



Transformer for Computer Vision

Wenhai Wang



Outline



- Transformer
- Transformer-Based Head
 - DEtection TRansformer (DETR) (*detection*)
 - Deformable DETR (*detection*)
 - Trans2Seg (*segmentation*)
- Transformer-Based Backbone
 - Vision Transformer (ViT)
 - Pyramid Vision (PVT)
- Future Work



Transformer



1. Multi-head Attention

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

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

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

2. Feed Forward

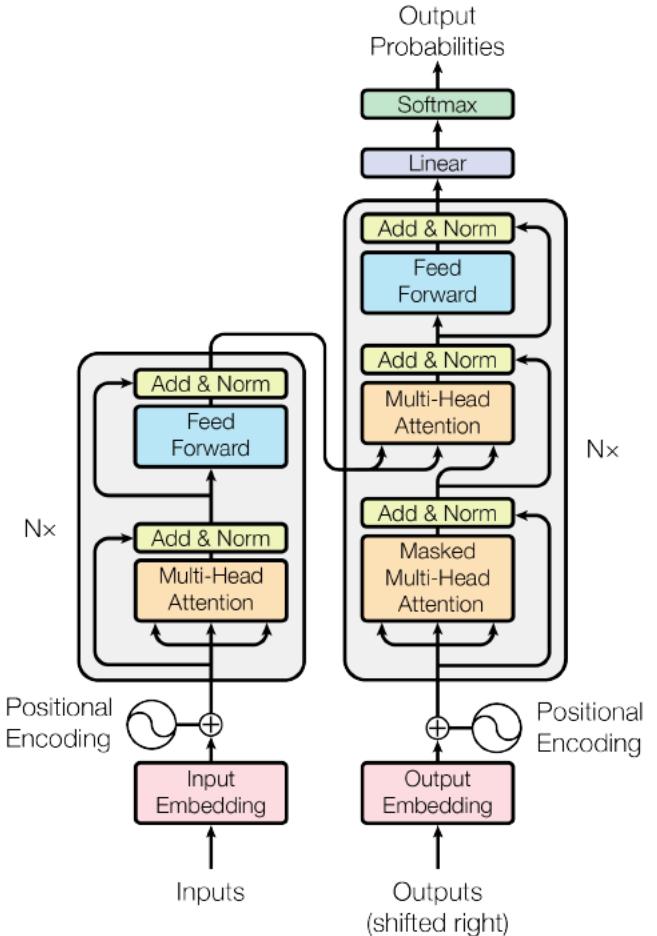
3. Position Embedding

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

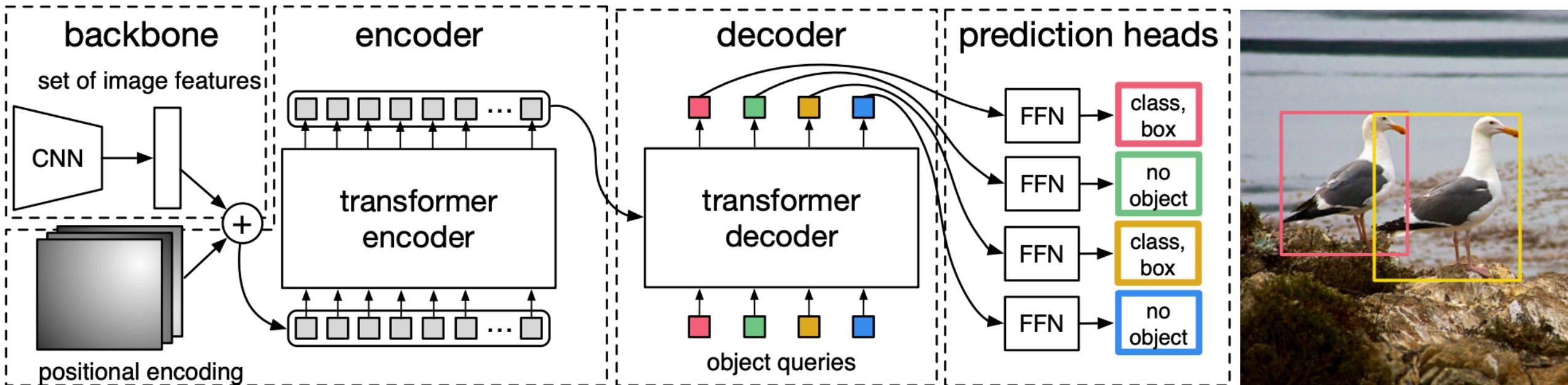
4. Dropout/Residual Connection/Layer Normalization (LN)

Vaswani, Ashish, et al. "Attention is all you need." in NeurIPS, 2017.





DEtection TRansformer (DETR)



1. *Object Detection -> Dictionary Lookup*
2. Query + Bipartite Matching Loss
3. No NMS

Carion, Nicolas, et al. "End-to-end object detection with transformers." In ECCV, 2020.



