A ConvNet for the 2020s

Доклад по статье

Петров Михаил, Бакалова Александра, Аланова Ширин
НИС Машинное обучение и приложения, 12.10.2022

Обзор статьи

Модели для обработки изображений

ConvNet:

- меньше параметров;
- translation equivariance.

Vision Transformer:

- более гибкая архитектура;
- лучше на классификации, но хуже на прочих задачах CV.

SWIN Transformer:

- shifted window — компромисс между архитектурами;
- успех во многих задачах.

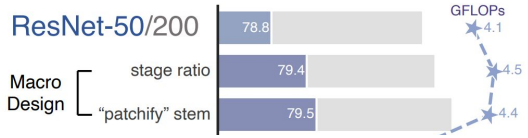
Цель работы

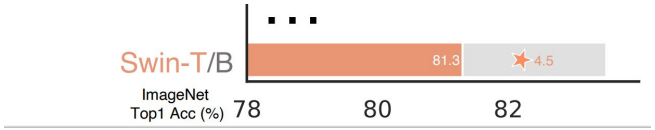
Как дизайн-решения, используемые при построении ViT, повлияют на качество свёрточных сетей?

Оттолкнёмся от ResNet и будем последовательно применять разные идеи, чтобы поднимать качество классификации (ImageNet-1K).

Macro Design

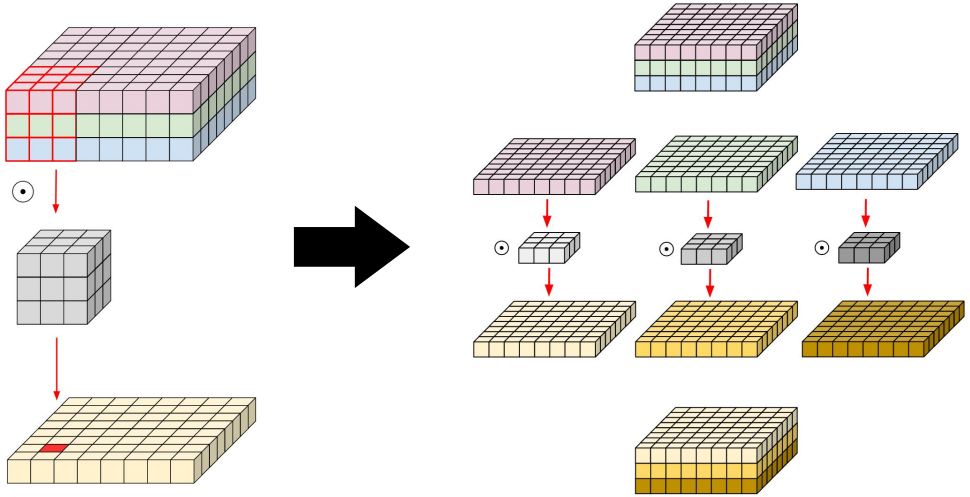
- Обучим обычный ResNet-50 так, как будто это трансформер: AdamW, больше эпох, больше аугментаций, регуляризация.
- Выровняем количество блоков между пулингами, как в Swin Transformer.
- Поставим свёртки 4x4, stride=4, как в Swin Transformer.

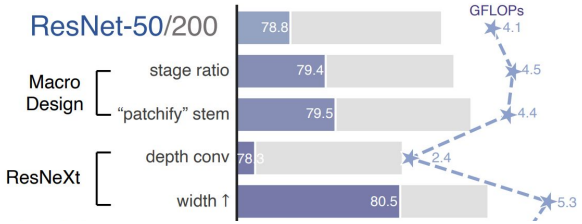


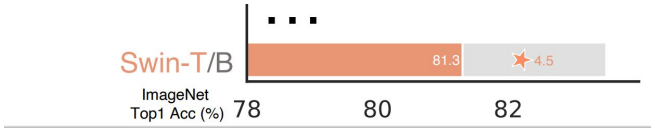


ResNeXt-ify

- Применим depthwise convolutions:

