Paper Review

InceptionNeXt: When Inception Meets ConvNeXt

Weihao Yu (National University of Singapore), et al.

CVPR 2024

Reviewed by Susang Kim

Contents

1.Introduction - Traditional Convolution Neural Networks

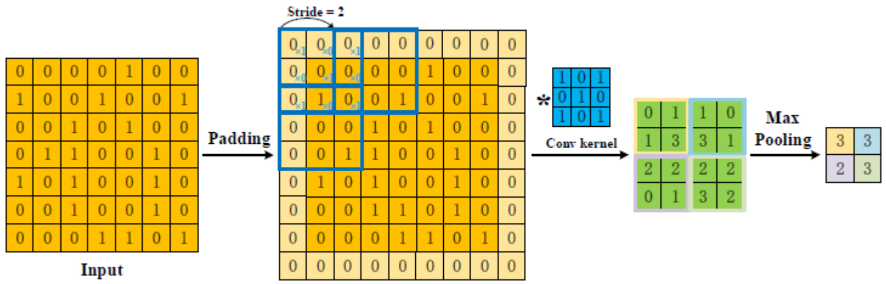
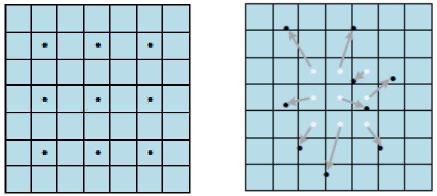


Fig. 1. Procedure of a two-dimensional CNN



Dilated&Deformable convolution kernel.

CNN has been making brilliant achievements, which have become one of the most representative neural networks.

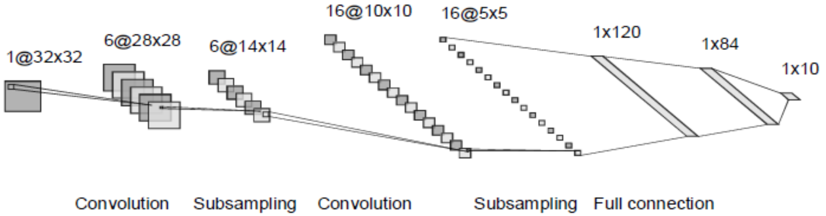
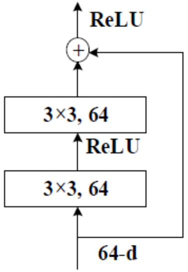
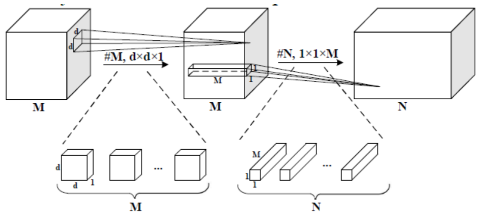
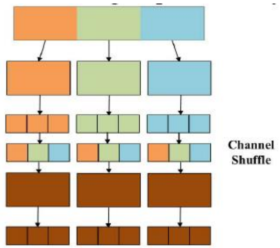


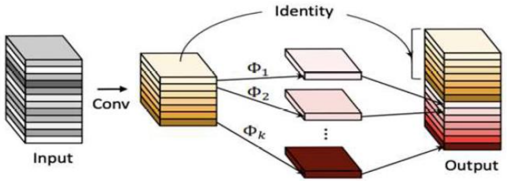
Fig. 5. Architecture of LeNet-5

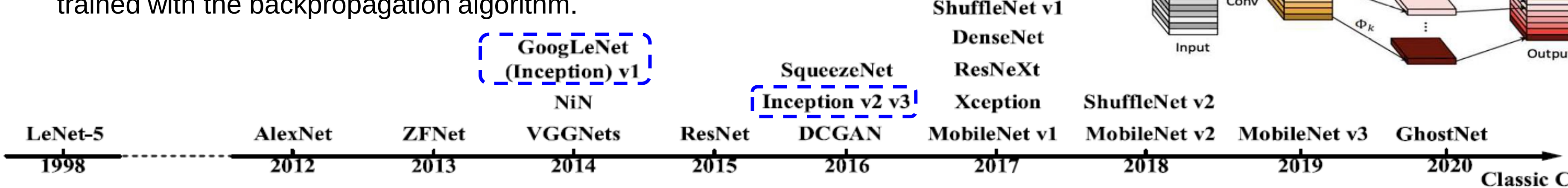






LeCun et al. proposed LeNet-5 in 1998, CNN trained with the backpropagation algorithm.

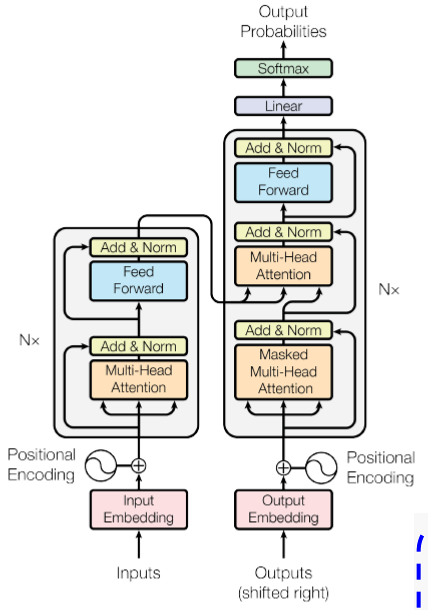


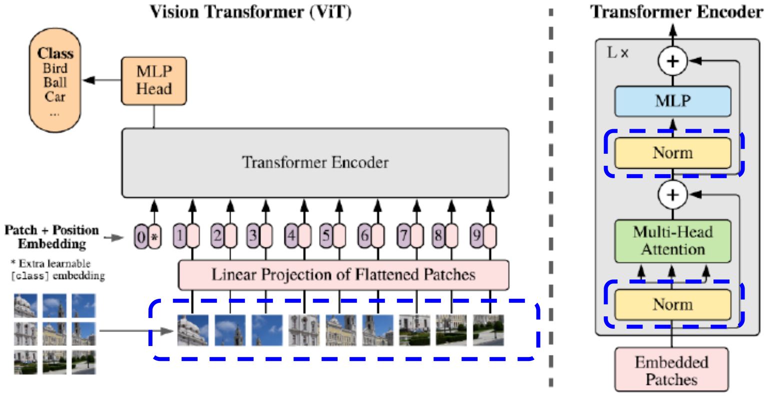


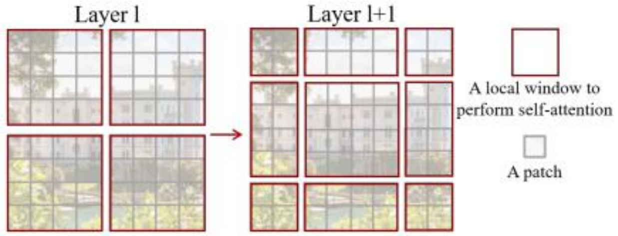
Classic CNN structures

Li, Zewen, et al. "A survey of convolutional neural networks: analysis, applications, and prospects." *IEEE transactions on neural networks (2021)*

1.Introduction – From Transformer to Vision Transformer







An illustration of the shifted window approach for computing self-attention in the proposed Swin Transformer architecture.