Deformable DETR



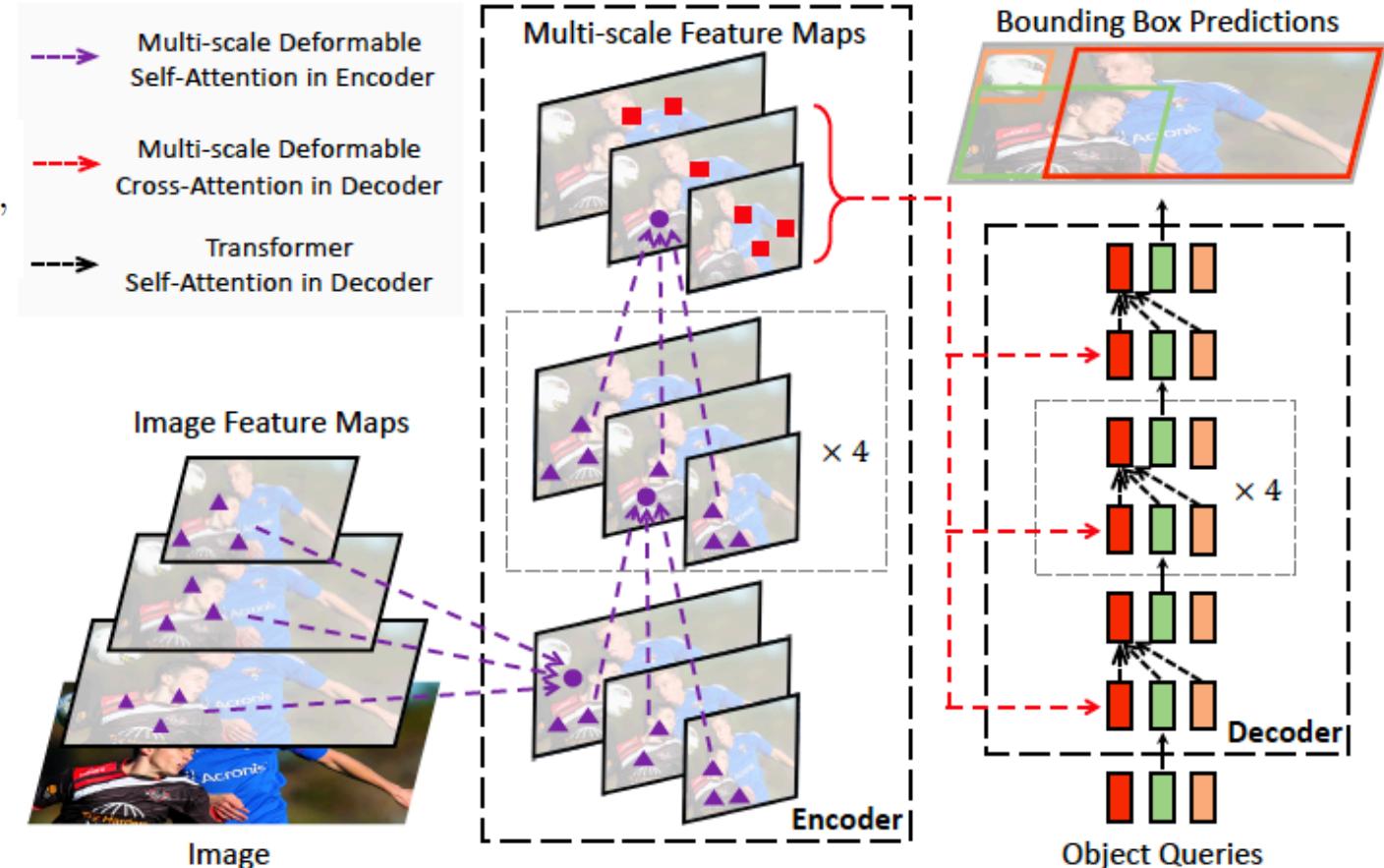
- DeformAttn

$$\sum_{m=1}^M \mathbf{W}_m \left[\sum_{k=1}^K A_{mqk} \cdot \mathbf{W}'_m \mathbf{x}(p_q + \Delta p_{mqk}) \right],$$

where $A_{mpk}, \Delta p_{mqk}$ is predicted by p_q .

- Advantages

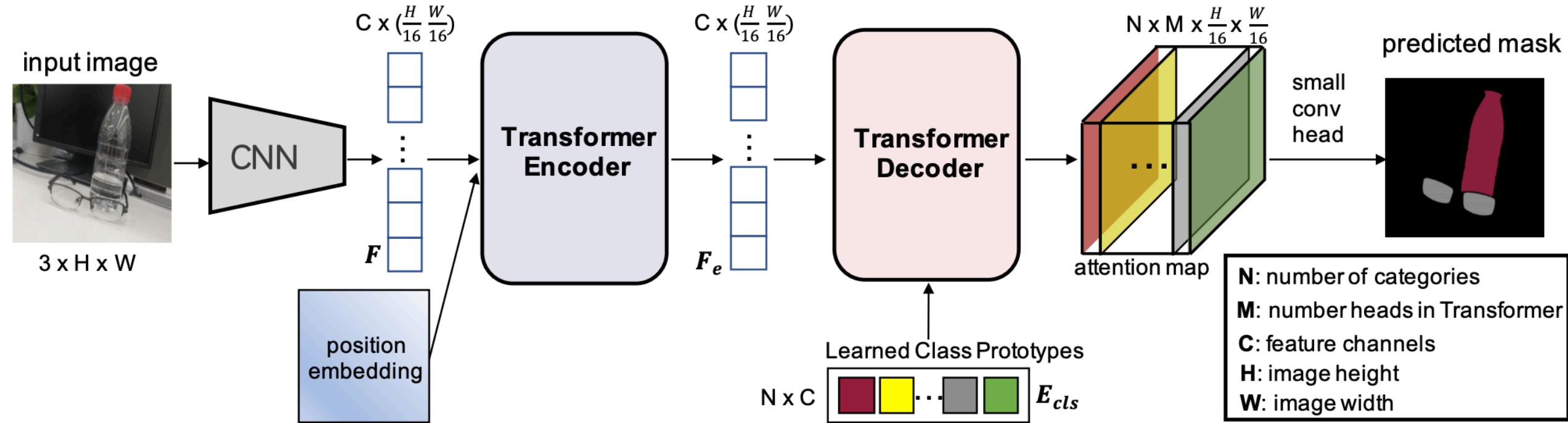
1. Faster Convergence Speed
2. Lower Computational Cost



Zhu, Xizhou, et al. "Deformable DETR: Deformable Transformers for End-to-End Object Detection." in ICLR, 2020.



Trans2Seg



Semantic Segmentation -> Dictionary Lookup

Xie, Enze, et al. "Segmenting transparent object in the wild with transformer." arXiv preprint arXiv:2101.08461 (2021).



