\mathbf{z}_L^0$。分类头在预训练时由一个具有一个隐藏层的多层感知机 (MLP) 实现，而在微调时由一个线性层实现。

Position embeddings are added to the patch embeddings to retain positional information. We use standard learnable 1D position embeddings, since we have not observed significant performance gains from using more advanced 2D-aware position embeddings (Appendix D.3). The resulting sequence of embedding vectors serves as input to the encoder.

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一维位置嵌入，因为我们没有观察到使用更先进的二维位置嵌入 (附录 D.3) 能显著提高性能。生成的嵌入向量序列作为编码器的输入。

The Transformer encoder (Vaswani et al. 2017) consists of alternating layers of multiheaded self-attention (MSA, see Appendix A) and MLP blocks (Eq. 2, 3). Layernorm (LN) is applied before every block, and residual connections after every block (Wang et al., 2019, Baevski & Auli, 2019).

Transformer 编码器 (Vaswani et al. 2017) 由交替的多头自注意力 (MSA，见附录 A) 和 MLP 块 (公式 2, 3) 组成。在每个块之前应用层归一化 (LN)，并在每个块之后添加残差连接 (Wang et al., 2019, Baevski & Auli, 2019)。

The MLP contains two layers with a GELU non-linearity.

MLP 包含两个具有 GELU 非线性的层。

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

$$\mathbf{z}'_\ell = \text{MSA}\left(\text{LN}\left(\mathbf{z}_{\ell-1}\right)\right) + \mathbf{z}_{\ell-1},\ \ell = 1 \ldots L \tag{2}$$

$$\mathbf{z}_\ell = \text{MLP}\left(\text{LN}\left(\mathbf{z}'_\ell\right)\right) + \mathbf{z}'_\ell,\ \ell = 1 \ldots L \tag{3}$$

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

Inductive bias. We note that Vision Transformer has much less image-specific inductive bias than CNNs. In CNNs, locality, two-dimensional neighborhood structure, and translation equivariance are baked into each layer throughout the whole model. In ViT, only MLP layers are local and translationally equivariant, while the self-attention layers are global. The two-dimensional neighborhood structure is used very sparingly: in the beginning of the model by cutting the image into patches and at fine-tuning time for adjusting the position embeddings for images of different resolution (as described below). Other than that, the position embeddings at initialization time carry no information about the 2D positions of the patches and all spatial relations between the patches have to be learned from scratch.

归纳偏差。我们注意到，视觉 Transformer 的图像特定归纳偏差远低于 CNN。在 CNN 中，局部性、二维邻域结构和平移等变性在整个模型的每一层中都被固化。而在 ViT 中，只有 MLP 层是局部的和具有平移等变性的，而自注意力层是全局的。二维邻域结构的使用非常有限: 在模型开始时通过将图像切割成补丁来实现，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补丁的二维位置的信息，所有补丁之间的空间关系必须从头学习。

Hybrid Architecture. As an alternative to raw image patches, the input sequence can be formed from feature maps of a CNN (LeCun et al., 1989). In this hybrid model, the patch embedding projection $\mathbf{E}$ (Eq. 1) is applied to patches extracted from a CNN feature map. As a special case, the patches can have spatial size 1×1, which means that the input sequence is obtained by simply flattening the spatial dimensions of the feature map and projecting to the Transformer dimension. The classification input embedding and position embeddings are added as described above.

混合架构。作为原始图像补丁的替代，输入序列可以由 CNN 的特征图 (LeCun et al., 1989) 形成。在这个混合模型中，补丁嵌入投影 $\mathbf{E}$ (公式 1) 应用于从 CNN 特征图中提取的补丁。作为一个特例，补丁可以具有空间大小 1×1，这意味着输入序列是通过简单地展平特征图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如上所述相加。

3.2 FINE-TUNING AND HIGHER RESOLUTION

3.2 微调与更高分辨率

Typically, we pre-train ViT on large datasets, and fine-tune to (smaller) downstream tasks. For this, we remove the pre-trained prediction head and attach a zero-initialized $D \times K$ feedforward layer, where K is the number of downstream classes. It is often beneficial to fine-tune at higher resolution than pre-training (Touvron et al. 2019, Kolesnikov et al. 2020). When feeding images of higher resolution, we keep the patch size the same, which results in a larger effective sequence length. The Vision Transformer can handle arbitrary sequence lengths (up to memory constraints), however, the pre-trained position embeddings may no longer be meaningful. We therefore perform 2D interpolation of the pre-trained position embeddings, according to their location in the original image. Note that this resolution adjustment and patch extraction are the only points at which an inductive bias about the 2D structure of the images is manually injected into the Vision Transformer.

通常，我们在大型数据集上对 ViT 进行预训练，并对 (较小的) 下游任务进行微调。为此，我们移除预训练的预测头，并附加一个零初始化的 $D \times K$ 前馈层，其中 K 是下游类别的数量。通常，在比预训练更高的分辨率下进行微调是有益的 (Touvron 等，2019；Kolesnikov 等，2020)。在输入更高分辨率的图像时，我们保持补丁大小不变，这导致有效序列长度更大。视觉变换器可以处理任意序列长度 (受内存限制)，然而，预训练的位置嵌入可能不再有意义。因此，我们根据它们在原始图像中的位置对预训练的位置嵌入进行 2D 插值。请注意，这种分辨率调整和补丁提取是将关于图像的 2D 结构的归纳偏差手动注入到视觉变换器中的唯一点。

4 EXPERIMENTS

4 实验

We evaluate the representation learning capabilities of ResNet, Vision Transformer (ViT), and the hybrid. To understand the data requirements of each model, we pre-train on datasets of varying size and evaluate many benchmark tasks. When considering the computational cost of pre-training the model, ViT performs very favourably, attaining state of the art on most recognition benchmarks at a lower pre-training cost. Lastly, we perform a small experiment using self-supervision, and show that self-supervised ViT holds promise for the future.

我们评估 ResNet、视觉变换器 (ViT) 和混合模型的表征学习能力。为了理解每个模型的数据需求，我们在不同大小的数据集上进行预训练，并评估多个基准任务。在考虑模型预训练的计算成本时，ViT 表现非常良好，在大多数识别基准上以较低的预训练成本达到了最先进的水平。最后，我们进行了一项小型实验，使用自监督学习，并展示了自监督 ViT 对未来的潜力。

4.1 SETUP

4.1 设置

Datasets. To explore model scalability, we use the ILSVRC-2012 ImageNet dataset with 1k classes and 1.3M images (we refer to it as ImageNet in what follows), its superset ImageNet-21k with 21k classes and 14M images (Deng et al. 2009), and JFT (Sun et al. 2017) with 18k classes and 303M high-resolution images. We de-duplicate the pre-training datasets w.r.t. the test sets of the downstream tasks following Kolesnikov et al. (2020). We transfer the models trained on these dataset to several benchmark tasks: ImageNet on the original validation labels and the cleaned-up ReaL labels (Beyer et al. 2020), CIFAR-10/100 (Krizhevsky, 2009), Oxford-IIIT Pets (Parkhi et al., 2012), and Oxford Flowers-102 (Nilsback & Zisserman, 2008). For these datasets, pre-processing follows Kolesnikov et al. (2020).

数据集。为了探索模型的可扩展性，我们使用 ILSVRC-2012 ImageNet 数据集，该数据集包含 1k 类和 1.3M 图像 (以下称为 ImageNet)，其超集 ImageNet-21k 包含 21k 类和 14M 图像 (Deng et al. 2009)，以及 JFT(Sun et al. 2017)，该数据集包含 18k 类和 303M 高分辨率图像。我们根据 Kolesnikov et al. (2020) 的方法，对预训练数据集进行去重，以避免与下游任务的测试集重复。我们将这些数据集上训练的模型转移到几个基准任务上: 在原始验证标签和清理后的 ReaL 标签 (Beyer et al. 2020) 上进行 ImageNet，CIFAR-10/100(Krizhevsky, 2009)，Oxford-IIIT Pets(Parkhi et al., 2012) 和 Oxford Flowers-102(Nilsback & Zisserman, 2008)。对于这些数据集，预处理遵循 Kolesnikov et al. (2020) 的方法。

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

模型	层	隐藏层大小 D	MLP 大小	头部	参数
ViT-基础	12	768	3072	12	86M
ViT-大型	24	1024	4096	16	307M
ViT-巨型	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

表 1: 视觉变换器模型变体的详细信息。

We also evaluate on the 19-task VTAB classification suite (Zhai et al. 2019b). VTAB evaluates low-data transfer to diverse tasks, using 1000 training examples per task. The tasks are divided into three groups: Natural - tasks like the above, Pets, CIFAR, etc. Specialized - medical and satellite imagery, and Structured - tasks that require geometric understanding like localization.

我们还在 19 个任务的 VTAB 分类套件上进行评估 (Zhai et al. 2019b)。VTAB 评估低数据传输到多样化任务的能力，每个任务使用 1000 个训练样本。这些任务分为三组: 自然 - 像上述任务、宠物、CIFAR 等；专业 - 医疗和卫星图像；以及结构化 - 需要几何理解的任务，如定位。

Model Variants. We base ViT configurations on those used for BERT (Devlin et al. 2019), as summarized in Table 1. The "Base" and "Large" models are directly adopted from BERT and we add the larger "Huge" model. In what follows we use brief notation to indicate the model size and the input patch size: for instance, ViT-L/16 means the "Large" variant with 16×16 input patch size. Note that the Transformer's sequence length is inversely proportional to the square of the patch size, thus models with smaller patch size are computationally more expensive.

模型变体。我们基于 BERT (Devlin et al. 2019) 使用的配置来构建 ViT 配置，如表 1 所总结的。"基础" 和 "大型" 模型直接采用自 BERT，并且我们添加了更大的 "巨大" 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 "大型" 变体。请注意，变换器的序列长度与补丁大小的平方成反比，因此具有较小补丁大小的模型在计算上更为昂贵。

For the baseline CNNs, we use ResNet (He et al. 2016), but replace the Batch Normalization layers (Ioffe & Szegedy, 2015) with Group Normalization (Wu & He, 2018), and used standardized convolutions (Qiao et al. 2019). These modifications improve transfer (Kolesnikov et al. 2020), and we denote the modified model "ResNet (BiT)". For the hybrids, we feed the intermediate feature maps into ViT with patch size of one "pixel". To experiment with different sequence lengths, we either (i) take the output of stage 4 of a regular ResNet50 or (ii) remove stage 4, place the same number of layers in stage 3 (keeping the total number of layers), and take the output of this extended stage 3. Option (ii) results in a 4x longer sequence length, and a more expensive ViT model.

