Swin Transformer and 5 Reasons to Use Transformer/Attention in Computer Vision

Han Hu

Microsoft Research Asia

June 20th, 2021

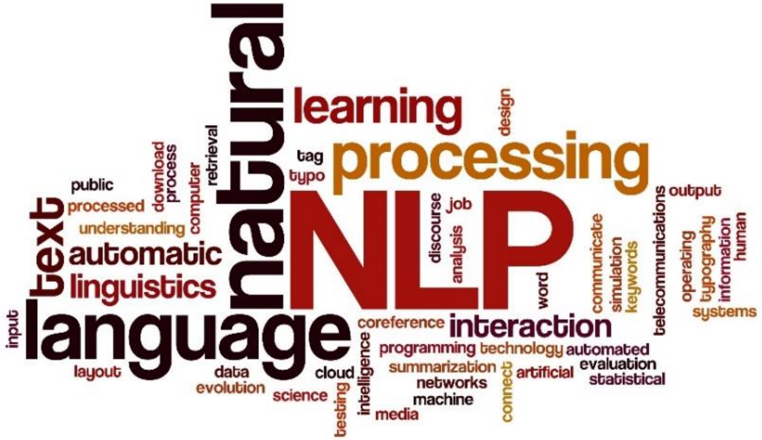
CVPR21, The 3rd Tutorial on "Learning Representations via Graph-structured Networks"

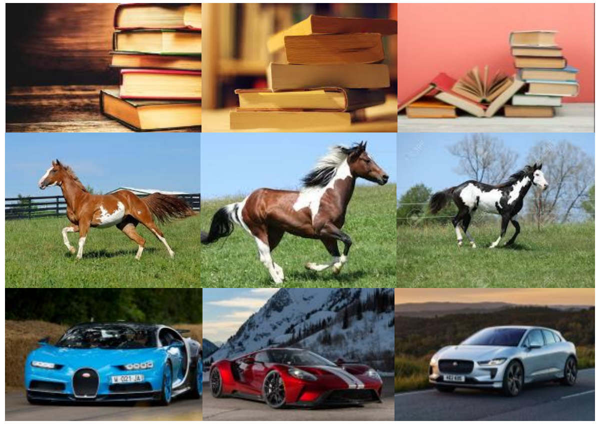
What is the role of Transformer for computer vision?

Transformer (2017.6)

refresh & dominate

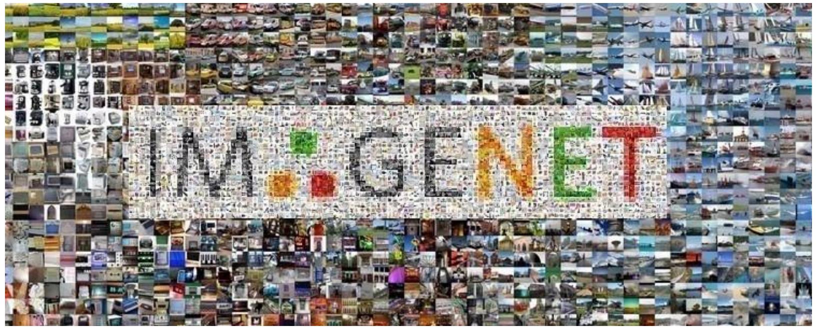
?