2017.6 | Transformer
Solely based on attention mechanism, the Transformer is proposed and shows great performance on NLP tasks.

2018.10 | BERT
Pre-training transformer models begin to be dominated in the field of NLP.

2020.5 | GPT-3
A huge transformer with 170B parameters, takes a big step towards general NLP model.

2020.5 | DETR
A simple yet effective framework for high-level vision by viewing object detection as a direct set prediction problem.

2020.7 | iGPT
The transformer model for NLP can also be used for image pre-training.

2020.10 | ViT
Pure transformer architectures work well for visual recognition.

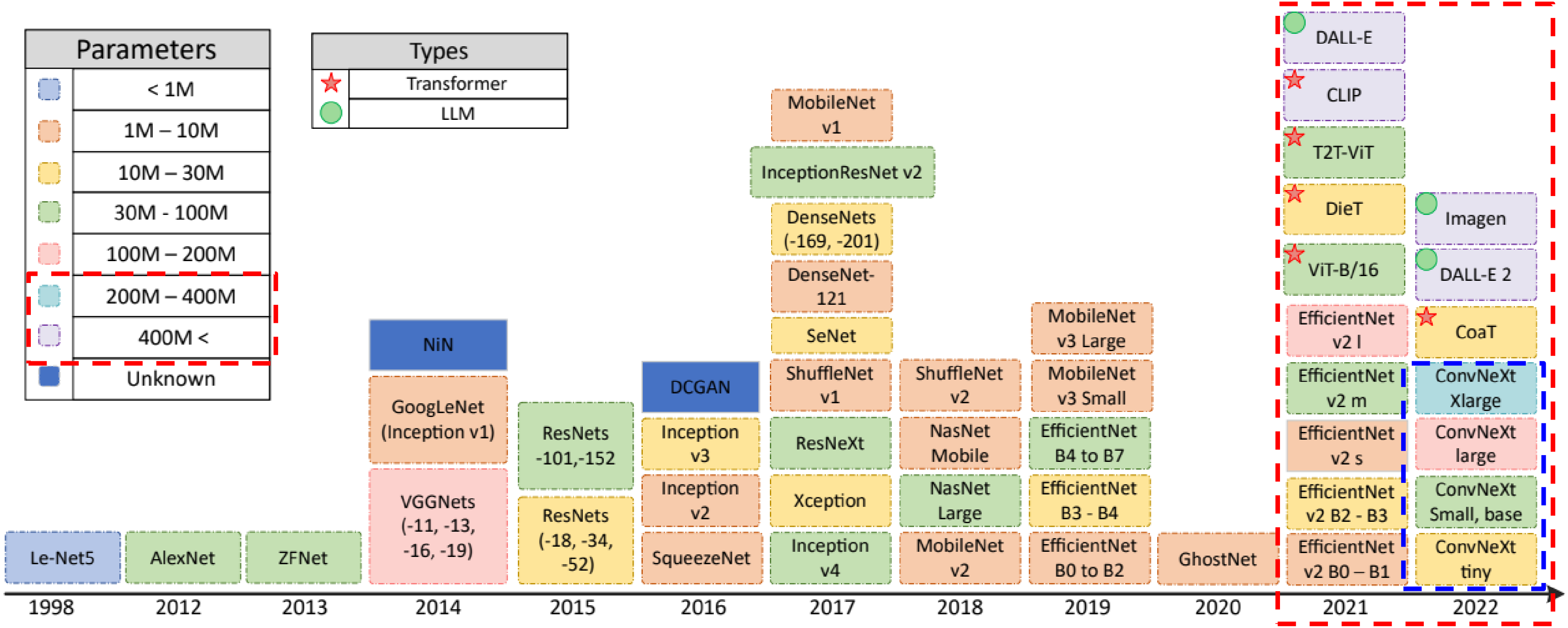
End of 2020 | IPT/SETR/CLIP
Applications of transformer model on low-level vision, segmentation and multimodality tasks, respectively.

2021 | ViT Variants
Variants of ViT models, e.g., DeiT, PVT, TNT, and Swin.

Han, Kai, et al. "A survey on vision transformer." IEEE TPAMI 2022.

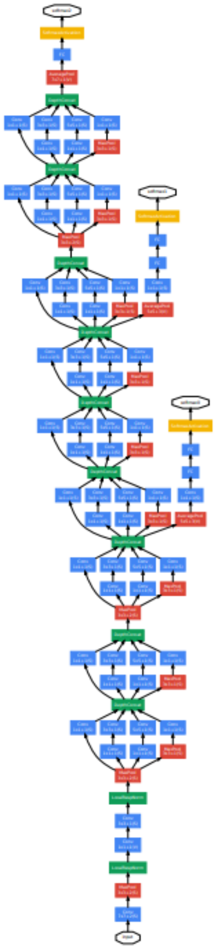
1.Introduction - Evolution of CNN Architectures

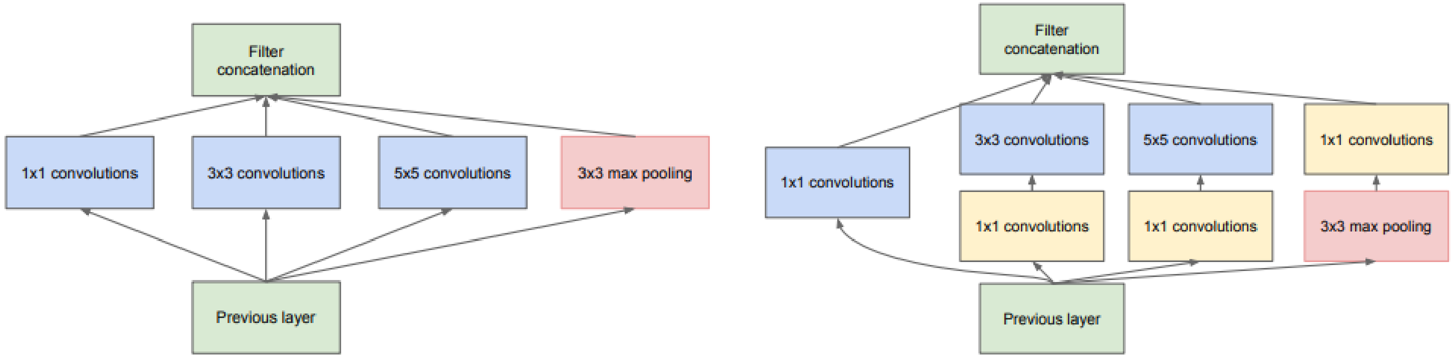
Since the early origins of CNNs, there has been a rapid evolution in CNN architectures over the past decade to enhance performance and efficiency.