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 (Wu & He, 2018)，并使用标准化卷积 (Qiao et al. 2019)。这些修改改善了迁移学习 (Kolesnikov et al. 2020)，我们将修改后的模型称为 "ResNet (BiT)"。对于混合模型，我们将中间特征图输入到补丁大小为一个 "像素" 的 ViT 中。为了实验不同的序列长度，我们要么 (i) 取常规 ResNet50 的第 4 阶段的输出，要么 (ii) 移除第 4 阶段，在第 3 阶段放置相同数量的层 (保持总层数不变)，并取这个扩展的第 3 阶段的输出。选项 (ii) 导致 4x 更长的序列长度，并且 ViT 模型的计算成本更高。

Training & Fine-tuning. We train all models, including ResNets, using Adam (Kingma & Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.999$, a batch size of 4096 and apply a high weight decay of 0.1, which we found to be useful for transfer of all models (Appendix D. 1 shows that, in contrast to common practices, Adam works slightly better than SGD for ResNets in our setting). We use a linear learning rate warmup and decay, see Appendix B. 1 for details. For fine-tuning we use SGD with momentum, batch size 512, for all models, see Appendix B.1.1 For ImageNet results in Table 2 we fine-tuned at higher resolution: 512 for ViT-L/16 and 518 for ViT-H/14, and also used Polyak & Juditsky (1992) averaging with a factor of 0.9999 (Ramachandran et al., 2019, Wang et al., 2020b).

训练与微调。我们使用 Adam (Kingma & Ba, 2015) 训练所有模型，包括 ResNets，使用 $\beta_1 = 0.9, \beta_2 = 0.999$ ，批量大小为 4096，并应用高达 0.1 的权重衰减，我们发现这对于所有模型的迁移学习是有用的 (附录 D.1 显示，与常见做法相反，在我们的设置中，Adam 的表现略优于 SGD)。我们使用线性学习率预热和衰减，详细信息见附录 B.1。对于微调，我们对所有模型使用带动量的 SGD，批量大小为 512，详见附录 B.1.1。对于表 2 中的 ImageNet 结果，我们在更高的分辨率下进行了微调:ViT-L/16 为 512，ViT-H/14 为 518，并且还使用了 Polyak & Juditsky (1992) 的平均法，因子为 0.9999 (Ramachandran et al., 2019, Wang et al., 2020b)。

Metrics. We report results on downstream datasets either through few-shot or fine-tuning accuracy. Fine-tuning accuracies capture the performance of each model after fine-tuning it on the respective dataset. Few-shot accuracies are obtained by solving a regularized least-squares regression problem that maps the (frozen) representation of a subset of training images to $\{-1, 1\}^K$ target vectors. This formulation allows us to recover the exact solution in closed form. Though we mainly focus on fine-tuning performance, we sometimes use linear few-shot accuracies for fast on-the-fly evaluation where fine-tuning would be too costly.

指标。我们通过少量样本或微调准确度报告下游数据集的结果。微调准确度捕捉每个模型在相应数据集上微调后的性能。少量样本准确度是通过解决一个正则化的最小二乘回归问题获得的，该问题将一部分训练图像的 (冻结) 表示映射到 $\{-1, 1\}^K$ 目标向量。这个公式使我们能够以封闭形式恢复精确解。尽管我们主要关注微调性能，但有时我们会使用线性少量样本准确度进行快速的实时评估，因为微调成本过高。

4.2 COMPARISON TO STATE OF THE ART

4.2 与最先进技术的比较

We first compare our largest models - ViT-H/14 and ViT-L/16 - to state-of-the-art CNNs from the literature. The first comparison point is Big Transfer (BiT) (Kolesnikov et al. 2020), which performs supervised transfer learning with large ResNets. The second is Noisy Student (Xie et al., 2020), which is a large EfficientNet trained using semi-supervised learning on ImageNet and JFT- $\overline{300\,\mathrm{M}}$ with the labels removed. Currently, Noisy Student is the state of the art on ImageNet and BiT-L on the other datasets reported here. All models were trained on TPUv3 hardware, and we report the number of TPUv3-core-days taken to pre-train each of them, that is, the number of TPU v3 cores (2 per chip) used for training multiplied by the training time in days.

我们首先将我们最大的模型 - ViT-H/14 和 ViT-L/16 - 与文献中的最先进 CNN 进行比较。第一个比较点是 Big Transfer (BiT) (Kolesnikov et al. 2020)，它使用大型 ResNet 进行监督迁移学习。第二个是 Noisy Student (Xie et al., 2020)，这是一个使用半监督学习在 ImageNet 和 JFT- $\overline{300\,\mathrm{M}}$ 上训练的大型 EfficientNet，标签已被移除。目前，Noisy Student 在 ImageNet 上是最先进的，而 BiT-L 在这里报告的其他数据集上也是如此。所有模型均在 TPUv3 硬件上训练，我们报告了预训练每个模型所需的 TPUv3 核心天数，即用于训练的 TPU v3 核心数量 (每个芯片 2 个) 乘以训练时间 (以天为单位)。

Table 2 shows the results. The smaller ViT-L/16 model pre-trained on JFT-300M outperforms BiT-L (which is pre-trained on the same dataset) on all tasks, while requiring substantially less computational resources to train. The larger model, ViT-H/14, further improves the performance, especially on the more challenging datasets - ImageNet, CIFAR-100, and the VTAB suite. Interestingly, this

表 2 显示了结果。较小的 ViT-L/16 模型在 JFT-300M 上进行预训练，在所有任务上都优于 BiT-L(后者在相同数据集上进行预训练)，同时所需的计算资源显著更少。较大的模型 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，这

	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	$\mathbf{90.72} \pm \mathbf{0.05}$	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	$\mathbf{99.50} \pm 0.06$	99.42 ± 0.03	99.15 ± 0.03	99.37±0.06	-
CIFAR-100	$\mathbf{94.55} \pm 0.04$	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	-
Oxford-IIIT Pets	$\mathbf{97.56} \pm 0.03$	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

	我们的-JFT (ViT-H/14)	我们的-JFT (ViT-L/16)	我们的-I21k (ViT-L/16)	BiT-L (ResNet152x4)	噪声学生 (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	$\mathbf{90.72} \pm \mathbf{0.05}$	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	$\mathbf{99.50} \pm 0.06$	99.42 ± 0.03	99.15 ± 0.03	99.37±0.06	-
CIFAR-100	$\mathbf{94.55} \pm 0.04$	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	-
牛津-印度理工学院宠物	$\mathbf{97.56} \pm 0.03$	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	-
牛津花卉-102	99.68 ± 0.02	99.74±0.00	99.61 ± 0.02	99.63±0.03	-
VTAB(19 个任务)	77.63 ±0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	-
TPUv3 核心天数	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).

表 2: 与流行图像分类基准的最先进技术比较。我们报告了准确率的均值和标准差，平均自三次微调运行。预训练于 JFT-300M 数据集的视觉变换器模型在所有数据集上都优于基于 ResNet 的基线，同时预训练所需的计算资源显著更少。预训练于较小的公共 ImageNet-21k 数据集的 ViT 表现也很好。*Touvron et al. (2020) 报告的结果略有改善 88.5% 。

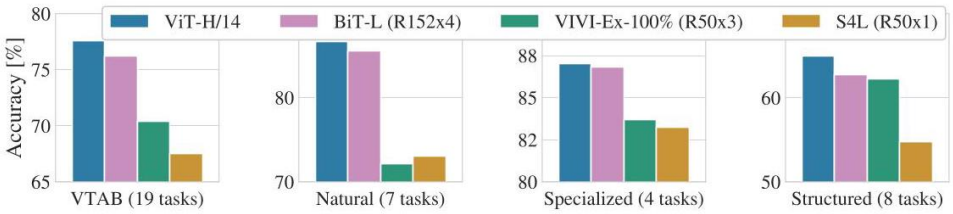


Figure 2: Breakdown of VTAB performance in Natural, Specialized, and Structured task groups.
图 2:VTAB 在自然、专业和结构任务组中的性能分解。

model still took substantially less compute to pre-train than prior state of the art. However, we note that pre-training efficiency may be affected not only by the architecture choice, but also other parameters, such as training schedule, optimizer, weight decay, etc. We provide a controlled study of performance vs. compute for different architectures in Section 4.4. Finally, the ViT-L/16 model pre-trained on the public ImageNet-21k dataset performs well on most datasets too, while taking fewer resources to pre-train: it could be trained using a standard cloud TPUv3 with 8 cores in approximately 30 days.

该模型在预训练时所需的计算资源显著低于之前的最先进水平。然而，我们注意到，预训练效率可能不仅受架构选择的影响，还受到其他参数的影响，例如训练计划、优化器、权重衰减等。我们在第 4.4 节提供了不同架构在性能与计算之间的受控研究。最后，基于公共 ImageNet-21k 数据集预训练的 ViT-L/16 模型在大多数数据集上表现良好，同时所需的预训练资源更少: 它可以在大约 30 天内使用标准的云 TPUv3(8 个核心) 进行训练。

Figure 2 decomposes the VTAB tasks into their respective groups, and compares to previous SOTA methods on this benchmark: BiT, VIVI - a ResNet co-trained on ImageNet and Youtube (Tschannen et al. 2020), and S4L - supervised plus semi-supervised learning on ImageNet (Zhai et al. 2019a). ViT-H/14 outperforms BiT-R152x4, and other methods, on the Natural and Structured tasks. On the Specialized the performance of the top two models is similar.

图 2 将 VTAB 任务分解为各自的组，并与该基准上的先前最先进方法进行了比较:BiT、VIVI - 一个在 ImageNet 和 Youtube 上共同训练的 ResNet(Tschannen 等，2020)，以及 S4L - 在 ImageNet 上进行的监督加半监督学习 (Zhai 等，2019a)。ViT-H/14 在自然和结构任务上超越了 BiT-R152x4 及其他方法。在专业任务上，前两种模型的性能相似。

4.3 PRE-TRAINING DATA REQUIREMENTS

4.3 预训练数据要求

The Vision Transformer performs well when pre-trained on a large JFT-300M dataset. With fewer inductive biases for vision than ResNets, how crucial is the dataset size? We perform two series of experiments.

