

Deep Learning & Engineering Applications

11. Vision Transformer

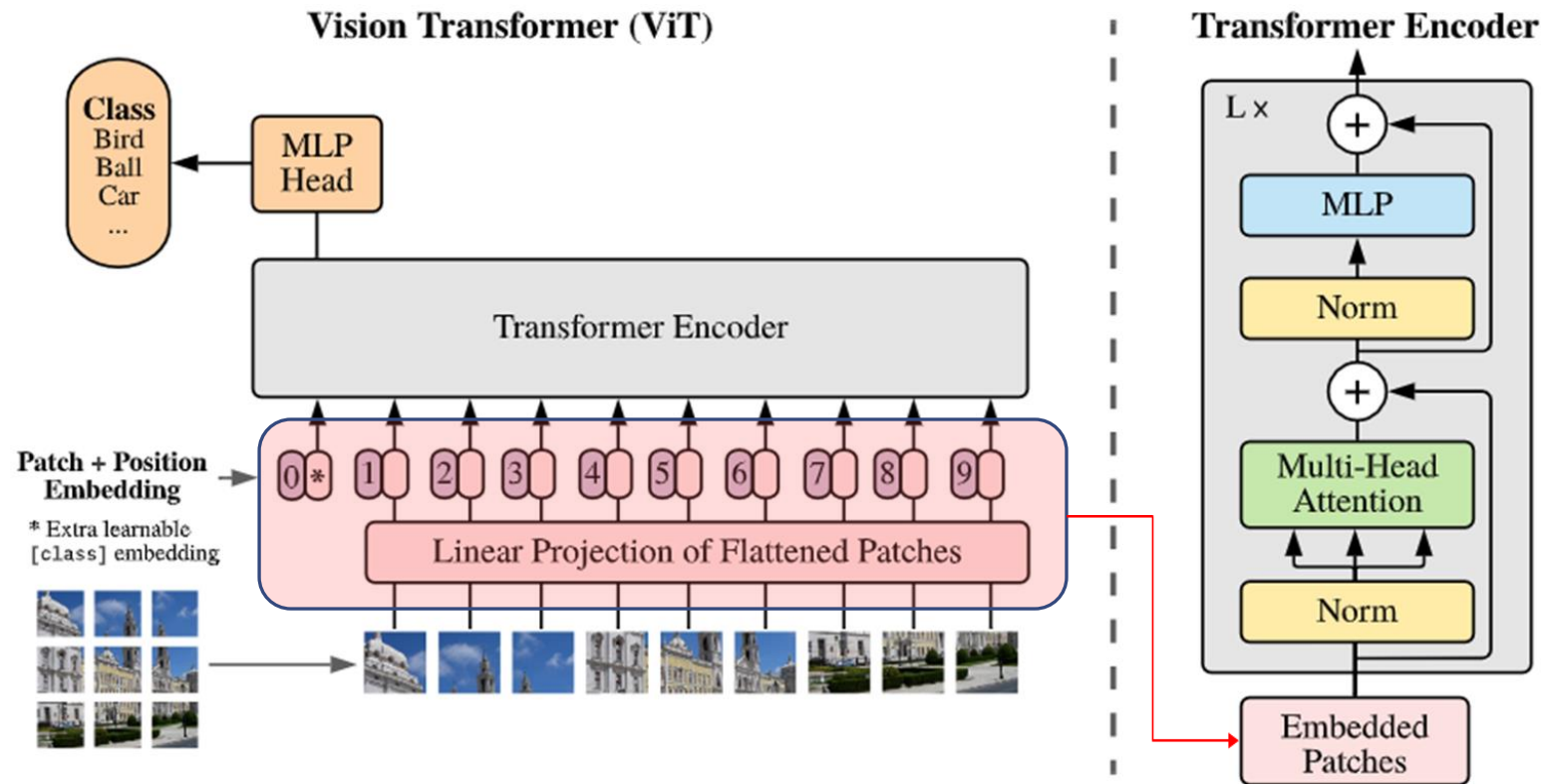
JIDONG J. YANG

COLLEGE OF ENGINEERING

UNIVERSITY OF GEORGIA

Vision Transformer (ViT)

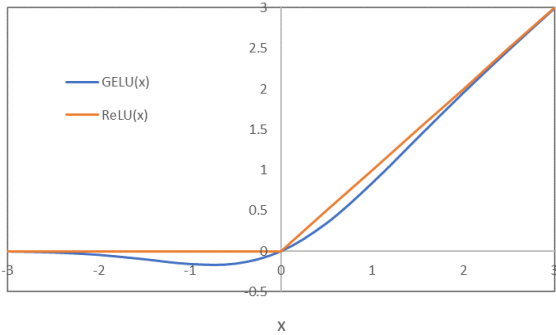
Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M



Dosovitskiy et al. (2020) "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." <https://arxiv.org/pdf/2010.11929>

