A ConvNet for the 2020s:2022

code:https://github.com/facebookresearch/ConvNeXt

摘要：自从transformer模型进入计算机视觉领域之后，在多种任务上取得SOTA，然而纯transformer在解决这些问题的时候依然有些缺陷，因此人们开始引入分层的transformer，将transformer发展成了通用的视觉主干。然而这种方式很大程度上归功于transformer结构的内在优越性，而不是卷积的归纳偏置。本文的工作是通过设计，“现代化”标准ResNet的设计，测试一个纯卷积神经网络可以达到的极限。

论文出发点和背景

卷积神经网络是过去十年的技术主导，由于其固有的归纳偏置等特性，使得其在各大视觉任务上面表现良好。同时自然语言处理的transformer击败了RNN，称为NLP任务的标配。2020年，transformer开始进入计算机视觉领域，并且通过VIT模型击败了当时的卷积模型，在各大数据集上获得了SOTA。但是如果纯transformer没有CNN的归纳偏置等特性的话，在很多方面也会遇到问题，最大的还是ViT的全局注意力设计，其复杂度与输入大小成二次方关系。
Swin transformer证明了卷积的本质在视觉transformer设计不仅不是没用的，反而是急需的

论文创新思路

1）宏观设计

Multi stage

swin transformer 的设计遵循多阶段设计，每个阶段都有不同的分辨率，在设计中有两个有意思的设计考虑：a. the stage compute ratio ,b. the stem cell 结构

我们将每个阶段的块的数量从ResNet-50中的（3、4、6、3）调整到（3、3、9、3）。这将模型精度从78.8%提高到79.4%。

Changing stem to “Patchify”

stem cell 的模块设计主要关注在网络开始时如何处理输入的图像

我们将ResNet-style的stem cell 替换为一个4×4，步幅为4放入卷积层，准确率从79.4%提升为79.5%。

2）ResNext

ResNext：使用更多的组，扩大宽度

我们：使用深度可分离卷积，深度可分离卷积是分组卷积的一种特殊情况，组的数量等于通道的数量，类似于自注意力机制中的加权和操作

3）倒置瓶颈

transformer：MLP block 的隐藏维度比输入维度宽四倍

4）大尺寸卷积核

transformer：非局部注意力机制

我们：使用大型卷积核7×7

5）微观设计

ReLU替换为GELU：GELU是relu的平滑变体

更少的激活函数：在每个块中只使用一个GELU

更少的归一化层：删除两个BN层，替换为只在Conv1×1之前只留下一个BN,之后准确率达到了81.4%

用LN替换为BN

单独的下采样层：2×2的卷积，步长为2代替3×3卷积部分的下采样

论文方法介绍

AdamW优化器、数据增强、扩展训练epochs，正则化方式：最终发现ResNet-50的性能上涨2.7%，证明传统卷积神经网络和vision transformer之间的性能差异很大一部分都是训练方式造成的

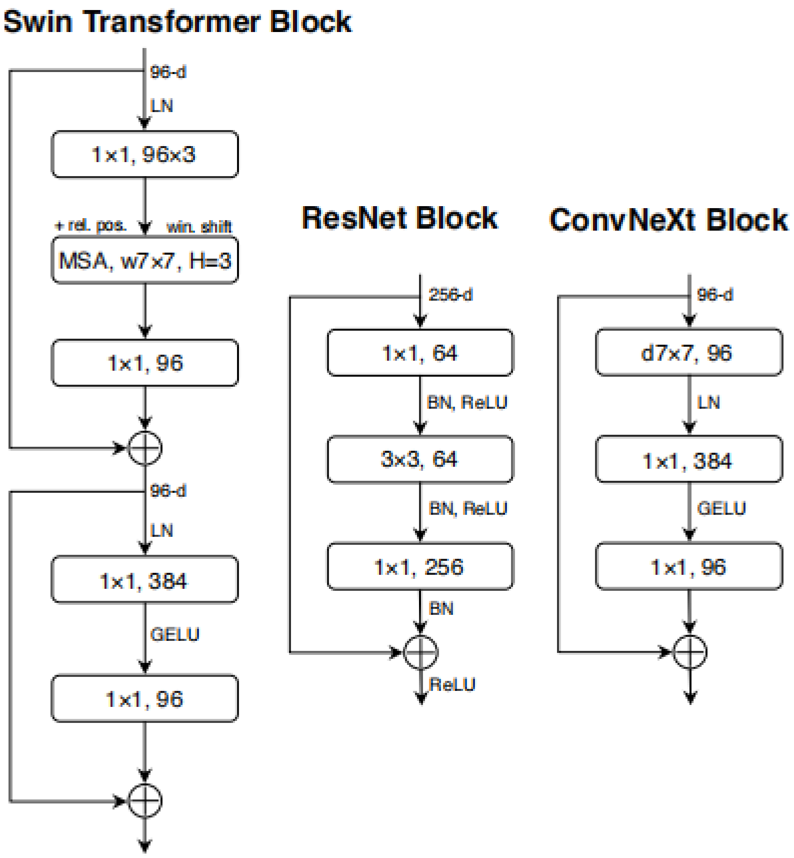


Figure 4. **Block designs** for a ResNet, a Swin Transformer, and a ConvNeXt. Swin Transformer's block is more sophisticated due to the presence of multiple specialized modules and two residual connections. For simplicity, we note the linear layers in Transformer MLP blocks also as "1×1 convs" since they are equivalent.

