Vision Transformer 技术与复现

摘要

基于卷积神经网络的深度学习技术已经在图像处理领域取得成功，而 Transformer 的出现为计算机视觉领域带来了新的可能。Vision Transformer（ViT）在最大程度保留 Transformer 架构的同时实现了自然语言处理到图像处理任务的迁移，通过将图像切片成小块将图形数据当作 token 进行处理。相较于卷积核的局部特征提取，Transformer 架构可以提供长距离依赖建模能力，提取图像中较远距离的注意力信息。本文对 ViT 进行了复现，复现实验及原论文中的实验证明，ViT 在大规模数据集预训练后迁移到下游任务效果媲美甚至优于卷积神经网络，但在数据量较少的情况下其表现不如卷积神经网络模型。

关键词：Transformer；ViT；注意力机制

1 引言

近年来，深度学习在计算机视觉领域取得了革命性的进展，尤其是在图像分类、语义分割和目标检测等任务中。传统的卷积神经网络（CNN）由于其优秀的局部特征提取能力，已经成为视觉任务的主流方法。然而，CNN 的局限性也逐渐显现，特别是在长距离依赖和全局特征建模方面。卷积操作的感受野虽然随着网络的加深逐渐增大，但在建模全局关系时仍存在一定的局限性，尤其是在复杂背景和多物体场景下，CNN 往往需要非常深的网络结构来捕捉长距离依赖，这会带来较大的计算开销和训练成本。

Transformer 的自注意力机制具有出色的全局特征建模能力，可以直接捕捉输入序列中各个位置之间的关系。随着 Transformer 在自然语言处理（NLP）领域的成功，研究者们开始尝试将其引入计算机视觉任务，Vision Transformer（ViT）[1] 应运而生。ViT 将图像切分为固定大小的块，并将每个块线性嵌入到高维空间中，然后通过 Transformer 编码器处理这些嵌入向量。它的提出打破了卷积操作的局限，并展示了 Transformer 在视觉任务中的巨大潜力，尤其在大规模数据集上，ViT 的表现超越了传统 CNN 模型。

本文旨在深入探讨 Vision Transformer 在计算机视觉中的应用，尤其是通过复现 ViT 模型，分析其在不同视觉任务中的表现。与传统的 CNN 模型相比，ViT 利用 Transformer 的全局依赖建模能力，不仅在图像分类中取得了优异成绩，也为其他计算机视觉任务提供了新的思路。通过阅读与实验，可以进一步理解 ViT 在视觉领域中的优势和局限，为 Transformer 在计算机视觉领域的应用提供理论支持。

2 相关工作

图像特征的提取原本依赖于人工设计的局部特征描述子，而卷积神经网络的出现大幅提高了图像处理性能，掀起了深度学习的研究热潮。多年以来基于卷积神经网络的算法蓬勃发展，而 Transformer 架构的出现则是为视觉领域注入了新的活力。

2.1 卷积神经网络

自上世纪九十年代以来，卷积神经网络（CNN）成为计算机视觉领域的核心技术之一。CNN 的优势在于其能够自动学习从原始图像中提取的特征，并通过多个卷积层逐步构建更高层次的抽象特征。这种基于局部感知的结构，能够有效捕捉图像中的局部模式，在图像分类、目标检测、语义分割等任务中取得了广泛应用。

在目标检测领域，经典的两阶段模型如 R-CNN [3] 和其改进版本 Fast R-CNN [2] 和 Faster R-CNN [6]，通过引入候选区域和卷积特征提取网络，极大地提高了目标检测的效率和准确性。随后，YOLO [5] 和 SSD [4] 等单阶段检测模型，通过全卷积网络的设计，实现了高速和高效的目标检测。这些模型通过卷积操作提取图像特征，并通过后续的分类和回归网络进行目标定位和识别。

然而，CNN 在长距离依赖建模和全局特征捕捉方面存在一定的局限性。在面对复杂的背景、尺度变化、以及物体间的关系时，CNN 可能无法有效地处理图像中远距离像素之间的依赖。为了克服这些局限，研究者开始寻求基于 Transformer 的解决方案。

