

Deep learning

Transformers

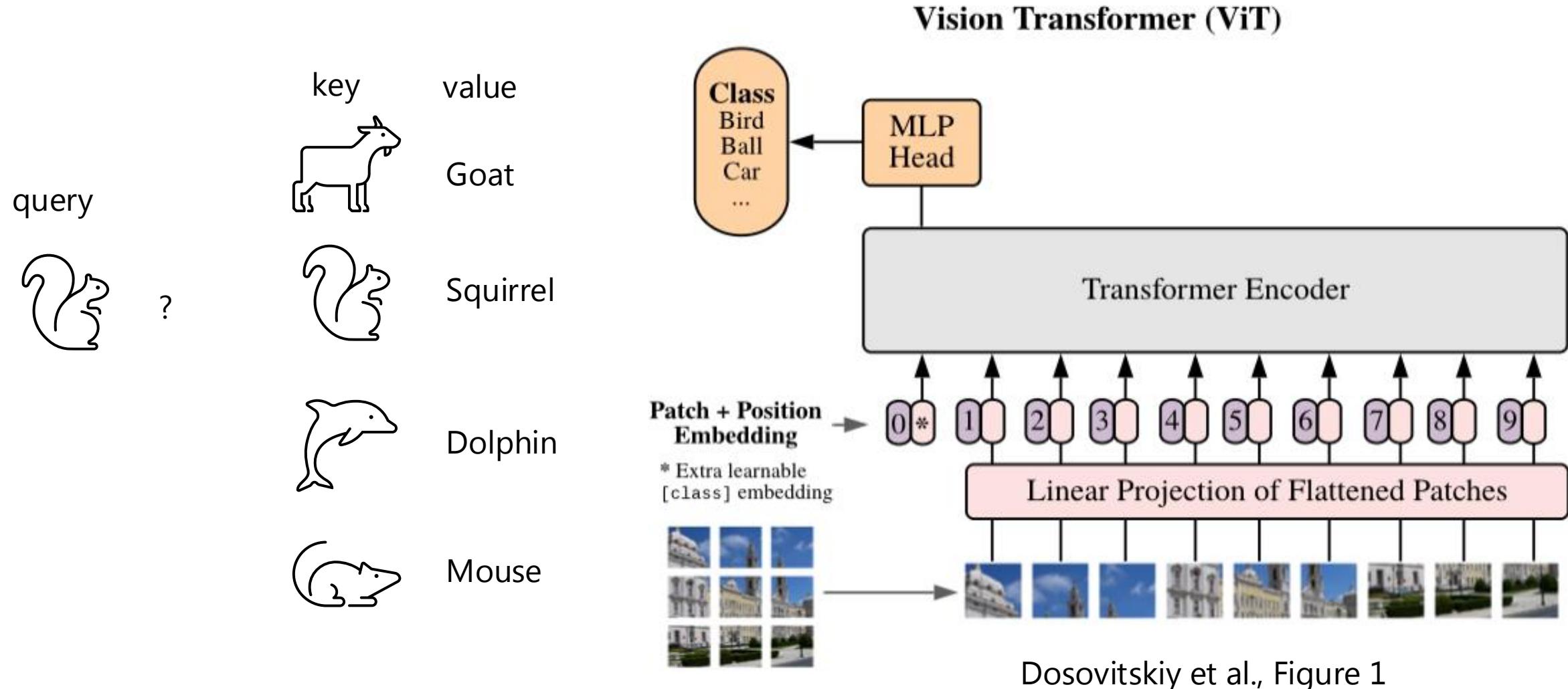
2024/12/18

Jon Sporring,
Department of Computer Science

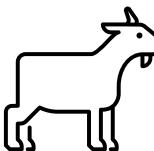
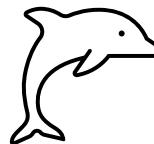
UNIVERSITY OF COPENHAGEN