Vision Transformer (ViT)

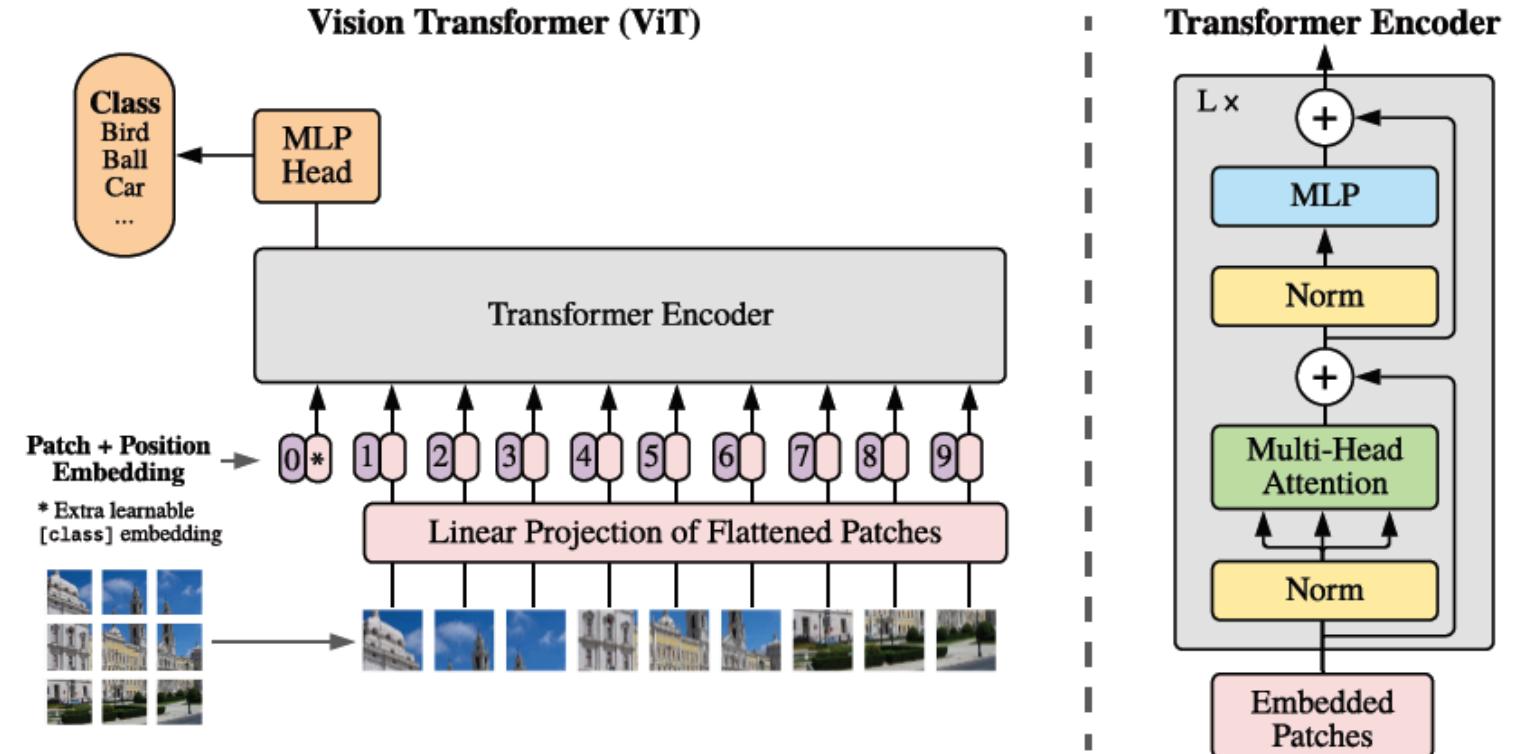


- Some Tricks

1. Randomly-Initialized Position Embedding

2. *Residual Connection w/ Drop Path*

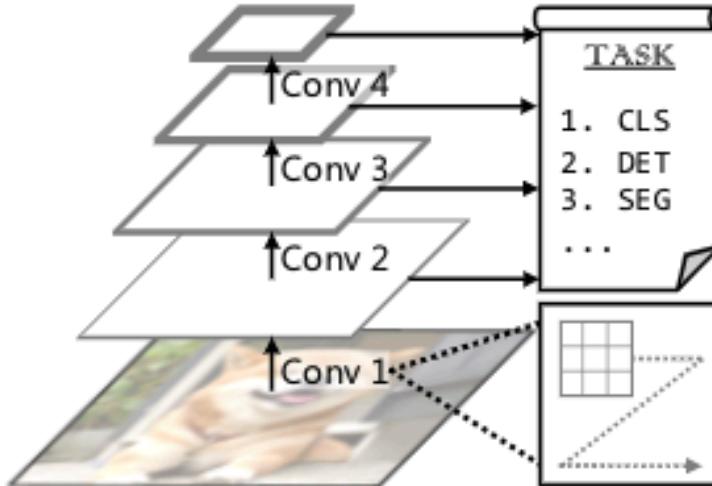
3. ReLU -> GeLU



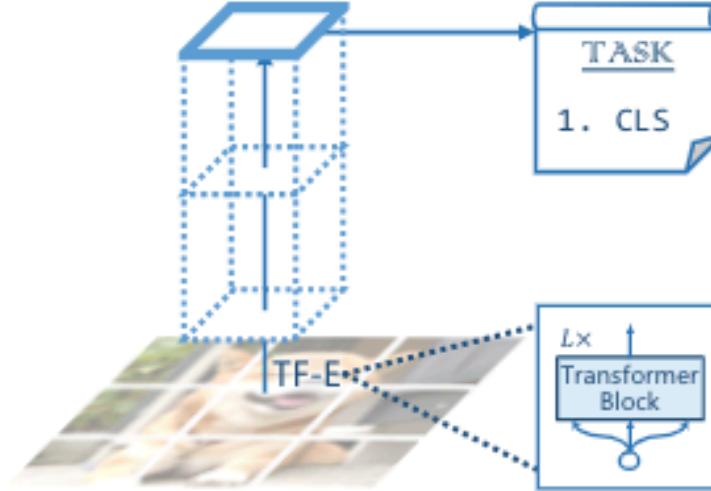
Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." in ICLR, 2020.