2.2 基于 Transformer 的神经网络

Transformer [7] 模型最初被提出用于自然语言处理（NLP）任务，在诸如机器翻译、文本生成、文本分类等任务中取得了显著成果。Transformer 模型的核心在于自注意力机制（Self-Attention），它能够直接计算输入序列中每个位置与其他位置之间的关系，从而捕捉长距离依赖信息。这一机制与传统的循环神经网络（RNN）和卷积神经网络不同，Transformer 不再依赖局部感知，能够更有效地建模全局特征。

在计算机视觉领域，Transformer 的引入也逐渐得到了关注。最初，研究者尝试将 Transformer 与卷积神经网络结合，构建混合模型，如 Convolutional Vision Transformer（CvT）[8]。这些模型在保持卷积神经网络高效的局部特征提取能力的同时，利用 Transformer 进行全局依赖建模，取得了更好的性能。

随着对 Transformer 理解的深入，研究者逐渐提出了将 Transformer 完全替代卷积层的纯视觉 Transformer 模型。ViT（Vision Transformer）便是这一方向的代表，它将图像直接划分为一系列固定大小的块（patch），然后将这些块线性嵌入并视作一个序列输入到 Transformer 中进行处理。通过这种方法，ViT 模型能够充分利用 Transformer 在长距离依赖建模上的优势，直接从图像中学习全局特征。由于 Transformer 的自注意力机制能够捕捉图像中远距离的依赖关系，ViT 在处理复杂图像结构时，尤其是在多物体、多背景的场景下，表现得尤为出色。

3 本文方法

3.1 本文方法概述

ViT 将 Transformer 架构引入计算机视觉领域，在 Transformer 基础上做了尽可能少的修改，其主要结构如图 1所示。ViT 将图像处理分为三个部分：Embedding、Transformer Encoder 和 MLP Head。Embedding 部分将图像切分成一个个 patch，将它们作为 token 进行处理；随后在每个 token 上拼接 class embedding 和位置编码，作为编码器的输入。Transformer 部分使用了多头注意力机制捕获图像块之间的全局依赖关系，堆叠多个编码器用于提取全局信息。最后，MLP Head 部分提取出用于分类的特征向量，输入到分类器中进行图像分类。

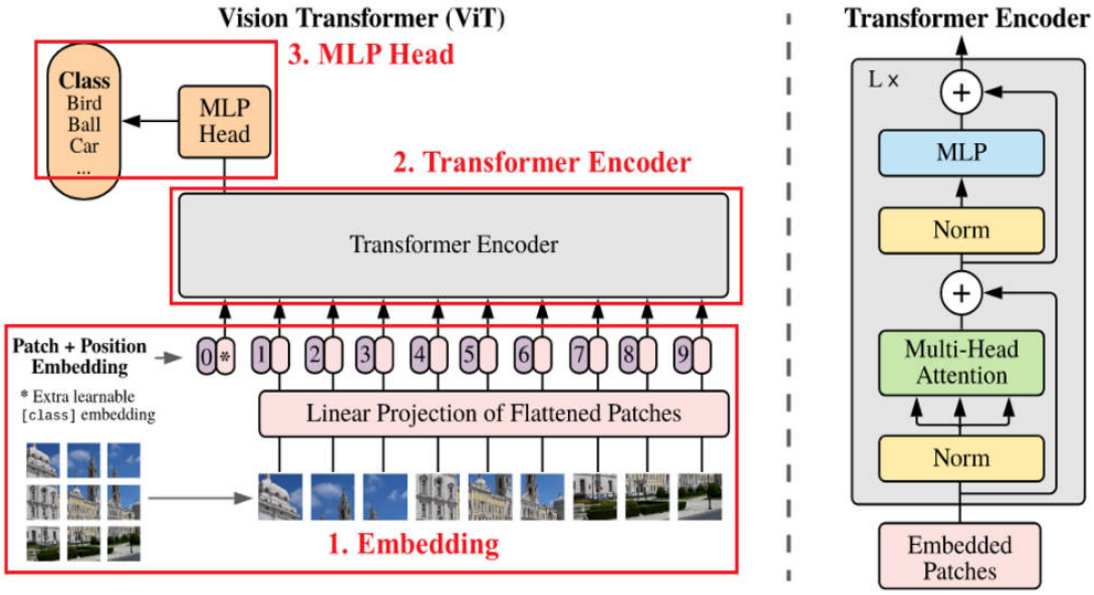

图 1. ViT 结构图

3.2 Embedding

对于应用在自然语言处理（NLP）领域的 Transformer，其输入是 token 向量的序列，所以需要将原始图像转换为可供输入的 token 以提取特征。由于图像中的像素数量可达数百万，以像素为基本单位提取注意力会带来庞大的计算量，故 ViT 选择将图像切分成一个个固定大小的 patch 进行处理。