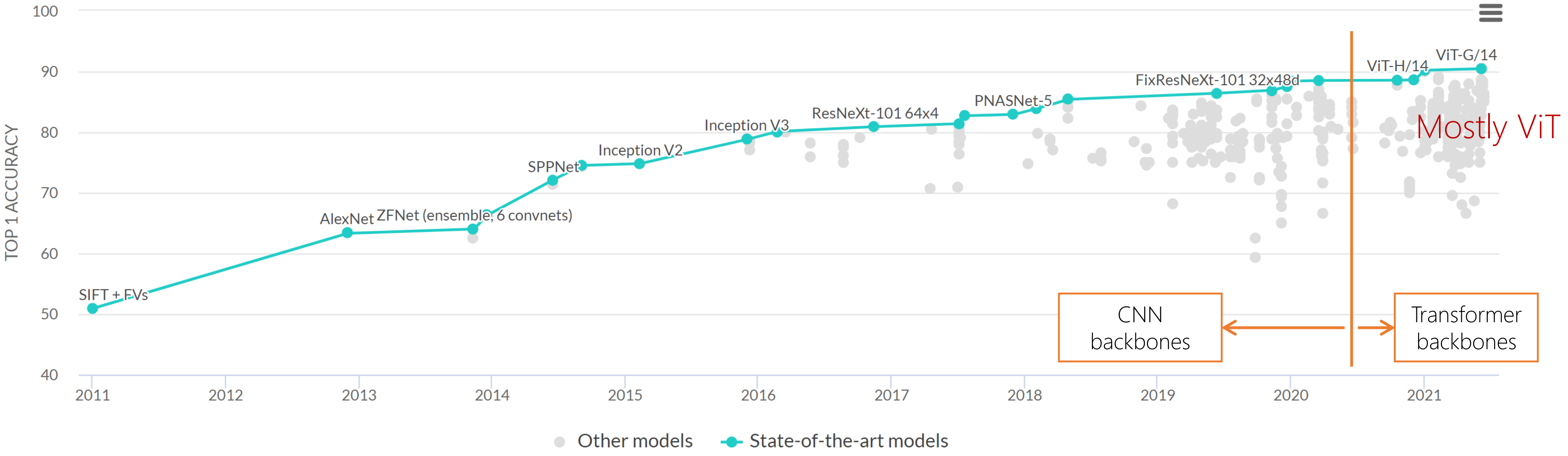




An answer: will also refresh & dominate CV

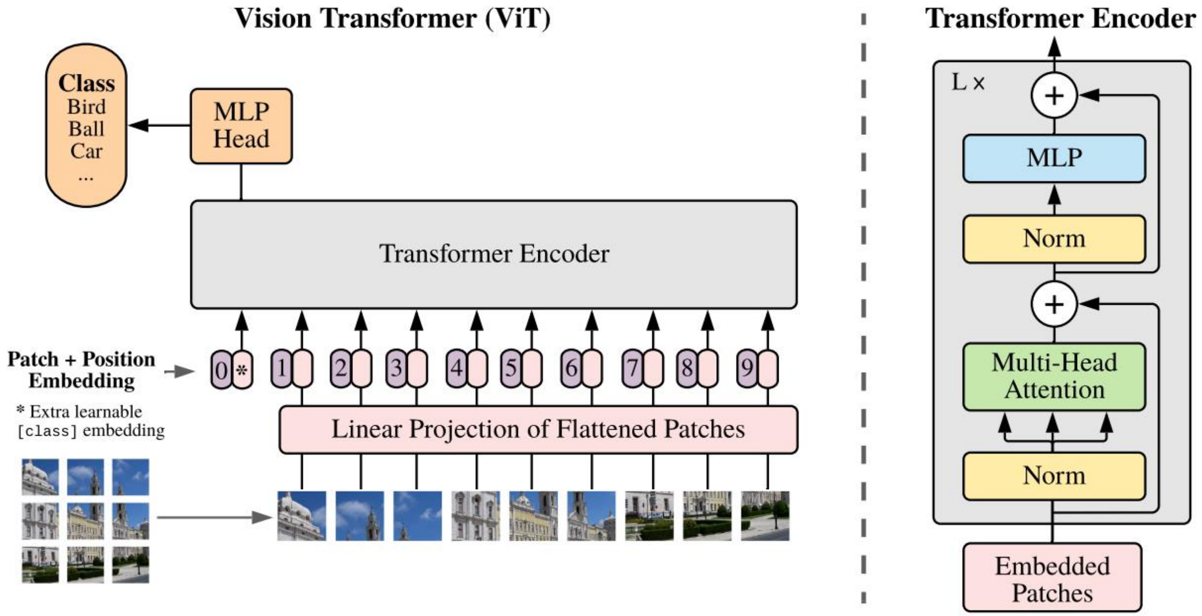
ImageNet-1K image classification





Vision Transformer (ViT, 10/2020)