YOUNESI, Abolfazl, et al. A Comprehensive Survey of Convolutions in Deep Learning: Applications, Challenges, and Future Trends. arXiv 2024.

2.Related Works - Inception(Going Deeper with Convolutions) (CVPR 2015)





(a) Inception module, naïve version (b) Inception module with dimension reductions

Figure 2: Inception module

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

(a) Siberian husky (b) Eskimo dog

It is necessary to distinguish between fine-grained visual categories like those in ImageNet

2.Related Works - Inception Transformer (NeurIPS 2022)

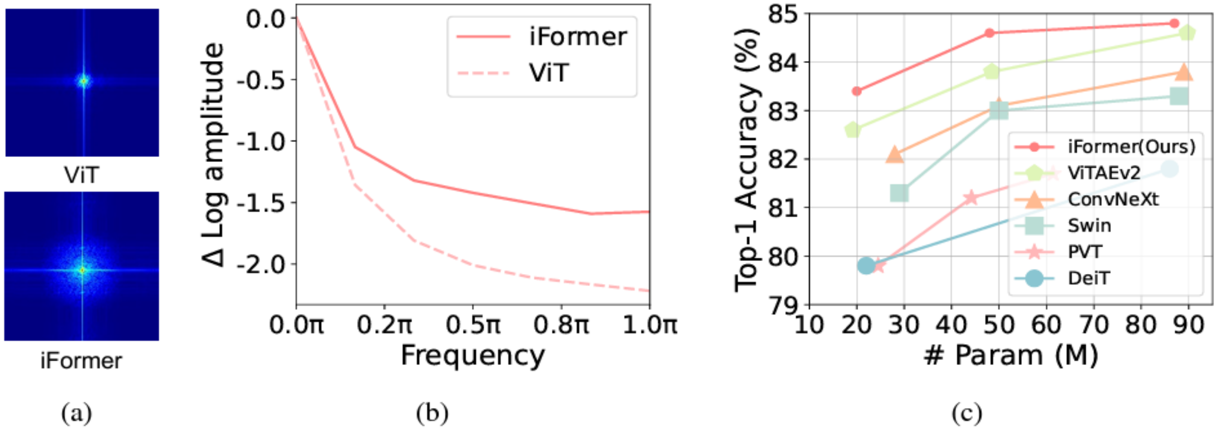
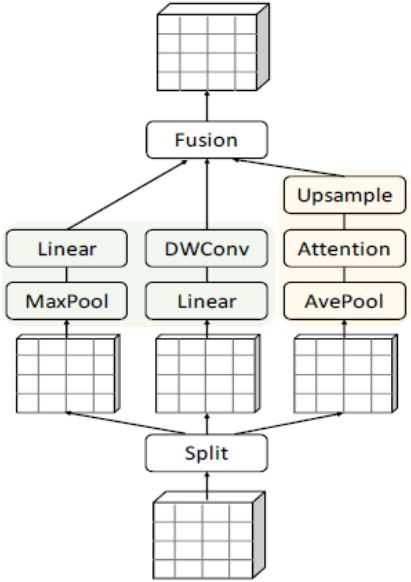
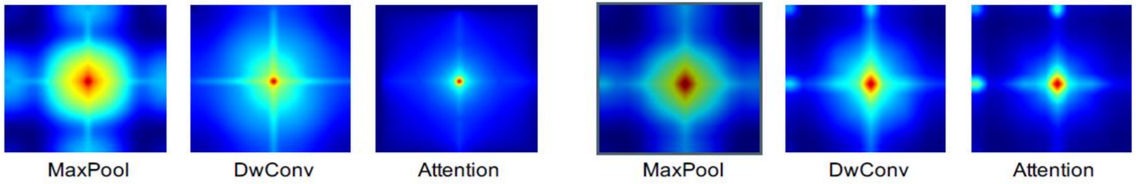


Figure 1: **(a) Fourier spectrum of ViT [18] and iFormer. (b) Relative log amplitudes of Fourier transformed feature maps. (c) Performance of models on ImageNet-1K validation set.** (a) and (b) show that iFormer captures more high-frequency signals.



Effectively learns comprehensive features with both high- and low-frequency information in visual data.

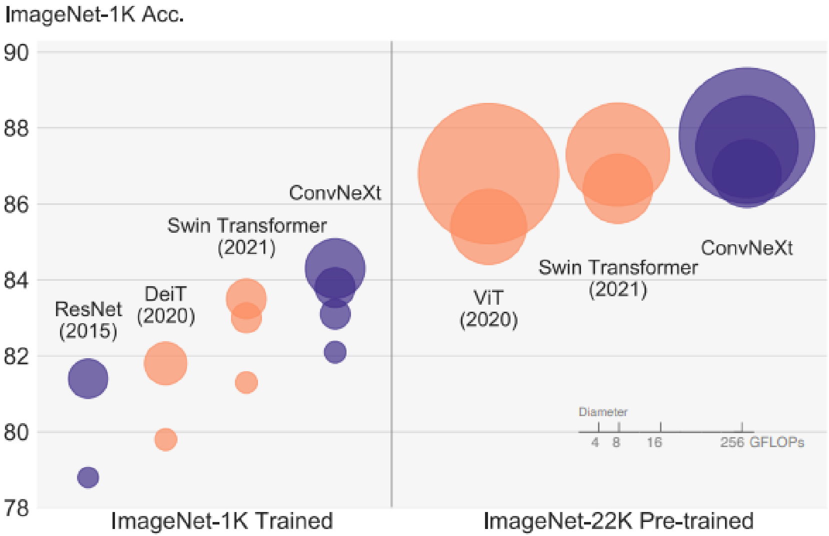
capturing both high and low frequencies. ViT mainly including global shapes and structures of a scene or object, but are not very powerful for learning high-frequencies,mainly including local edges and textures.