图 2是 Embedding 步骤的流程图。ViT 使用大小为 16x16，步距为 16 的卷积核用于裁剪原始图片（224x224x3），得到 14x14x768 大小的特征层，展平后得到 196x768 大小的 token 矩阵，其中每一行代表一个 patch 的特征。随后，增加一行 class token 与生成的 token 矩阵拼接，专门用于图像的分类。最后，由于 Transformer 本身不具备位置感知能力，故需要添加位置编码（Position Embedding），将位置编码与 token 矩阵进行叠加操作后可以得到编码器的输入矩阵。

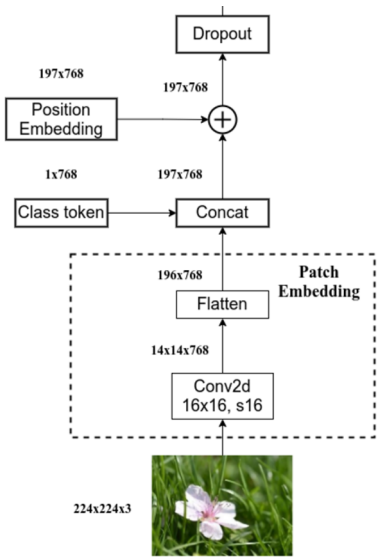

图 2. Embedding 流程图

3.3 Transformer Encoder

ViT 的骨干部分仅采用了 Transformer 中的编码器组（Encoder），并对每个编码器（encoder) 的结构进行了略微调整。每个编码器的结构如图 3所示。

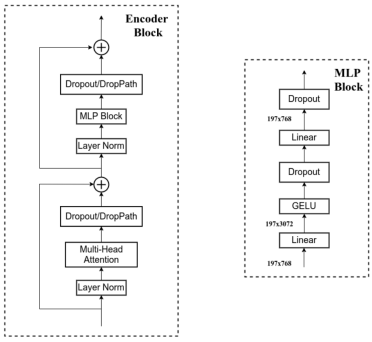

图 3. encoder 结构图

首先，Encoder Block 由两个相似的残差结构组成，除了中间模块不同以外其他部分相同。假设中间处理为 F、输出为 Y，来自 Embedding 步骤的输入 X 首先经过 Layer Normalization，经过 F 后再进行 Dropout 处理，最终与原始输入进行残差相加。每个残差结构可以用公式表示为：

$$Y = X + Dropout(F(LN(X))) \tag{1}$$

多头注意力机制用于提取 token 矩阵中每个 patch 相对于其他 patches 的注意力，获取图像块之间的依赖关系。MLP Block 由两个全连接层和 GELU 激活函数组成，使用 Dropout 防止过拟合。第一个全连接层会将输入节点的个数翻 4 倍用于高维筛选，随后还原为原节点数目。

3.4 MLP Head

MLP Head 部分由可选的 Pre-Logits 和用于分类的全连接层组成，其主要结构如图 4所示。

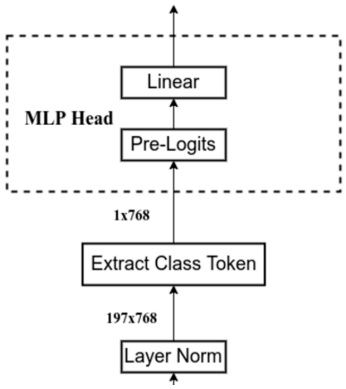

图 4. MLP Head 结构图

对于 Transformer Encoder 输出的结果，经过 Layer Norm 后取出 class token 对应的分类信息，随后送入 MLP Head。当把模型迁移到一般的数据集时，只需要一个全连接层；当把模型迁移到较大的数据集，如 ImageNet21K 时，需要额外添加 Pre-Logits，Pre-Logits 往往由一个全连接层与 tanh 激活函数串联而成。

