“A ConvNet for the 2020s”论文阅读报告

姓名：陈鸿绪　　学号：PB21000224　　日期: 6.29.2024

一. 背景简介：

Transformer[1]架构的提出对自然语言处理（NLP）领域带来了深远的影响，Transformer 架构摒弃了传统的循环神经网络（RNN）或长短时记忆网络（LSTM）结构，引入了全新的自注意力机制（Self-Attention Mechanism），这一创新使得文本数据的处理更为高效，Transformer 在处理长序列时能够更好地捕捉长距离依赖关系，从而克服了传统 RNN 在处理长序列时存在的梯度消失和梯度爆炸问题。

Transformer 架构在 NLP 领域的革命性突破让计算机视觉（CV）领域也开始了新架构的尝试，以基于自注意力机制架构的 Vision Transformer[2]（ViT）成功地将 Transformer 引入到 CV，而在这之前，卷积神经网络（CNN）在计算机视觉任务中一直占据主导地位。然而 ViT 的出现证明了纯 Transformer 架构在图像分类等任务上也能表现出色，直接挑战了 CNN 的统治地位。在研究人员 ViT 的研究基础上陆续提出了 Transformer in Transformer（TNT）、Data-efficient image Transformers[3]（DeiT）、Swing Transformer[4]（SwinT，注意后续其与 Swin-T 的区别）等多种改进的 ViT 变体。其中拥有可变感受野、移位窗口自注意力机制、线性计算复杂度的 SwinT 以其巧妙的设计和在 ImageNet 数据集上的 Sota（提出的时候）水平再次证明了自注意力机制的强大。在 Transformer 的冲击下的 2020s，卷积神经网络似乎尽显颓势。

二. 问题提出：

在计算机视觉领域，卷积的统治地位并非偶然。滑动窗口策略本质上与视觉处理的需求相契合，归纳偏置和平移不变性特质使得 CNN 在该领域的应用中极为适宜。尽管 ViT 曾面临挑战，即普通 ViT 的感受野是固定的，不能关注到一个 patch 中的内部信息，但这些困难通过引入类似卷积中滑动窗口策略的 Swin Transformer 得到了解决。这一事实进一步表现出卷积在视觉处理中的重要性，同时也启发我们思考如何将 Transformer 的优势与卷积网络相结合。基于 Transformer 的尝试往往伴随复杂的系统设计和非常大的算力代价，是否可以将这些设计思想移植到结构简单的 CNN 网络上，因为我们知道 CNN 之所以会显现颓势主要是因为 Transformer 基于其多头自注意力机制的优越的尺度行为。

在经过以上分析后，我们自然会考虑架构简单的 CNN 真的是弱于 Transformer，是否可以得到比 SwinT 性能更加优越纯卷积神经网络。

三. 论文简介：

Facebook AI Research 和 UC Berkeley 的联合论文“A ConvNet for the 2020s”[5]探究了卷积神经网络的极限，这篇文章发表在了 2022 年 CVPR 会议上。该工作提出了 ConvNeXt 纯卷积网络，其在设计网络时想法理念对标于 Swin Transformer，作者在此实验中发现了几个导致性能差异的关键组件。实验证明，ConvNeXt 相比 SwinT 拥有更快推理速度与更高准确率，在 ImageNet 22K 上 ConvNeXt-XL 达到了 87.8%的准确率（当时的 Sota），从而证明了纯 CNN 结构并没有达到它的瓶颈。下面我将具体阐述论文的提出方法、实验结果与分析。

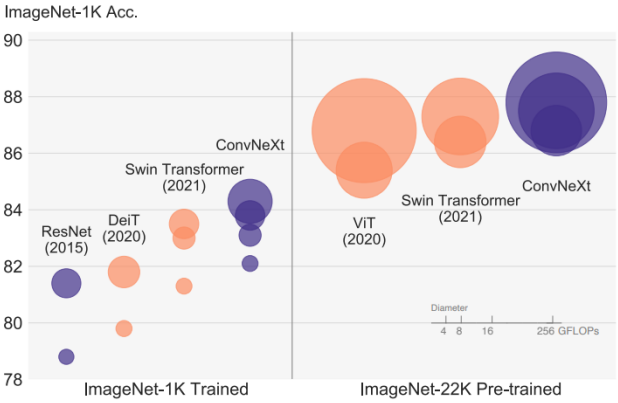

图 1. ImageNet-1K classification results

四. 提出方法:

作者使用 ViT 的训练策略重新训练得到 ResNet50[6]模型，其效果显著优于原始结果，所以将该结果认定为后续实验的 baseline。ConvNeXt 设计的主要创新点分为下面五个：

1. **Macro Design:** 对于 SwinT 设计遵循了过去 ConvNets 的设计规则，即使用了 multi-stage design，所以为了对标 SwinT 设计，ConvNeXt 的设计在这部分着重关注两个部分：stage compute ratio 和 “stem cell” structure。

   1) Stage compute ratio：原始 ResNet50 的 block 堆叠次数为(3, 4, 6, 3)，比例近似为 1:1:2:1。参考 Swin-T 的堆叠 block 次数比例为 1:1:3:1，Swin-L 的比例为 1:1:9:1，明显在 SwinT 在设计时其 stage3 的 block 占比相对其它 stage 较大，所以作者据此将原始 ResNet50 的堆叠次数改变成为(3, 3, 9, 3)，FLOPs 与 Swin-T 相似。最后模型准确率从 78.8%提升至 79.4%。

   2) Changing stem to “Patchify”: 在之前的卷积神经网络中，一般最初下采样 stem 一般使用一个 7×7 卷积核、stride 为 2 的卷积层加上 strade 为 2 的 Maxpooling 下采样组成，该 stem 可以使得完成 4×的下采样。但是 ViT 采取了更加激进的 patchify 方案，这对应着更大的卷积核，且相邻窗口没有交集（SwinT 也采取了该方案）。对标 SwinT，作者将 stem 替换成了 patchify 方案，替换之后准确率从 79.4%上升到 79.5%，且 FLOPs 也略微降低。

2. **ResNeXt-ify：**这部分作者主要运用了 ResNeXt[7]的思想，即采用组卷积（卷积核被分成了不同组），采用组卷积后 ResNeXt 做到了比原始 ResNet 更好的 FLOPs 与 accuracy 之间的平衡。作者采用了深度可分离卷积设计方案（即组卷积在 group 数和 channel 数相同的特殊情况），作者采取这种动机在于他们认为组卷积某种程度上类似于自注意力机制的加权和。这种设计有效减少了网络的 FLOPs。同时网络宽度也改变成了与 Swin-T 相同的通道数(64 到 96)，最后这些改变使得模型准确率上升到 80.5%。

3. **Inverted Bottleneck：**Inverted Bottleneck 指的是 MLP block 的隐藏层维度比输入向量维度宽 4 倍，是 Transformer 采用重要的一个设计。作者在模型设计中引入了 Inverted Bottleneck，最后在小模型上准确率从 80.5%上升至 80.6%，但在大模型（如 ResNet-200 / Swin-B regime），这步使得准确率得到较为显著的提升，从 81.9%提升至 82.6%同时也减少了 FLOPs。

4. **Large Kernel Sizes：**过去 ConvNets 中曾使用过较大的卷积核尺寸，但 VGGNet[8]普及了堆叠 3×3 小尺寸卷积层，所以现代 GPU 对 3×3 卷积做了很多的优化。考察 SwinT 的自注意力块中引入的局部窗口，其大小为 7×7，这远大于 3×3 卷积核大小，所以我们需

要重新审视 ConvNets 中使用大卷积层的做法。

1) 上移 depthwise conv layer：为了探索大尺寸卷积核，作者将模型向上移动 depthwise 卷积层，这是在 Transformers 中较为明显类似的设计，因为 MSA block 也被放置在 MLP 层之前。这样改动准确率下降到 79.9%，其 FLOPs 略微减少。

2) 加大卷积尺寸：作者接着将 depthwise 卷积层的卷积核增大至 7×7，作者在尝试其它尺寸时发现 7 达到了饱和性能。准确率从原本 3×3 卷积核的 79.9%上升到 7×7 卷积核的 80.6%。

5. **Micro Design：**这部分主要关注微观尺寸上模型的变化，即这些探索主要聚焦于网络层，着重关注激活函数以及 normalization 层的选择。

1) 替换激活函数 ReLU 为 GELU：在 transformer 的原始论文中，ReLU 作为激活函数。但在 BERT、OpenAI 的 GPT-2、ViTs 模型以及先进的 transformer 模型中，高斯误差线性单元（GULE，可以看作 ReLU 的变体）得以应用。所以作者采取 GELU 代替 ReLU，尽管准确率保持一致。