当在大型 JFT-300M 数据集上进行预训练时，视觉变换器表现良好。与 ResNets 相比，它对视觉的归纳偏差较少，那么数据集的大小有多重要呢? 我们进行了两系列实验。

First, we pre-train ViT models on datasets of increasing size: ImageNet, ImageNet-21k, and JFT-300 M . To boost the performance on the smaller datasets, we optimize three basic regularization parameters - weight decay, dropout, and label smoothing. Figure 3 shows the results after fine-tuning to ImageNet (results on other datasets are shown in Table [5]^2 . When pre-trained on the smallest dataset, ImageNet, ViT-Large models underperform compared to ViT-Base models, despite (moderate) regularization. With ImageNet-21k pre-training, their performances are similar. Only with JFT-300M, do we see the full benefit of larger models. Figure 3 also shows the performance

首先，我们在不断增大的数据集上预训练 ViT 模型:ImageNet、ImageNet-21k 和 JFT- 300 M 。为了提高在较小数据集上的性能，我们优化了三个基本的正则化参数 - 权重衰减、dropout 和标签平滑。图 3 显示了在 ImageNet 上微调后的结果 (其他数据集的结果见表 [5]^2)。当在最小的数据集 ImageNet 上进行预训练时，ViT-Large 模型的表现不如 ViT-Base 模型，尽管进行了 (适度的) 正则化。通过 ImageNet-21k

预训练后，它们的性能相似。只有在 JFT-300M 上，我们才能看到更大模型的全部优势。图 3 还显示了性能

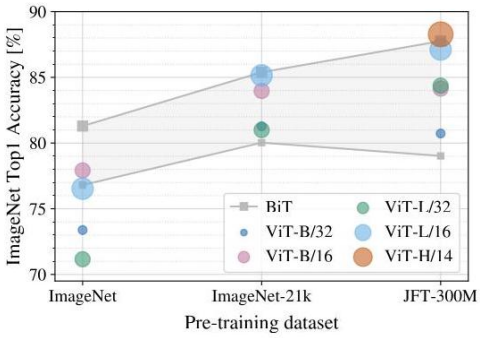


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

图 3: 迁移到 ImageNet。当大型 ViT 模型在小数据集上预训练时，其表现不如 BiT ResNets(阴影区域)，但在较大数据集上预训练时，它们表现出色。同样，随着数据集的增长，较大的 ViT 变体超过较小的变体。

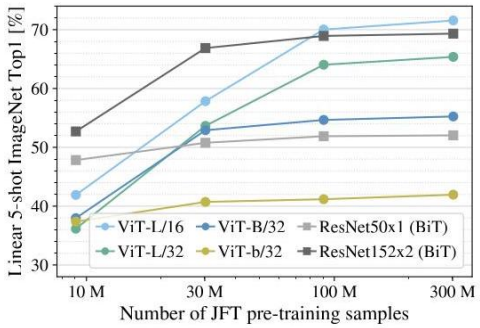
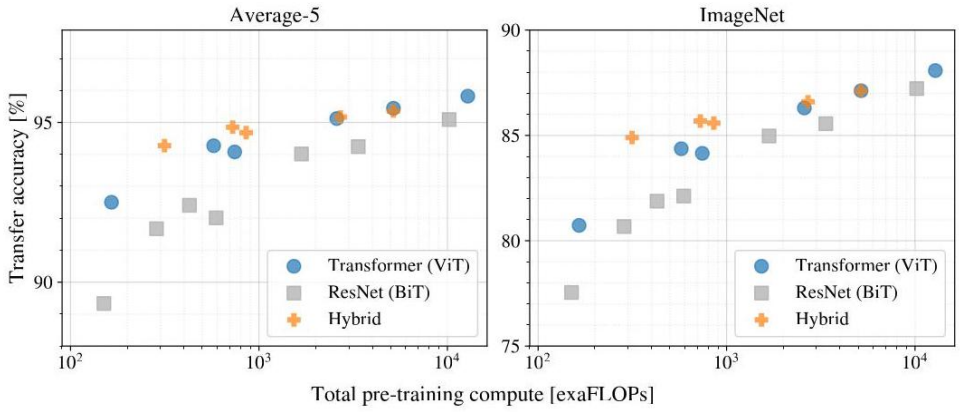


Figure 4: Linear few-shot evaluation on Ima-geNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

图 4: 在 ImageNet 上进行的线性少样本评估与预训练规模的对比。ResNets 在较小的预训练数据集上表现更好，但比 ViT 更早达到性能平台期，而 ViT 在较大的预训练数据集上表现更佳。ViT-b 是将所有隐藏维度减半的 ViT-B。



[2] Note that the ImageNet pre-trained models are also fine-tuned, but again on ImageNet. This is because the resolution increase during fine-tuning improves the performance.

[2] 请注意，ImageNet 预训练模型也经过微调，但仍然是在 ImageNet 上。这是因为微调过程中分辨率的提高改善了性能。

Figure 5: Performance versus cost for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

图 5: 不同架构的性能与成本比较: 视觉变换器、ResNets 和混合模型。视觉变换器通常在相同的计算预算下优于 ResNets。混合模型在较小模型尺寸上优于纯变换器，但在较大模型上这种差距消失。

region spanned by BiT models of different sizes. The BiT CNNs outperform ViT on ImageNet, but with the larger datasets, ViT overtakes.

不同大小的 BiT 模型所覆盖的区域。BiT CNN 在 ImageNet 上的表现优于 ViT，但在更大的数据集上，ViT 超越了它。

Second, we train our models on random subsets of 9M, 30M , and 90M as well as the full JFT- 300M dataset. We do not perform additional regularization on the smaller subsets and use the same hyper-parameters for all settings. This way, we assess the intrinsic model properties, and not the effect of regularization. We do, however, use early-stopping, and report the best validation accuracy achieved during training. To save compute, we report few-shot linear accuracy instead of full fine-tuning accuracy. Figure 4 contains the results. Vision Transformers overfit more than ResNets with comparable computational cost on smaller datasets. For example, ViT-B/32 is slightly faster than ResNet50; it performs much worse on the 9M subset, but better on 90M+ subsets. The same is true for ResNet 152×2 and ViT-L/16. This result reinforces the intuition that the convolutional inductive bias is useful for smaller datasets, but for larger ones, learning the relevant patterns directly from data is sufficient, even beneficial.

其次，我们在 9M, 30M 的随机子集和 90M 以及完整的 JFT- 300M 数据集上训练我们的模型。我们没有对较小的子集进行额外的正则化，并且对所有设置使用相同的超参数。通过这种方式，我们评估的是模型的内在属性，而不是正则化的影响。然而，我们确实使用了早停，并报告了训练期间达到的最佳验证准确率。为了节省计算资源，我们报告少样本线性准确率，而不是完整微调的准确率。图 4 包含了结果。在较小的数据集上，视觉变换器的过拟合程度超过了计算成本相当的 ResNet。例如，ViT-B/32 的速度略快于 ResNet50；但在 9M 子集上的表现要差得多，而在 90M+ 子集上的表现则更好。ResNet 152×2 和 ViT-L/16 也是如此。这个结果强化了这样一种直觉: 卷积归纳偏置对于较小的数据集是有用的，但对于较大的数据集，从数据中直接学习相关模式就足够了，甚至是有益的。

Overall, the few-shot results on ImageNet (Figure 4), as well as the low-data results on VTAB (Table 2) seem promising for very low-data transfer. Further analysis of few-shot properties of ViT is an exciting direction of future work.

总体而言，在 ImageNet 上的少样本结果 (图 4) 以及在 VTAB 上的低数据结果 (表 2) 似乎对非常低数据的迁移是有前景的。对 ViT 的少样本特性的进一步分析是未来工作的一个令人兴奋的方向。

4.4 SCALING STUDY

4.4 扩展研究

We perform a controlled scaling study of different models by evaluating transfer performance from JFT-300M. In this setting data size does not bottleneck the models' performances, and we assess performance versus pre-training cost of each model. The model set includes: 7 ResNets, R50x1, R50x2 R101x1, R152x1, R152x2, pre-trained for 7 epochs, plus R152x2 and R200x3 pre-trained for 14 epochs; 6 Vision Transformers, ViT-B/32, B/16, L/32, L/16, pre-trained for 7 epochs, plus L/16 and H/14 pre-trained for 14 epochs; and 5 hybrids, R50+ViT-B/32, B/16, L/32, L/16 pre-trained for 7 epochs, plus R50+ViT-L/16 pre-trained for 14 epochs (for hybrids, the number at the end of the model name stands not for the patch size, but for the total dowsampling ratio in the ResNet backbone).

我们通过评估从 JFT-300M 的迁移性能，对不同模型进行了受控的规模研究。在这种情况下，数据规模并不限制模型的性能，我们评估了每个模型的性能与预训练成本之间的关系。模型集包括:7 个 ResNet，R50x1、R50x2、R101x1、R152x1、R152x2，预训练 7 个周期，以及 R152x2 和 R200x3 预训练 14 个周期；6 个视觉变换器，ViT-B/32、B/16、L/32、L/16，预训练 7 个周期，以及 L/16 和 H/14 预训练 14 个周期；以及 5 个混合模型，R50+ViT-B/32、B/16、L/32、L/16 预训练 7 个周期，以及 R50+ViT-L/16 预训练 14 个周期 (对于混合模型，模型名称末尾的数字并不代表补丁大小，而是 ResNet 主干中的总下采样比例)。

Figure 5 contains the transfer performance versus total pre-training compute (see Appendix D. 4 for details on computational costs). Detailed results per model are provided in Table 6 in the Appendix. A few patterns can be observed. First, Vision Transformers dominate ResNets on the performance/compute trade-off. ViT uses approximately $2 - 4\times$ less compute to attain the same performance (average over 5

datasets). Second, hybrids slightly outperform ViT at small computational budgets, but the difference vanishes for larger models. This result is somewhat surprising, since one might expect convolutional local feature processing to assist ViT at any size. Third, Vision Transformers appear not to saturate within the range tried, motivating future scaling efforts.

图 5 显示了迁移性能与总预训练计算量的关系 (有关计算成本的详细信息，请参见附录 D.4)。每个模型的详细结果在附录的表 6 中提供。可以观察到一些模式。首先，视觉变换器在性能/计算权衡上优于 ResNet。ViT 使用大约 $2-4\times$ 更少的计算量来获得相同的性能 (在 5 个数据集上的平均值)。其次，在小计算预算下，混合模型略微优于 ViT，但对于较大的模型，这种差异消失。这一结果有些令人惊讶，因为人们可能会期望卷积局部特征处理在任何规模下都能帮助 ViT。第三，视觉变换器在尝试的范围内似乎没有饱和，这为未来的规模扩展工作提供了动力。

4.5 INSPECTING VISION TRANSFORMER

4.5 检查视觉变换器

To begin to understand how the Vision Transformer processes image data, we analyze its internal representations. The first layer of the Vision Transformer linearly projects the flattened patches into a lower-dimensional space (Eq. 1). Figure 7 (left) shows the top principal components of the the learned embedding filters. The components resemble plausible basis functions for a low-dimensional representation of the fine structure within each patch.

为了开始理解视觉变换器如何处理图像数据，我们分析其内部表示。视觉变换器的第一层将展平的图块线性投影到一个低维空间中 (方程 1)。图 7(左) 显示了学习到的嵌入滤波器的主要成分。这些成分类似于对每个图块内部细结构的低维表示的合理基函数。

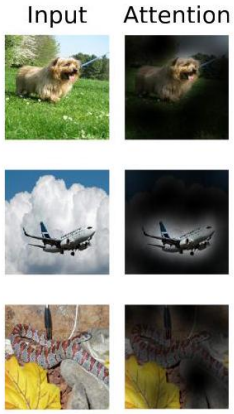


Figure 6: Representative examples of attention from the output token to the input space. See Appendix D. 6 for details.

图 6: 从输出标记到输入空间的注意力代表性示例。有关详细信息，请参见附录 D.6。

After the projection, a learned position embedding is added to the patch representations. Figure 7 (center) shows that the model learns to encode distance within the image in the similarity of position em-beddings, i.e. closer patches tend to have more similar position em-beddings. Further, the row-column structure appears; patches in the same row/column have similar embeddings. Finally, a sinusoidal structure is sometimes apparent for larger grids (Appendix D). That the position embeddings learn to represent 2D image topology explains why hand-crafted 2D-aware embedding variants do not yield improvements (Appendix D.3).

