# Title：

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

# Abstract：

虽然现在Transformer是NLP领域的标准，但是在CV领域用的还是很少的。现有CV领域还是保持整体结构不变，只是用自注意力机制取代其中的一个block而已。本文证明了这种对CNN的依赖是完全不必要的。在大规模的训练集上做训练，再迁移到中小型的数据集上，ViT能够获得和最好的CNN相媲美的结果。

# Intro：

Transformer已经是自然语言中的必选模型了，现在的大规模的趋势是在大的数据集上做预训练，然后再在小的数据集上做微调（BERT）。随着模型和数据集的增长，现在还没有任何饱和的现象。

在CV，主流还是CNN，但是自注意力如何用在CV内呢？一些工作是结合CNN和attention（用resnet输出的特征图当作序列长度），另一些工作是全用自注意力（孤立注意力：局部小窗口；轴注意力：在H和W分别作自注意力）。都是减少序列长度的操作，因为CV输入的像素点太多了，相对于BERT的512长度的序列来说。之前的工作虽然理论非常高效，但是没有硬件加速，所以无法训练千亿参数级的大Transformer模型，所以目前做的最好的还是基于ResNet的模型。

ViT是怎么做的呢？它把图片分为一个个patch，每个patch是16\*16的“单词”，所以输入变成了224/16 = 14，14\*14=196长的序列了，可以接受。它的训练是有监督的。它的encoder和BERT完全一样。

在ImageNet上，如果不加很好的约束，ViT比ResNet要弱几个点，因为Transformer比传统的CNN少一些归纳偏置（一种先验知识或提前做好的假设：平移不变性和局部性），这种先验知识让CNN可以在相对少的数据集就做到很好的效果。所以作者在更大的数据集上做了预训练（14M-300M image）。果然效果很好，能取得和ResNet相近或更好的结果。

# Conclusion：

本文直接拿NLP领域的标准Transformer来做计算机视觉问题。跟前人工作不一样的是在位置编码这里用了图像特有的归纳偏置。相当于一系列的图像块作为单词的句子进入Transformer。这种ViT在进行大规模预训练后工作的出奇的好，在很多benchmark上超过了现有最好的方法，而且相对便宜。

# Related Work

Transformer在NLP领域是大规模预训练后再微调，现已有BERT和GPT。

直接在像素层面的Transformer不现实，如果想用，只能做一些近似。算法复杂度高是因为用了整张图作输入，相关的一些工作是用各种方法减少序列长度（详细看论文），还有相当多的工作是关注CNN和Transformer结合和多模态的工作。

# Method

模型的设计上尽可能按照最原始的transformer来做，这样的好处是能把NLP成功的架构和其非常高效的实现都能拿过来用。

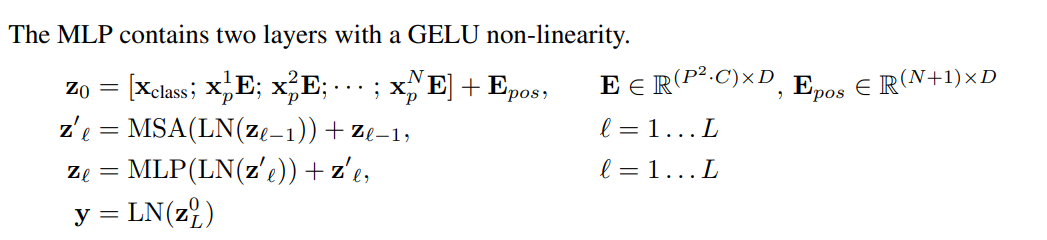
## Vision Transformer（ViT）

模型总览图

前向操作：

224\*224\*3的图片进入后，分为16\*16\*3=768的patch，加入位置编码和Extra learnable {class}embedding（cls token 来自BERT）后得到输出。全连接层的维度是768\*768（前面的768是不变的，后面是可变的（隐藏层）），所以整体长度就是197\*768进入encoder。通过LN后，因为有多头注意力，所以Q、K、V的维度都是197\*768/12=197\*64，进入多头注意力后，再把输出拼接起来，成为197\*768。后面再进入MLP，一般是放大4倍，所以成了197\*3012，最后再投射回去，成为197\*768输出。

整体过程：



## Inductive bias

ViT比CNN要少很多图像特有的归纳偏置，比如局部性和平移不变性。对于ViT来说，只有mlp这一层是有这两个性质的。注意力是全局的，2D的信息没有怎么使用。

## Hybrid Architecture

用CNN输出的特征图作为输入：14\*14=196

## class token/position encoder

因为在本文中，作者想和原始的Transformer保持尽可能一致，所以运用了原始的class token，作为全局的对句子/图片理解的特征。后面接一个MLP，用tanh当激活函数。

但是在CV其实不是这么做的，在ResNet中，14\*14的特征图输出时做了一个全局的Global Average Pooling（GAP），把特征图拉直成一个向量，可以把这个向量理解成全局的对于图片的特征。为什么在ViT中不用呢？作者做了实验，发现两者都可以，但是本文为了和Transformer保持一致，所以用了class token。

1D的位置编码和2D的位置编码（将长度开根号，把patch表示成矩阵的形式（11，12，13…33）,用两个D/2的向量分别表示横坐标和纵坐标，最后把他们concat起来）作者也试过，都可以，为了和Transformer保持一致，所以用了1D的位置编码。