实际效果

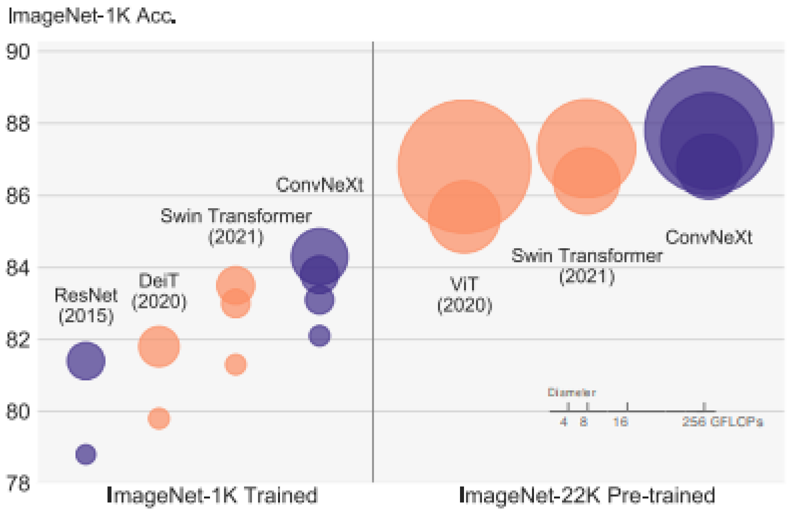


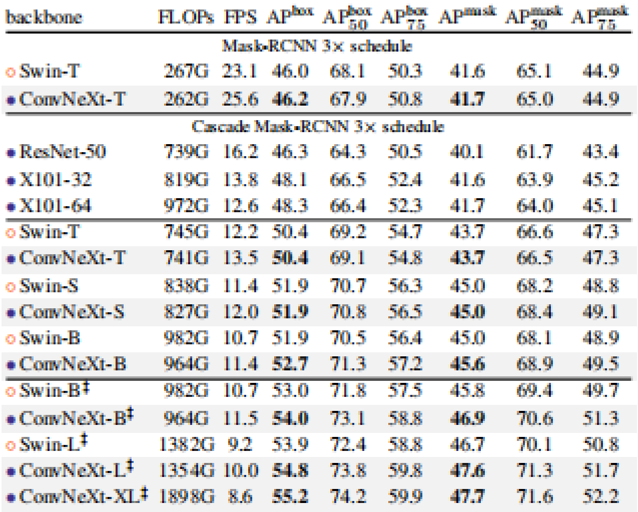
Figure 1. **ImageNet-1K classification** results for ● ConvNets and ○ vision Transformers. Each bubble's area is proportional to FLOPs of a variant in a model family. ImageNet-1K/22K models here take 224^2/384^2 images respectively. ResNet and ViT results were obtained with improved training procedures over the original papers. We demonstrate that a standard ConvNet model can achieve the same level of scalability as hierarchical vision Transformers while being much simpler in design.