Phil Wang (2020). Vision Transformer – Pytorch. <https://github.com/lucidrains/vit-pytorch>

Transformer Encoder



2 FC layers with a Gaussian Error Linear Unit (GELU) nonlinearity, $GELU(x) = x\Phi(x)$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \quad \ell = 1 \dots L$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0)$$

$$\text{MSA}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_o$$

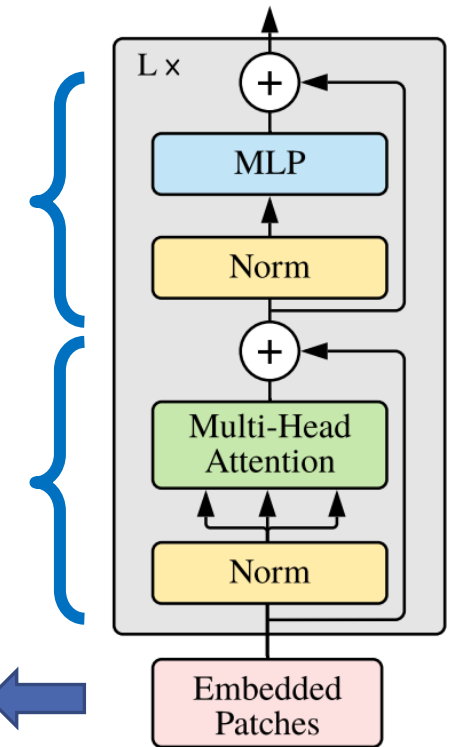
$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{head}_i = \text{Attn}(QW_i^Q, KW_i^K, VW_i^V)$$

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \quad \ell = 1 \dots L$$

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \dots; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{\text{pos}}, \quad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$$

Transformer Encoder



[Hendrycks & Gimpel, 2016] Gaussian Error Linear Units (GELUs) <https://arxiv.org/pdf/1606.08415.pdf>

ViT vs. CNN

ViT has less image-specific inductive bias than CNN.

Position embeddings at initialization time carry no information about the 2D positions of the patches and all spatial relations between the patches are learned from scratch.

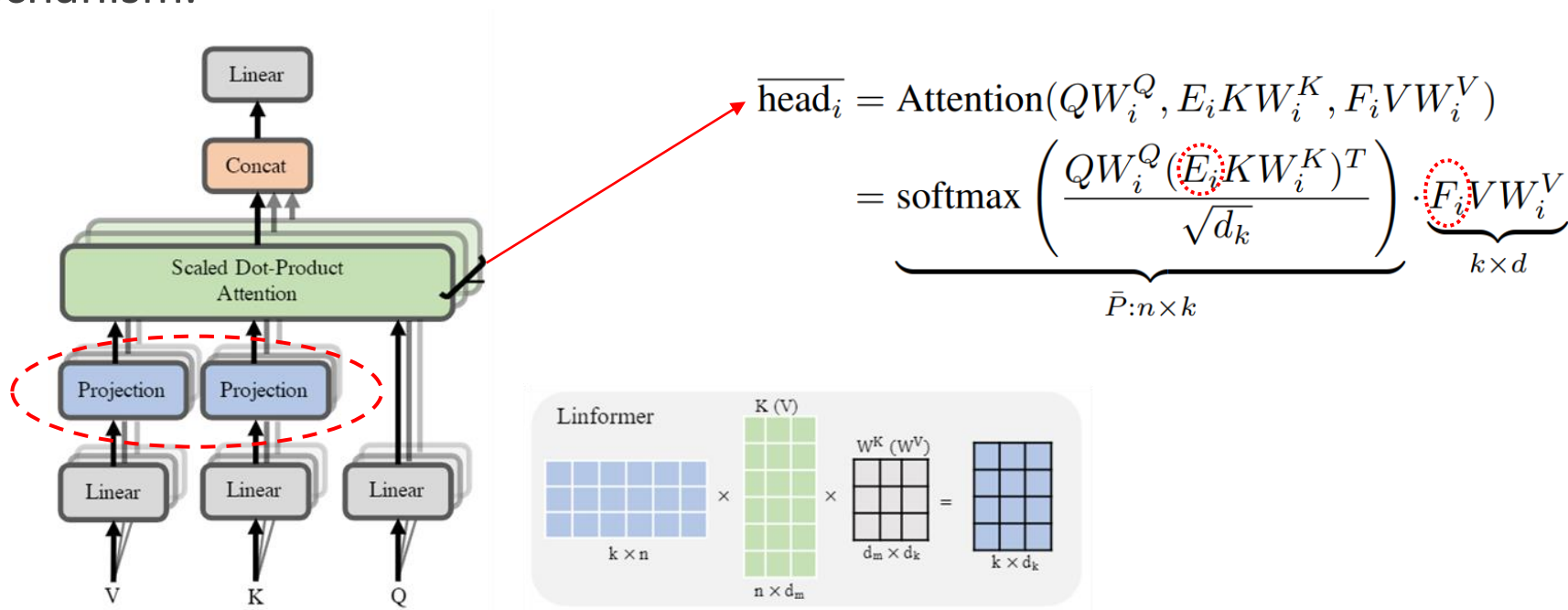
As an alternative to raw image patches, the input sequence can be formed from feature maps of a CNN (some variants of ViT using CNN features will be discussed shortly).