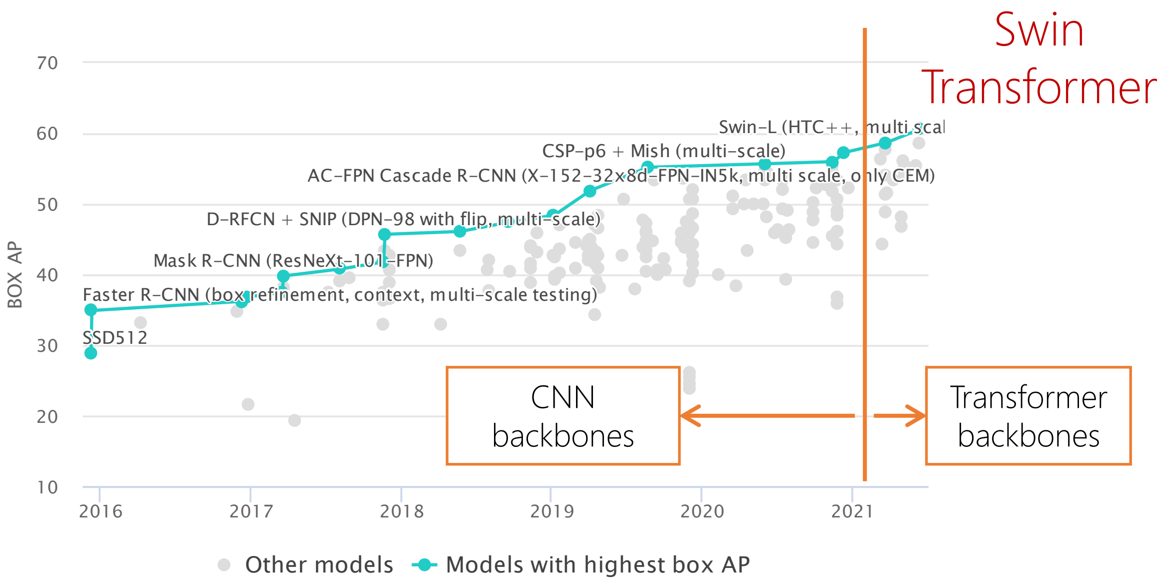
- SOTA performance on Image classification



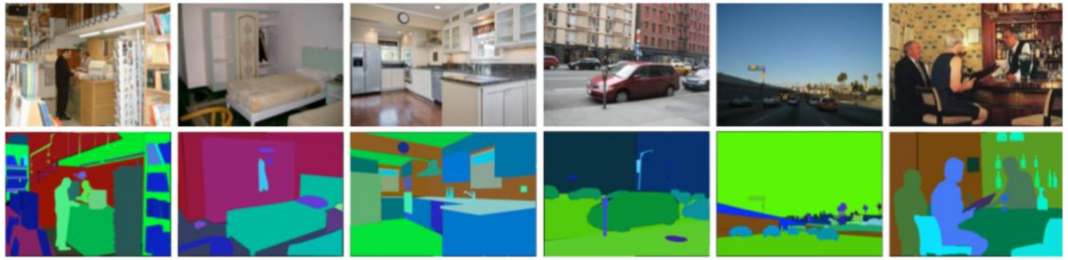
Alexey Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR' 2021