model	image size	#param.	FLOPs	throughput (image / s)	IN-1K top-1 acc.
ImageNet-1K trained models					
● RegNetY-16G [54]	224^2	84M	16.0G	334.7	82.9
● EffNet-B7 [71]	600^2	66M	37.0G	55.1	84.3
● EffNetV2-L [72]	480^2	120M	53.0G	83.7	85.7
○ DeiT-S [73]	224^2	22M	4.6G	978.5	79.8
○ DeiT-B [73]	224^2	87M	17.6G	302.1	81.8
○ Swin-T	224^2	28M	4.5G	757.9	81.3
● ConvNeXt-T	224^2	29M	4.5G	774.7	**82.1**
○ Swin-S	224^2	50M	8.7G	436.7	83.0
● ConvNeXt-S	224^2	50M	8.7G	447.1	**83.1**
○ Swin-B	224^2	88M	15.4G	286.6	83.5
● ConvNeXt-B	224^2	89M	15.4G	292.1	**83.8**
○ Swin-B	384^2	88M	47.1G	85.1	84.5
● ConvNeXt-B	384^2	89M	45.0G	95.7	**85.1**
● ConvNeXt-L	224^2	198M	34.4G	146.8	**84.3**
● ConvNeXt-L	384^2	198M	101.0G	50.4	**85.5**
ImageNet-22K pre-trained models					
● R-101x3 [39]	384^2	388M	204.6G	-	84.4
● R-152x4 [39]	480^2	937M	840.5G	-	85.4
● EffNetV2-L [72]	480^2	120M	53.0G	83.7	86.8
● EffNetV2-XL [72]	480^2	208M	94.0G	56.5	87.3
○ ViT-B/16 (☎) [67]	384^2	87M	55.5G	93.1	85.4
○ ViT-L/16 (☎) [67]	384^2	305M	191.1G	28.5	86.8
● ConvNeXt-T	224^2	29M	4.5G	774.7	**82.9**
● ConvNeXt-T	384^2	29M	13.1G	282.8	**84.1**
● ConvNeXt-S	224^2	50M	8.7G	447.1	**84.6**
● ConvNeXt-S	384^2	50M	25.5G	163.5	**85.8**
○ Swin-B	224^2	88M	15.4G	286.6	85.2
● ConvNeXt-B	224^2	89M	15.4G	292.1	**85.8**
○ Swin-B	384^2	88M	47.0G	85.1	86.4
● ConvNeXt-B	384^2	89M	45.1G	95.7	**86.8**
○ Swin-L	224^2	197M	34.5G	145.0	86.3
● ConvNeXt-L	224^2	198M	34.4G	146.8	**86.6**
○ Swin-L	384^2	197M	103.9G	46.0	87.3
● ConvNeXt-L	384^2	198M	101.0G	50.4	**87.5**
● ConvNeXt-XL	224^2	350M	60.9G	89.3	**87.0**
● ConvNeXt-XL	384^2	350M	179.0G	30.2	**87.8**

Table 1. **Classification accuracy on ImageNet-1K.** Similar to Transformers, ConvNeXt also shows promising scaling behavior with higher-capacity models and a larger (pre-training) dataset. Inference throughput is measured on a V100 GPU, following [45]. On an A100 GPU, ConvNeXt can have a much higher throughput than Swin Transformer. See Appendix E. (☎)ViT results with 90-epoch AugReg [67] training, provided through personal communication with the authors.

model	#param.	FLOPs	throughput (image / s)	training mem. (GB)	IN-1K acc.
○ ViT-S	22M	4.6G	978.5	4.9	79.8
● ConvNeXt-S (*iso.*)	22M	4.3G	1038.7	4.2	79.7
○ ViT-B	87M	17.6G	302.1	9.1	81.8
● ConvNeXt-B (*iso.*)	87M	16.9G	320.1	7.7	82.0
○ ViT-L	304M	61.6G	93.1	22.5	82.6
● ConvNeXt-L (*iso.*)	306M	59.7G	94.4	20.4	82.6

Table 2. **Comparing isotropic ConvNeXt and ViT.** Training memory is measured on V100 GPUs with 32 per-GPU batch size.



backbone	FLOPs	FPS	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
Mask-RCNN 3× schedule								
○ Swin-T	267G	23.1	46.0	68.1	50.3	41.6	65.1	44.9
● ConvNeXt-T	262G	25.6	**46.2**	67.9	50.8	**41.7**	65.0	44.9
Cascade Mask-RCNN 3× schedule								
● ResNet-50	739G	16.2	46.3	64.3	50.5	40.1	61.7	43.4
● X101-32	819G	13.8	48.1	66.5	52.4	41.6	63.9	45.2
● X101-64	972G	12.6	48.3	66.4	52.3	41.7	64.0	45.1
○ Swin-T	745G	12.2	50.4	69.2	54.7	43.7	66.6	47.3
● ConvNeXt-T	741G	13.5	**50.4**	69.1	54.8	**43.7**	66.5	47.3
○ Swin-S	838G	11.4	51.9	70.7	56.3	45.0	68.2	48.8
● ConvNeXt-S	827G	12.0	**51.9**	70.8	56.5	**45.0**	68.4	49.1
○ Swin-B	982G	10.7	51.9	70.5	56.4	45.0	68.1	48.9
● ConvNeXt-B	964G	11.4	**52.7**	71.3	57.2	**45.6**	68.9	49.5
○ Swin-B‡	982G	10.7	53.0	71.8	57.5	45.8	69.4	49.7
● ConvNeXt-B‡	964G	11.5	**54.0**	73.1	58.8	**46.9**	70.6	51.3
○ Swin-L‡	1382G	9.2	53.9	72.4	58.8	46.7	70.1	50.8
● ConvNeXt-L‡	1354G	10.0	**54.8**	73.8	59.8	**47.6**	71.3	51.7
● ConvNeXt-XL‡	1898G	8.6	**55.2**	74.2	59.9	**47.7**	71.6	52.2

个人理解

感觉就是仿ViT中的设计策略，去设计CNN的策略，把ViT中体现到的和 CNN一样的思想的那一块，验证，说明CNN也可以做到这个准确率。并不是CNN不行。然后就把所有的相似点扯到一块，做了一个实验，验证其有效性。感觉就是很多trick堆叠，没有昨天看的RepVGG重参数化惊艳。