[Raghu et al., 2021] compared ViT with CNN.

[Raghu et al., 2021] Do Vision Transformers See Like Convolutional Neural Networks? <https://arxiv.org/abs/2108.08810>

Linformer

Linformer adds two low rank matrices to reduce the dimensions of the multi-head self-attention mechanism.

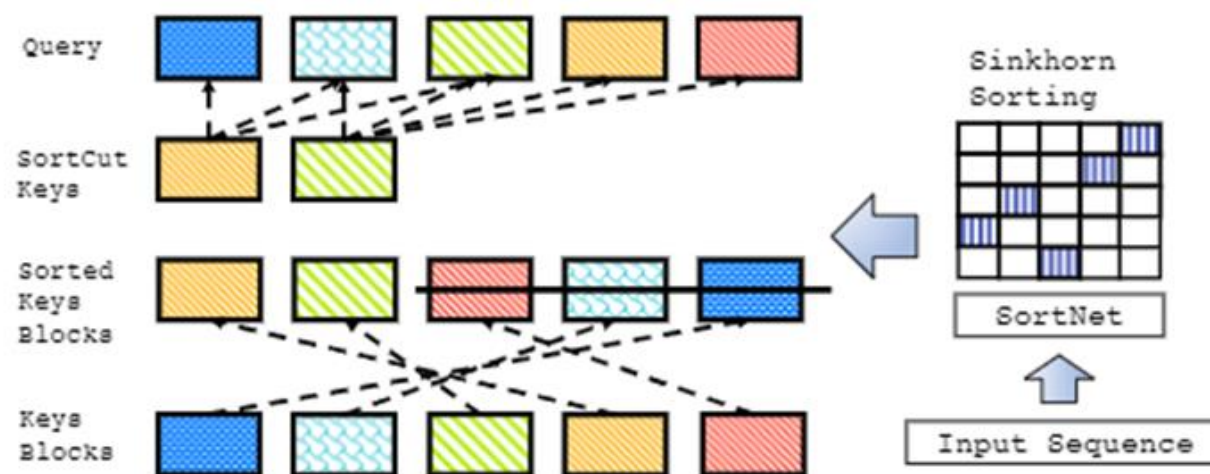


Phil Wang (2020). x-transformers. <https://github.com/lucidrains/x-transformers>

Wang et al. (2020) "Linformer: Self-Attention with Linear Complexity," arXiv:2006.04768 [cs.LG] <https://arxiv.org/pdf/2006.04768>

Sinkhorn

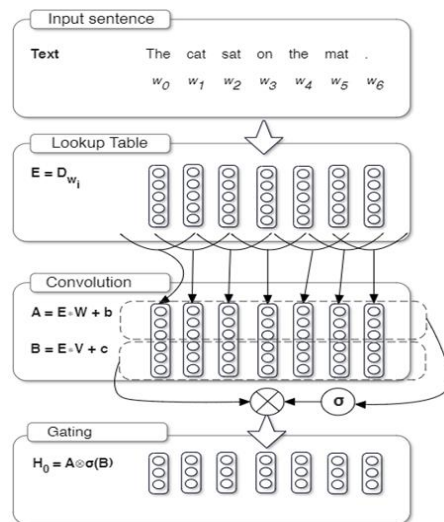
Sinkhorn sorts and cuts key sequences to enable efficient quasi-global local attention. It includes a parameterized sorting network, using “Sinkhorn normalization” to sample a permutation matrix that matches the most relevant blocks of keys to the blocks of queries.



Tay et al. (2020) “Sparse Sinkhorn Attention,” arXiv:2002.11296 [cs.LG]
<https://arxiv.org/pdf/2002.11296>

Encoder + Feed Forward GLU Variants + Residual Attention (we refer to this as “Encoder”)

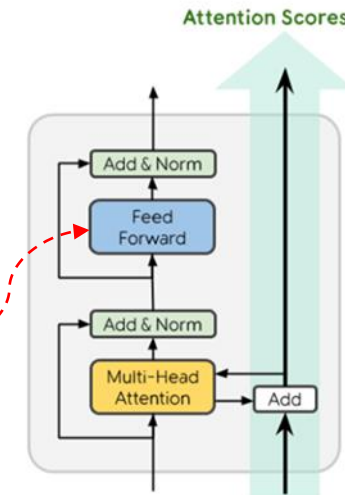
Transformer encoder with Gated Linear Units Variants (in Feed Forward) and Residual Attention is used.