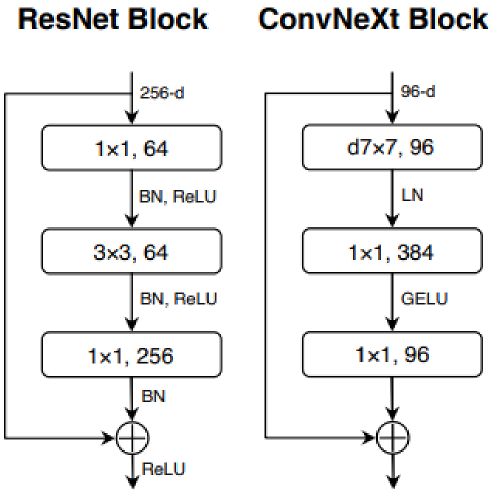


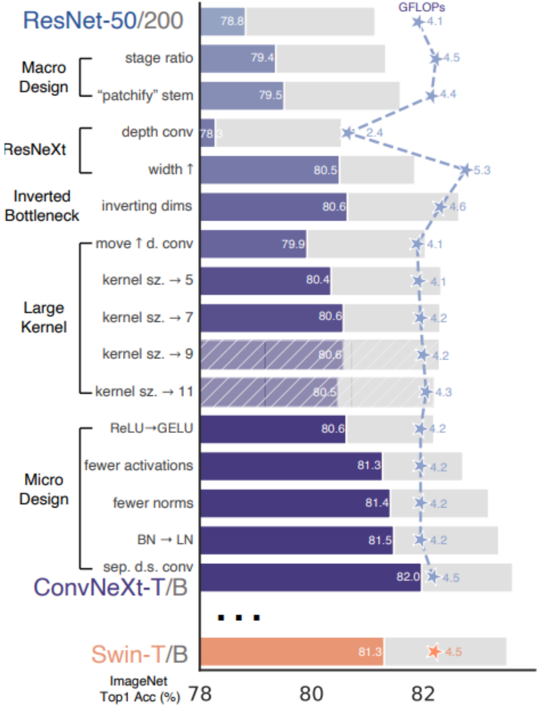
high-frequency max-pooling operation and convolution

low-frequency mixer is implemented by a vanilla self-attention in ViTs.

2.Related Works - ConvNeXt : A ConvNet for the 2020s (CVPR 2022)







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

Liu, Zhuang, et al. "A convnet for the 2020s." CVPR 2022.

3.Method – Motivation (InceptionNeXt)

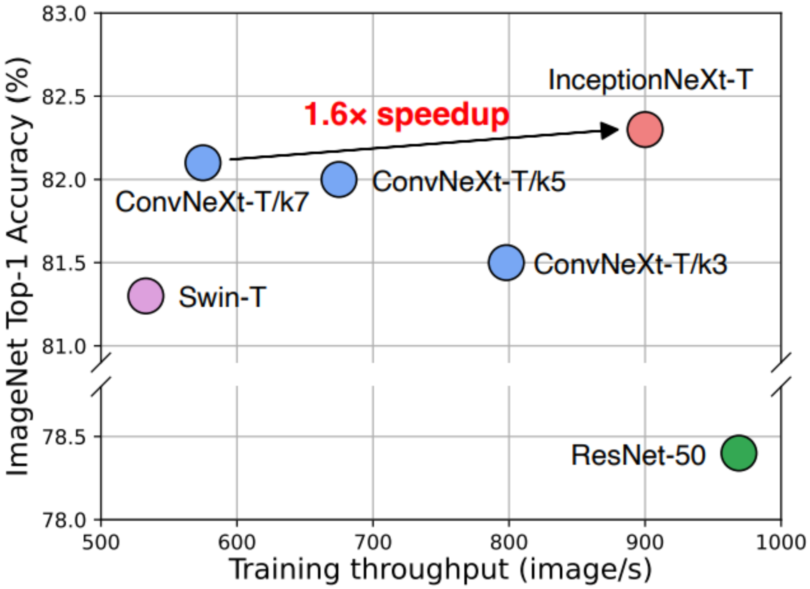
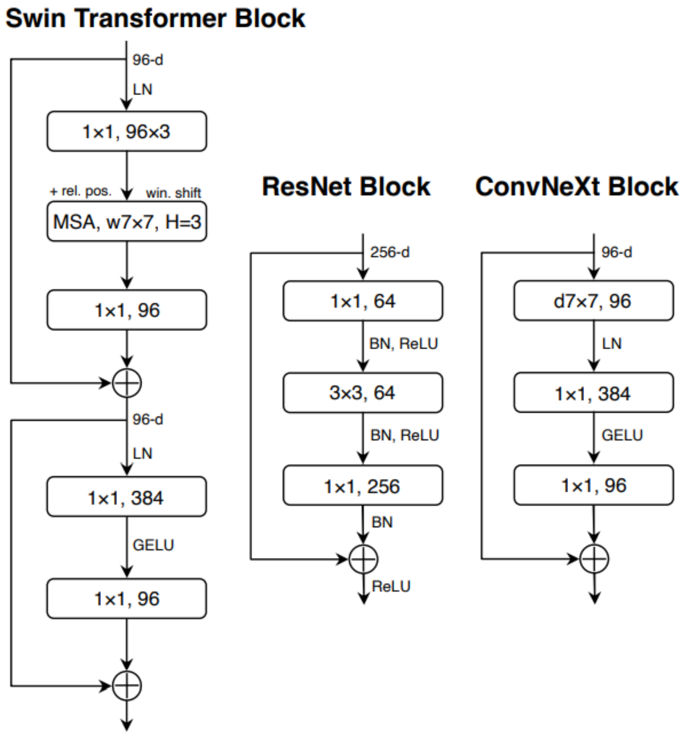
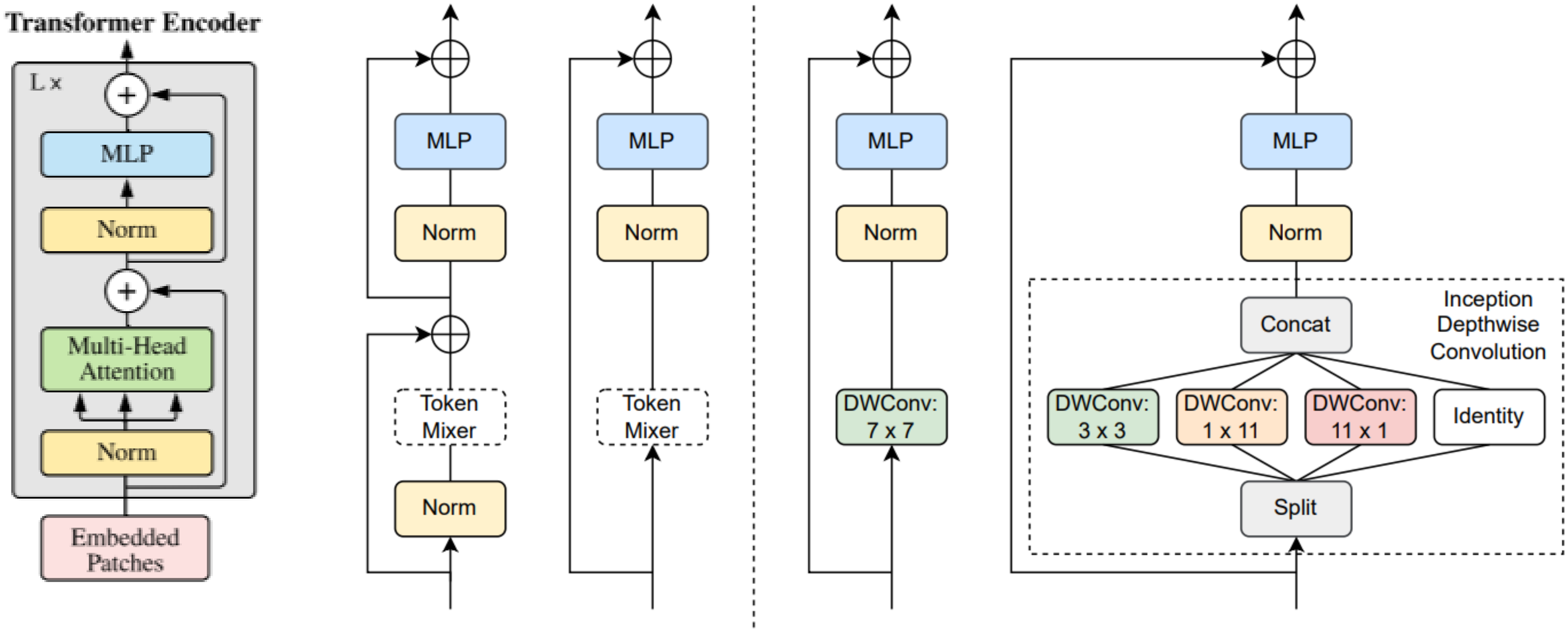