在投影之后，学习到的位置嵌入被添加到图块表示中。图 7(中) 显示模型学习在位置嵌入的相似性中编码图像内的距离，即更靠近的图块往往具有更相似的位置嵌入。此外，行列结构显现；同一行/列中的图块具有相似的嵌入。最后，对于较大的网格，有时会明显出现正弦结构 (附录 D)。位置嵌入学习表示 2D 图像拓扑解释了为什么手工制作的二维感知嵌入变体没有带来改进 (附录 D.3)。

Self-attention allows ViT to integrate information across the entire image even in the lowest layers. We investigate to what degree the network makes use of this capability. Specifically, we compute the average distance in image space across which information is integrated, based on the attention weights (Figure 7, right). This "attention distance" is analogous to receptive field size in CNNs. We find that

some heads attend to most of the image already in the lowest layers, showing that the ability to integrate information globally is indeed used by the model. Other attention heads have consistently small attention distances in the low layers. This highly localized attention is less pronounced in hybrid models that apply a ResNet before the Transformer (Figure 7, right), suggesting that it may serve a similar function as early convolutional layers in CNNs. Further, the attention distance increases with network depth. Globally, we find that the model attends to image regions that are semantically relevant for classification (Figure 6).

自注意力使得 ViT 能够在最低层中整合整个图像的信息。我们研究网络在多大程度上利用了这一能力。具体而言，我们基于注意力权重计算信息整合的平均距离 (图 7，右)。这个“注意力距离”类似于 CNN 中的感受野大小。我们发现某些头在最低层中就已经关注到大部分图像，这表明模型确实利用了全球整合信息的能力。其他注意力头在低层中的注意力距离始终较小。这种高度局部化的注意力在应用 ResNet 于 Transformer 之前的混合模型中不那么明显 (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，随着网络深度的增加，注意力距离也在增加。总体而言，我们发现模型关注于对分类具有语义相关性的图像区域 (图 6)。

4.6 SELF-SUPERVISION

4.6 自监督学习

Transformers show impressive performance on NLP tasks. However, much of their success stems not only from their excellent scalability but also from large scale self-supervised pre-training (Devlin

Transformers 在自然语言处理任务上表现出色。然而，它们成功的原因不仅在于其出色的可扩展性，还在于大规模的自监督预训练 (Devlin)。

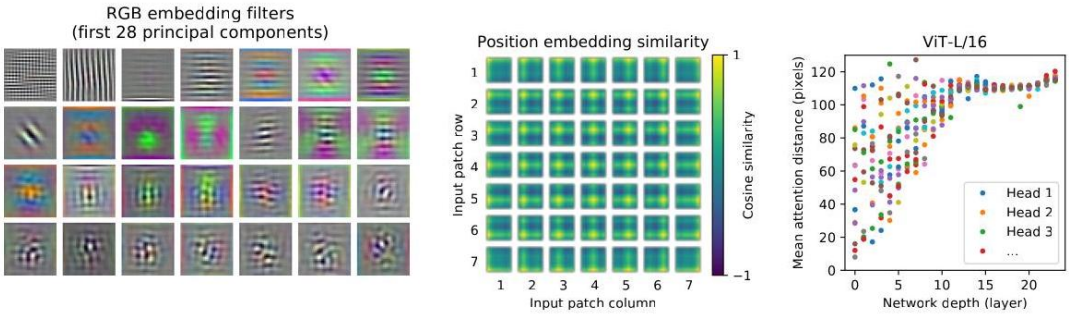


Figure 7: Left: Filters of the initial linear embedding of RGB values of ViT-L/32. Center: Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches. Right: Size of attended area by head and network depth. Each dot shows the mean attention distance across images for one of 16 heads at one layer. See Appendix D. 6 for details.

图 7: 左:ViT-L/32 的 RGB 值初始线性嵌入的滤波器。中:ViT-L/32 的位置嵌入的相似性。方块显示了具有指定行和列的补丁的位置嵌入与所有其他补丁的位置嵌入之间的余弦相似性。右: 按头和网络深度划分的关注区域大小。每个点显示了在某一层中 16 个头之一的图像平均注意力距离。有关详细信息，请参见附录 D.6。

et al. 2019 Radford et al. 2018). We also perform a preliminary exploration on masked patch prediction for self-supervision, mimicking the masked language modeling task used in BERT. With self-supervised pre-training, our smaller ViT-B/16 model achieves 79.9% accuracy on ImageNet, a significant improvement of 2% to training from scratch, but still 4% behind supervised pre-training. Appendix B.1.2 contains further details. We leave exploration of contrastive pre-training (Chen et al., 2020b, He et al., 2020, Bachman et al., 2019, Hénaff et al., 2020) to future work.

et al. 2019 Radford et al. 2018)。我们还对自我监督的掩码补丁预测进行了初步探索，模仿了 BERT 中使用的掩码语言建模任务。通过自我监督的预训练，我们较小的 ViT-B/16 模型在 ImageNet 上达到了 79.9% 的准确率，相较于从头训练提高了 2%，但仍比监督预训练低 4%。附录 B.1.2 包含更多细节。我们将对对比预训练 (Chen et al., 2020b, He et al., 2020, Bachman et al., 2019, Hénaff et al., 2020) 的探索留待未来工作。

5 CONCLUSION

5 结论

We have explored the direct application of Transformers to image recognition. Unlike prior works using self-attention in computer vision, we do not introduce image-specific inductive biases into the architecture apart from the initial patch extraction step. Instead, we interpret an image as a sequence of patches and process it by a standard Transformer encoder as used in NLP. This simple, yet scalable, strategy works surprisingly well when coupled with pre-training on large datasets. Thus, Vision Transformer matches or exceeds the state of the art on many image classification datasets, whilst being relatively cheap to pre-train.

我们探讨了 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，我们没有在架构中引入图像特定的归纳偏置，除了最初的补丁提取步骤。相反，我们将图像解释为补丁序列，并通过标准的 Transformer 编码器进行处理，正如在自然语言处理 (NLP) 中使用的那样。这种简单而可扩展的策略在与大规模数据集的预训练结合时表现得相当出色。因此，视觉 Transformer 在许多图像分类数据集上的表现与当前最先进的技术相匹配或超过，同时预训练的成本相对较低。

While these initial results are encouraging, many challenges remain. One is to apply ViT to other computer vision tasks, such as detection and segmentation. Our results, coupled with those in Carion et al. (2020), indicate the promise of this approach. Another challenge is to continue exploring self-supervised pre-training methods. Our initial experiments show improvement from self-supervised pre-training, but there is still large gap between self-supervised and large-scale supervised pretraining. Finally, further scaling of ViT would likely lead to improved performance.

尽管这些初步结果令人鼓舞，但仍然面临许多挑战。其中之一是将 ViT 应用于其他计算机视觉任务，如检测和分割。我们的结果与 Carion et al. (2020) 的结果相结合，表明这种方法的潜力。另一个挑战是继续探索自我监督的预训练方法。我们的初步实验显示自我监督预训练带来了改进，但自我监督与大规模监督预训练之间仍存在较大差距。最后，进一步扩大 ViT 的规模可能会带来性能的提升。

ACKNOWLEDGEMENTS

致谢

The work was performed in Berlin, Zürich, and Amsterdam. We thank many colleagues at Google for their help, in particular Andreas Steiner for crucial help with the infrastructure and the open-source release of the code; Joan Puigcerver and Maxim Neumann for help with the large-scale training infrastructure; Dmitry Lepikhin, Aravindh Mahendran, Daniel Keysers, Mario Lučić, Noam Shazeer, and Colin Raffel for useful discussions.

该工作在柏林、苏黎世和阿姆斯特丹进行。我们感谢谷歌的许多同事的帮助，特别感谢安德烈亚斯 · 施泰纳在基础设施和代码开源发布方面的关键帮助；感谢乔安 · 普伊克塞尔和马克西姆 · 诺伊曼在大规模训练基础设施方面的帮助；感谢德米特里 · 列皮金、阿拉文德 · 马亨德兰、丹尼尔 · 凯瑟斯、马里奥 · 卢奇奇、诺姆 · 沙泽尔和科林 · 拉费尔的有益讨论。

REFERENCES

参考文献

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In ACL, 2020.

Samira Abnar 和 Willem Zuidema. 在变压器中量化注意力流. 发表在 ACL, 2020.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In NeurIPS, 2019.

Philip Bachman, R Devon Hjelm 和 William Buchwalter. 通过最大化视图间的互信息来学习表示. 发表在 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: Pretraining 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 pretraining. 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. S^4 L : 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

模型	数据集	训练周期	基础学习率	学习率衰减	权重衰减	随机失活
ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	线性	0.1	0.0
ViT-L/32	JFT-300M	7	$6 \cdot 10^{-4}$	线性	0.1	0.0
ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	线性	0.1	0.0
ViT-H/14	JFT-300M	14	$3 \cdot 10^{-4}$	线性	0.1	0.0
R50x{1,2}	JFT-300M	7	10^{-3}	线性	0.1	0.0
R101x1	JFT-300M	7	$8 \cdot 10^{-4}$	线性	0.1	0.0
R152x{1,2}	JFT-300M	7	$6 \cdot 10^{-4}$	线性	0.1	0.0
R50+ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	线性	0.1	0.0
R50+ViT-L/32	JFT-300M	7	$2 \cdot 10^{-4}$	线性	0.1	0.0
R50+ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	线性	0.1	0.0
ViT-B/{16,32}	ImageNet-21k	90	10^{-3}	线性	0.03	0.1
ViT-L/{16,32}	ImageNet-21k	30/90	10^{-3}	线性	0.03	0.1
ViT-*	ImageNet	300	$3 \cdot 10^{-3}$	余弦	0.3	0.1

Table 3: Hyperparameters for training. All models are trained with a batch size of 4096 and learning rate warmup of 10k steps. For ImageNet we found it beneficial to additionally apply gradient clipping at global norm 1. Training resolution is 224.

表 3: 训练的超参数。所有模型均以批量大小 4096 进行训练，并且学习率预热为 10k 步。对于 ImageNet，我们发现额外应用全局范数为 1 的梯度裁剪是有益的。训练分辨率为 224。

APPENDIX

附录

A MULTIHEAD SELF-ATTENTION

多头自注意力

Standard qkv self-attention (SA, Vaswani et al. (2017)) is a popular building block for neural architectures. For each element in an input sequence $\mathbf{z} \in \mathbb{R}^{N \times D}$, we compute a weighted sum over all values $\mathbf{v}$ in the sequence. The attention weights A_{ij} are based on the pairwise similarity between two elements of the sequence and their respective query $\mathbf{q}^i$ and key $\mathbf{k}^j$ representations.