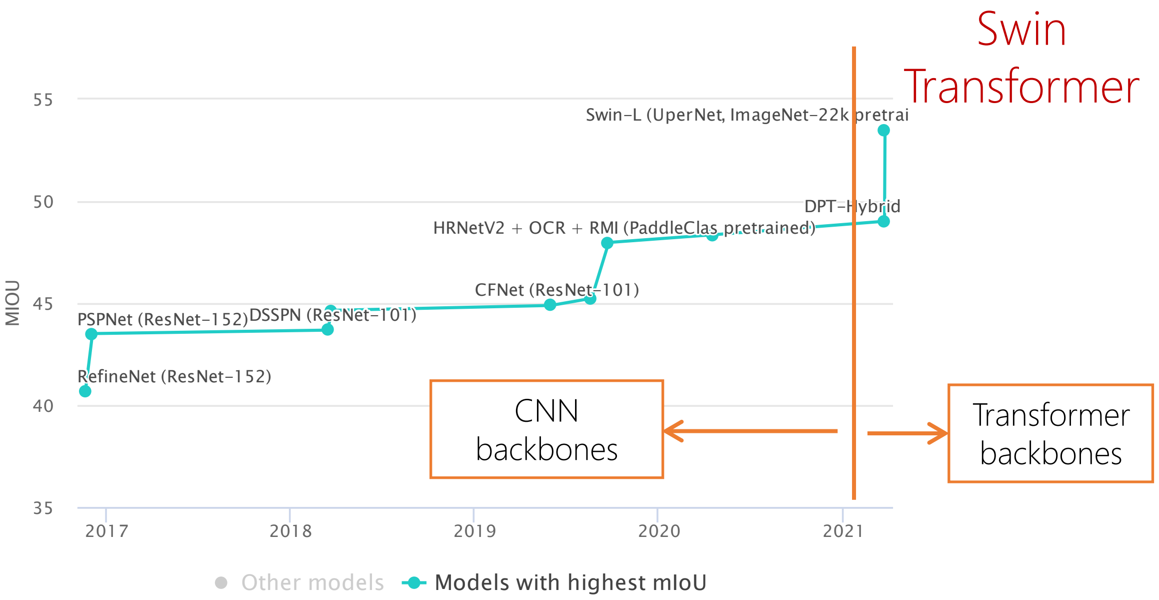
An answer: will also refresh & dominate CV

COCO object detection



ADE20K semantic segmentation

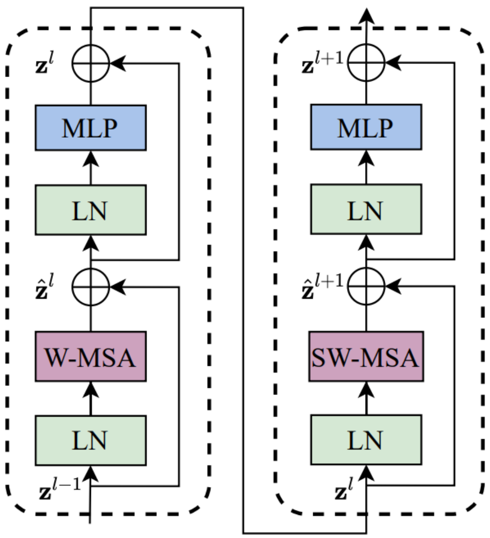




Swin Transformer (03/2021)

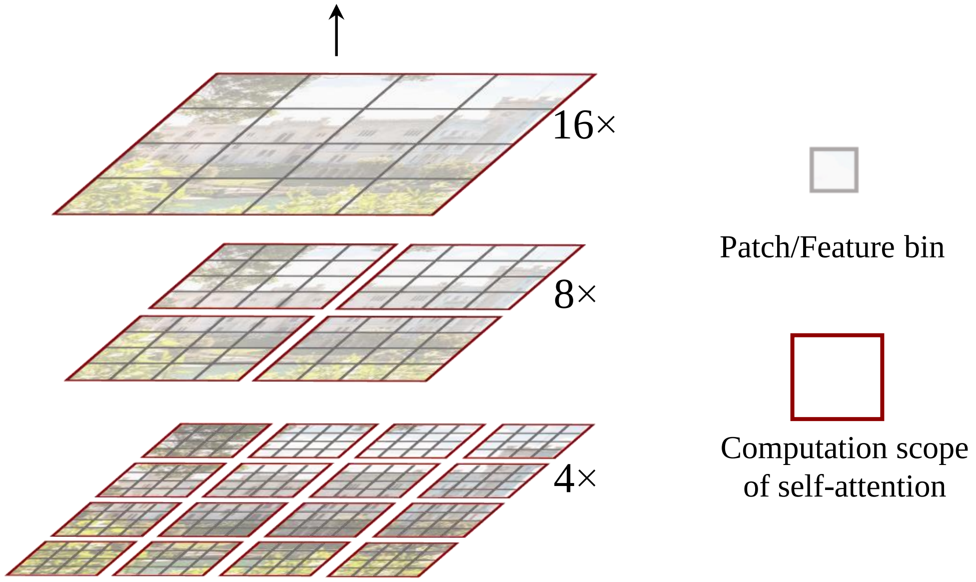
- SOTA performance on object detection and semantic segmentation