Figure 1: **Trade-off between accuracy and training throughput.** All models are trained under the DeiT training hyperparameters [61, 37, 38, 69]. The training throughput is measured on an A100 GPU with batch size of 128. ConvNeXt-T/kn means variants with depthwise convolution kernel size of $n \times n$. **InceptionNeXt-T enjoys both ResNet-50's speed and ConvNeXt-T's accuracy.**

Inspired by the long-range modeling ability of ViTs, large-kernel convolutions are widely adopted. Although such depthwise operator only consumes a few FLOPs, it largely harms the model efficiency on powerful computing devices due to the high memory access costs.

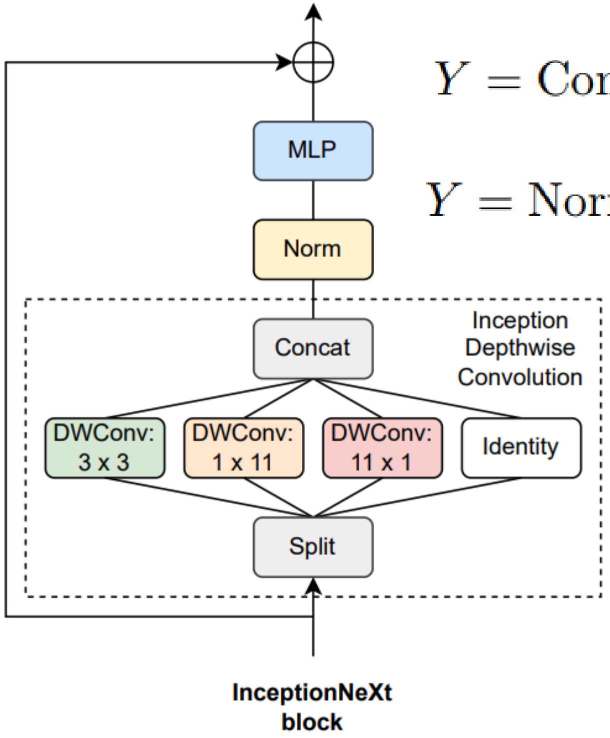


3.Method - Block illustration of InceptionNeXt and others



Vision Transformer **MetaFormer block** **MetaNeXt block** **ConvNeXt block** **InceptionNeXt block**

4. Experiments - Cross-domain FAS Performance



$$Y = \mathrm{Conv}_{1\times 1}^{rC\rightarrow C}\{\sigma[\mathrm{Conv}_{1\times 1}^{C\rightarrow rC}(Y)]\} + X$$

$$Y = \mathrm{Norm}(X')$$

$$X' = \mathrm{Concat}(X'_{\mathrm{hw}}, X'_{\mathrm{w}}, X'_{\mathrm{h}}, X'_{\mathrm{id}})$$

$$\begin{aligned} X'_{\mathrm{hw}} &= \mathrm{DWConv}_{k_s\times k_s}^{g\rightarrow g} g(X_{\mathrm{hw}}), \\ X'_{\mathrm{w}} &= \mathrm{DWConv}_{1\times k_b}^{g\rightarrow g} g(X_{\mathrm{w}}), \\ X'_{\mathrm{h}} &= \mathrm{DWConv}_{k_b\times 1}^{g\rightarrow g} g(X_{\mathrm{h}}), \\ X'_{\mathrm{id}} &= X_{\mathrm{id}}. \end{aligned}$$

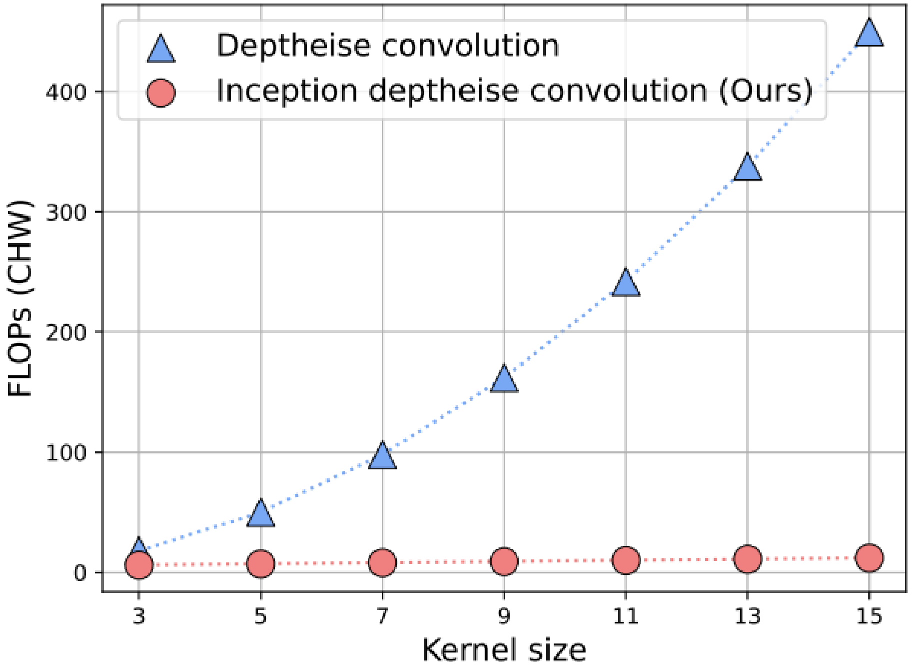
$$\begin{aligned} X_{\mathrm{hw}}, X_{\mathrm{w}}, X_{\mathrm{h}}, X_{\mathrm{id}} &= \mathrm{Split}(X) \\ &= X_{:,:g}, X_{:g:2g}, X_{:2g:3g}, X_{:3g:} \end{aligned}$$