标准的 qkv 自注意力 (SA, Vaswani 等人 (2017)) 是神经架构中的一个流行构建块。对于输入序列中的每个元素 $\mathbf{z} \in \mathbb{R}^{N \times D}$ ，我们计算序列中所有值 $\mathbf{v}$ 的加权和。注意力权重 A_{ij} 基于序列中两个元素之间的成对相似性以及它们各自的查询 $\mathbf{q}^i$ 和键 $\mathbf{k}^j$ 表示。

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

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

$$\mathrm{SA}\left(\mathbf{z}\right) = A\mathbf{v}. \tag{7}$$

Multihead self-attention (MSA) is an extension of SA in which we run k self-attention operations, called "heads", in parallel, and project their concatenated outputs. To keep compute and number of parameters constant when changing k, D_h (Eq. 5) is typically set to D/k .

多头自注意力 (MSA) 是 SA 的扩展，其中我们并行运行 k 自注意力操作，称为 "头"，并投影它们的连接输出。为了在改变 k, D_h 时保持计算和参数数量不变，通常将其设置为 D/k 。

$$\mathrm{MSA}\left(\mathbf{z}\right)=\left[\mathrm{SA}_1\left(z\right);\mathrm{SA}_2\left(z\right);\cdots;\mathrm{SA}_k\left(z\right)\right]\mathbf{U}_{msa}\ \mathbf{U}_{msa}\in\mathbb{R}^{k\cdot D_h\times D} \tag{8}$$

B EXPERIMENT DETAILS

B 实验细节

B.1 TRAINING

B.1 训练

Table 3 summarizes our training setups for our different models. We found strong regularization to be key when training models from scratch on ImageNet. Dropout, when used, is applied after every dense layer except for the the qkv-projections and directly after adding positional- to patch embeddings. Hybrid models are trained with the exact setup as their ViT counterparts. Finally, all training is done on resolution 224.

表 3 总结了我们不同模型的训练设置。我们发现强正则化在从头开始训练模型时对 ImageNet 至关重要。Dropout 在使用时应用于每个密集层之后，除了 qkv 投影和在添加位置到补丁嵌入之后直接应用。混合模型的训练设置与其 ViT 对应模型完全相同。最后，所有训练均在分辨率 224 上进行。

B.1.1 FINE-TUNING

B.1.1 微调

We fine-tune all ViT models using SGD with a momentum of 0.9 . We run a small grid search over learning rates, see learning rate ranges in Table 4 To do so, we use small sub-splits from the training set (10% for Pets and Flowers, 2% for CIFAR, 1% ImageNet) as development set and train on the remaining data. For final results we train on the entire training set and evaluate on the respective test data. For fine-tuning ResNets and hybrid models we use the exact same setup, with the only exception of ImageNet where we add another value 0.06 to the learning rate sweep. Additionally,

我们使用动量为 0.9 的 SGD 对所有 ViT 模型进行微调。我们对学习率进行小规模网格搜索，学习率范围见表 4。为此，我们使用来自训练集的小子集 (宠物和花卉的 10%，CIFAR 的 2%，ImageNet 的 1%) 作为开发集，并在剩余数据上进行训练。对于最终结果，我们在整个训练集上进行训练，并在相应的测试数据上进行评估。对于 ResNets 和混合模型的微调，我们使用完全相同的设置，唯一的例外是 ImageNet，我们在学习率范围中增加了另一个值 0.06。此外，

Dataset	Steps	Base LR
ImageNet	20000	$\{0.003, 0.01, 0.03, 0.06\}$
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)	2500	0.01

数据集	步骤	基础学习率
ImageNet	20000	$\{0.003, 0.01, 0.03, 0.06\}$
CIFAR100	10 000	$\{0.001, 0.003, 0.01, 0.03\}$
CIFAR10	10 000	$\{0.001, 0.003, 0.01, 0.03\}$
牛津-印度理工学院宠物	500	$\{0.001, 0.003, 0.01, 0.03\}$
牛津花卉-102	500	$\{0.001, 0.003, 0.01, 0.03\}$
VTAB(19 个任务)	2500	0.01

Table 4: Hyperparameters for fine-tuning. All models are fine-tuned with cosine learning rate decay, a batch size of 512, no weight decay, and grad clipping at global norm 1 . If not mentioned otherwise, fine-tuning resolution is 384.

表 4: 微调的超参数。所有模型均使用余弦学习率衰减进行微调，批量大小为 512，无权重衰减，梯度裁剪为全局范数 1。如果没有特别说明，微调分辨率为 384。

for ResNets we also run the setup of Kolesnikov et al. (2020) and select the best results across this run and our sweep. Finally, if not mentioned otherwise, all fine-tuning experiments run at 384 resolution (running fine-tuning at different resolution than training is common practice (Kolesnikov et al., 2020)).

对于 ResNets，我们还运行 Kolesnikov 等人 (2020) 的设置，并选择该运行和我们的搜索中最佳结果。最后，如果没有特别说明，所有微调实验均在 384 分辨率下进行 (在与训练不同的分辨率下进行微调是常见做法 (Kolesnikov 等人，2020))。

When transferring ViT models to another dataset, we remove the whole head (two linear layers) and replace it by a single, zero-initialized linear layer outputting the number of classes required by the target dataset. We found this to be a little more robust than simply re-initializing the very last layer.

在将 ViT 模型迁移到另一个数据集时，我们移除整个头部 (两个线性层)，并用一个单一的、零初始化的线性层替换它，该层输出目标数据集所需的类别数量。我们发现这种方法比简单地重新初始化最后一层要更稳健一些。

For VTAB we follow the protocol in Kolesnikov et al. (2020), and use the same hyperparameter setting for all tasks. We use a learning rate of 0.01 and train for 2500 steps (Tab. 4). We chose this setting by running a small sweep over two learning rates and two schedules, and selecting the setting with the highest VTAB score on the 200-example validation sets. We follow the pre-processing used in Kolesnikov et al. (2020), except that we do not use task-specific input resolutions. Instead we find that Vision Transformer benefits most from a high resolution (384×384) for all tasks.

对于 VTAB，我们遵循 Kolesnikov 等人 (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 (表 4)。我们通过对两个学习率和两个调度进行小范围的测试来选择这个设置，并选择在 200 个样本的验证集上获得最高 VTAB 分数的设置。我们遵循 Kolesnikov 等人 (2020) 中使用的预处理，除了我们不使用特定任务的输入分辨率。相反，我们发现 Vision Transformer 在所有任务中都最受益于高分辨率 (384×384) 。

B.1.2 SELF-SUPERVISION

B.1.2 自我监督

We employ the masked patch prediction objective for preliminary self-supervision experiments. To do so we corrupt 50% of patch embeddings by either replacing their embeddings with a learnable [mask] embedding (80%) , a random other patch embedding (10%) or just keeping them as is (10%). This setup is very similar to the one used for language by Devlin et al. (2019). Finally, we predict the 3-bit, mean color (i.e., 512 colors in total) of every corrupted patch using their respective patch representations.

我们采用掩码补丁预测目标进行初步的自我监督实验。为此，我们通过将补丁嵌入的某些部分替换为可学习的 [mask] 嵌入 (80%) 、随机其他补丁嵌入 (10%) 或保持原样 (10%) 来破坏 50% 的补丁嵌入。这个设置与 Devlin 等人 (2019) 用于语言的设置非常相似。最后，我们使用各自的补丁表示预测每个损坏补丁的 3 位平均颜色 (即，总共 512 种颜色)。

We trained our self-supervised model for 1M steps (ca. 14 epochs) with batch size 4096 on JFT. We use Adam, with a base learning rate of $2 \cdot 10^{-4}$, warmup of 10k steps and cosine learning rate decay. As prediction targets for pretraining we tried the following settings: 1) predicting only the mean, 3 bit color (i.e.,1 prediction of 512 colors),2) predicting a 4×4 downsized version of the 16×16 patch with 3bit colors in parallel (i.e., 16 predictions of 512 colors), 3) regression on the full patch using L2 (i.e., 256 regressions on the 3 RGB channels). Surprisingly, we found that all worked quite well, though L2 was slightly worse. We report final results only for option 1) because it has shown best few-shot performance. We also experimented with 15% corruption rate as used by Devlin et al. 2019) but results were also slightly worse on our few-shot metrics.

我们在 JFT 上训练了我们的自监督模型 1M 步 (大约 14 个周期)，批量大小为 4096。我们使用 Adam 优化器，基础学习率为 $2 \cdot 10^{-4}$ ，预热步骤为 10k ，并采用余弦学习率衰减。作为预训练的预测目标，我们尝试了以下设置:1) 仅预测均值，3 位颜色 (即，512 种颜色的 1 个预测)，2) 并行预测 4×4 降级版本的 16×16 补丁，使用 3 位颜色 (即，512 种颜色的 16 个预测)，3) 使用 L2 对完整补丁进行回归 (即，对 3 个 RGB 通道进行 256 次回归)。令人惊讶的是，我们发现所有方法都表现得相当不错，尽管 L2 的效

果稍差。我们仅报告选项 1) 的最终结果，因为它显示了最佳的少量样本性能。我们还实验了 Devlin 等人 (2019) 使用的 15% 损坏率，但在我们的少量样本指标上结果也稍差。

Lastly, we would like to remark that our instantiation of masked patch prediction doesn't require such an enormous amount of pretraining nor a large dataset such as JFT in order to lead to similar performance gains on ImageNet classification. That is, we observed diminishing returns on downstream performance after 100k pretraining steps, and see similar gains when pretraining on ImageNet.

最后，我们想指出，我们的掩蔽补丁预测实例化并不需要如此巨大的预训练量或像 JFT 这样的大型数据集，就能在 ImageNet 分类上获得类似的性能提升。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 上进行预训练时也看到类似的提升。

C ADDITIONAL RESULTS

C 附加结果

We report detailed results corresponding to the figures presented in the paper. Table 5 corresponds to Figure 3 from the paper and shows transfer performance of different ViT models pre-trained on datasets of increasing size: ImageNet, ImageNet-21k, and JFT-300M. Table 6 corresponds to

我们报告与论文中呈现的图形相对应的详细结果。表 5 对应于论文中的图 3，显示了在不同大小数据集 (ImageNet、ImageNet-21k 和 JFT-300M) 上预训练的不同 ViT 模型的迁移性能。表 6 对应于

		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

		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	-
	牛津花卉-102	89.49	85.43	89.66	86.36	-
	牛津-印度理工学院宠物	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
	牛津花卉-102	99.38	99.11	99.61	99.19	99.51
	牛津-印度理工学院宠物	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
	牛津花卉-102	99.56	99.27	99.56	99.45	99.68
	牛津-印度理工学院宠物	95.80	93.40	97.11	95.83	97.56

Table 5: Top1 accuracy (in %) of Vision Transformer on various datasets when pre-trained on ImageNet, ImageNet-21k or JFT300M. These values correspond to Figure 3 in the main text. Models are fine-tuned at 384 resolution. Note that the ImageNet results are computed without additional techniques (Polyak averaging and 512 resolution images) used to achieve results in Table 2

表 5: 在 Im-ageNet、ImageNet-21k 或 JFT300M 上预训练时，Vision Transformer 在各种数据集上的 Top1 准确率 (以% 计)。这些值对应于主文本中的图 3。模型在 384 分辨率下进行微调。请注意，ImageNet 的结果是在没有使用额外技术 (Polyak 平均和 512 分辨率图像) 来实现表 2 中结果的情况下计算的。