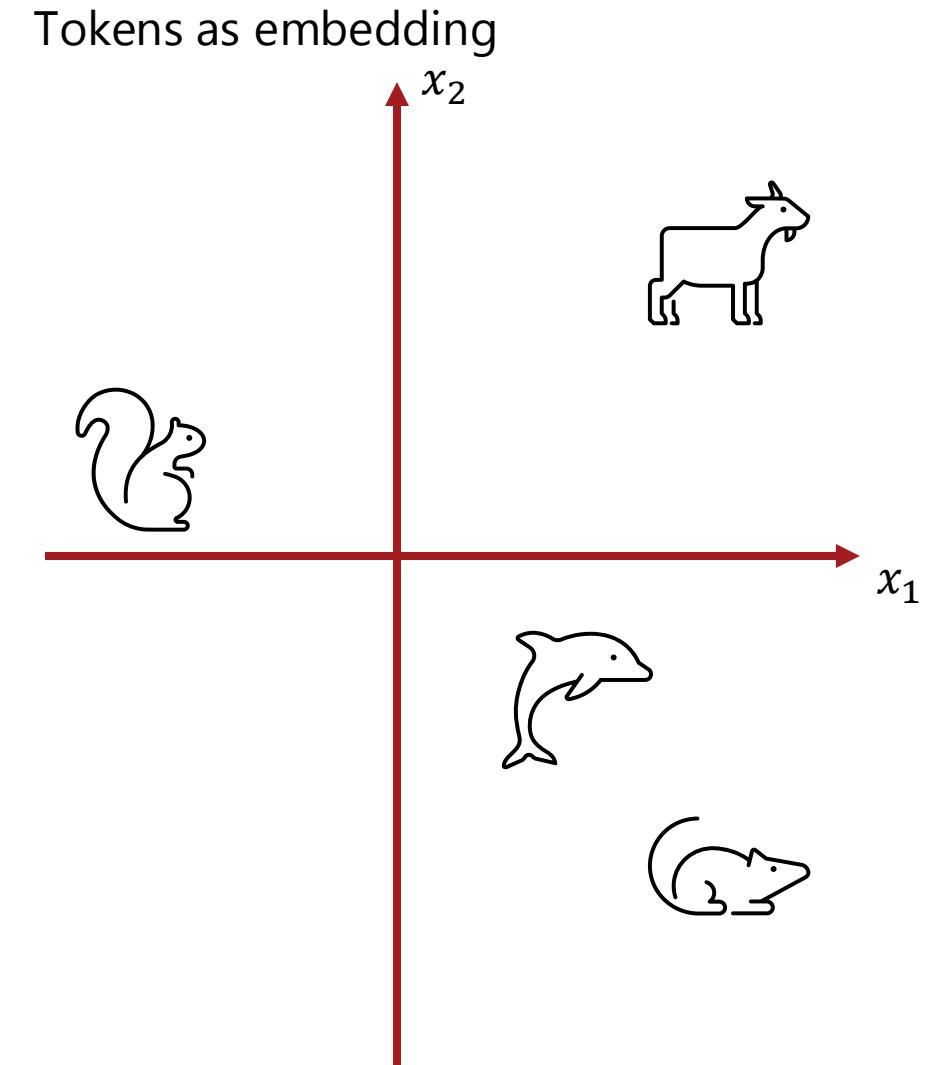


Transformer networks – Vision Transformer ViT: Vaswani ea (2017) & Dosovitskiy ea (2020)

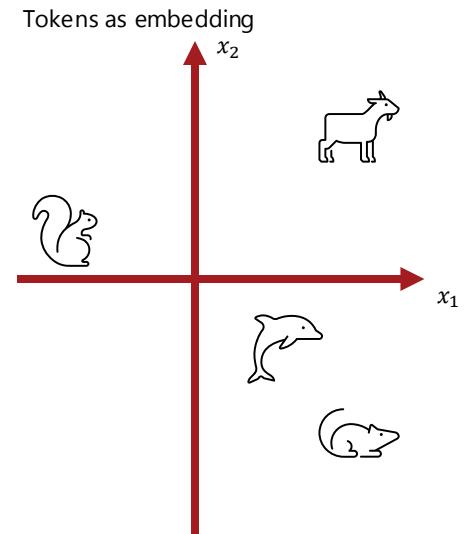
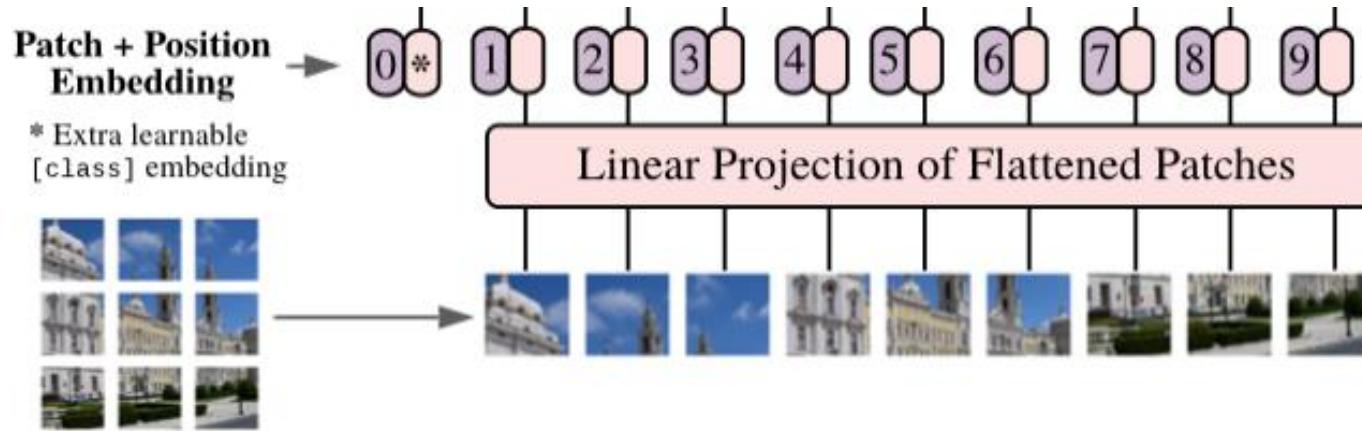


Tokens: Cut into patches, flatten and linearly embed in lower dimensional space

	key	value
query		
	 ? 	Goat
		Squirrel
		Dolphin
		Mouse



Embeddings++



$$I \in \mathbb{R}^{H \times W \times C}, I_p \in \mathbb{R}^{M \times M \times C}, \text{Flatten: } \mathbb{R}^{M \times M \times C} \rightarrow \mathbb{R}^{M^2 C}, \text{Token: } \mathbb{R}^{M^2 C} \rightarrow \mathbb{R}^D$$

Convention: row vectors

$$\mathbf{x}_j = \text{Flatten}(I_j) \in \mathbb{R}^{M^2 C}$$

$$\mathbf{t}_j = \text{Token}(\mathbf{x}_j) = \mathbf{x}_j \mathbf{E} \in \mathbb{R}^D$$