2) 更少的激活函数：考虑到 Transformer 中并不是每一个模型都接有激活函数，所以作者采取适当减少激活函数的方案，发现准确率从 80.6%上升至 81.3%。

3) 更少的 Normalization 层：考虑到 Transformer 中，Normalization 层的使用有限，所以作者也减少了 ConNeXT Block 中的 Normalization 层的使用，只对 depthwise 卷积层后的 Normalization 层保留，最终准确率上升到了 81.4%。超过了 Swin-T。

4) 替代 Batch Normalization（BN）为 Layer Normalization（LN）：考虑到 Transformer 中基本使用的都为 LN，作者将 BN 替换成了 LN，发现准确率小幅度提升到了 81.5%。

5) 分离出下采样层：考虑 SwinT 设计采用了 Patch Merging 操作，作者为 ConvNext 单独设计了下采样层，即 LN 后接 2×2 卷积层，进行了此设计后准确率提升至 82.0%。

五. 实验结果及分析：

实验训练的模型根据数据集分为了两类，一类是直接在 ImageNet-1K 上进行训练，另一类是通过在 ImageNet-22K 上预训练后再通过在 ImageNet-1K 上微调的模型。对上面两类模型在 ImageNet-1K 上进行分类任务准确率预测。同时也进行了 ConvNeXt vs ViT 的消融实验。最后将 ConvNeXt 运用在特定下游任务上评估其性能。

1. ImageNet-1K（在 ImageNet-1K 上训练的若干模型）

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**

表 1：ImageNet-1K trained models 准确率

作者在上面表 1 展示了当时最新的 Transformer 变体（DeiT 和 SwinT）以及来自架构搜索的两种 ConvNet（RegNets、EfficientNets[9]和 EfficientNetsV2[10]）的结果对比，注意上述参数量相近的之间进行比较。ConvNeXt 在准确性与计算量权衡方面，以及推理吞吐量方面，与两个强大的 ConvNet 基线（RegNet[11]和 EfficientNet）相比具有一定竞争力。

此外，ConvNeXt 全面超越具有相似复杂度的 Swin Transformer，甚至有时其优势非常显著（例如，ConvNeXt-T 比 Swin-T 高出 0.8%的准确率）。然而 ConvNeXt 没有使用诸如移位窗口或相对位置偏差等专用模块，同时相比之下，ConvNeXt 吞吐量也有所提高。

特别地，我们需要注意结果中 ConvNeXt-B 在测试图像在 384×384 分辨率下，它比 Swin-B 高出 0.6%（84.5%到 85.1%），但推理吞吐量高出 12.5%（85.1 到 95.7 图像/秒）。当分辨率从 224×224 增加到 384×384 时，ConvNeXt-B 相对于 Swin-B 的浮点运算/吞吐量优势会变得更大。此外，当我们进一步扩展到 ConvNeXt-L 时，可以从上表观察到结果提高到 85.5%。

2. ImageNet-22K（在 ImageNet-22K 上预训练的若干模型）

model	image size	#param.	FLOPs	throughput (image / s)	IN-1K top-1 acc.
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**

表 2：ImageNet-22K pre-trained models 准确率

上表展示了使用 ImageNet-22K 预训练微调后的模型结果。学界普遍认为 vision Transformers 的归纳偏差较少，因此在更大规模的数据集上进行预训练时，性能会比基于纯卷积的 ConvNets 更好。

然而事实上，上表结果表明，当使用大型数据集进行预训练时，作者所设计的卷积神经网络并不逊色于基于 Transformer 的模型，ConvNeXts 的性能仍然与参数量级一致的 SwinT 相当或更好，并且吞吐量略高。

此外，我们注意 ConvNeXt-XL 模型在图像 384×384 的分辨率下达到了 87.8%的准确率，相比 ConvNeXt-L 的 87.5%有了显著的改进，这证明了 ConvNeXts 是可扩展的架构。 在 ImageNet-1K 上，EfficientNetV2-L 通过一系列模块配置达到了顶级性能。然而，通过模型在 ImageNet-22K 上的预训练，ConvNeXt 能够超越 EfficientNetV2（87.8%超出 87.3%），这进一步证明了大规模训练对 ConvNeXts 性能表现的重要性。

3. ConvNeXts 与 ViT 消融对比实验

作者希望在这个部分探究 ConvNeXt 块设计是否可泛化到 ViT 风格的等轴架构中。所以作者构建了 ConvNeXt-S/B/L，使用与 ViT-S/B/L（384/768/1024）相同的特征维度，最后得到了在 ImageNet-1K 数据集的若干结果，如下表所示：