Transformer
(strong modeling power)



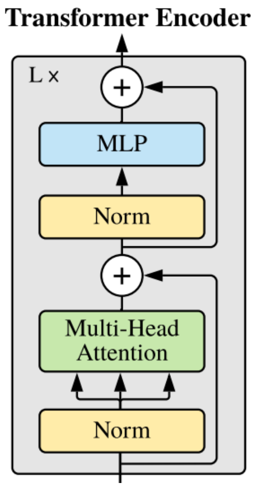
+

good priors for visual signals
(hierarchy / locality / translation invariance)



Ze Liu et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. Arxiv 2021

4 years unleash the power of Transformer in CV



Reason I: General modeling capability

Reason II: Complement convolution

2019.4

2021.1

Reason V: Scalability

2017.06

2017.11

Reason III: Strong modeling power

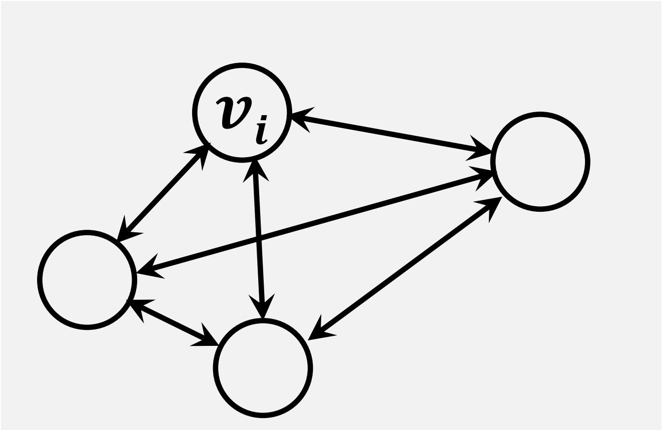
Reason IV: Better connect vision and language