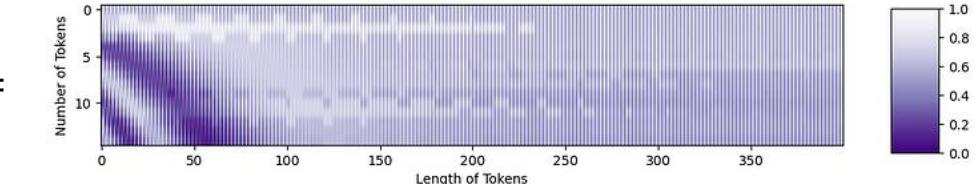
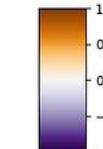
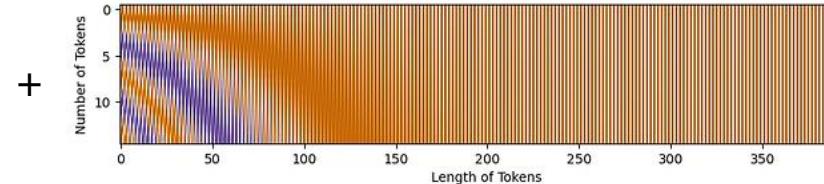
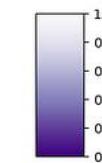
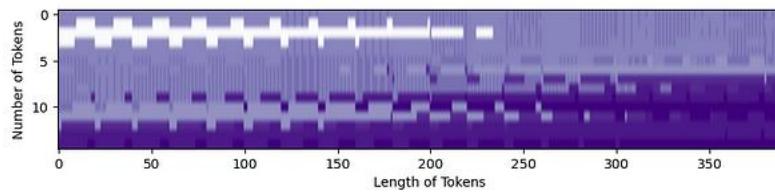
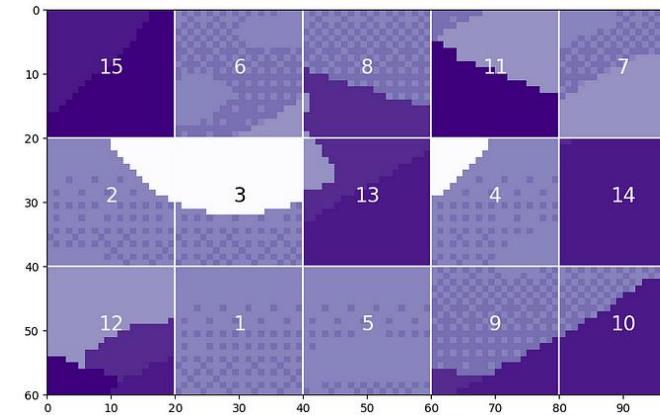
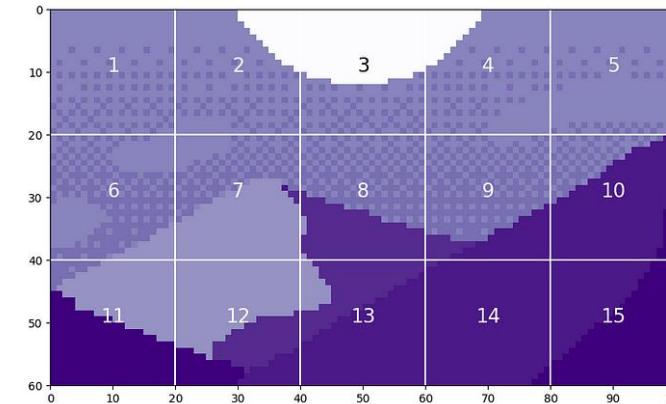
$$\mathbf{z}^0 = [\mathbf{x}_{\text{class}}; \mathbf{t}_1; \mathbf{t}_2; \dots; \mathbf{t}_N] + \text{PositionEmbedding}() \in \mathbb{R}^{(N+1) \times D}$$

Positional Embedding: Vision depends on position

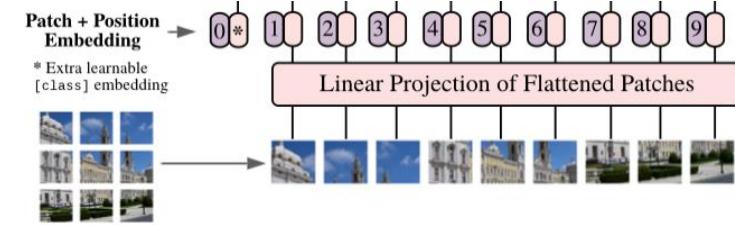
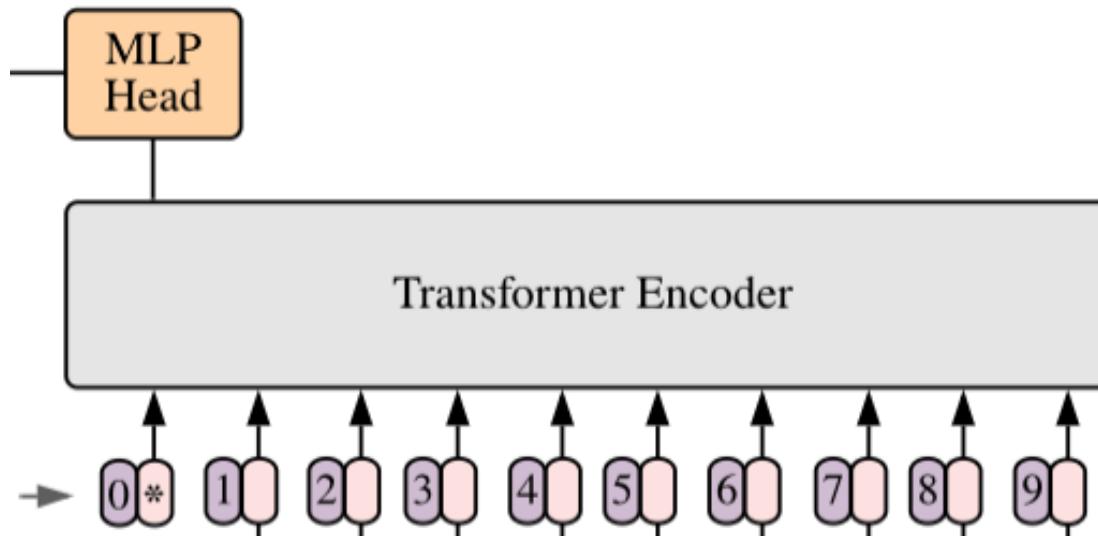
Yuan et al (2021). *Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet*

Figures from

<https://towardsdatascience.com/position-embeddings-for-vision-transformers-explained-a6f9add341d5>



Encoding



A sequence of attention layers: $l = 1 \dots L$

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{t}_1; \mathbf{t}_2; \dots; \mathbf{t}_N] \in \mathbb{R}^{(N+1) \times D}$$

$$\mathbf{z}'_l = \text{MSA}(\text{LN}(\mathbf{z}'_{l-1})) + \mathbf{z}'_{l-1} \in \mathbb{R}^{(N+1) \times D}$$

$$\mathbf{z}_l = \text{MLP}_2(\text{LN}(\mathbf{z}_l)) + \mathbf{z}'_l \in \mathbb{R}^{(N+1) \times D}$$