4 复现细节

4.1 与已有开源代码对比

本文使用的复现代码源自 https://github.com/WZMIAOMIAO/deep-learning-for-image-processing/tree/master/pytorch_classification/vision_transformer，并没有进行太大改动，按照其使用方法在花分类数据集上测试了 ViT 在有预训练与没有预训练情况下迁移到下游子任务上的性能，并在网络上下载图片进行模型的效果测试。

4.2 实验环境搭建

本次实验的目的在于复现 ViT-Base/16 模型，使用的 IDE 为 pytorch。下载代码后按照 environment.txt 配置 conda 环境，并下载花分类数据集和 ViT-Base/16 对应的预训练权重。数据集的下载地址参考 README.md，模型权重的地址可以查找 vit_model.py。

随后，依照 README 中的步骤修改 train.py 中的数据集路径和预训练权重路径，如图 5所示。运行 train.py 脚本开始训练，可以指定学习率、epoch 以及是否冻结权重。

模型训练完毕后，可以使用 predict.py 脚本预测图片。下载数据集中的某一类别图片后，修改 predict.py 脚本的图片路径和权重，即可进行模型效果的验证，程序将同时输出各个类别的预测概率。

```
# 数据集所在根目录
# https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz
parser.add_argument( *name_or_flags: '--data-path', type=str,
                    default="./flower_photos")    数据集根目录
parser.add_argument( *name_or_flags: '--model-name', default='', help='create model name')

# 预训练权重路径，如果不想载入就设置为空字符                                    预训练权重文件
parser.add_argument( *name_or_flags: '--weights', type=str, default='./vit_base_patch16_224_in21k.pth',
                    help='initial weights path')
# 是否冻结权重
parser.add_argument( *name_or_flags: '--freeze-layers', type=bool, default=True)    是否冻结权重
parser.add_argument( *name_or_flags: '--device', default='cuda:0', help='device id (i.e. 0 or 0,1 or cpu)')
```

图 5. train.py 中修改数据集与权重路径

5 实验结果分析

5.1 复现结果展示

复现一共进行了三次实验，旨在探究 ViT 模型的特性及将其迁移到下游任务的表现。

第一次实验使用了 ImageNet21K 的预训练权重，冻结主体权重仅训练 MLP Head，在 train.py 脚本上训练 10 个 epoch 后在测试集上的准确率为 98.1%。