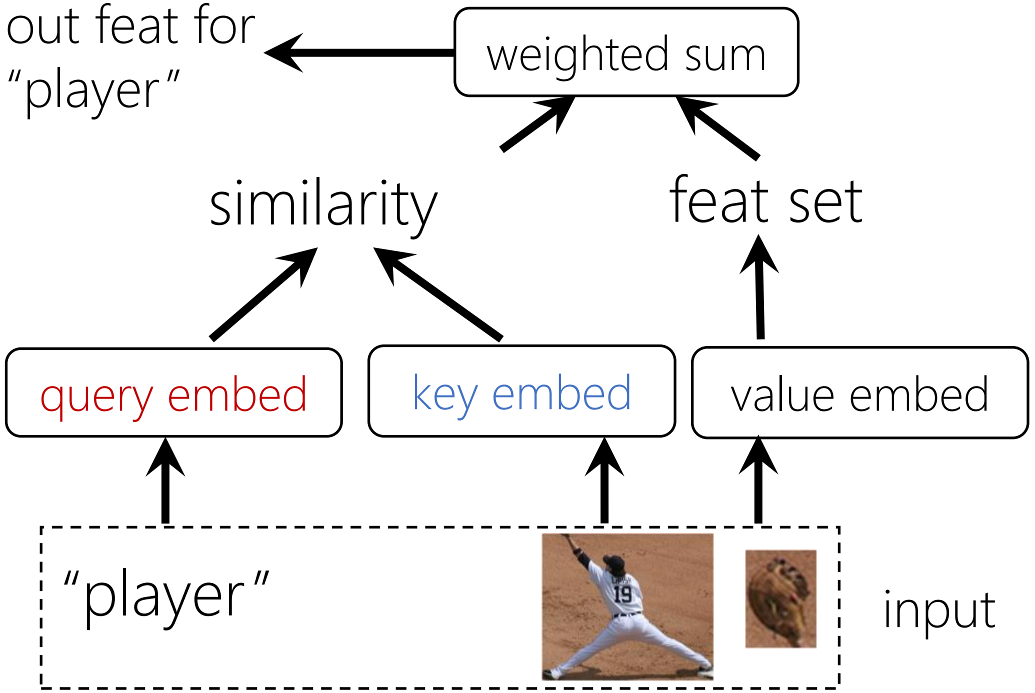
2021.6

Swin Transformer: a general-purpose backbone

Reason I to use Transformer in computer vision

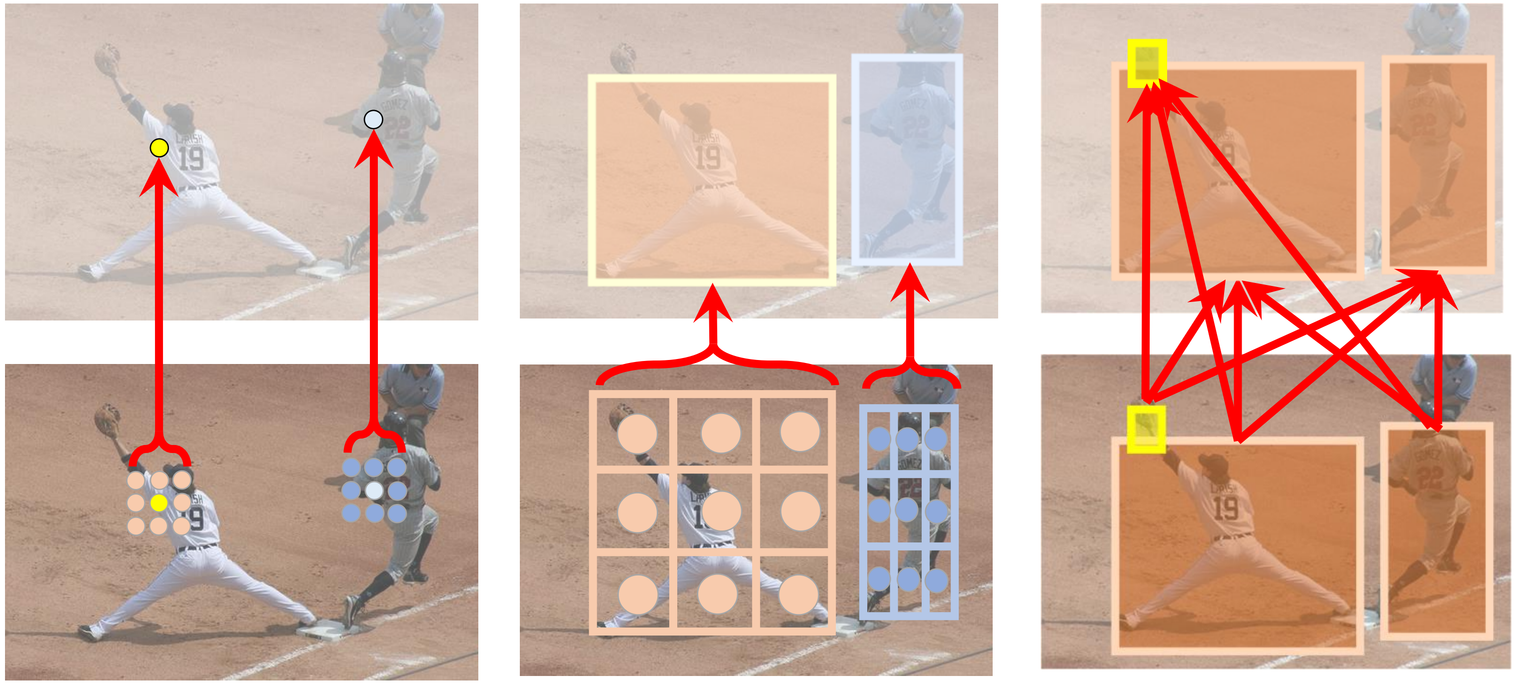
- General modeling capability
  - All concepts (concrete or abstract) and their relationships can be modeled by a graph
  - Modeling arbitrary relationship via <u>verification</u>, which is hard by CNN