Gated Linear Units (GLU)

$$\begin{aligned} \text{FFN}_{\text{GLU}}(x, W, V, W_2) &= (\sigma(xW) \otimes xV)W_2 \\ \text{FFN}_{\text{Bilinear}}(x, W, V, W_2) &= (xW \otimes xV)W_2 \\ \text{FFN}_{\text{ReLU}}(x, W, V, W_2) &= (\max(0, xW) \otimes xV)W_2 \\ \text{FFN}_{\text{GELU}}(x, W, V, W_2) &= (\text{GELU}(xW) \otimes xV)W_2 \\ \text{FFN}_{\text{SwiGLU}}(x, W, V, W_2) &= (\text{Swish}_1(xW) \otimes xV)W_2 \end{aligned}$$

GLU Variants



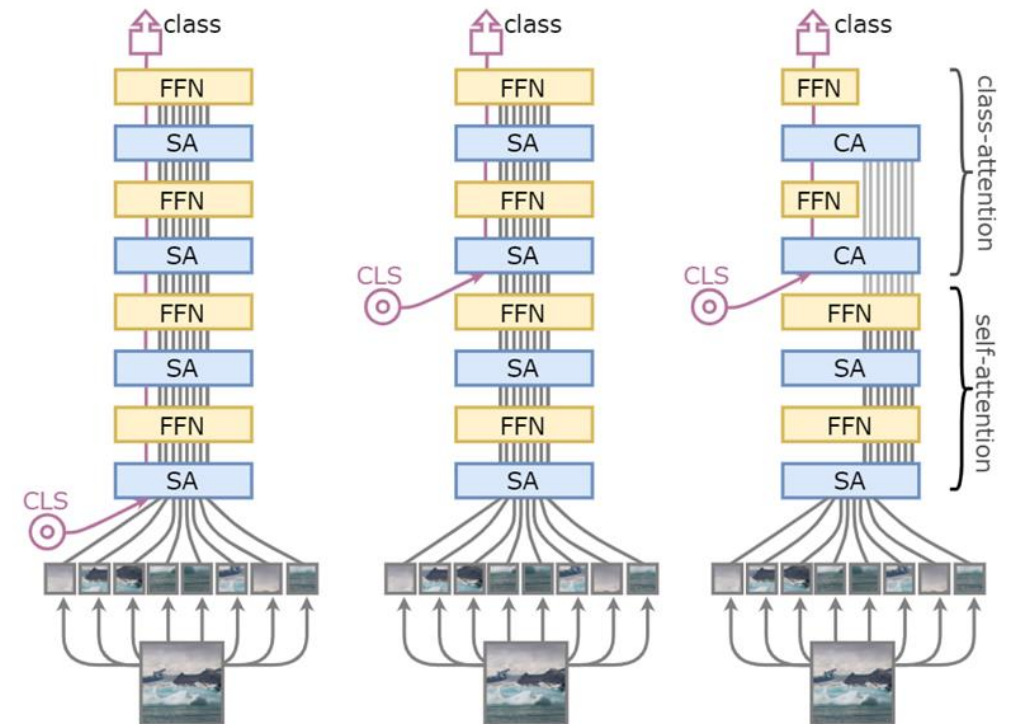
Residual Attention

Dauphin et al. (2017), “Language Modeling with Gated Convolutional Networks,” arXiv:1612.08083v3 [cs.CL] <https://arxiv.org/pdf/1612.08083>
Shazeer (2020), “GLU Variants Improve Transformer,” arXiv: 2002.05202 [cs.LG] <https://arxiv.org/pdf/2002.05202>
He et al. (2020), RealFormer: Transformer Likes Residual Attention,” <https://arxiv.org/pdf/2012.11747>

Class-Attention in Image Transformers (CaiT)

Vision Transformer with per-channel multiplication of the output of the residual blocks and adding the class token after a few layers of self-attention and FFN.