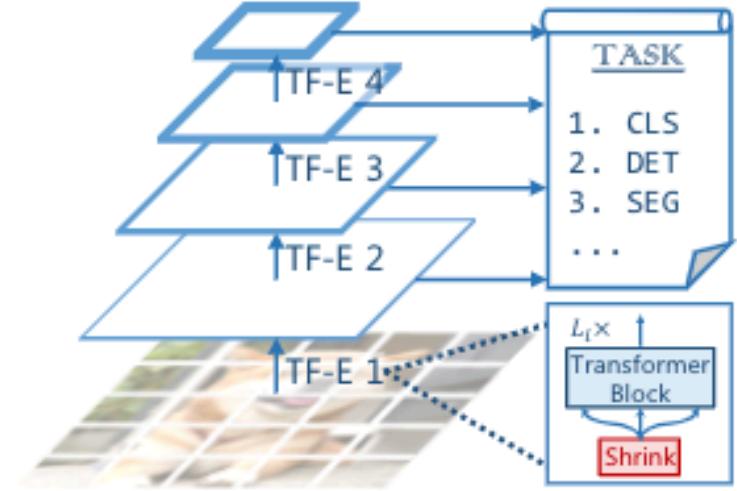
Pyramid Vision Transformer (PVT)



(a) CNNs: VGG [41], ResNet [15], etc.



(b) Vision Transformer [10]



(c) Pyramid Vision Transformer (ours)

- Limitations of CNN
- 1. Local Receptive Field

- Limitations of ViT
- 1. Columnar Structure
- 2. Single-Scale/Low-Resolution Output
- 3. *Unsuitable for Detection/Segmentation*

Wang, Wenhui, et al. "Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions." arXiv preprint arXiv:2102.12122 (2021).



Overall Architecture

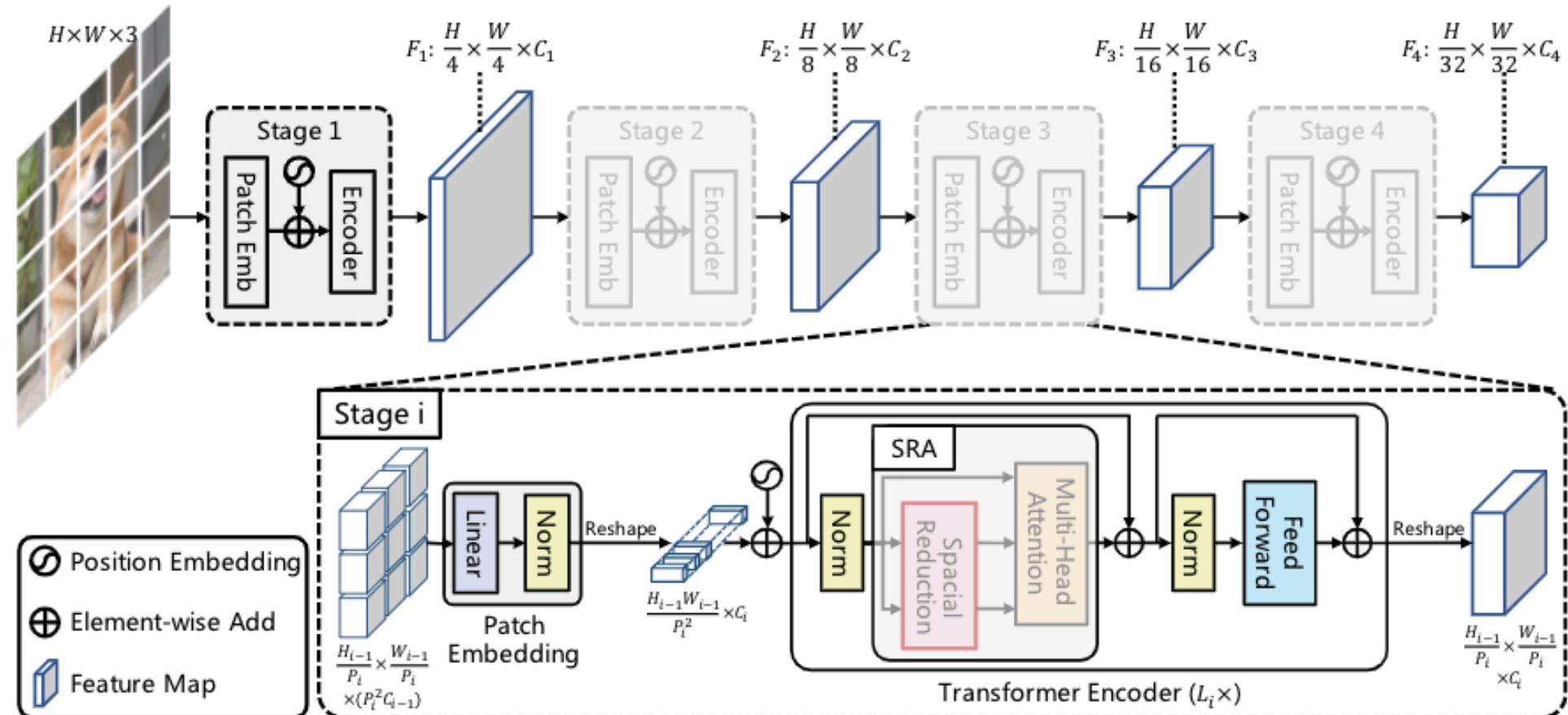


- Key Points

1. Four Stages for *Pyramid Structure*

2. Each Stage:
 - (1) Patch Emb.
 - (2) Transformer Enc.

3. Spatial-Reduction Attention (SRA) for *high-resolution feature map*

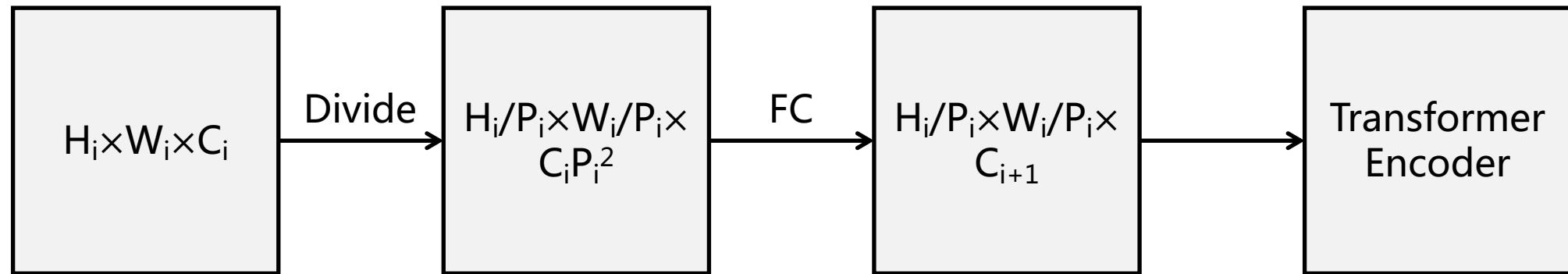




How PVT obtains the feature pyramid?