Model	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	164
ViT-B/16	7	84.15	88.85	99.00	91.87	95.80	99.56	743
ViT-L/32	7	84.37	88.28	99.19	92.52	95.83	99.45	574
ViT-L/16	7	86.30	89.43	99.38	93.46	96.81	99.66	2586
ViT-L/16	14	87.12	89.99	99.38	94.04	97.11	99.56	5172
ViT-H/14	14	88.08	90.36	99.50	94.71	97.11	99.71	12826
ResNet50x1	7	77.54	84.56	97.67	86.07	91.11	94.26	150
ResNet50x2	7	82.12	87.94	98.29	89.20	93.43	97.02	592
ResNet101x1	7	80.67	87.07	98.48	89.17	94.08	95.95	285
ResNet152x1	7	81.88	87.96	98.82	90.22	94.17	96.94	427
ResNet152x2	7	84.97	89.69	99.06	92.05	95.37	98.62	1681
ResNet152x2	14	85.56	89.89	99.24	91.92	95.75	98.75	3362
ResNet200x3	14	87.22	90.15	99.34	93.53	96.32	99.04	10212
R50x1+ViT-B/32	7	84.90	89.15	99.01	92.24	95.75	99.46	315
R50x1+ViT-B/16	7	85.58	89.65	99.14	92.63	96.65	99.40	855
R50x1+ViT-L/32	7	85.68	89.04	99.24	92.93	96.97	99.43	725
R50x1+ViT-L/16	7	86.60	89.72	99.18	93.64	97.03	99.40	2704
R50x1+ViT-L/16	14	87.12	89.76	99.31	93.89	97.36	99.11	5165

模型	训练周期	ImageNet	ImageNet ReaL	CIFAR-10	CIFAR-100	宠物	花卉	exaFLOPs
ViT-B/32	7	80.73	86.27	98.61	90.49	93.40	99.27	164
ViT-B/16	7	84.15	88.85	99.00	91.87	95.80	99.56	743
ViT-L/32	7	84.37	88.28	99.19	92.52	95.83	99.45	574
ViT-L/16	7	86.30	89.43	99.38	93.46	96.81	99.66	2586
ViT-L/16	14	87.12	89.99	99.38	94.04	97.11	99.56	5172
ViT-H/14	14	88.08	90.36	99.50	94.71	97.11	99.71	12826
ResNet50x1	7	77.54	84.56	97.67	86.07	91.11	94.26	150
ResNet50x2	7	82.12	87.94	98.29	89.20	93.43	97.02	592
ResNet101x1	7	80.67	87.07	98.48	89.17	94.08	95.95	285
ResNet152x1	7	81.88	87.96	98.82	90.22	94.17	96.94	427
ResNet152x2	7	84.97	89.69	99.06	92.05	95.37	98.62	1681
ResNet152x2	14	85.56	89.89	99.24	91.92	95.75	98.75	3362
ResNet200x3	14	87.22	90.15	99.34	93.53	96.32	99.04	10212
R50x1+ViT-B/32	7	84.90	89.15	99.01	92.24	95.75	99.46	315
R50x1+ViT-B/16	7	85.58	89.65	99.14	92.63	96.65	99.40	855
R50x1+ViT-L/32	7	85.68	89.04	99.24	92.93	96.97	99.43	725
R50x1+ViT-L/16	7	86.60	89.72	99.18	93.64	97.03	99.40	2704
R50x1+ViT-L/16	14	87.12	89.76	99.31	93.89	97.36	99.11	5165

Table 6: Detailed results of model scaling experiments. These correspond to Figure 5 in the main paper.

表 6: 模型扩展实验的详细结果。这些结果对应于主论文中的图 5。

Figure 5 from the paper and shows the transfer performance of ViT, ResNet, and hybrid models of varying size, as well as the estimated computational cost of their pre-training.

论文中的图 5 显示了不同规模的 ViT、ResNet 和混合模型的迁移性能，以及它们预训练的估计计算成本。

D ADDITIONAL ANALYSES

D 额外分析

D. 1 SGD vs. ADAM FOR RESNETS

D. 1 SGD 与 ADAM 在 ResNets 中的比较

ResNets are typically trained with SGD and our use of Adam as optimizer is quite unconventional. Here we show the experiments that motivated this choice. Namely, we compare the fine-tuning performance of two ResNets - 50x1 and 152x2 - pre-trained on JFT with SGD and Adam. For SGD, we use the hyperparameters recommended by Kolesnikov et al. (2020). Results are presented

ResNets 通常使用 SGD 进行训练，而我们使用 Adam 作为优化器则相当不寻常。在这里，我们展示了促使这一选择的实验。具体而言，我们比较了在 JFT 上使用 SGD 和 Adam 预训练的两个 ResNets - 50x1 和 152x2 的微调性能。对于 SGD，我们使用 Kolesnikov 等人 (2020) 推荐的超参数。结果已呈现。

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

数据集	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
牛津-印度理工学院宠物	91.11	91.00	95.37	94.79
牛津花卉-102	94.26	92.06	98.62	99.32
平均	89.33	88.79	94.01	93.72

Table 7: Fine-tuning ResNet models pre-trained with Adam and SGD.
表 7: 使用 Adam 和 SGD 预训练的 ResNet 模型的微调。

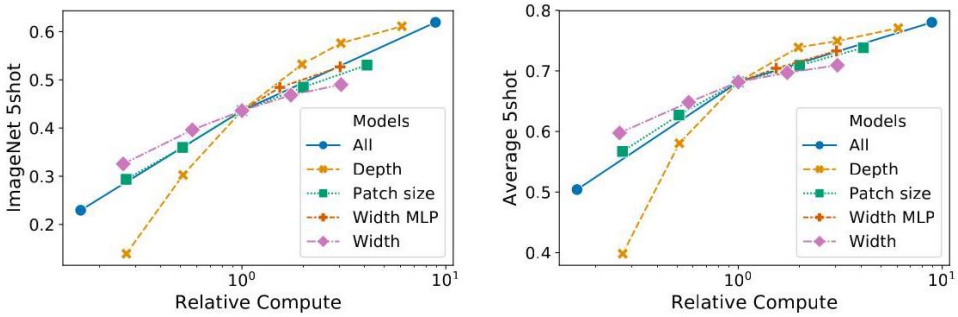


Figure 8: Scaling different model dimensions of the Vision Transformer.
图 8: 扩展 Vision Transformer 的不同模型维度。

in Table 7. Adam pre-training outperforms SGD pre-training on most datasets and on average. This justifies the choice of Adam as the optimizer used to pre-train ResNets on JFT. Note that the absolute numbers are lower than those reported by Kolesnikov et al. (2020), since we pre-train only for 7 epochs, not 30 .

在表 7 中。Adam 预训练在大多数数据集上以及平均而言优于 SGD 预训练。这证明了选择 Adam 作为在 JFT 上预训练 ResNets 的优化器的合理性。请注意，绝对数字低于 Kolesnikov 等人 (2020) 报告的数字，因为我们仅预训练了 7 个周期，而不是 30 个。

D.2 TRANSFORMER SHAPE

D.2 变换器形状

We ran ablations on scaling different dimensions of the Transformer architecture to find out which are best suited for scaling to very large models. Figure 8 shows 5-shot performance on ImageNet for different configurations. All configurations are based on a ViT model with 8 layers, $D = 1024$, $D_{MLP} = 2048$ and

a patch size of 32, the intersection of all lines. We can see that scaling the depth results in the biggest improvements which are clearly visible up until 64 layers. However, diminishing returns are already visible after 16 layers. Interestingly, scaling the width of the network seems to result in the smallest changes. Decreasing the patch size and thus increasing the effective sequence length shows surprisingly robust improvements without introducing parameters. These findings suggest that compute might be a better predictor of performance than the number of parameters, and that scaling should emphasize depth over width if any. Overall, we find that scaling all dimensions proportionally results in robust improvements.

我们对 Transformer 架构的不同维度进行了消融实验，以找出哪些最适合扩展到非常大的模型。图 8 显示了不同配置下在 ImageNet 上的 5-shot 性能。所有配置均基于具有 8 层的 ViT 模型，$D = 1024$ ，$D_{MLP} = 2048$ 和补丁大小为 32，所有线的交点。我们可以看到，增加深度带来了最大的改进，这在 64 层之前是明显可见的。然而，在 16 层之后，收益递减已经显现。有趣的是，扩展网络的宽度似乎导致的变化最小。减小补丁大小，从而增加有效序列长度，显示出令人惊讶的稳健改进，而没有引入参数。这些发现表明，计算可能是性能的更好预测指标，而不是参数的数量，并且如果有扩展，应该强调深度而非宽度。总体而言，我们发现按比例扩展所有维度会导致稳健的改进。

D.3 Positional EMBEDDING

D.3 位置嵌入

We ran ablations on different ways of encoding spatial information using positional embedding. We tried the following cases:

我们对使用位置嵌入编码空间信息的不同方法进行了消融实验。我们尝试了以下几种情况:

- Providing no positional information: Considering the inputs as a bag of patches.
- 不提供位置相关信息: 将输入视为补丁的集合。
- 1-dimensional positional embedding: Considering the inputs as a sequence of patches in the raster order (default across all other experiments in this paper).
- 一维位置嵌入: 将输入视为按光栅顺序排列的补丁序列 (本文所有其他实验的默认设置)。
- 2-dimensional positional embedding: Considering the inputs as a grid of patches in two dimensions. In this case, two sets of embeddings are learned, each for one of the axes, X -embedding, and Y -embedding, each with size $D/2$. Then, based on the coordinate on
- 二维位置嵌入: 将输入视为二维补丁网格。在这种情况下，学习了两组嵌入，每组对应一个轴，X -嵌入和 Y -嵌入，每个的大小为 $D/2$ 。然后，基于坐标

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

正位嵌入	默认/主干	每一层	每一层-共享
无正位嵌入	0.61382	不适用	不适用
一维正位嵌入	0.64206	0.63964	0.64292
2-D 位置嵌入	0.64001	0.64046	0.64022
相对位置嵌入	0.64032	不适用	不适用

Table 8: Results of the ablation study on positional embeddings with ViT-B/16 model evaluated on ImageNet 5-shot linear.