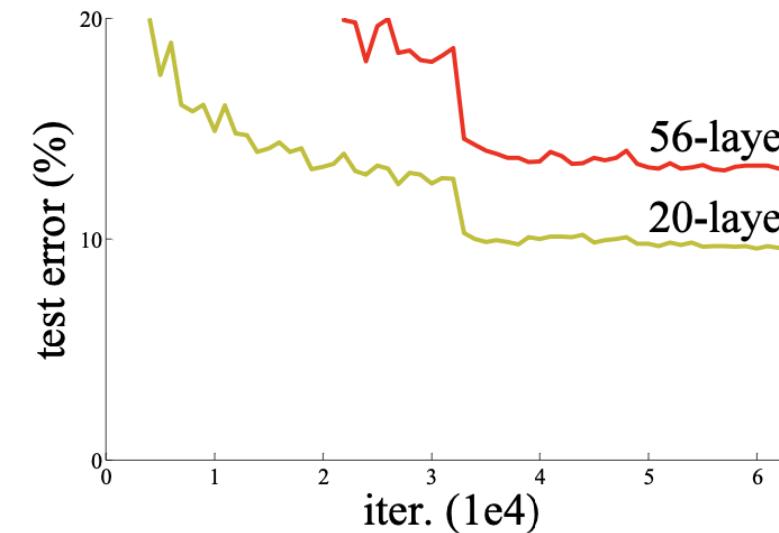
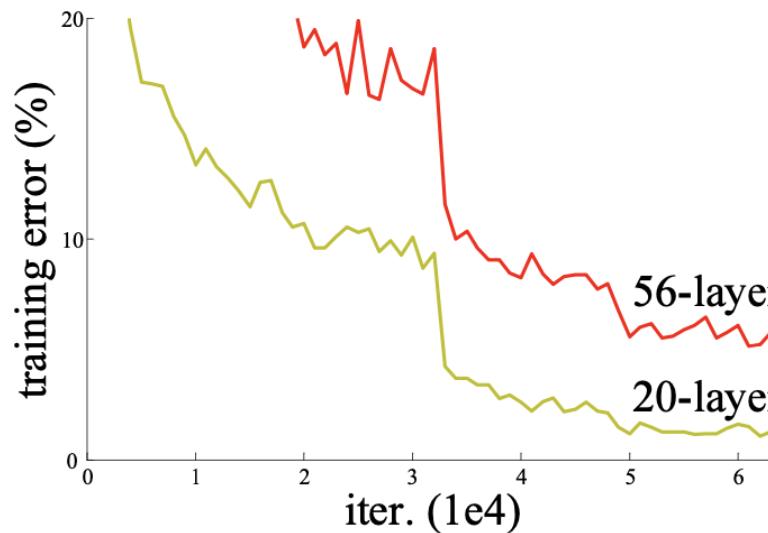
MSA: Multiheaded self-attention
 MLP_n: n-layer Multilayer perceptron
 LN: Layer normalization

Residual learning

Residual learning for optimizing deeper networks

He et al, "Deep Residual Learning for Image Recognition, CVPR 2016

Observation: Deeper networks = poorer convergence



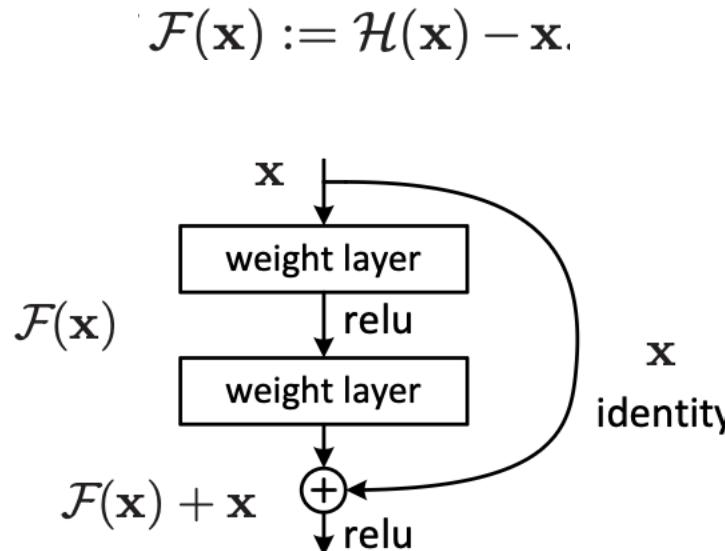
He et al. Figure 1

Residual learning for optimizing deeper networks

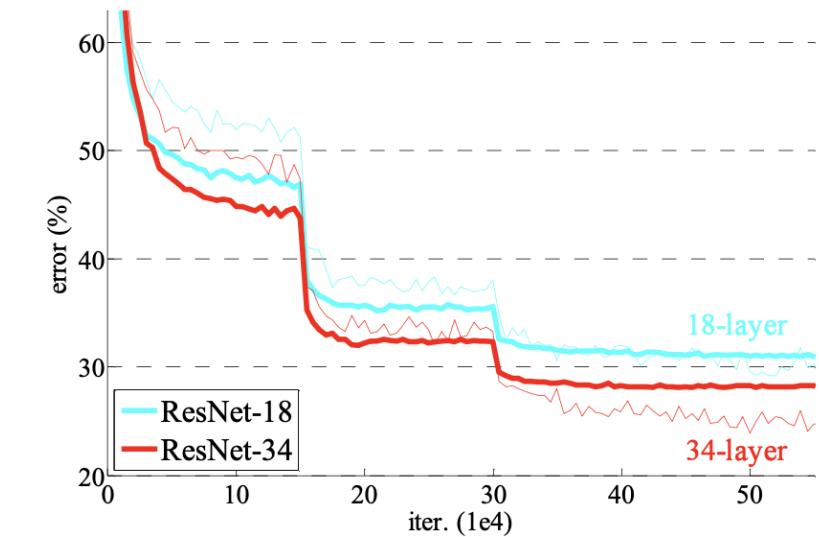
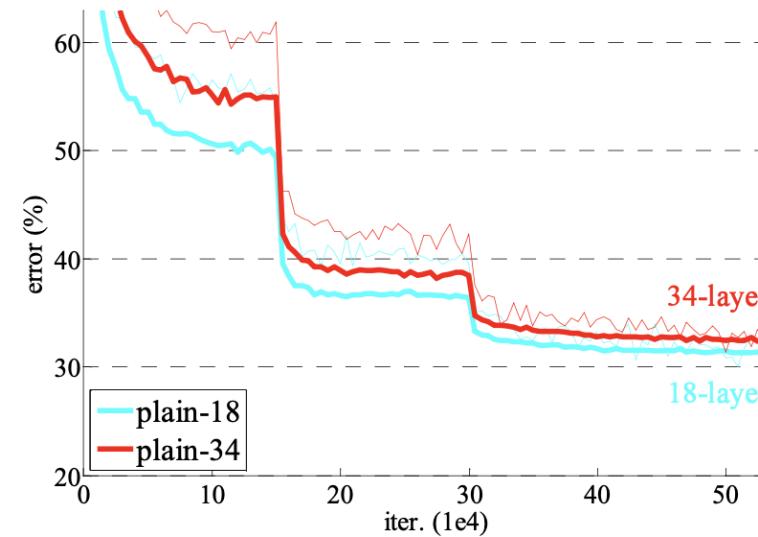
He et al, "Deep Residual Learning for Image Recognition, CVPR 2016