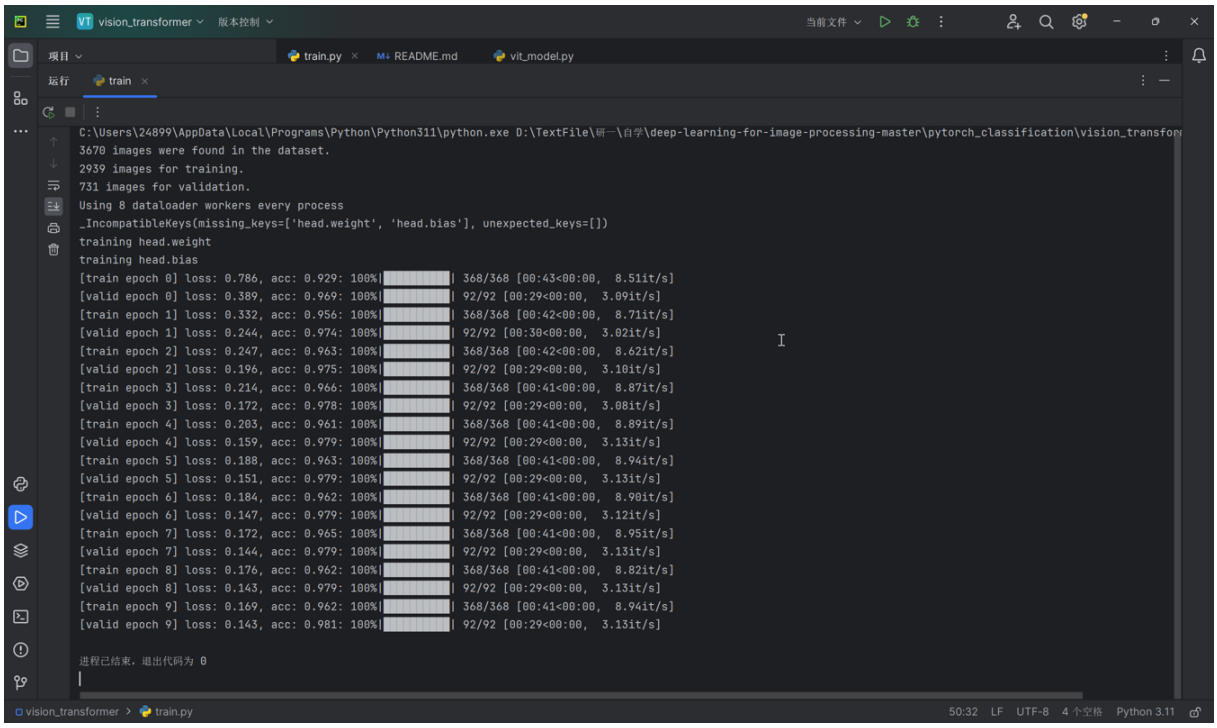

图 6. 冻结权重训练 10 个 epoch

随后从网络上下载一张向日葵的图片，使用 predict.py 脚本测试迭代 10 个 epoch 后的模型的运行效果。模型成功输出各个类别的预测概率，并将预测结果结合图像输出。