表 8: 使用 ViT-B/16 模型在 ImageNet 5-shot 线性评估的位置信息消融研究结果。

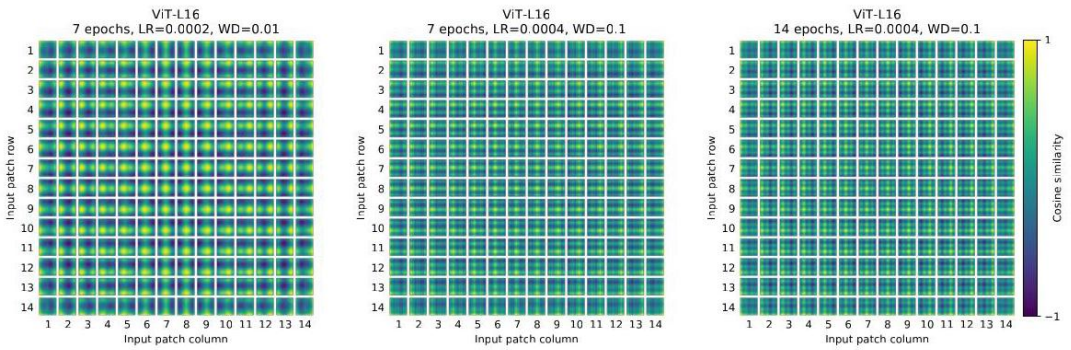


Figure 9: Position embeddings of models trained with different hyperparameters.
图 9: 使用不同超参数训练的模型的位置信息嵌入。

the path in the input, we concatenate the X and Y embedding to get the final positional embedding for that patch.
在输入路径中，我们将 X 和 Y 嵌入连接起来，以获得该补丁的最终位置嵌入。

- Relative positional embeddings: Considering the relative distance between patches to encode the spatial information as instead of their absolute position. To do so, we use 1- dimensional Relative Attention, in which we define the relative distance all possible pairs of patches. Thus, for every given pair (one as query, and the other as key/value in the attention mechanism), we have an offset $p_q - p_k$, where each offset is associated with an embedding. Then, we simply run extra attention, where we use the original query (the content of query), but use relative positional embeddings as keys. We then use the log-its from the relative attention as a bias term and add it to the logits of the main attention (content-based attention) before applying the softmax.

- 相对位置嵌入: 考虑补丁之间的相对距离，以编码空间信息，而不是它们的绝对位置。为此，我们使用一维相对注意力，在其中定义所有可能的补丁对之间的相对距离。因此，对于每一对给定的补丁 (一个作为查询，另一个作为键/值在注意力机制中)，我们有一个偏移量 $p_q - p_k$ ，每个偏移量都与一个嵌入相关联。然后，我们简单地运行额外的注意力，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，我们使用来自相对注意力的对数值作为偏置项，并将其添加到主注意力 (基于内容的注意力) 的对数值中，然后再应用 softmax。

In addition to different ways of encoding spatial information, we also tried different ways of incorporating this information in our model. For the 1-dimensional and 2-dimensional positional embeddings, we tried three different cases: (1) add positional embeddings to the inputs right after the stem of them model and before feeding the inputs to the Transformer encoder (default across all other experiments in this paper); (2) learn and add positional embeddings to the inputs at the beginning of each layer; (3) add a learned positional embeddings to the inputs at the beginning of each layer (shared between layers).
除了编码空间信息的不同方式外，我们还尝试了将这些信息纳入我们模型的不同方法。对于一维和二维位置嵌入，我们尝试了三种不同的情况:(1) 在模型的主干之后立即将位置嵌入添加到输入中，然后再将输入馈送到 Transformer 编码器 (本文所有其他实验的默认设置)；(2) 在每一层的开始学习并添加位置嵌入到输入中；(3) 在每一层的开始将学习到的位置嵌入添加到输入中 (在层之间共享)。

Table 8 summarizes the results from this ablation study on a ViT-B/16 model. As we can see, while there is a large gap between the performances of the model with no positional embedding and models with positional embedding, there is little to no difference between different ways of encoding positional information. We speculate that since our Transformer encoder operates on patch-level inputs, as opposed to pixel-level, the differences in how to encode spatial information is less important. More precisely, in patch-level inputs, the spatial dimensions are much smaller than the original pixel-level inputs, e.g., 14×14 as opposed to 224×224 , and learning to represent the spatial relations in this resolution is equally easy for these different positional encoding strategies. Even so, the specific pattern of position embedding similarity learned by the network depends on the training hyperparameters (Figure 9).
表 8 总结了关于 ViT-B/16 模型的消融研究结果。正如我们所看到的，虽然没有位置嵌入的模型与具有位置嵌入的模型之间的性能差距很大，但不同编码位置信息的方法之间几乎没有差异。我们推测，由于我们的 Transformer 编码器在补丁级输入上操作，而不是像素级输入，因此编码空间信息的方式差异不那么重要。更准确地说，在补丁级输入中，空间维度远小于原始像素级输入，例如，14×14 与 224×224 相对，并且在这种分辨率下学习表示空间关系对这些不同的位置编码策略来说同样容易。即便如此，网络学习到的位置嵌入相似性的具体模式依赖于训练超参数 (图 9)。

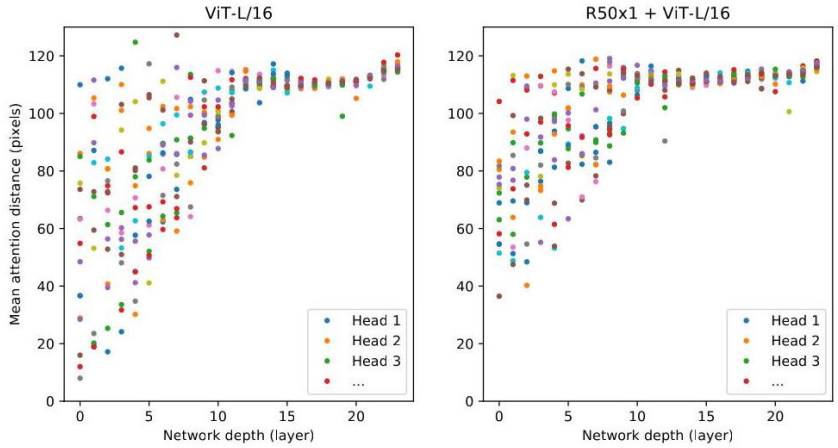


Figure 10: Size of attended area by head and network depth. Attention distance was computed for 128 example images by averaging the distance between the query pixel and all other pixels, weighted by the attention weight. Each dot shows the mean attention distance across images for one of 16 heads at one layer. Image width is 224 pixels.

图 10: 按头和网络深度计算的关注区域大小。通过对 128 个示例图像的查询像素与所有其他像素之间的距离进行加权平均，计算了注意力距离。每个点表示在某一层中 16 个头之一的图像平均注意力距离。图像宽度为 224 像素。

D.4 EMPIRICAL COMPUTATIONAL COSTS

D.4 实证计算成本

We are also interested in real-world speed of the architectures on our hardware, which is not always well predicted by theoretical FLOPs due to details like lane widths and cache sizes. For this purpose, we perform timing of inference speed for the main models of interest, on a TPUv3 accelerator; the difference between inference and backprop speed is a constant model-independent factor.

我们还对这些架构在我们的硬件上的实际速度感兴趣，由于车道宽度和缓存大小等细节，理论上的 FLOPs 并不总是能很好地预测这一点。为此，我们在 TPUv3 加速器上对主要感兴趣模型的推理速度进行了计时；推理速度与反向传播速度之间的差异是一个与模型无关的常数因子。

Figure 11 (left) shows how many images one core can handle per second, across various input sizes. Every single point refers to the peak performance measured across a wide range of batch-sizes. As can be seen, the theoretical bi-quadratic scaling of ViT with image size only barely starts happening for the largest models at the largest resolutions.

图 11(左) 显示了一个核心每秒可以处理多少图像，涵盖了各种输入大小。每一个点都表示在广泛的批量大小范围内测得的峰值性能。可以看出，ViT 随着图像大小的理论二次扩展仅在最大分辨率下的最大模型中才刚刚开始出现。

Another quantity of interest is the largest batch-size each model can fit onto a core, larger being better for scaling to large datasets. Figure 11 (right) shows this quantity for the same set of models. This shows that large ViT models have a clear advantage in terms of memory-efficiency over ResNet models.

另一个关注的量是每个模型可以适配到核心上的最大批量大小，较大的批量大小在扩展到大数据集时更具优势。图 11(右) 显示了同一组模型的这一量。这表明，大型 ViT 模型在内存效率方面明显优于 ResNet 模型。

D.5 AXIAL ATTENTION

D.5 轴向注意力

Axial Attention (Huang et al. 2020, Ho et al. 2019) is a simple, yet effective technique to run self-attention on large inputs that are organized as multidimensional tensors. The general idea of axial attention is to perform multiple attention operations, each along a single axis of the input tensor, instead of applying

1-dimensional attention to the flattened version of the input. In axial attention, each attention mixes information along a particular axis, while keeping information along the other axes independent. Along this line, Wang et al. (2020b) proposed the AxialResNet model in which all the convolutions with kernel size 3×3 in a ResNet50 are replaced by axial self-attention, i.e. a row and column attention, augmented by relative positional encoding. We have implemented AxialResNet as a baseline model 3

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输入上运行自注意力。轴向注意力的总体思路是沿着输入张量的单个轴执行多个注意力操作，而不是对输入的扁平版本应用一维注意力。在轴向注意力中，每个注意力沿特定轴混合信息，同时保持其他轴上的信息独立。在这一方面，Wang et al.(2020b) 提出了 AxialResNet 模型，其中 ResNet50 中所有卷积核大小为 3×3 的卷积被轴向自注意力替代，即行和列注意力，并增强了相对位置编码。我们已将 AxialResNet 实现为基线模型 3。

Moreover, we have modified ViT to process inputs in the 2-dimensional shape, instead of a 1- dimensional sequence of patches, and incorporate Axial Transformer blocks, in which instead of

此外，我们已修改 ViT 以处理二维形状的输入，而不是一维的补丁序列，并结合了轴向变换器块，其中而不是

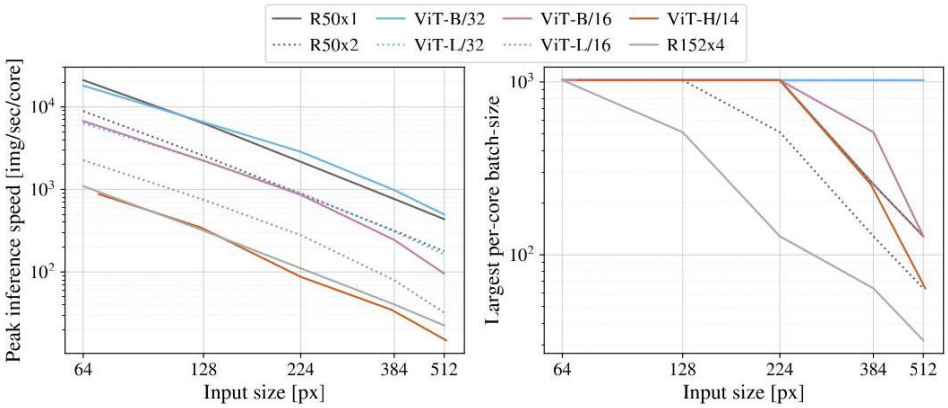


Figure 11: Left: Real wall-clock timings of various architectures across input sizes. ViT models have speed comparable to similar ResNets. Right: Largest per-core batch-size fitting on device with various architectures across input sizes. ViT models are clearly more memory-efficient.