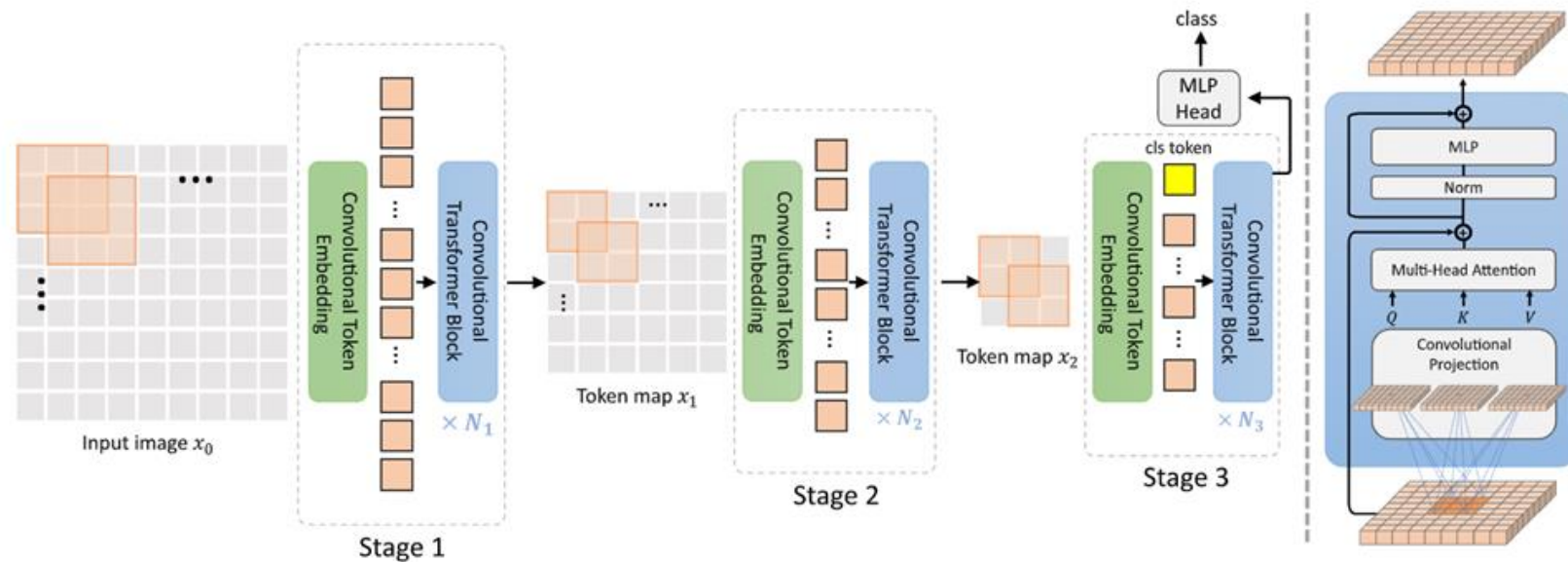
Separate the class attention from self-attention between patches.



Touvron et al. (2020) "Going deeper with Image Transformers," arXiv:2103.17239 [cs.CV] <https://arxiv.org/pdf/2103.17239>

Convolutional Vision Transformer (CvT)

Replaces Linear Projection of patches with a convolutional layer.
CLS token is only being added in the last stage.
Removal of positional embedding.