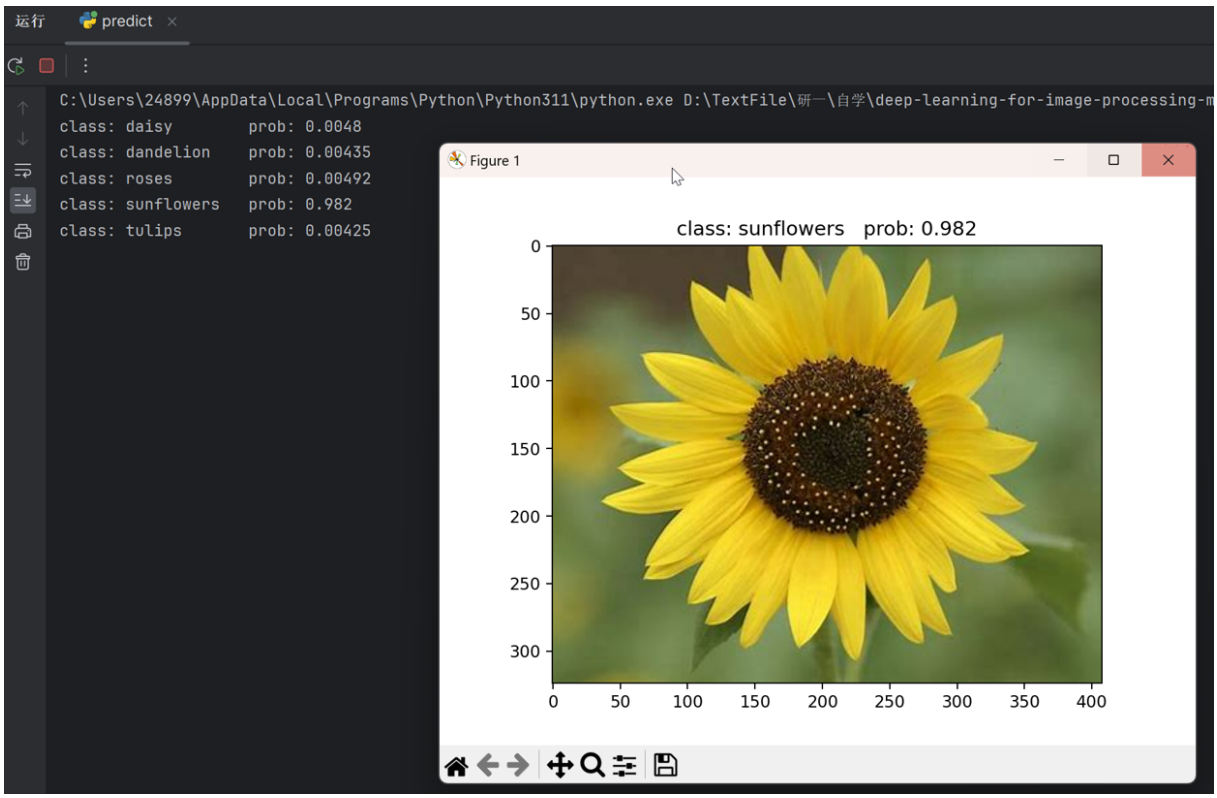

图 7. 模型预测图像类别的效果

第二次实验使用了 ImageNet21K 的预训练权重，不冻结权重直接训练模型，训练 10 个 epoch 之后模型在测试集上的分类准确率为 97.8%。由于第一次实验是在笔记本（GEFORCE RTX 4060）上运行，但后续实验是在服务器端（GEFORCE RTX 4090）上运行的，所以界面 UI 会有所不同。

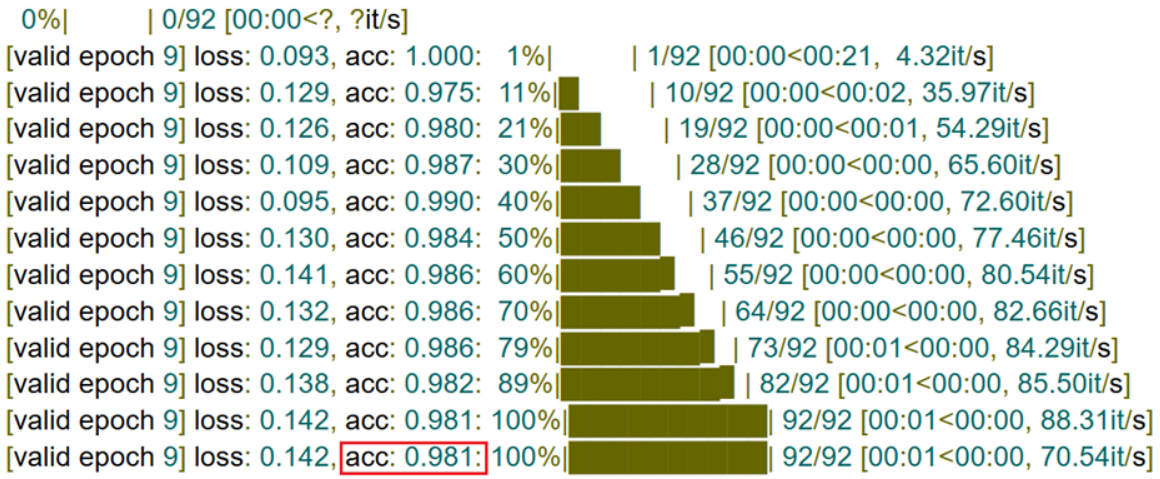

图 8. 不冻结权重训练 10 个 epoch

第三次实验不使用预训练权重从头开始训练，在训练集上训练了 48 个 epoch，测试集上分类准确率为 72.6%，存在明显下降。这与训练 24 个 epoch 后的结果相差并不大。