Observation: Deeper networks = poorer convergence

Solution: Train on residuals



He et al. Figure 2



He et al. Figure 4

Multilayer perceptron (MLP) and Layer Normalization (LN)

Ba, Kiros, and Hinton, Layer normalization, 2016

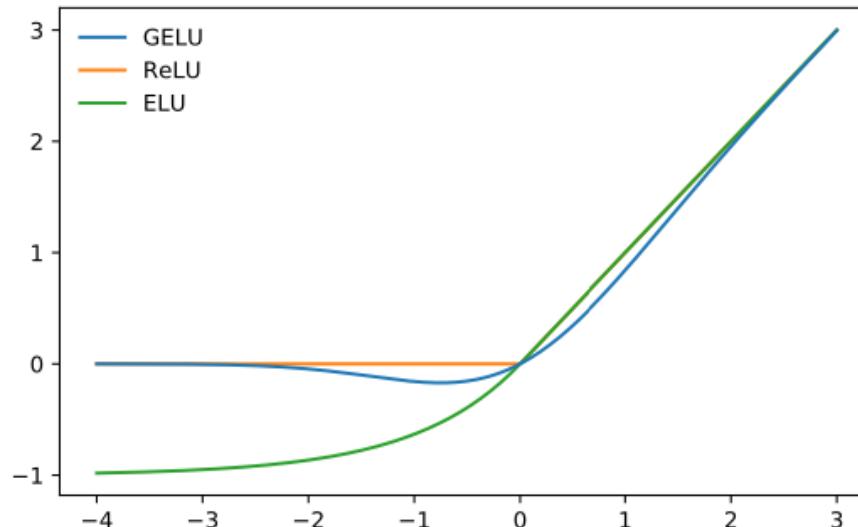
$$\text{LN}(\mathbf{z}) = \frac{\mathbf{z} - \bar{\mathbf{z}}}{\sqrt{\text{Var}(\mathbf{z}) + \epsilon}}$$

A sequence of attention layers: $l = 1 \dots L$

$$\begin{aligned}\mathbf{z}_0 &= [\mathbf{x}_{\text{class}}; \mathbf{t}_1; \mathbf{t}_2; \dots; \mathbf{t}_N] \in \mathbb{R}^{(N+1) \times D} \\ \mathbf{z}'_l &= \text{MSA}(\text{LN}(\mathbf{z}'_{l-1})) + \mathbf{z}'_{l-1} \in \mathbb{R}^{(N+1) \times D} \\ \mathbf{z}_l &= \text{MLP}(\text{LN}(\mathbf{z}'_l)) + \mathbf{z}'_l \in \mathbb{R}^{(N+1) \times D}\end{aligned}$$

MLP: Single layer with two GELU

GELU: Hendryks & Gimpel, Gaussian Error Linear Units (GELU)