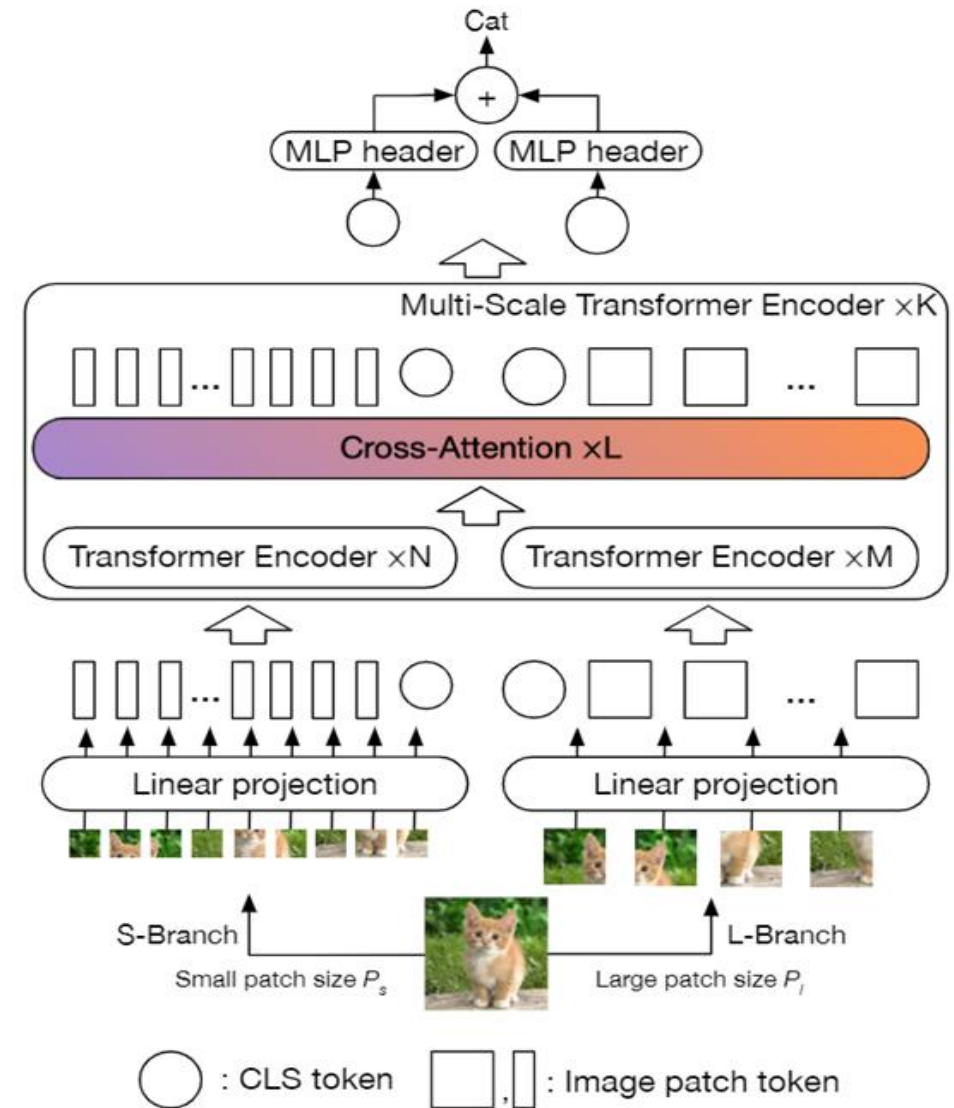


Wu et al. (2021) "CvT: Introducing Convolutions to Vision Transformers," arXiv:2103.15808 [cs.CV] <https://arxiv.org/pdf/2103.15808>

Cross ViT

Cross vision transformer processes images in two different patch sizes. Then a cross attention is used instead of traditional multi-head attention.



























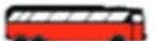









It allows the CLS token interact with patch tokens in the other branch.



Chen et al. (2021) "CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification," arXiv:2103.14899 [cs.CV] <https://arxiv.org/pdf/2103.14899>

Engineering Application Example – Vehicle Classification

Federal Highway Administration (FHWA) Vehicle Classification

Class 1 Motorcycle		Class 5 Two axle, six tire, single unit	  	Class 9 5-axle tractor semitrailer	 
Class 2 Passenger cars	   	Class 6 Three axle, single unit	   	Class 10 Six or more axle, single trailer	 
Class 3 Four tire, single unit	  	Class 7 Four or more axle, single unit	    	Class 11 5 or less axle, multi trailer	
Class 4 Buses	  	Class 8 Four or less axle, single trailer	  	Class 12 Six axle, multi-trailer	 
				Class 13 Seven or more axle, multi-trailer	  

What are the most important features for vehicle classification?

Wheel Information

Wheel detectors:

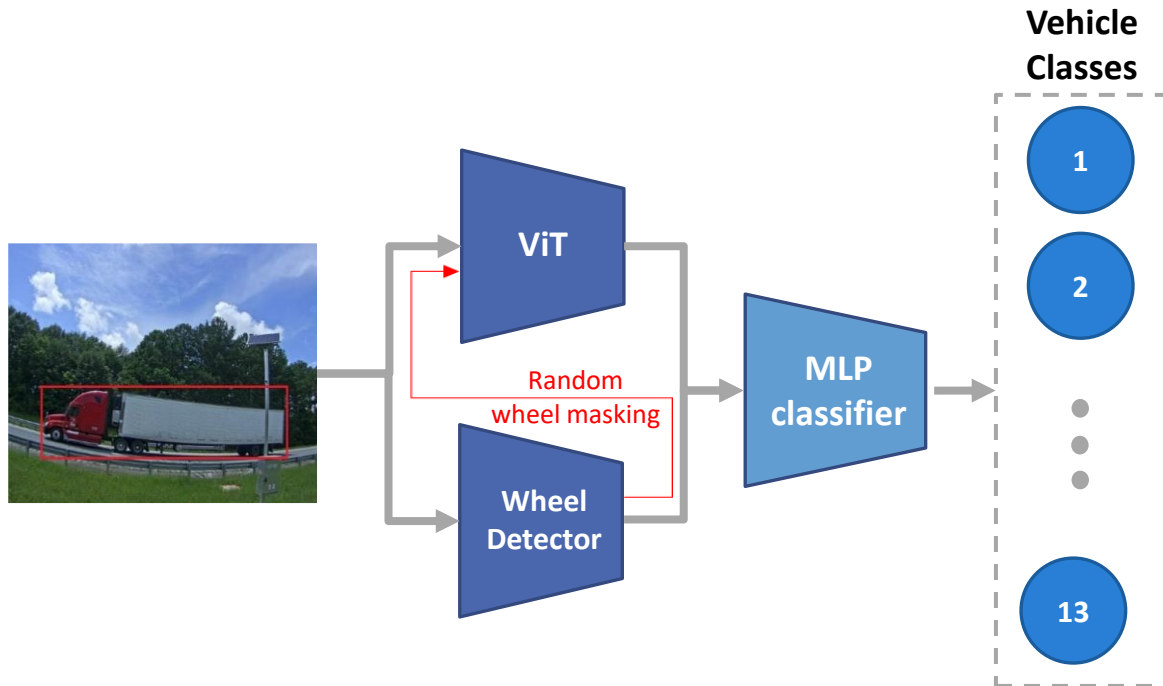
- Faster RCNN with backbones:
 - MobileNet V2
 - ResNet
- YOLOv4



Wheel Bounding Boxes

Composite Model (ViT + Wheel Detector)

- Original images vs Cropped images
- Wheel info vs No wheel Info
- Random wheel masking vs No wheel masking