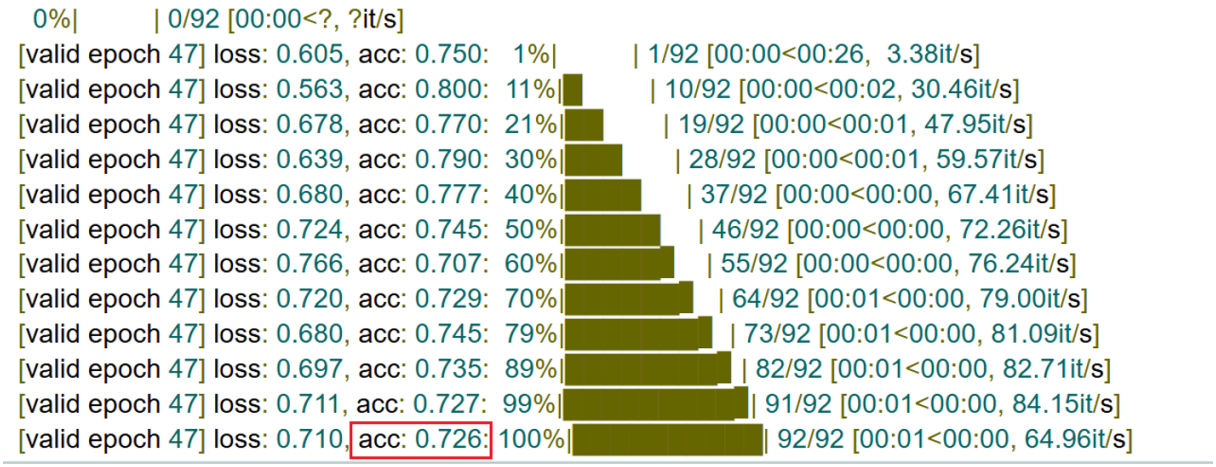

图 9. 从零开始训练 48 个 epoch

5.2 结果分析

原论文中的实验结果如图 10所示，横轴代表不同的预训练数据集（括号中为训练模型），纵轴为迁移到的下游任务数据集。可以看出，ViT 模型的表现与耗时都比当时最先进的 BiT（以 ResNet 作为 Backbone）更加优秀，在 JFT-300M 等大型数据集上进行预训练后迁移到其他数据集上，ViT 的表现不逊色于卷积神经网络算法。

	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

图 10. 原论文中 ViT 使用不同数据集预训练后在不同下游任务上的表现

图 11为预训练数据集样本数与 Top 1 查准率的关系。可以看出，在训练数据规模小于 30M 时，ViT 的表现不如 ResNet。由此可见 ViT 对大型数据集和预训练的依赖。

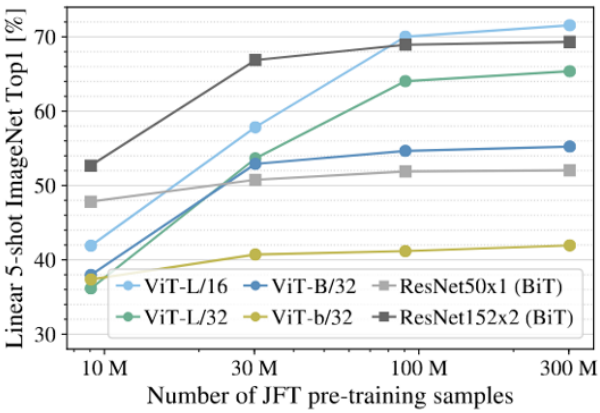

图 11. ViT 和 ResNet 模型使用不同预训练样本规模时的表现

对于复现的结果：使用预训练权重与从零开始训练相差 25.2%，说明 ViT 对大数据集上的预训练较为依赖。虽然 ViT 在迁移学习后性能媲美甚至优于同时期的卷积神经网络算法，但数据集较小及迭代次数较少时性能存在一定差距。

6 总结与展望

本文首先介绍了图像处理领域的发展，以 Transformer 为主体的 ViT 相较卷积神经网络有着长距离依赖建模的优势，能够捕捉图像中远距离的依赖关系。随后从三个方面对 ViT 进行介绍，Embedding 部分将原始图像切分成小的 patch，加入位置编码后形成 token 矩阵；Transformer Encoder 采用多头注意力机制和残差结构提取注意力信息；MLP Head 则是用于图像分类。接着进行了论文的复现，搭建环境后在花分类数据集上测试了 ViT-Base/16 模型的训练效果。最后结合原文结果分析，得出结论：ViT 在大规模数据集上进行预训练后应用于下游任务，实现的分类效果略优于卷积神经网络；而它在小数据集上的训练效果不如 CNN，比较依赖于数据量。

未来 ViT 可以在提高计算效率、改进位置编码以及提高模型在低资源环境上的表现等方面优化，可以将 ViT 与图神经网络等其他架构进行结合，或用作其他后续任务的框架。ViT 在计算机视觉方面的应用前景仍然广阔，在大规模图像分类、多模态学习等方面依旧有着巨大的发展空间。

参考文献

[1] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[2] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[5] J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[7] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[8] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021.