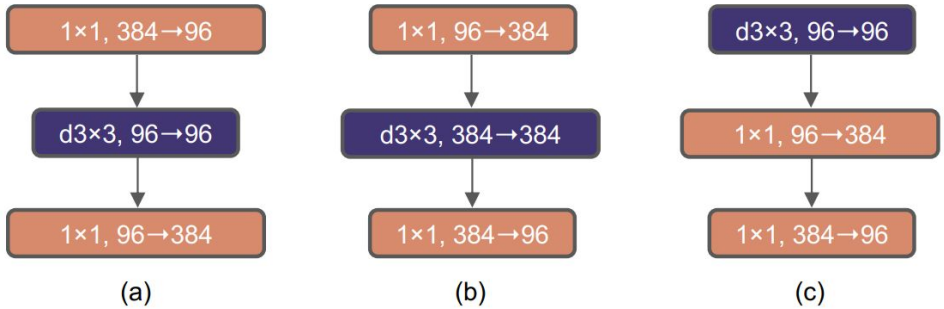


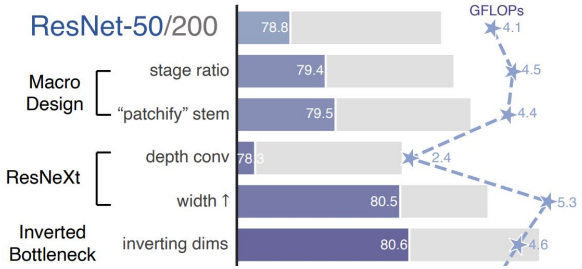


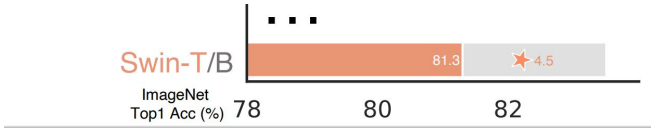


Inverted Bottleneck

- Поменяем канальные размерности свёрток и их последовательность в блоке:

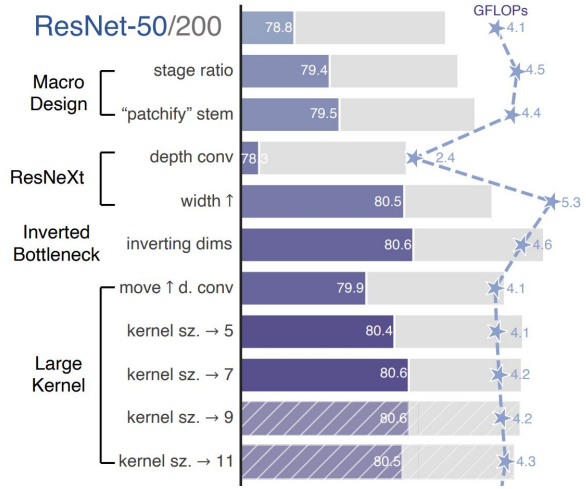


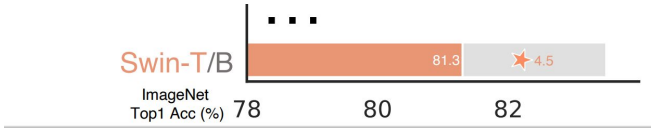




Large Kernel

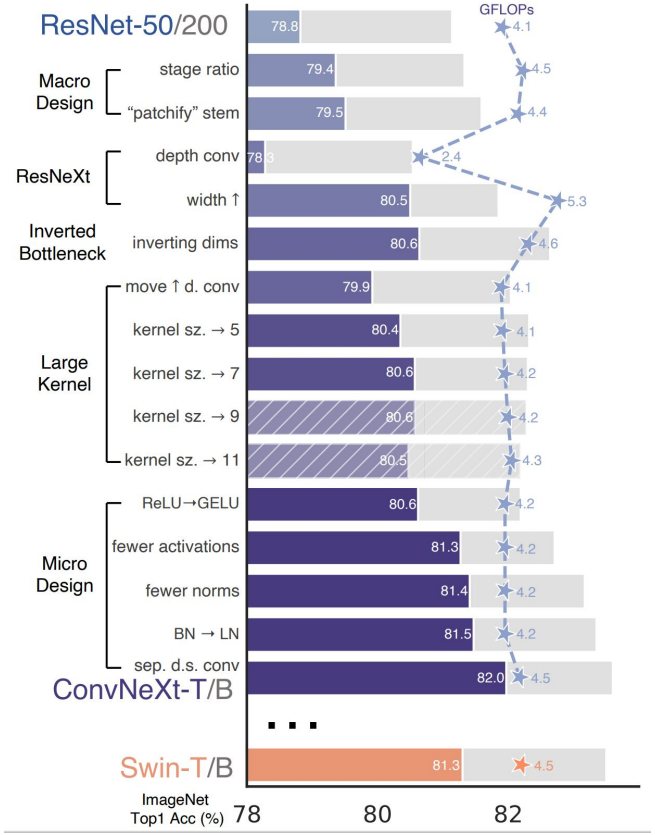
- Поэкспериментируем с размером фильтров для depthwise convolutions, чтобы достичь сопоставимого с трансформерами поля восприятия.





Micro Design

- Опять же, смотрим на детали архитектуры трансформеров и повторяем их в нашей модели.
- Так как это последний этап модификации, следим за количеством FLOPs для корректного сравнения.