Model Type	Network	Top-1 Acc (%)							
		0 wheel masking		1 wheel masking		2 wheel masking		3 wheel masking	
		Original	Cropped	Original	Cropped	Original	Cropped	Original	Cropped
Vision Transformer	ViT	87.5	90.4	87.7	90.1	87.0	90.1	86.8	90.2
	Linformer	84.1	89.2	83.3	89.1	83.9	88.5	84.6	89.1
	Sinkhorn	86.3	90.6	86.5	89.6	86.3	89.8	86.5	89.4
	Encoder	87.3	89.9	87.6	90.7	86.0	90.4	86.3	89.8
	CaiT	81.7	88.3	82.7	88.2	82.9	87.9	82.4	87.3
	CvT	83.6	87.2	84.2	87.5	83.7	86.6	82.4	86.0
	CrossViT	85.8	89.2	86.9	89.6	86.2	88.9	84.8	89.6
Composite Model	ViT+YOLOv4	88.0	91.8	89.2	93.2	88.9	91.7	88.3	91.1
	Linformer+YOLOv4	85.3	90.4	85.6	91.4	86.7	90.4	87.0	90.3
	Sinkhorn+YOLOv4	89.1	92.2	88.7	92.5	88.5	91.7	88.4	91.6
	Encoder+YOLOv4	88.9	91.7	88.9	93.0	87.8	91.4	88.0	91.7
	CaiT+YOLOv4	85.6	91.5	86.4	91.3	86.8	91.1	86.1	90.4
	CvT+YOLOv4	84.3	87.6	84.6	88.0	84.6	87.6	83.0	87.2
	CrossViT+YOLOv4	87.7	91	88.9	91.5	87.5	90.4	87.6	91.2

Note: the wheel masking was performed in a random fashion.

Some recently emerged variants of transformer

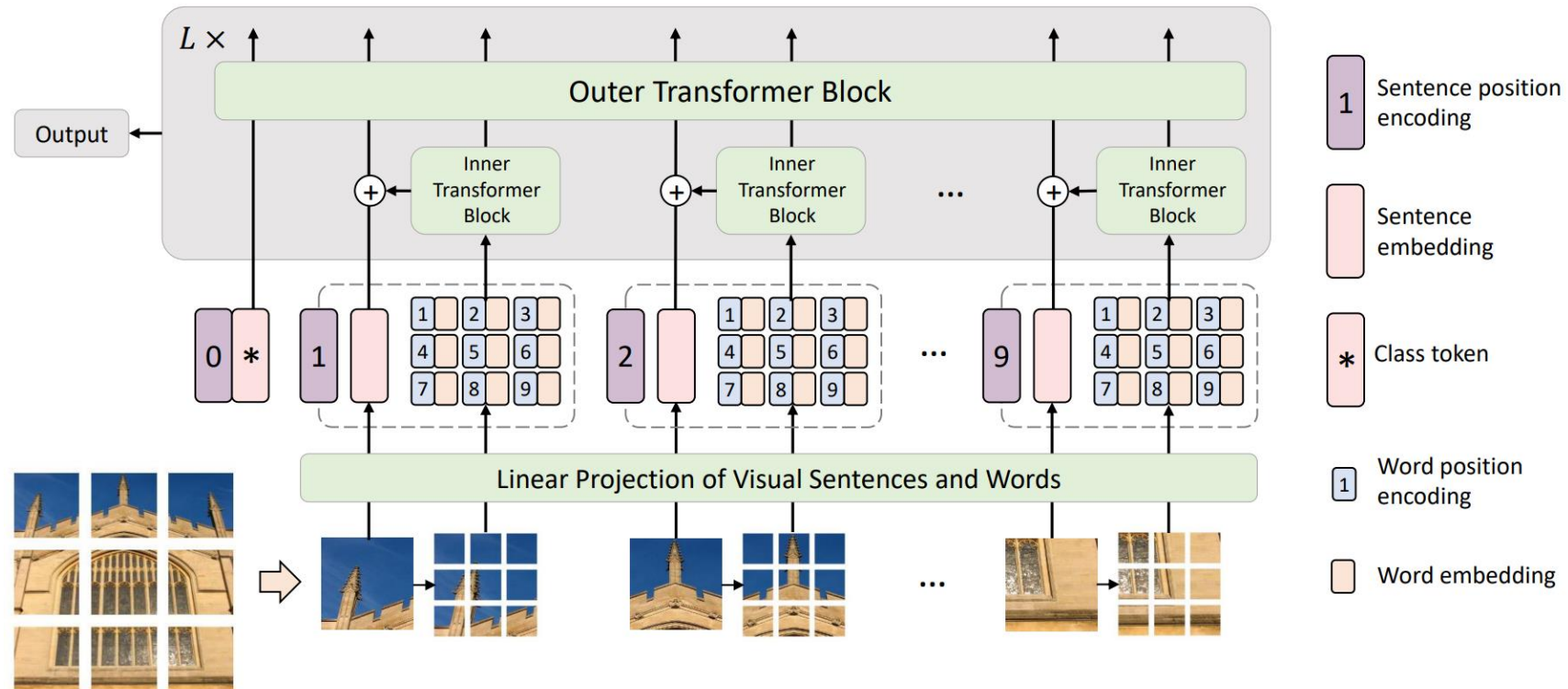
Transformer in Transformer (TNT)