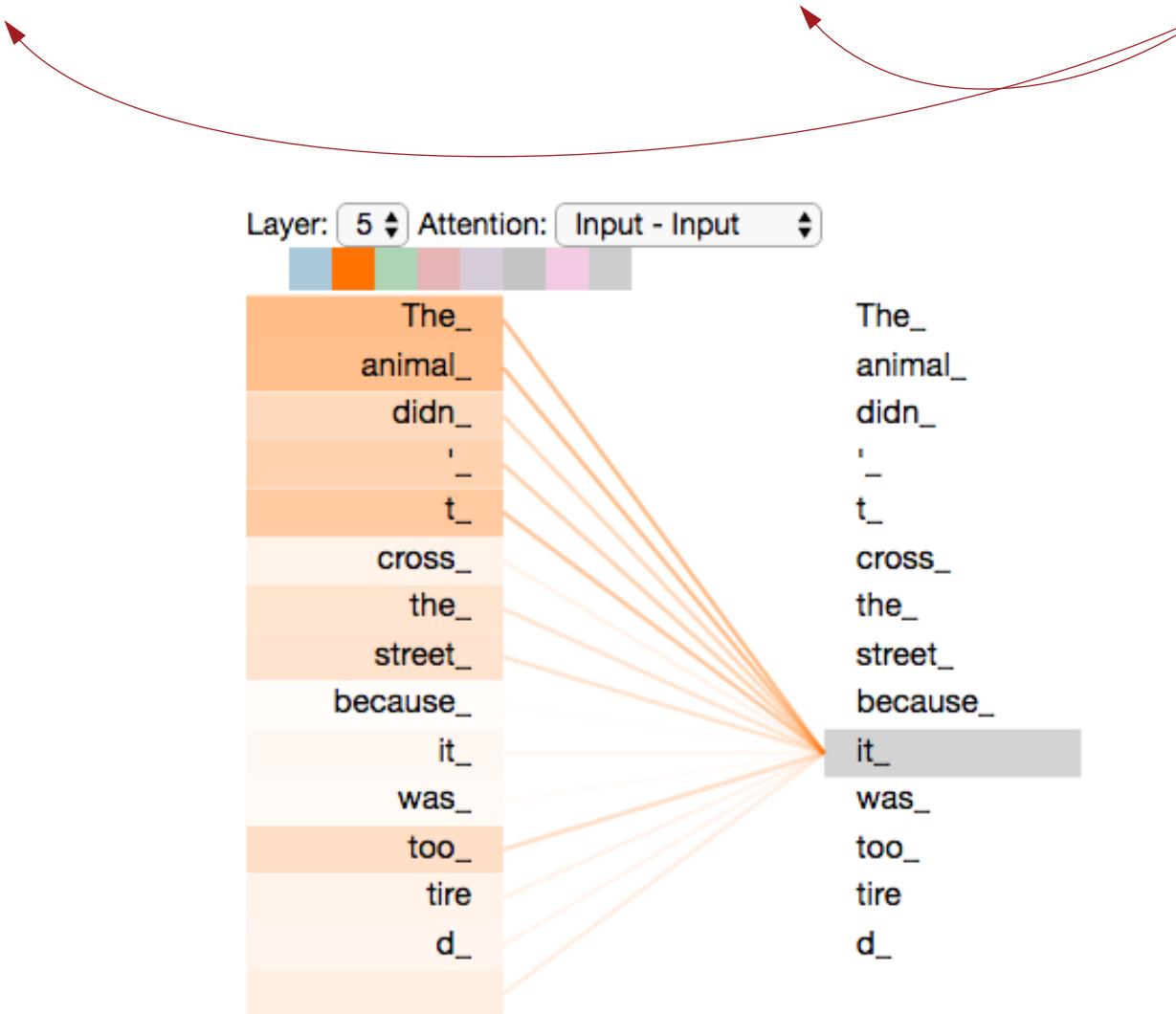


Hendryks & Gimpel, Figure 1

$$\begin{aligned}\text{GELU}(x) &= \frac{x}{2} \left[1 + \text{erf} \left(\frac{x}{\sqrt{2}} \right) \right] \\ &\simeq x/2(1 + \tanh \left((x + 0.044715x^3) \sqrt{\frac{2}{\pi}} \right)\end{aligned}$$

Attention:

The animal didn't cross the street because it was too tired



Self-attention head

$I \in \mathbb{R}^{H \times W \times C}, I_p \in \mathbb{R}^{M \times M \times C}$, Flatten: $\mathbb{R}^{M \times M \times C} \rightarrow \mathbb{R}^{M^2 C}$, Token: $\mathbb{R}^{M^2 C} \rightarrow \mathbb{R}^D$

$\mathbf{z} \in \mathbb{R}^{(N+1) \times D}$

Query $\mathbf{q}_j = \mathbf{z}_{j*} U_q \in \mathbb{R}^{D_h}$
 Key $\mathbf{k}_j = \mathbf{z}_{j*} U_k \in \mathbb{R}^{D_h}$
 Value $\mathbf{v}_j = \mathbf{z}_{j*} U_v \in \mathbb{R}^{D_h}$

$U_* \in \mathbb{R}^{D \times D_h}$ learnable

$$\mathbf{a}_{ij} = \text{softmax}\left(\frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{D_h}}\right)$$

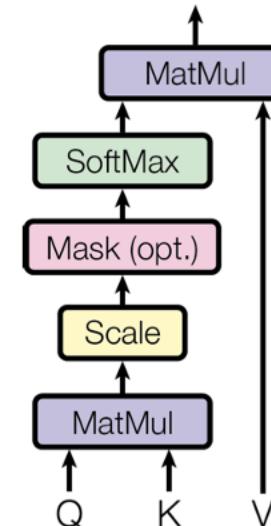
$$\text{softmax}(\{s_j\}) = \left\{ \frac{e^{s_j}}{\sum_{j=1}^n e^{s_j}} \right\}$$

$$\text{SA}(\mathbf{z}_{i*}) = \sum_j \mathbf{a}_{ij} \mathbf{v}_j \in \mathbb{R}^{D_h}$$

Query how one embedding (\mathbf{q}) matches keys (\mathbf{k}), which are all other embeddings.

Softmax select the best matching querys to their cosine difference

Output is now computed as a weighted sum of values (\mathbf{v}).



Vaswani, Fig 2

Multihead self-attention

$I \in \mathbb{R}^{H \times W \times C}, I_p \in \mathbb{R}^{M \times M \times C}$, Flatten: $\mathbb{R}^{M \times M \times C} \rightarrow \mathbb{R}^{M^2 C}$, Token: $\mathbb{R}^{M^2 C} \rightarrow \mathbb{R}^D$

$\mathbf{z} \in \mathbb{R}^{(N+1) \times D}$

Query	$Q = \{q_j\} \in \mathbb{R}^{(N+1) \times D_h}$	←
Key	$K = \{k_j\} \in \mathbb{R}^{(N+1) \times D_h}$	
Value	$V = \{v_j\} \in \mathbb{R}^{(N+1) \times D_h}$	

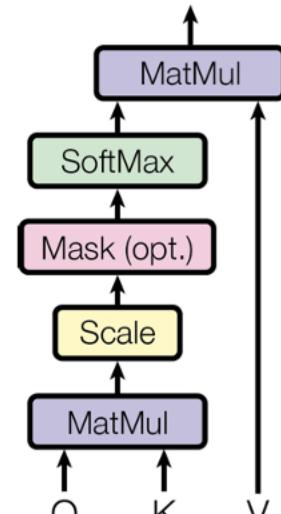
$U_* \in \mathbb{R}^{D \times D_h}$ learnable

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_h}}\right) \in \mathbb{R}^{(N+1) \times (N+1)}$$