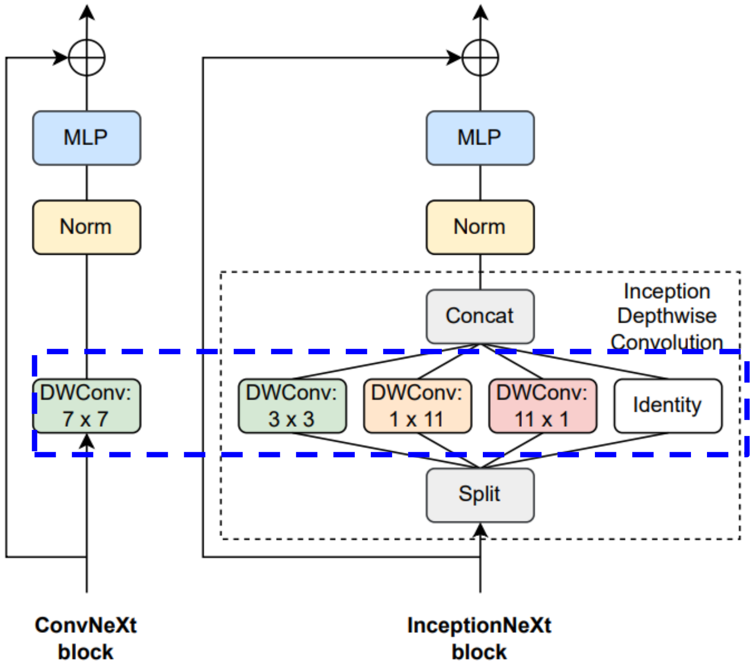
$$X' = \mathrm{TokenMixer}(X) = \mathrm{DWConv}_{k\times k}^{C\rightarrow C}(X)$$

3.Method - Complexity of different types of convolution

Conv. type	Params	FLOPs
Conventional conv.	k^2C^2	$2k^2C^2HW$
Depthwise conv.	k^2C	$2k^2CHW$
Inception dep. conv.	$\boxed{(2k+9)C/8}$	$(2k+9)CHW/4$

K=11
11x1 + 11x1 = 2k
3x3 = 9
Identy = 0





4. Experiments - Performance of models trained on ImageNet-1K

Model	Mixing Type	Image (size)	Params (M)	MACs (G)	Throughput (img/second) Train	Inference	Top-1 (%)
DeiT-S [61]	Attn	224^2	22	4.6	1227	3781	79.8
T2T-ViT-14 [76]	Attn	224^2	22	4.8	–	–	81.5
TNT-S [18]	Attn	224^2	24	5.2	–	–	81.5
Swin-T [37]	Attn	224^2	29	4.5	564	1768	81.3
Focal-T [73]	Attn	224^2	29	4.9	–	–	82.2
ResNet-50 [20, 69]	Conv	224^2	26	4.1	969	3149	78.4
RSB-ResNet-50 [20, 69]	Conv	224^2	26	4.1	969	3149	79.8
RegNetY-4G [46, 69]	Conv	224^2	21	4.0	670	2694	81.3
FocalNet-T [72]	Conv	224^2	29	4.5	–	–	82.3
ConvNeXt-T [38]	Conv	224^2	29	4.5	575	2413 (1943)	82.1
InceptionNeXt-T (Ours)	Conv	224^2	28	4.2	901 (+57%)	2900 (+20%)	82.3 (+0.2)
T2T-ViT-19 [76]	Attn	224^2	39	8.5	–	–	81.9
PVT-Medium [65]	Attn	224^2	44	6.7	–	–	81.2
Swin-S [37]	Attn	224^2	50	8.7	359	1131	83.0
Focal-S [73]	Attn	224^2	51	9.1	–	–	83.5
RSB-ResNet-101 [20, 69]	Conv	224^2	45	7.9	620	2057	81.3
RegNetY-8G [46, 69]	Conv	224^2	39	8.0	689	1326	82.1
FocalNet-S [72]	Conv	224^2	50	8.7	–	–	83.5
ConvNeXt-S [38]	Conv	224^2	50	8.7	361	1535 (1275)	83.1
InceptionNeXt-S (Ours)	Conv	224^2	49	8.4	521 (+44%)	1750 (+14%)	83.5 (+0.4)
RSB-ResNet-152 [20, 69]	Conv	224^2	60	11.6	437	1457	81.8
RegNetY-16G [46, 69]	Conv	224^2	84	15.9	322	1100	82.2
RepLKNet-31B [13]	Conv	224^2	79	15.3	–	–	83.5
FocalNet-B [72]	Conv	224^2	89	15.4	–	–	83.9
ConvNeXt-B [38]	Conv	224^2	89	15.4	267	1122 (969)	83.8
InceptionNeXt-B (Ours)	Conv	224^2	87	14.9	375 (+40%)	1244 (+11%)	84.0 (+0.2)
ViT-Base/16 [16]	Attn	384^2	87	55.4	130	359	77.9
DeiT-B [61]	Attn	384^2	86	55.4	131	361	83.1
Swin-B [37]	Attn	384^2	88	47.1	104	296	84.5
RepLKNet-31B [13]	Conv	384^2	79	45.1	–	–	84.8
ConvNeXt-B [38]	Conv	384^2	89	45.0	95	393 (337)	85.1
InceptionNeXt-B (Ours)	Conv	384^2	87	43.6	139 (+46%)	428 (+9%)	85.2 (+0.1)

Fairly compared with the widely-used baselines. (Swin and ConvNeXt)

The throughputs are measured on an A100 GPU with batch size of 128 and full precision (FP32).

The numbers in gray color are reported by ConvNeXt paper. (Inference – image/second)

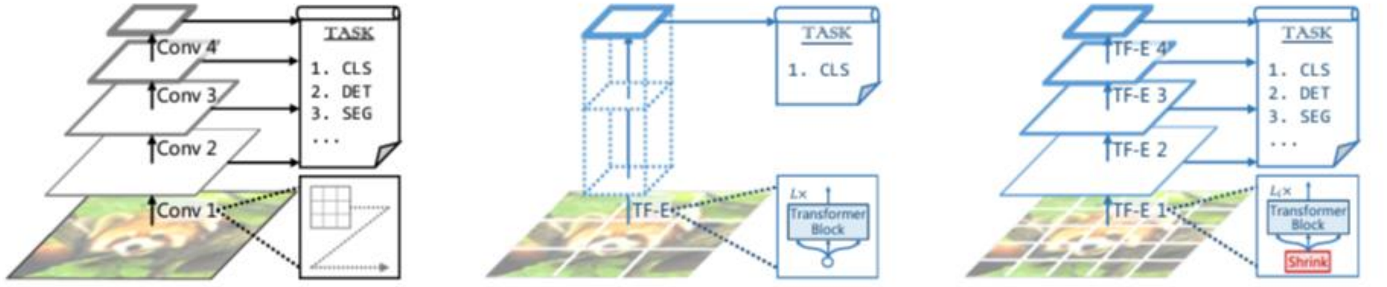
1.6×/1.2× training/inference throughputs than ConvNeXts.