Результаты

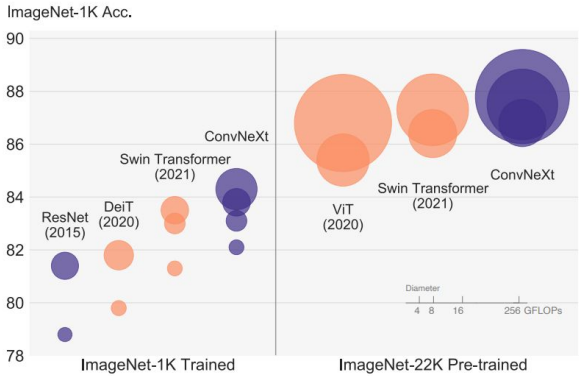


Figure 1. **ImageNet-1K classification** results for • ConvNets and ○ vision Transformers. Each bubble's area is proportional to FLOPs of a variant in a model family. ImageNet-1K/22K models here take 224^2/384^2 images respectively. ResNet and ViT results were obtained with improved training procedures 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**

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

Table 3. **COCO object detection and segmentation results** using Mask-RCNN and Cascade Mask-RCNN. ‡ indicates that the model is pre-trained on ImageNet-22K. ImageNet-1K pre-trained Swin results are from their Github repository [3]. AP numbers of the ResNet-50 and X101 models are from [45]. We measure FPS on an A100 GPU. FLOPs are calculated with image size (1280, 800).

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

Table 4. **ADE20K validation results** using UperNet [85]. ‡ indicates IN-22K pre-training. Swins' results are from its GitHub repository [2]. Following Swin, we report mIoU results with multi-scale testing. FLOPs are based on input sizes of (2048, 512) and (2560, 640) for IN-1K and IN-22K pre-trained models, respectively.

Результат

Свёртки всё ещё актуальны!

(если их правильно оформить)

Рецензия

A ConvNet for the 2020s

Zhuang Liu[1,2*] Hanzi Mao[1] Chao-Yuan Wu[1] Christoph Feichtenhofer[1] Trevor Darrell[2] Saining Xie[1†]
[1]Facebook AI Research (FAR) [2]UC Berkeley
Code: https://github.com/facebookresearch/ConvNeXt

Abstract

The "Roaring 20s" of visual recognition began with the introduction of Vision Transformers (ViTs), which quickly superseded ConvNets as the state-of-the-art image classification model. A vanilla ViT, on the other hand, faces difficulties when applied to general computer vision tasks such as object detection and semantic segmentation. It is the hierarchical Transformers (e.g., Swin Transformers) that reintroduced several ConvNet priors, making Transformers practically viable as a generic vision backbone and demonstrating remarkable performance on a wide variety of vision tasks. However, the effectiveness of such hybrid approaches is still largely credited to the intrinsic superiority of Transformers, rather than the inherent inductive biases of convolutions. In this work, we reexamine the design spaces and test the limits of what a pure ConvNet can achieve. We gradually "modernize" a standard ResNet toward the design of a vision Transformer, and discover several key components that contribute to the performance difference along the way. The outcome of this exploration is a family of pure ConvNet models dubbed ConvNeXt. Constructed entirely from standard ConvNet modules, ConvNeXts compete favorably with Transformers in terms of accuracy and scalability, achieving 87.8% ImageNet top-1 accuracy and outperforming Swin Transformers on COCO detection and ADE20K segmentation, while maintaining the simplicity and efficiency of standard ConvNets.

1. Introduction

Looking back at the 2010s, the decade was marked by the monumental progress and impact of deep learning. The primary driver was the renaissance of neural networks, particularly convolutional neural networks (ConvNets). Through

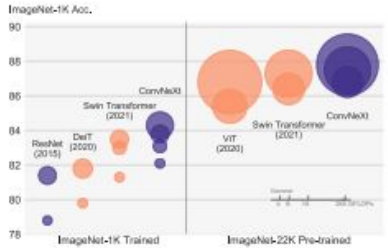


Figure 1. **ImageNet-1K classification** results for • ConvNets and ○ vision Transformers. Each bubble's area is proportional to FLOPs of a variant in a model family. ImageNet-1K/22K models here take 224^2/384^2 images respectively. ResNet and ViT results were obtained with improved training procedures 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.

visual feature learning. The introduction of AlexNet [40] precipitated the "ImageNet moment" [59], ushering in a new era of computer vision. The field has since evolved at a rapid speed. Representative ConvNets like VGGNet [64], Inceptions [68], ResNe(X)t [28, 87], DenseNet [36], MobileNet [34], EfficientNet [71] and RegNet [54] focused on different aspects of accuracy, efficiency and scalability, and popularized many useful design principles.