$$\text{SA}(\mathbf{z}) = \mathbf{AV} \in \mathbb{R}^{(N+1) \times D_h}$$

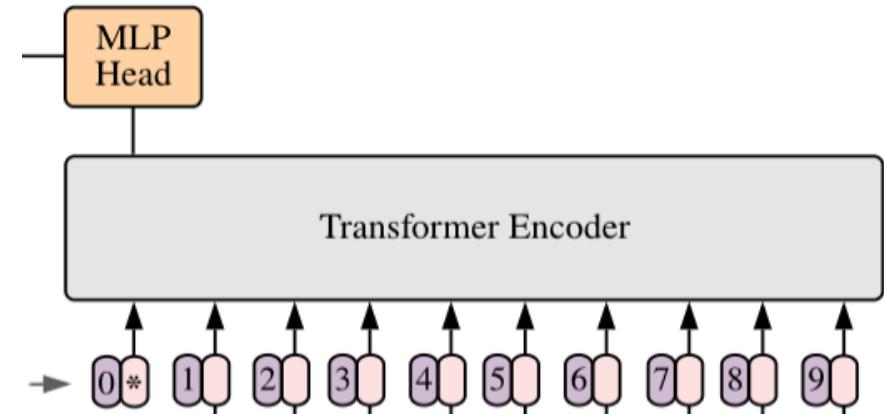
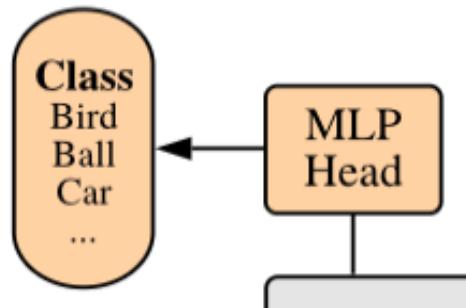
$$\mathbf{U}_{msa} \in \mathbb{R}^{kD_h \times D}$$

$$\text{MSA}(\mathbf{z}) = [\text{SA}_1(\mathbf{z}), \text{SA}_2(\mathbf{z}), \dots, \text{SA}_k(\mathbf{z})] \mathbf{U}_{msa} \in \mathbb{R}^{(N+1) \times D}$$



Vaswani, Fig 2

Encoding



A sequence of attention layers: $l = 1 \dots L$

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{t}_1; \mathbf{t}_2; \dots; \mathbf{t}_N] \in \mathbb{R}^{2(N+1) \times D}$$

$$\mathbf{z}'_l = \text{MSA}(\text{LN}(\mathbf{z}'_{l-1})) + \mathbf{z}'_{l-1} \in \mathbb{R}^{2(N+1) \times D}$$

$$\mathbf{z}_l = \text{MLP}_2(\text{LN}(\mathbf{z}'_l)) + \mathbf{z}'_l \in \mathbb{R}^{2(N+1) \times D}$$

$\text{Class} = \text{MLP}_1(\mathbf{z}_L)$

Inductive bias [Dosoviskiy]

Inductive bias. We note that Vision Transformer has much less image-specific inductive bias than CNNs. In CNNs, locality, two-dimensional neighborhood structure, and translation equivariance are baked into each layer throughout the whole model. In ViT, only MLP layers are local and translationally equivariant, while the self-attention layers are global. The two-dimensional neighborhood

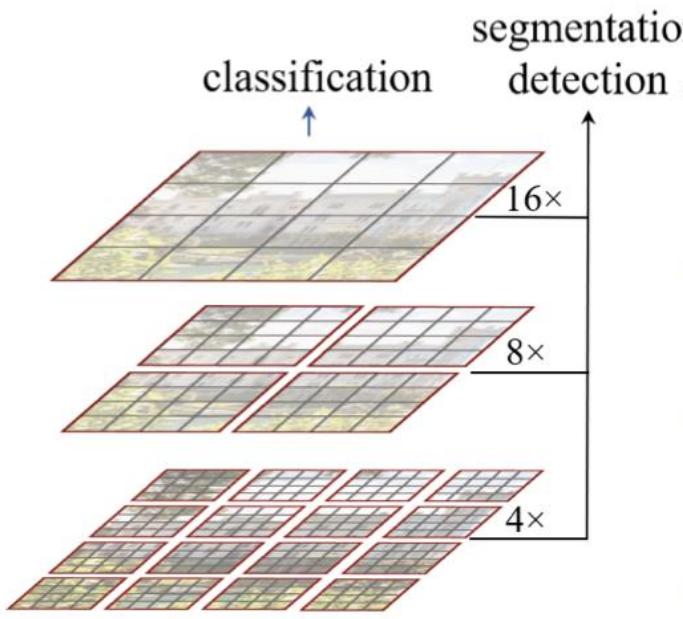
ImageNet: <https://image-net.org/>
14*10⁶ images, 2*10⁴ categories, 10⁶ images with
bounding boxes



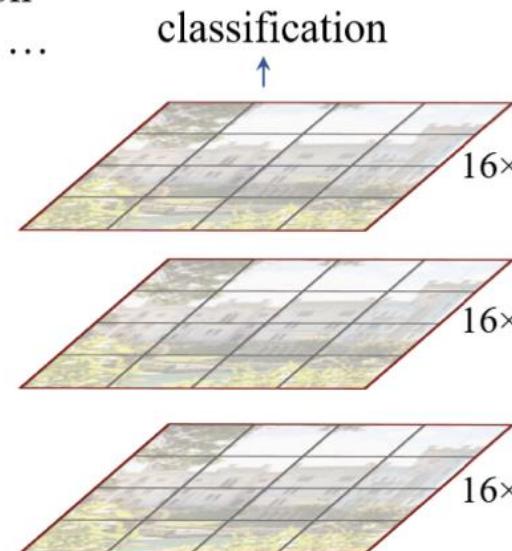
© 2020 Stanford Vision Lab, Stanford University, Princeton University imagenet.help.desk@gmail.com Copyright infringement