4. Experiments – isotropic architecture

Model	Params (M)	MACs (G)	Top-1 (%)
DeiT-S [61]	22	4.6	**79.8**
MetaNeXt-Attn	22	4.6	3.9
ConvNeXt-S (*iso.*) [38]	22	4.3	79.7
InceptionNeXt-S (*iso.*)	22	4.2	79.7
DeiT-B [61]	87	17.6	81.8
ConvNeXt-S (*iso.*) [38]	87	16.9	82.0
InceptionNeXt-S (*iso.*)	86	16.8	**82.1**

Table 3: **Comparison among ViT, isotropic ConvNeXt and InceptionNeXt.** MetaNeXt-Attn is instantiated from MetaNeXt with token mixer of self-attention [63].

Besides the 4-stage framework, another notable one is ViT-style isotropic architecture which has only one stage. To match the parameters and MACs of DeiT, we construct InceptionNeXt (iso.) following ConvNeXt.

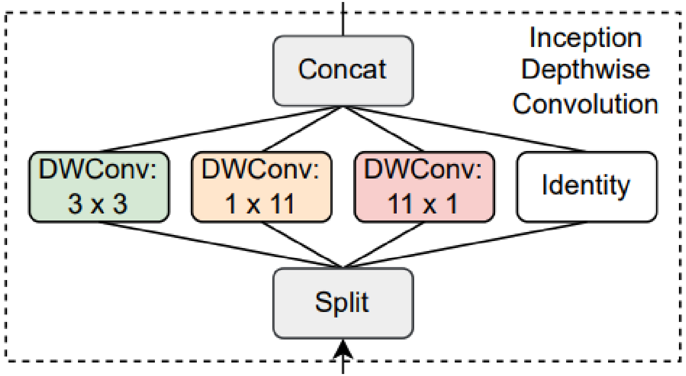


(a) CNNs: VGG [54], ResNet [22], *etc.*   (b) Vision Transformer [13]   (c) Pyramid Vision Transformer (ours)

Wang, Wenhai, et al. "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions." ICCV 2021.

4. Experiments - Ablation for InceptionNeXt on ImageNet-1K

Ablation	Variant	Params (M)	MACs (G)	Throughput Train	Throughput Inference	Top-1 (%)
Baseline	None (InceptionNeXt-T)	28.1	4.2	901	2900	82.3
Branch	Remove horizontal band kernel	28.0	4.2	947	3093	81.9
	Remove vertical band kernel	28.0	4.2	954	3173	81.9
	Remove small band kernel	28.0	4.2	940	3004	82.0
	horizontal and vertical band kernel in parallel → in sequence	28.1	4.2	903	2971	82.1
Band kernel size	Band kernel size 11 → 7	28.0	4.2	905	2946	82.1
	Band kernel size 11 → 9	28.1	4.2	904	2916	82.1
	Band kernel size 11 → 13	28.1	4.2	896	2895	82.0
Convolution branch ratio	Conv. branch ratio 1/8 → 1/4	28.1	4.2	834	2499	82.2
	Conv. branch ratio 1/8 → 1/16	28.0	4.2	936	3097	81.8




```
class InceptionDWConv2d(nn.Module):
    def __init__(self, in_channels,
        square_kernel_size=3, band_kernel_size=11,
        branch_ratio=1/8):
        super().__init__()

        gc = int(in_channels * branch_ratio) # channel
            number of a convolution branch

    self.split_indexes = (gc, gc, gc, in_channels
        - 3 * gc)
```

4. Experiments - Semantic segmentation

Backbone	UperNet		
	Params (M)	MACs (G)	mIoU (%)
Swin-T [37]	60	945	45.8
ConvNeXt-T [38]	60	939	46.7
InceptionNeXt-T	56	933	**47.9**
Swin-S [37]	81	1038	49.5
ConvNeXt-S [38]	82	1027	49.6
InceptionNeXt-S	78	1020	**50.0**
Swin-B [37]	121	1188	49.7
ConvNeXt-B [38]	122	1170	49.9
InceptionNeXt-B	115	1159	**50.6**

Table 5: **Performance of Semantic segmentation with UperNet [70] on ADE20K [84] validation set.** Images are cropped to 512×512 for training. The MACs are measured with input size of 512×2048.

Backbone	Semantic FPN		
	Params (M)	MACs (G)	mIoU (%)
ResNet-50 [20]	29	46	36.7
PVT-Small [65]	28	45	39.8
PoolFormer-S24 [74]	23	39	40.3
InceptionNeXt-T	28	44	**43.1**
ResNet-101 [20]	48	65	38.8
ResNeXt-101-32x4d [71]	47	65	39.7
PVT-Medium [65]	48	61	41.6
PoolFormer-S36 [74]	35	48	42.0
PoolFormer-M36 [74]	60	68	42.4
InceptionNeXt-S	50	65	**45.6**
PVT-Large [65]	65	80	42.1
ResNeXt-101-64x4d [71]	86	104	40.2
PoolFormer-M48 [74]	77	82	42.7
InceptionNeXt-B	85	100	**46.4**