The full dominance of ConvNets in computer vision was not a coincidence: in many application scenarios, a "sliding window" strategy is intrinsic to visual processing, particularly when working with high-resolution images. ConvNets have several built-in inductive biases that make them well-suited to a wide variety of computer vision applications. The

ith several kernel sizes, includ-
twork's performance increases
% (7×7), while the network's
. Additionally, we observe that
zes reaches a saturation point at
ior in the large capacity model
nodel does not exhibit further
ernel size beyond 7×7.
e conv in each block.
concluded our examination of
nacro scale. Intriguingly, a sig-
choices taken in a vision Trans-
onvNet instantiations.

gate several other architectural
– most of the explorations here
ocusing on specific choices of
nalization layers.

LU One discrepancy between
es is the specifics of which ac-
Numerous activation functions
e, but the Rectified Linear Unit
ely used in ConvNets due to its
LU is also used as an activation
sformer paper [77]. The Gaus-
LU [32], which can be thought
ReLU, is utilized in the most
uding Google's BERT [18] and
most recently, ViTs. We find
d with GELU in our ConvNet
ays unchanged (80.6%).

One minor distinction between
block is that Transformers have
fewer activation functions. Consider a Transformer block with key/query/value linear embedding layers, the projection layer, and two linear layers in an MLP block. There is only one activation function present in the MLP block. In

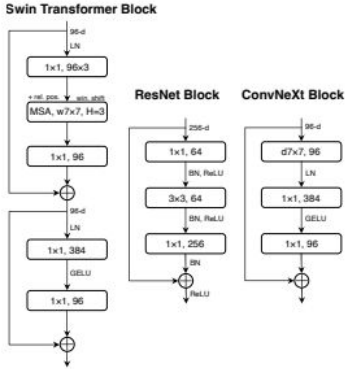


Figure 4. **Block designs** for a ResNet, a Swin Transformer, and a ConvNeXt. Swin Transformer's block is more sophisticated due to the presence of multiple specialized modules and two residual connections. For simplicity, we note the linear layers in Transformer MLP blocks also as "1×1 convs" since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

that we have even fewer normalization layers per block than Transformers, as empirically we find that adding one additional BN layer at the beginning of the block does not improve the performance.

Substituting BN with LN. BatchNorm [38] is an essential component in ConvNets as it improves the convergence and reduces overfitting. However, BN also has many intricacies that can have a detrimental effect on the model's performance [84]. There have been numerous attempts at developing alternative normalization [60, 75, 83] techniques, but BN has remained the preferred option in most vision

IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2022

☆ 4.3k stars

Авторы статьи

Zhuang Liu,
FAIR, UC Berkeley
h-index: 16

Learning Efficient Convolutional Networks through Network Slimming

Densely Connected Convolutional Networks

Hanzi Mao,
FAIR
h-index: 5

Exploring Plain Vision Transformer Backbones for Object Detection

Chao-Yuan Wu,
FAIR
h-index: 17

Christoph Feichtenhofer,
FAIR
h-index: 30

Trevor Darrell,
UC Berkeley
h-index: 148

Научный руководитель Zhuang Liu

Saining Xie,
FAIR
h-index: 27

Таймлайн

- An image is worth 16x16 words: Transformers for image recognition at scale. In ICLR, 2021, Dosovitskiy A. et al
- Swin transformer: Hierarchical vision transformer using shifted windows. ICCV 2021, Liu Z. et al
- A convnet for the 2020s. In CVPR 2022, Liu Z. et al

Цитирования и близкие работы

- ConvMixer: large kernels, depthwise convolutions, patchify layer.
  Patches are all you need? arXiv, 2022. Trockman A., et al.

- Local attention is equivalent to dynamic depthwise convolutions.
  Demystifying local vision transformer: Sparse connectivity, weight sharing, and dynamic weight. arXiv, 2021. Han Q. et al.

- Среди 341 цитирований есть очень близкие работы
  Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. CVPR– 2022, Ding X. et al.
  MixFormer: Mixing Features across Windows and Dimensions
  DynaMixer: A Vision MLP Architecture with Dynamic Mixing

Сильные стороны

- Добились результата
- Объяснили свои действия
- Выложили код
- Много экспериментов:
  - Сравнили с Swin Transformer, DeiT, RegNet, EfficientNetV2 с разными размерами моделей, а также с предобучением и без предобучения на большем датасете
  - Сравнили с Swin Transformer на задачах детекции и сегментации на датасетах COCO и ADE20K
  - Измерили устойчивость
  - Измерили время инференса на A100 GPUs

Слабые стороны

- Сравнили с Swin Transformer на небольшом наборе задач
- При введении нового улучшения смотрели только на top-1 accuracy