图 11: 左: 不同架构在输入大小上的实际墙钟时间。ViT 模型的速度与类似的 ResNet 相当。右: 在各种架构和输入大小下，设备上最大每核心批量大小的适配。ViT 模型显然更具内存效率。

a self-attention followed by an MLP, we have a a row-self-attention plus an MLP followed by a column-self-attention plus an MLP.

一个自注意力层后接一个多层感知机，我们有一个行自注意力层加一个多层感知机，后接一个列自注意力层加一个多层感知机。

---

[3] Our implementation is based on the open-sourced PyTorch implementation in https://github.com/ csrhddlam/axial-deeplab. In our experiments, we reproduced the scores reported in (Wang et al. 2020b) in terms of accuracy, however, our implementation, similar to the open-source implementation, is very slow on TPUs. Therefore, we were not able to use it for extensive large-scale experiments. These may be unlocked by a carefully optimized implementation.

[3] 我们的实现基于开源的 PyTorch 实现，网址为 https://github.com/csrhddlam/axial-deeplab。在我们的实验中，我们在准确性方面重现了 (Wang et al. 2020b) 报告的分数，然而，我们的实现与开源实现类似，在 TPU 上非常缓慢。因此，我们无法将其用于广泛的大规模实验。这些可能通过经过精心优化的实现来解锁。

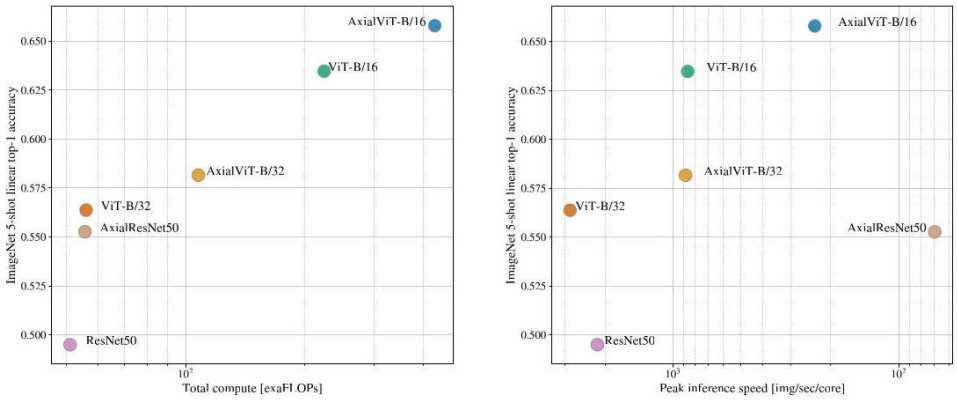


Figure 12: Performance of Axial-Attention based models, in terms of top-1 accuracy on ImageNet 5-shot linear, versus their speed in terms of number of FLOPs (left) and inference time (left).

图 12: 基于轴向注意力模型的性能，以 ImageNet 5-shot 线性任务的 top-1 准确率为标准，与其在 FLOPs 数量 (左) 和推理时间 (左) 方面的速度进行比较。

Figure 12, present the performance of Axial ResNet, Axial-ViT-B/32 and Axial-ViT-B/16 on ImageNet 5 shot linear, when pretrained on JFT dataset, verses the pretraining compute, both in terms of number of FLOPs and inference time (example per seconds). As we can see, both Axial-ViT-B/32 and Axial-ViT-B/16 do better than their ViT-B counterpart in terms of performance, but it comes at the cost of more compute. This is because in Axial-ViT models, each Transformer block with global self-attention is replaced by two Axial Transformer blocks, one with row and one with column self-attention and although the sequence length that self-attention operates on is smaller in axial case, there is a extra MLP per Axial-ViT block. For the AxialResNet, although it looks reasonable in terms of accuracy/compute trade-off (Figure 12, left), the naive implementation is extremely slow on TPUs (Figure 12, right).

图 12，展示了在 JFT 数据集上预训练的轴向 ResNet、Axial-ViT-B/32 和 Axial-ViT-B/16 在 ImageNet 5-shot 线性任务上的性能，与预训练计算量进行对比，包括 FLOPs 数量和推理时间 (每秒示例)。可以看出，Axial-ViT-B/32 和 Axial-ViT-B/16 在性能上优于其 ViT-B 对应模型，但这需要更多的计算资源。这是因为在轴向 ViT 模型中，每个具有全局自注意力的 Transformer 块被两个轴向 Transformer 块替代，一个具有行自注意力，另一个具有列自注意力，尽管在轴向情况下，自注意力操作的序列长度较小，但每个 Axial-ViT 块都有一个额外的多层感知机。对于 AxialResNet，尽管在准确性与计算资源的权衡上看起来合理 (图 12，左)，但其简单实现在 TPU 上极其缓慢 (图 12，右)。

D.6 Attention Distance

D.6 注意力距离

To understand how ViT uses self-attention to integrate information across the image, we analyzed the average distance spanned by attention weights at different layers (Figure 10). This ”attention distance” is analogous to receptive field size in CNNs. Average attention distance is highly variable across heads in lower layers, with some heads attending to much of the image, while others attend to small regions at or near the query location. As depth increases, attention distance increases for all heads. In the second half of the network, most heads attend widely across tokens.

为了理解 ViT 如何利用自注意力在图像中整合信息，我们分析了不同层次上注意力权重所跨越的平均距离 (图 10)。这种 “注意力距离” 类似于 CNN 中的感受野大小。较低层次上，平均注意力距离在各个头之间高度可变，有些头关注于图像的大部分区域，而其他头则关注于查询位置附近的小区域。随着深度的增加，所有头的注意力距离也随之增加。在网络的后半部分，大多数头在标记之间广泛关注。

D.7 ATTENTION MAPS

D.7 注意力图

To compute maps of the attention from the output token to the input space (Figures 6 and 13), we used Attention Rollout (Abnar & Zuidema, 2020). Briefly, we averaged attention weights of ViT-L/16 across

all heads and then recursively multiplied the weight matrices of all layers. This accounts for the mixing of attention across tokens through all layers.

为了计算从输出标记到输入空间的注意力图 (图 6 和 13)，我们使用了注意力展开 (Abnar & Zuidema, 2020)。简而言之，我们对 ViT-L/16 的所有头的注意力权重进行了平均，然后递归地乘以所有层的权重矩阵。这考虑了通过所有层在标记之间的注意力混合。

D.8 VTAB BREAKDOWN

D.8 VTAB 分析

Table 9 shows the scores attained on each of the VTAB-1k tasks.

表 9 显示了在每个 VTAB-1k 任务中获得的分数。

Table 9: Breakdown of VTAB-1k performance across tasks.

表 9:VTAB-1k 各任务性能的分析。

	Suppose	2023-09-30	2023-03	2023-09-30	1-millo-mass	2023-09-30	2023-03-30	Jul-2018	2023-03-30	1-dimensions	2023-09-30	Supervisor	1-23-23	2023-09-30	1-2023-03-30	2023-09-30	Sep-23	2023-09-03	2023-03-30	1-30-30
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

	假设	2023-09-30	2023-03	2023-09-30	1-毫克质量	2023-09-30	2023-03-30	2018 年 7 月	2023-03-30	1 维	2023-09-30	监督者	1-23-23	2023-09-30	1-2023-03-30	2023-09-30	2023 年 9 月	2023-09-03	2023-03-30	1-30-30
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

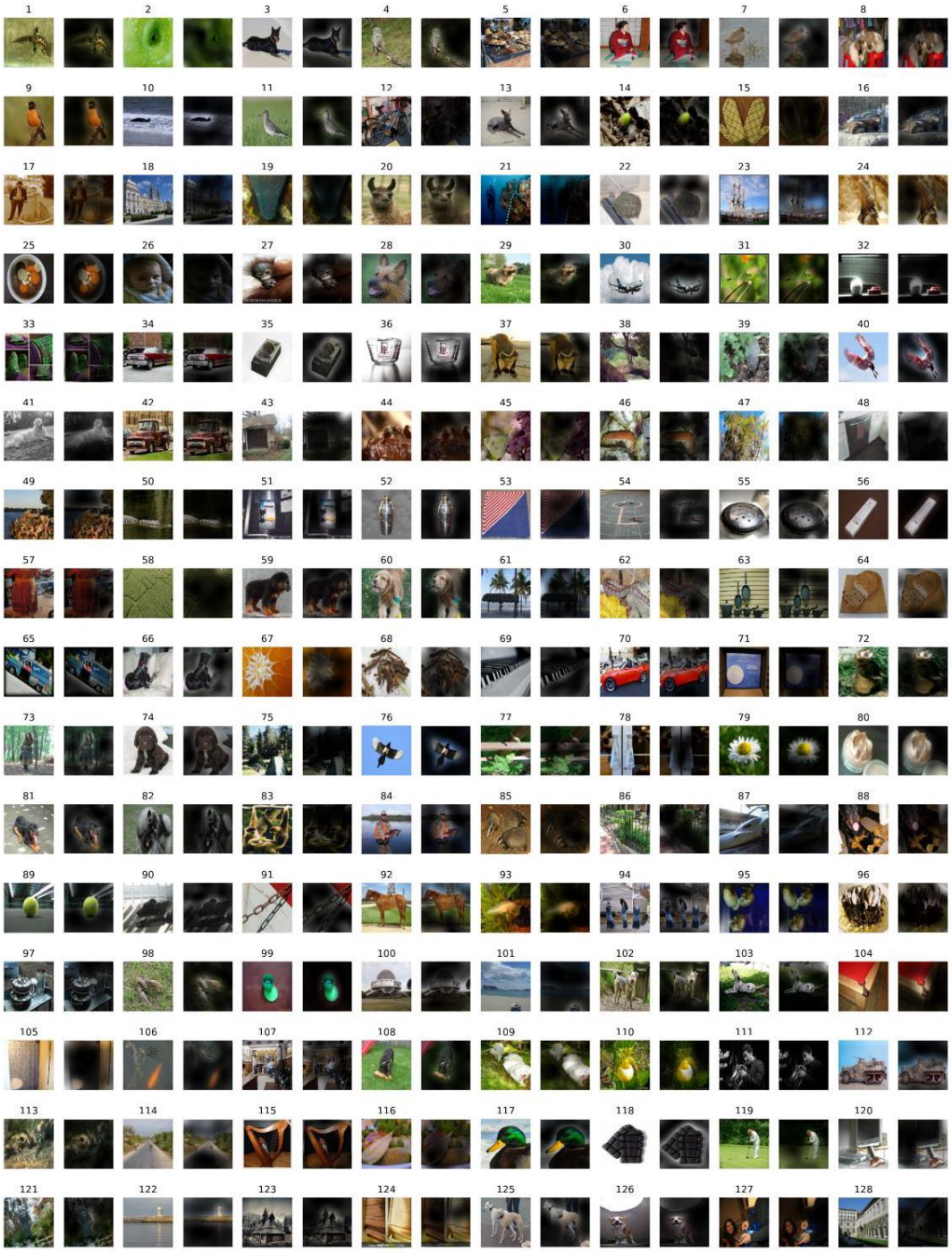


Figure 13: Further example attention maps as in Figure 6 (random selection).
图 13: 进一步的示例注意力图，如图 6 所示 (随机选择)。