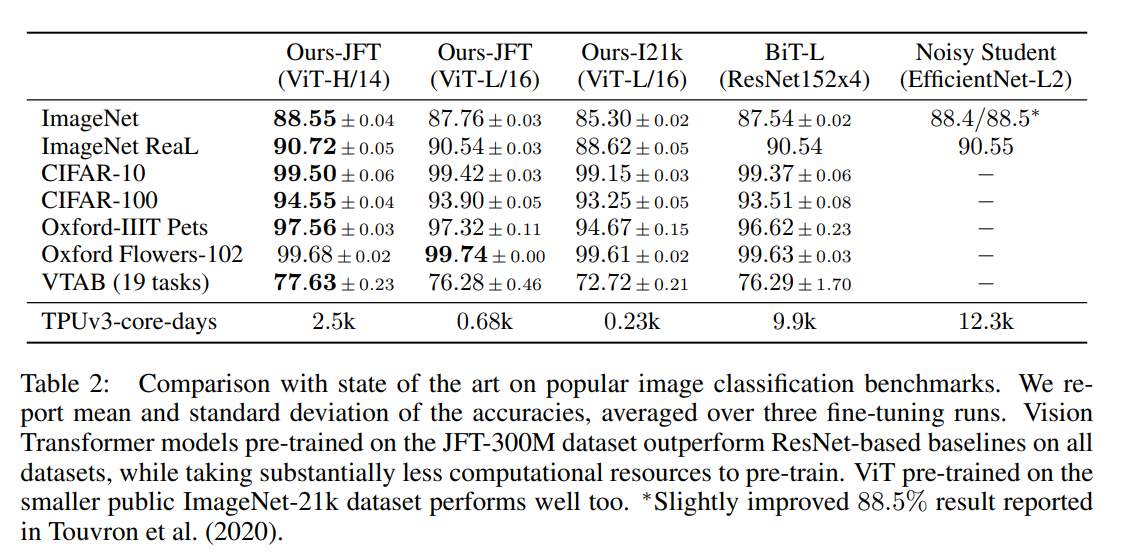
## Fine-tune and higher resolution

如果在微调的时候能够用更大的图像尺寸，如256\*256/320\*320，那就会得到更好的结果。但是ViT不太好改输入分辨率和位置编码。

# Experiment

本章主要对比残差网络和ViT的表征学习能力，他们在不同大小的数据集上做预训练，然后再很多的数据集上做测试。最后作者做了子监督训练的实验，表现还可以，比较有潜力。

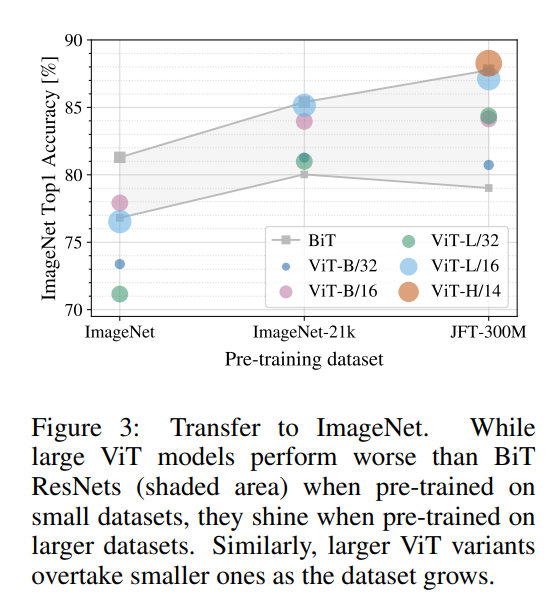
## COMPARISON TO STATE OF THE ART

下表是已经在大规模数据集上做预训练了，然后在下面数据集上做微调的表现：

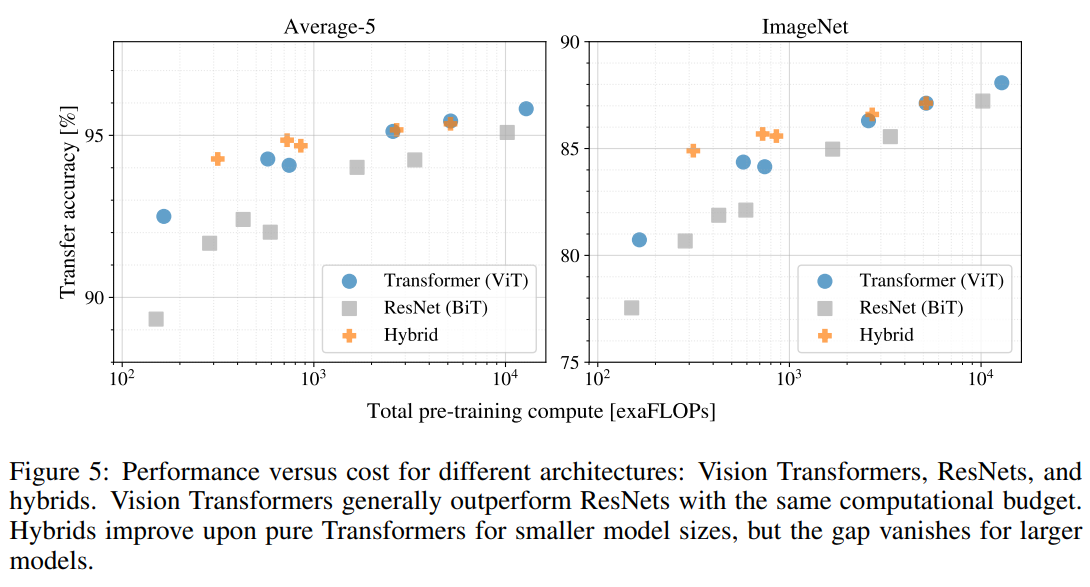
ViT训练起来更便宜

## PRE-TRAINING DATA REQUIREMENTS

ViT到底需要多少数据才能训练的比较好？



但数据集不断增大的时候，ViT表现越来越好



在同等的计算复杂度下，Transformer比ResNet要好