- A nested structure

Swin (**S**hifted **W**indow) Transformer

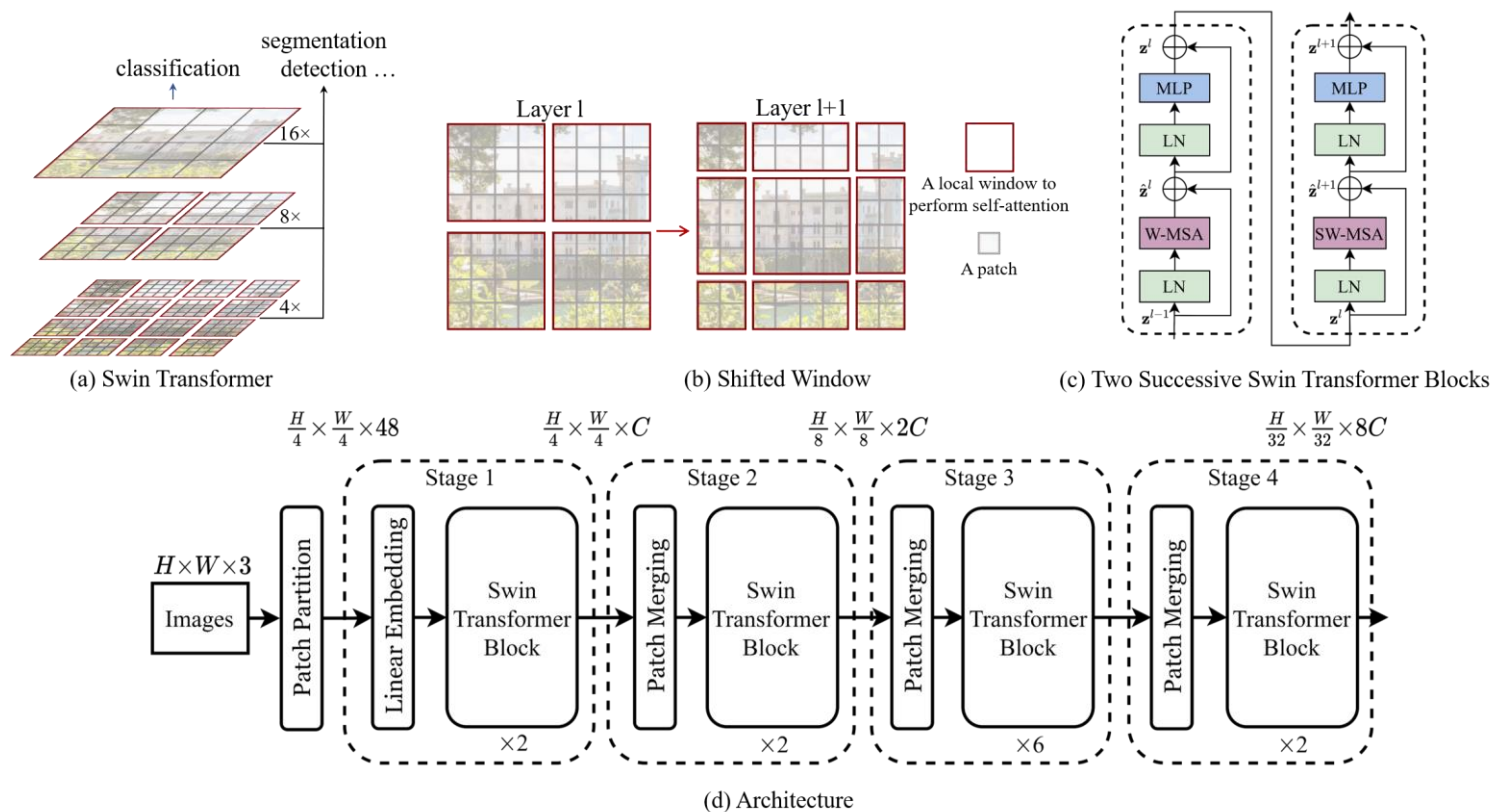
- A hierarchical Transformer, where representation is computed with shifted windows.

Transformer in Transformer (TNT)



[Han et al., 2021] <https://arxiv.org/pdf/2103.00112.pdf> https://github.com/huawei-noah/CV-Backbones/tree/master/tnt_pytorch

Swin Transformer



[Liu et al., 2021] <https://arxiv.org/pdf/2103.14030.pdf> <https://github.com/microsoft/Swin-Transformer>