- Adjusting the *patch size* (P_i) in Stage i



The process of the patch embedding in Stage i



Spatial-Reduction Attention



- SRA

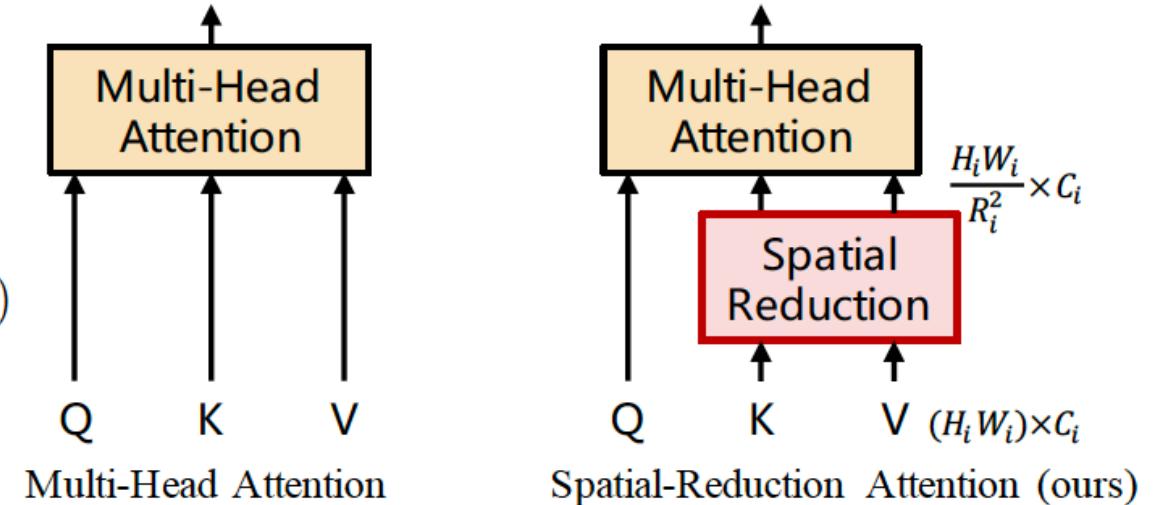
$$\text{SRA}(Q, K, V) = \text{Concat}(\text{head}_0, \dots, \text{head}_{N_i})W^O$$

$$\text{head}_j = \text{Attention}(QW_j^Q, \text{SR}(K)W_j^K, \text{SR}(V)W_j^V)$$

$$\text{SR}(\mathbf{x}) = \text{Norm}(\text{Reshape}(\mathbf{x}, R_i)W^S)$$

$$\text{Attention}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \text{Softmax}\left(\frac{\mathbf{q}\mathbf{k}^\top}{\sqrt{d_{\text{head}}}}\right)\mathbf{v}$$

Compared to MHA, the resource cost of our SRA is R_i^2 times lower!





Detailed settings

- P_i : the patch size of the stage i ;
- C_i : the channel number of the output of the stage i ;
- L_i : the number of encoder layers in the stage i ;
- R_i : the reduction ratio of the SRA in the stage i ;
- N_i : the head number of the SRA in the stage i ;
- E_i : the expansion ratio of the feed-forward layer [51] in the stage i ;