State-of-the-art 2024

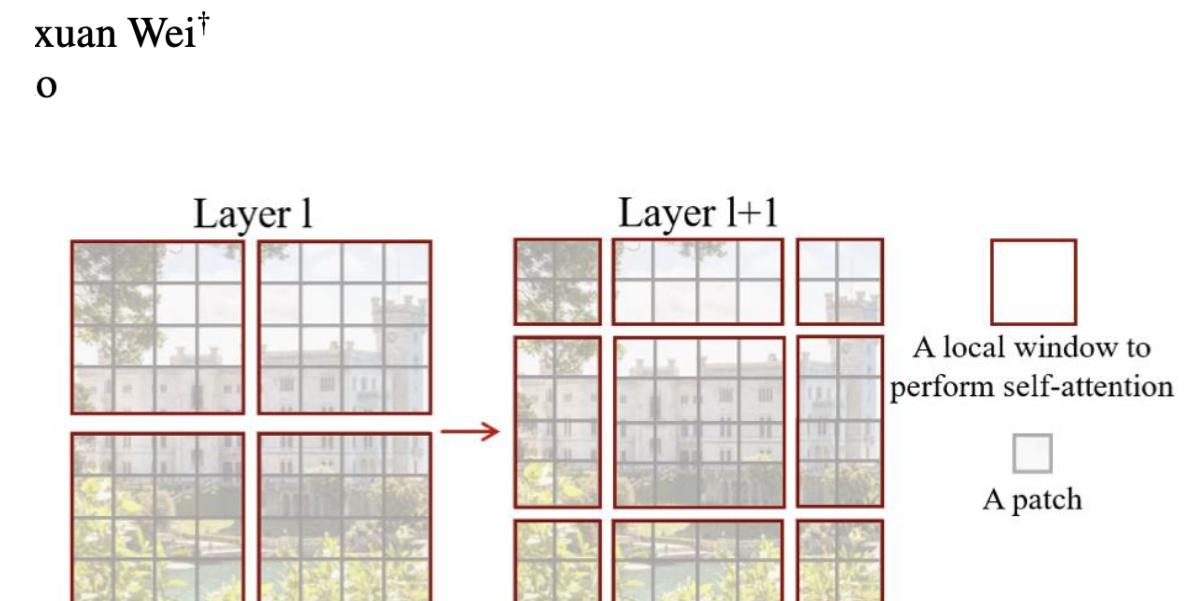
Swin Transformer: Hierarchical Vision Transformer using Shifted Windows



(a) Swin Transformer (ours)



(b) ViT



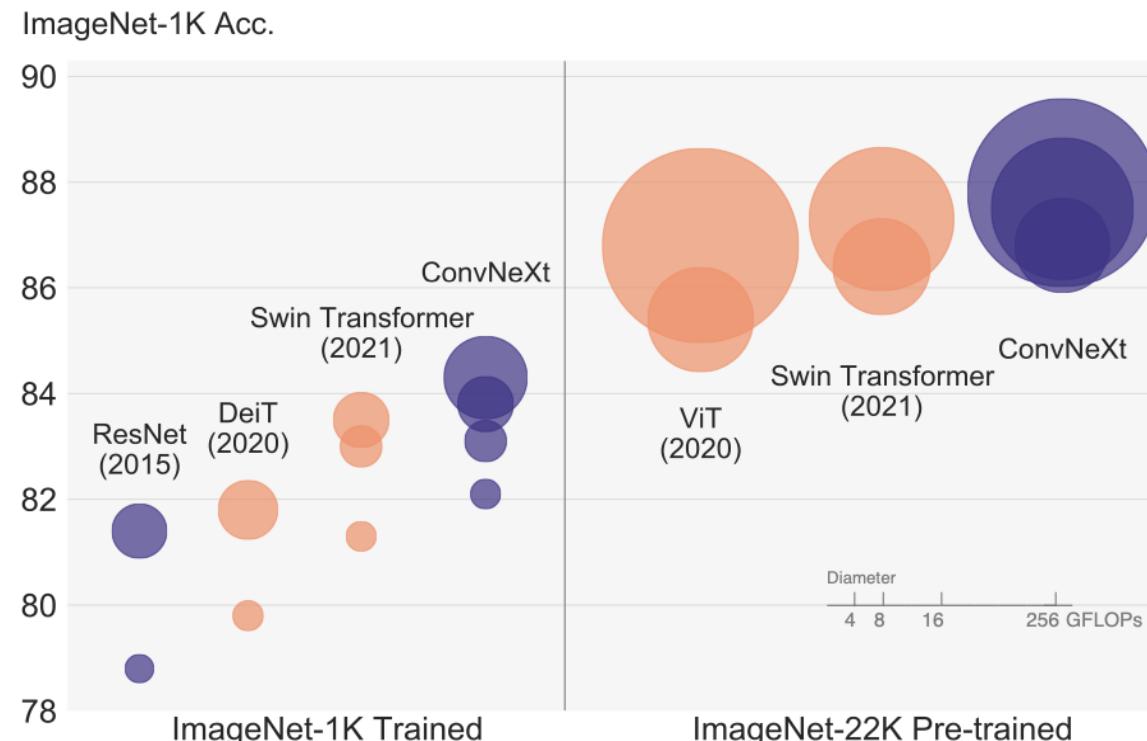
ConvNeXt

A ConvNet for the 2020s

Zhuang Liu^{1,2*} Hanzi Mao¹ Chao-Yuan Wu¹ Christoph Feichtenhofer¹ Trevor Darrell² Saining Xie^{1†}

¹Facebook AI Research (FAIR) ²UC Berkeley

Code: <https://github.com/facebookresearch/ConvNeXt>



Papers with code <https://paperswithcode.com/>

The image displays three separate browser tabs from the [PapersWithCode](https://paperswithcode.com/) website:

- Vision Transformer**: A model by [rwightman / pytorch-image-models](#). Last updated on Feb 14, 2021. Model: `vit_large_patch16_224`. Key stats: Parameters 304 Million, FLOPs 119 Billion, File Size 1.16 GB. Training Data: JFT-300M, ImageNet. Training Resources: TPUv3. Training Techniques: SGD with Momentum, Cosine Annealing, Gradient Clip. Architecture: Layer Normalization, Multi-Head Attention, Tanh Activation, Cross-Attention, Attention Dropout, Dropout, Scaled Dot-Attention, GELU, Convolution.
- Swin Transformer: Hierarchical Vision Transformer using Shifted Windows**: A paper by [ICCV 2021](#) (Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo). Last updated on Feb 14, 2021. The paper discusses a hierarchical Transformer architecture designed for computer vision, addressing challenges like large variance in visual entities and high resolution pixels compared to words. It proposes shifted windows to limit self-attention computation to non-overlapping local windows while maintaining global context through window connections. The hierarchical design allows for linear computational complexity with respect to image size, making it suitable for various vision tasks like classification, detection, and segmentation.
- ConvNeXt Explained**: A method by [Liu et al.](#) introduced in [A ConvNet for the 2020s](#). Last updated on Mar 28, 2024. The paper presents a new convolutional neural network architecture called ConvNeXt, which is designed to be efficient and perform well on various computer vision tasks. It features a hierarchical structure of residual blocks with depthwise separable convolutions and a novel window-based attention mechanism. The results show improved performance on benchmarks like ImageNet-1K, COCO, and ADE20K compared to previous state-of-the-art models.