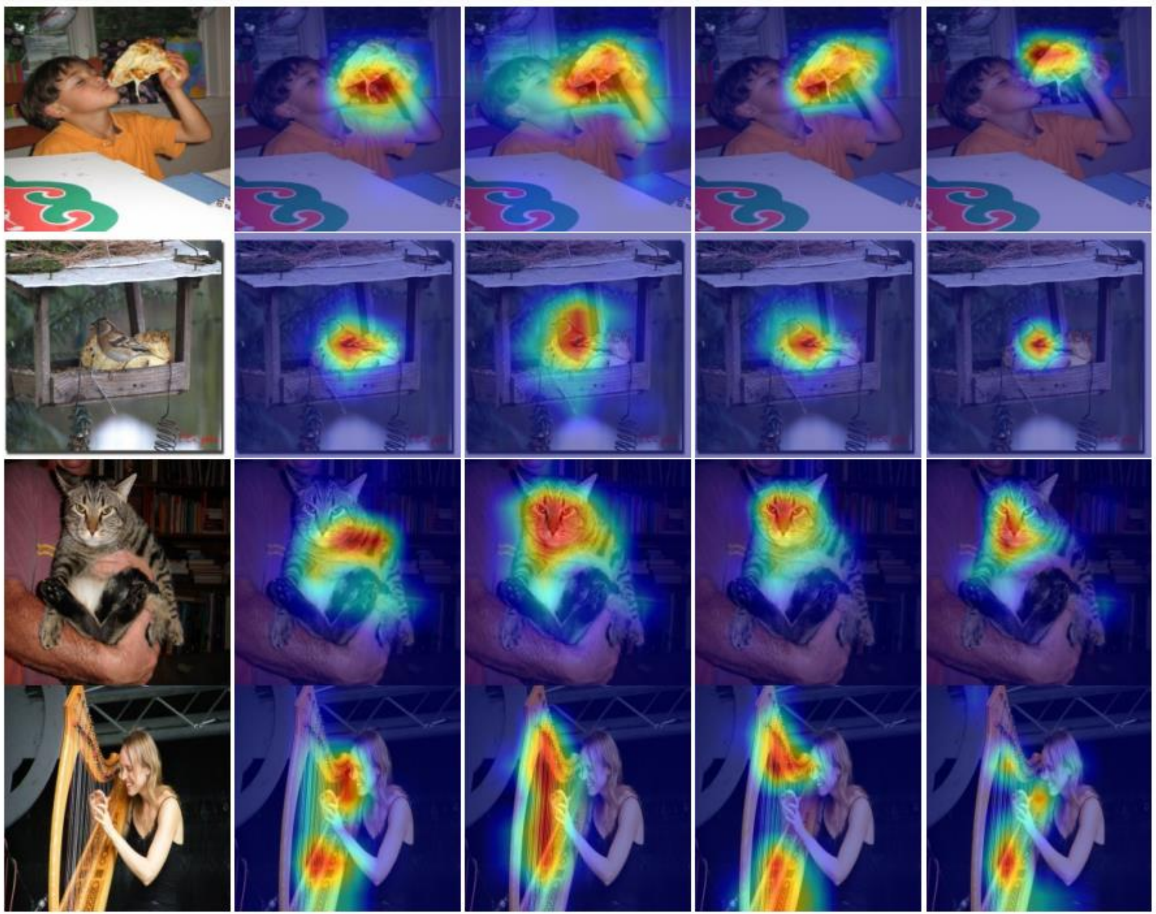
Table 6: **Performance of Semantic segmentation with Semantic FPN [29] on ADE20K [84] validation set.** Images are cropped to 512×512 for training. The MACs are measured with input size of 512×512.

4. Experiments - Preliminary experiments based on ConvNeXt-T

Kernel size of DWConv	Convolution ratio	Params (M)	MACs (G)	Throughput		Top-1 (%)
				Train	Inference	
7×7	1.0	28.6	4.5	575	2413	82.1*
5×5	1.0	28.4	4.4	675	2704	82.0
3×3	1.0	28.3	4.4	798	2802	81.5
3×3	$1/2$	28.3	4.4	818	2740	81.4
3×3	$3/8$	28.3	4.4	847	2762	81.4
3×3	$1/4$	28.3	4.4	871	2808	81.3
3×3	$1/8$	28.3	4.4	901	2833	80.8
3×3	$1/16$	28.3	4.4	916	2846	80.1

Table 10: **Preliminary experiments based on ConvNeXt-T.** Convolution ratio means the ratio of channels to be processed by depthwise convolution while the other channels keep unchanged. Throughputs are measured on an A100 GPU with batch size of 128 and full precision (FP32). * The result is reported in ConvNeXt paper [38].

4. Experiments - Qualitative results



Input RSB-ResNet-50 [20, 69] Swin-T [61] ConvNeXt-T [38] InceptionNeXt-T

4. Experiments - Configurations of InceptionNeXt models.

Stage	#Tokens	Layer Specification		InceptionNeXt		
				T	S	B
1	$\frac{H}{4} \times \frac{W}{4}$	Down-sampling	Kernel Size	4×4, stride 4		
			Embed. Dim.	96		128
		InceptionNeXt Block	Kernel size	3×3, 1×11, 11×1		
			Conv. group ratio	1/8		
			MLP Ratio	4		
			# Block	3		
2	$\frac{H}{8} \times \frac{W}{8}$	Down-sampling	Kernel Size	2×2, stride 2		
			Embed. Dim.	192		256
		InceptionNeXt Block	Kernel size	3×3, 1×11, 11×1		
			Conv. group ratio	1/8		
			MLP Ratio	4		
			# Block	3		
3	$\frac{H}{16} \times \frac{W}{16}$	Down-sampling	Kernel Size	2×2, stride 2		
			Embed. Dim.	384		512
		InceptionNeXt Block	Kernel size	3×3, 1×11, 11×1		
			Conv. group ratio	1/8		
			MLP Ratio	4		
			# Block	9	27	
4	$\frac{H}{32} \times \frac{W}{32}$	Down-sampling	Kernel Size	2×2, stride 2		
			Embed. Dim.	768		1024
		InceptionNeXt Block	Kernel size	3×3, 1×11, 11×1		
			Conv. group ratio	1/8		
			MLP Ratio	3		
			# Block	3		
Global average pooling, MLP						
Parameters (M)				4.2	8.4	14.9
MACs (G)				28.1	49.4	86.7

InceptionNeXt has similar model configurations to Swin and ConvNeXt.

5.Conclusion

(+) It is an effective and efficient CNN architecture that enjoys a better trade-off between the practical speed and the performance than previous network architectures.

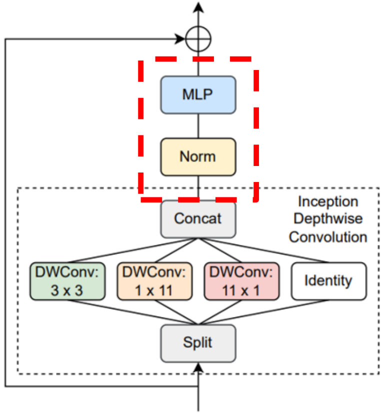
(+) It is noticed the speed-up ratios of InceptionNeXt in inference is smaller than that during training.

(+) Extensive experimental results demonstrate the superior performance and the high practical efficiency

(-) What if InceptionNeXt applied to other architecture(e.g, Swin, ConvFormer) instead of just ConvNeXt?"

(-) Compared for Semantic segmentation with UperNet and Semantic FPN, but there are no experiments on Object Detection.

(-) This study is focused only on the token mixer, but how about reviewing it from the perspective of the overall architecture as well, such as MLP or other modules?



InceptionNeXt block

Thanks

Any Questions?

You can send mail to
Susang Kim(healess1@gmail.com)