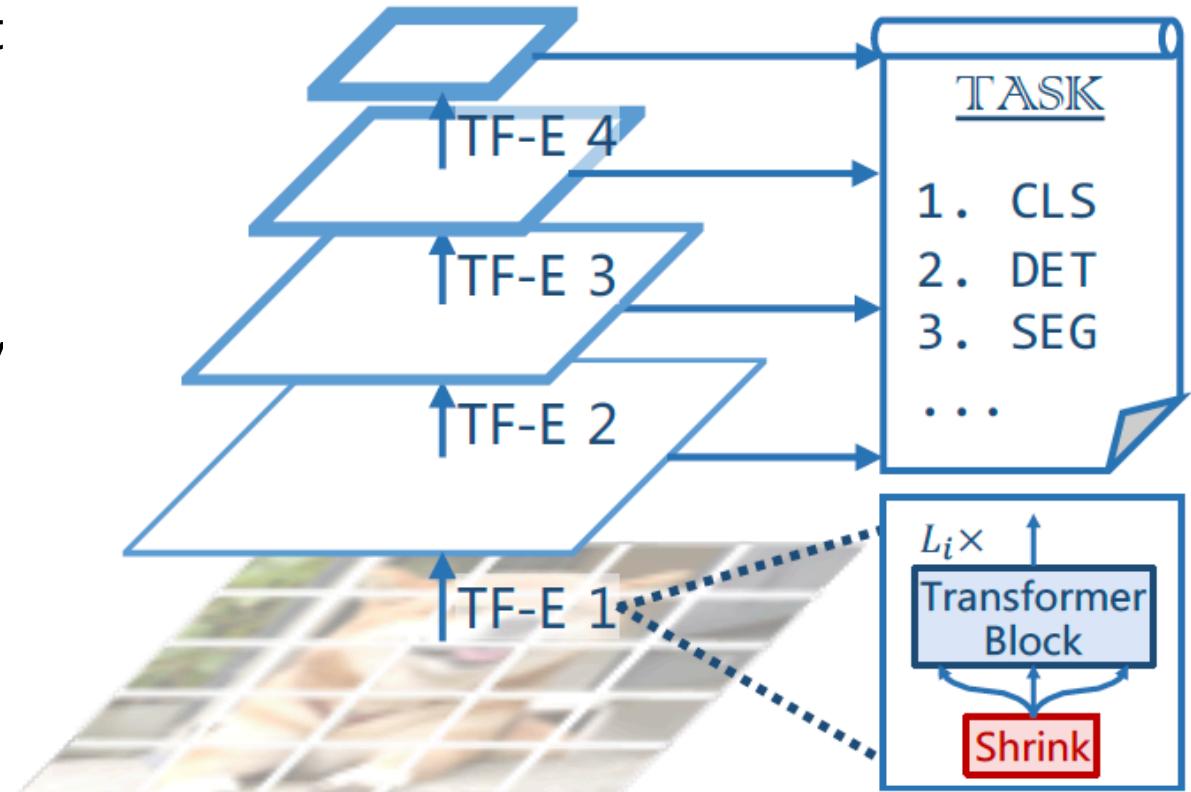
	Output Size	Layer Name	PYT-Tiny	PYT-Small	PYT-Medium	PYT-Large
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	Patch Embedding		$P_1 = 4; C_1 = 64$		
		Transformer Encoder	$\begin{bmatrix} R_1 = 8 \\ N_1 = 1 \\ E_1 = 8 \end{bmatrix} \times 2$	$\begin{bmatrix} R_1 = 8 \\ N_1 = 1 \\ E_1 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} R_1 = 8 \\ N_1 = 1 \\ E_1 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} R_1 = 8 \\ N_1 = 1 \\ E_1 = 8 \end{bmatrix} \times 3$
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	Patch Embedding		$P_2 = 2; C_2 = 128$		
		Transformer Encoder	$\begin{bmatrix} R_2 = 4 \\ N_2 = 2 \\ E_2 = 8 \end{bmatrix} \times 2$	$\begin{bmatrix} R_2 = 4 \\ N_2 = 2 \\ E_2 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} R_2 = 4 \\ N_2 = 2 \\ E_2 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} R_2 = 4 \\ N_2 = 2 \\ E_2 = 8 \end{bmatrix} \times 8$
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	Patch Embedding		$P_3 = 2; C_3 = 320$		
		Transformer Encoder	$\begin{bmatrix} R_3 = 2 \\ N_3 = 5 \\ E_3 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} R_3 = 2 \\ N_3 = 5 \\ E_3 = 4 \end{bmatrix} \times 6$	$\begin{bmatrix} R_3 = 2 \\ N_3 = 5 \\ E_3 = 4 \end{bmatrix} \times 18$	$\begin{bmatrix} R_3 = 2 \\ N_3 = 5 \\ E_3 = 4 \end{bmatrix} \times 27$
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	Patch Embedding		$P_4 = 2; C_4 = 512$		
		Transformer Encoder	$\begin{bmatrix} R_4 = 1 \\ N_4 = 8 \\ E_4 = 4 \end{bmatrix} \times 2$	$\begin{bmatrix} R_4 = 1 \\ N_4 = 8 \\ E_4 = 4 \end{bmatrix} \times 3$	$\begin{bmatrix} R_4 = 1 \\ N_4 = 8 \\ E_4 = 4 \end{bmatrix} \times 3$	$\begin{bmatrix} R_4 = 1 \\ N_4 = 8 \\ E_4 = 4 \end{bmatrix} \times 3$

Table 1: **Detailed settings of PVT series.** The design follows the two rules of ResNet [15]: (1) with the growth of network depth, the hidden dimension gradually increases, and the output resolution progressively shrinks; (2) the major computation resource is concentrated in Stage 3.



Advantages

1. Multi-Scale/High-Resolution Output
2. *As versatile as ResNet, can be apply to detection/segmentation*
3. Making pure Transformer detection/segmentation possible, for example
 - (1) PVT + DETR
 - (2) PVT + Trans2Seg





Performance



- PVT-S vs. R50

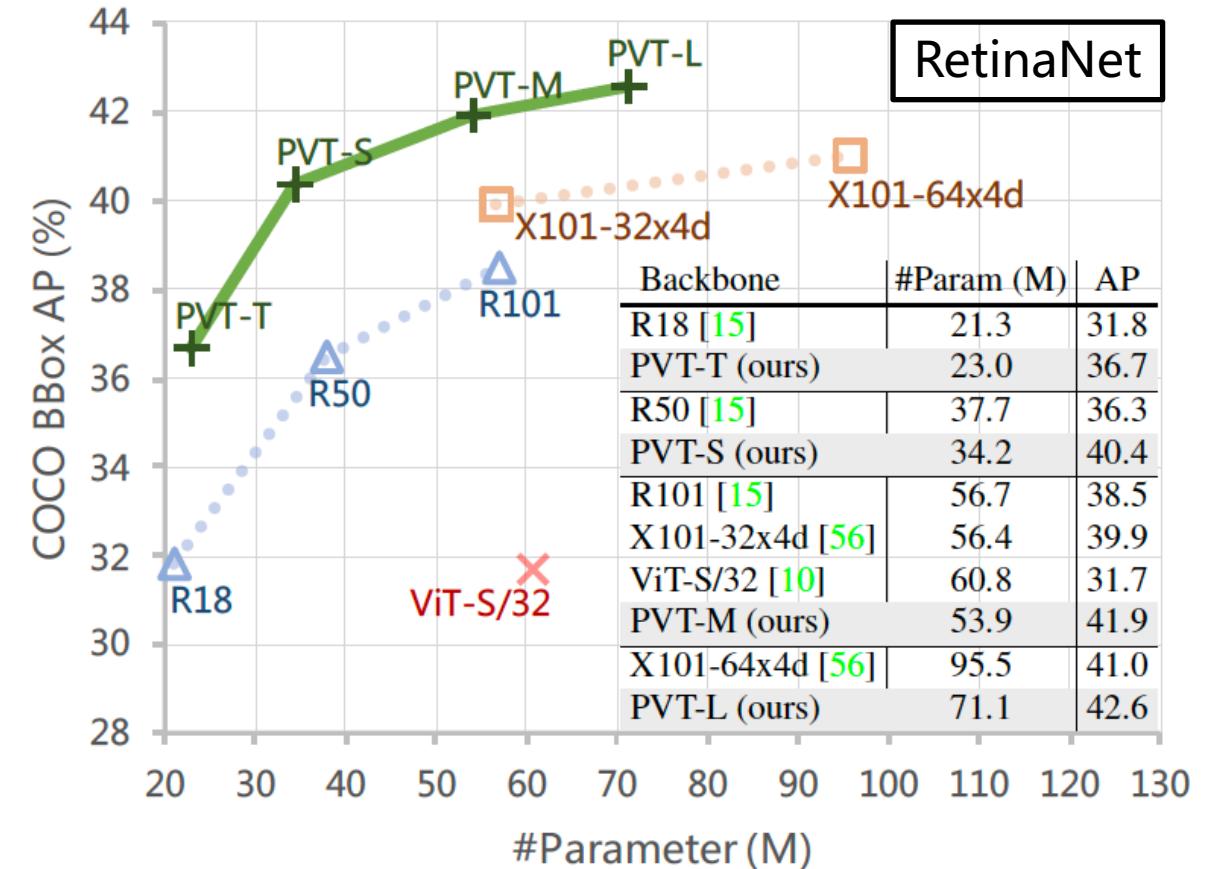
AP: 40.4 vs. 36.3 (4.1 better)