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

表 3. Comparing isotropic ConvNeXt and ViT

由上表，我们可以看见相同维度的 ConvNeXt 和 ViT 相比具有较强的竞争力，ConvNeXt-L 和 ConvNeXt-B 在 IN-1K acc 上都不比相应的 ViT-L 与 ViT-B 低，且相同维度 ConvNeXt 训练占用显存是比 ViT 小的。所以可以认为 ConvNeXt 通常可以与 ViT 相媲美，这表明当用于非层次化模型时，ConvNeXt 块设计具有竞争力，ConvNeXt 块设计是可以泛化到 ViT 风格的等轴架构中的。

4. 下游任务表现

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

表 4. COCO object detection and segmentation results

可以从上表看出对于 COCO 物体检测和分割任务，相同维度而言，基于 ConvNeXt 的模型测试指标几乎都是相当或者优于基于 SwinT 模型的测试指标的。当扩大到更大的模型（在 ImageNet-22K 上预训练的 ConvNeXt-B/L/XL）时，可以看出 ConvNeXt 在很多情况下是显著优于 Swin Transformer（例如，+1.0 AP）。

backbone	input crop.	mIoU	#param.	FLOPs
ImageNet-1K pre-trained				
○ Swin-T	512^2	45.8	60M	945G
● ConvNeXt-T	512^2	**46.7**	60M	939G
○ Swin-S	512^2	49.5	81M	1038G
● ConvNeXt-S	512^2	**49.6**	82M	1027G
○ Swin-B	512^2	49.7	121M	1188G
● ConvNeXt-B	512^2	**49.9**	122M	1170G
ImageNet-22K pre-trained				
○ Swin-B‡	640^2	51.7	121M	1841G
● ConvNeXt-B‡	640^2	**53.1**	122M	1828G
○ Swin-L‡	640^2	53.5	234M	2468G
● ConvNeXt-L‡	640^2	**53.7**	235M	2458G
● ConvNeXt-XL‡	640^2	**54.0**	391M	3335G

表 5. ADE20K validation results

作者还使用 UperNet[12]在 ADE20K 语义分割任务上评估了 ConvNeXt 主干网络。所有模型变体都进行了 160K 次迭代训练，批次大小为 16。在表 5 中，可以看出多尺度测试下的验证 mIoU 指标。上表显示 ConvNeXt 模型可以在不同模型容量下实现具有竞争力的性能，进一步验证了 ConvNeXt 模型架构设计的有效性。

六. 与课内所学的关联&阅读收获

在深度学习导论课程中，我们学习了卷积神经网络和基于 Transformer 架构的网络模型，也知道了在学界开始逐渐开始形成一种刻板印象：认为传统卷积已经在 CV 领域输给基于自注意力机制的 Transformer。这篇论文工作的出现说明了纯卷积网络其实还没有达到性能的尽头，只要它设计得当，它仍然和 Transformer 架构有相当的竞争力。

通过学习这篇论文的工作，我认识到在这个 Transformer 架构盛行的时代，需要重新审视传统卷积，纯卷积神经网络不仅拥有其简单直观的架构，还拥有仍然没有被完全开发的强大性能。同时通过论文中在大规模数据集（如 ImageNet-22K）上进行预训练的实验，我明白了大规模预训练对于提高模型性能的重要性。除了架构设计外，论文还强调了训练策略对于提高模型性能的重要性。通过采用更长的预热周期和更精细的训练设置，ConvNeXt 能够在相同的计算资源下获得更好的性能。

参考文献

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[2] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[3] Touvron H, Cord M, Douze M, et al. Training data-efficient image transformers & distillation through attention[C]//International conference on machine learning. PMLR, 2021: 10347-10357.

[4] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.

[5] Liu Z, Mao H, Wu C Y, et al. A convnet for the 2020s[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 11976-11986.

[6] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[7] Xie S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1492-1500.

[8] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[9] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]//International conference on machine learning. PMLR, 2019: 6105-6114.

[10] Tan M, Le Q. Efficientnetv2: Smaller models and faster training[C]//International conference on machine learning. PMLR, 2021: 10096-10106.

[11] Radosavovic I, Kosaraju R P, Girshick R, et al. Designing network design spaces[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10428-10436.

[12] Xiao T, Liu Y, Zhou B, et al. Unified perceptual parsing for scene understanding[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 418-434.