#P: 34.2 vs. 37.7

- PVT-L vs. X101-64x4d

AP: 42.6 vs. 41.0 (1.6 better)

#P: 71.1 vs. 95.5 (20% fewer)





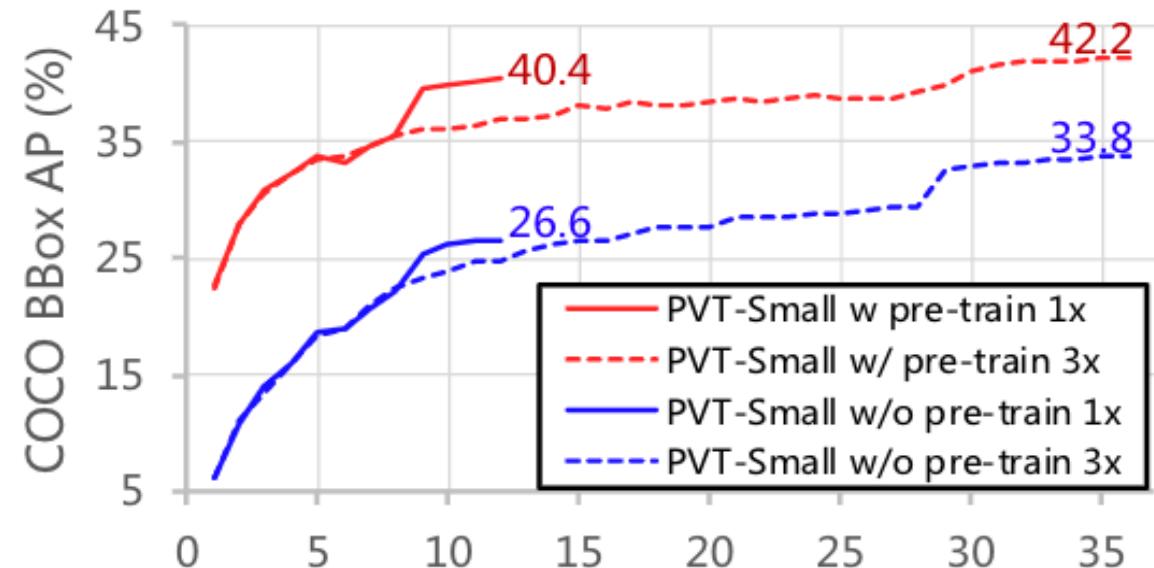
Ablation Study



1. *Going deeper is better than going wider*
2. *Pretrained weights can help PVT converge faster and better*

Method	#Param (M)	Top-1	RetinaNet 1x		
			AP	AP ₅₀	AP ₇₅
Wider PVT-Small	46.8	19.3	40.8	61.8	43.3
Deeper PVT-Small	44.2	18.8	41.9	63.1	44.3

Table A3: **Deeper vs. Wider.** “Top-1” denotes the top-1 error on the ImageNet validation set. “AP” denotes the bounding box AP on COCO val2017. The deeper model obtains better performance than the wider model under comparable parameter number.

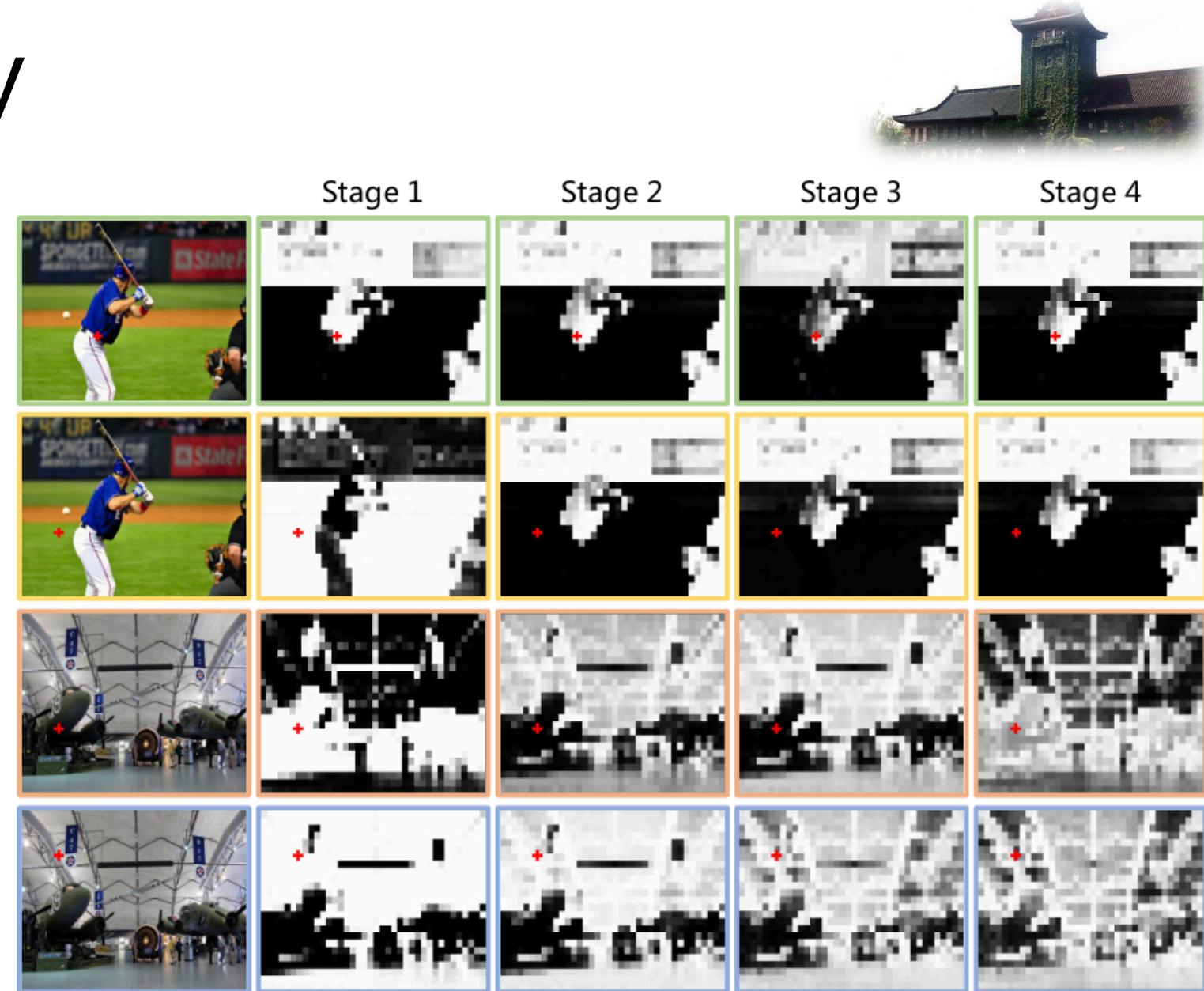




Ablation Study

1. The attention layer at the *shallow* stage mainly focuses on *appearance feature* (e.g., color)

2. The attention layer at the *deep* stage encodes *the global pattern of entire image*, and is *insensitive to the query pixel*





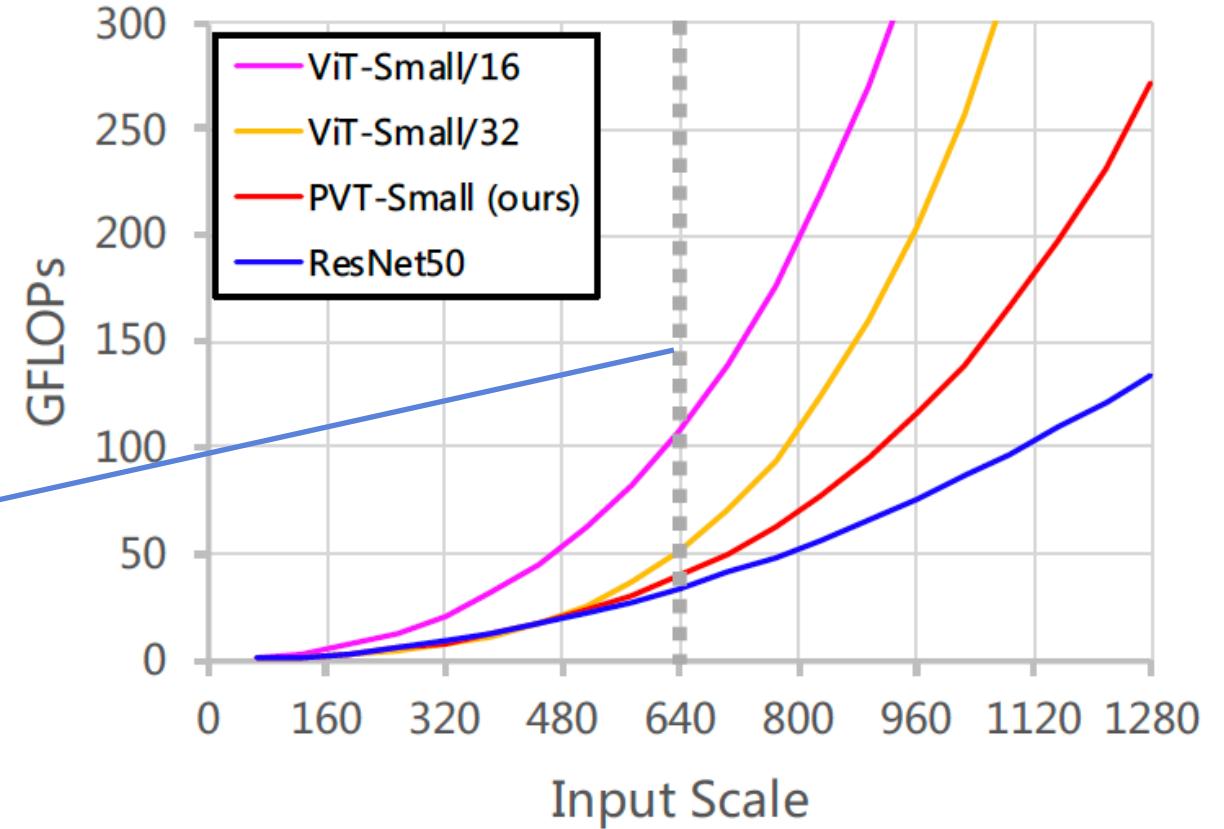
Computation Cost



- FLOPs Growth Rate

$\text{ViT-S/16} > \text{ViT-S/32} > \text{PVT-S} > \text{R50}$

PVT is more suitable for medium-resolution input (640×640 pixels)





Future Work



- Efficient Attention Layer;
- Position Embedding for 2D/3D Images;
- Pure Transformer Vision Models;
- Transformer + NAS/Pruning/Distillation/Quantification;
- Multimodal Transformer (*e.g.*, CV+NLP);
- ...



Thanks
Q&A