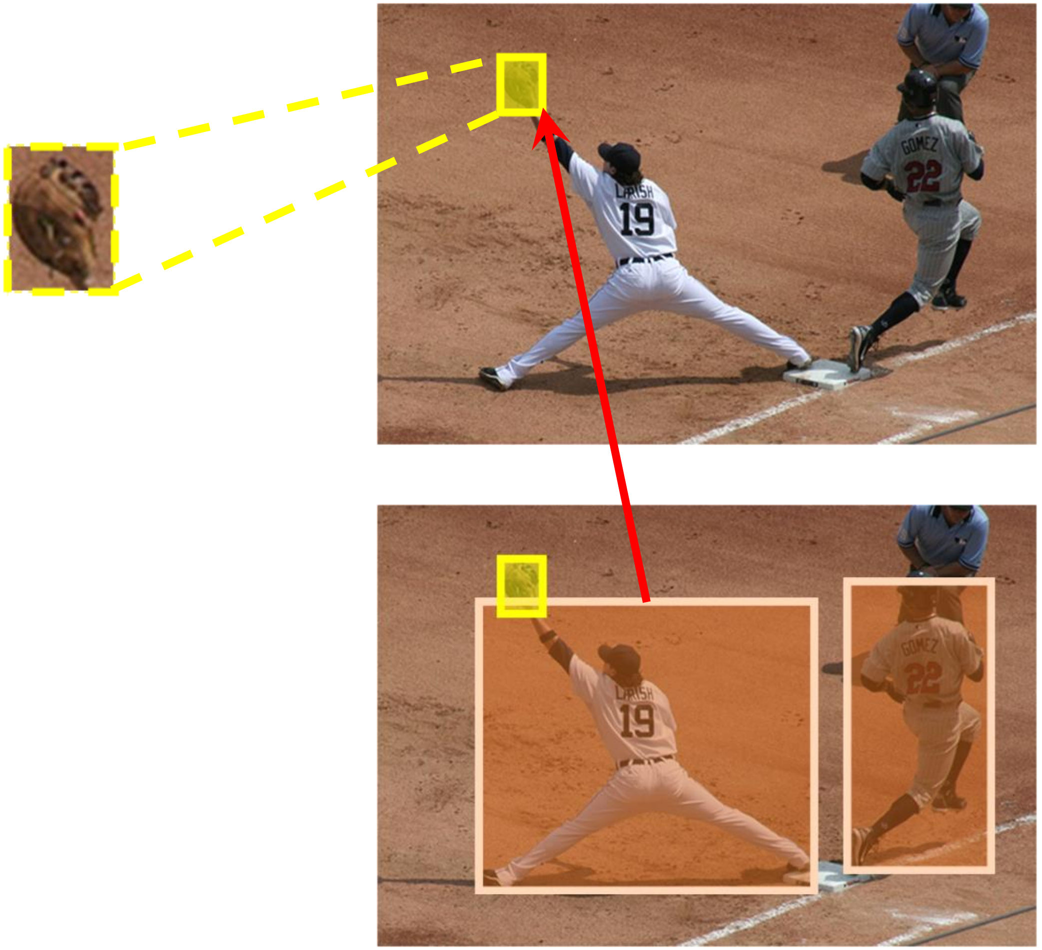
Reason I to use Transformer in computer vision

- General modeling capability
  - Can model all of pixel-to-pixel, object-to-pixel, object-to-object relationships



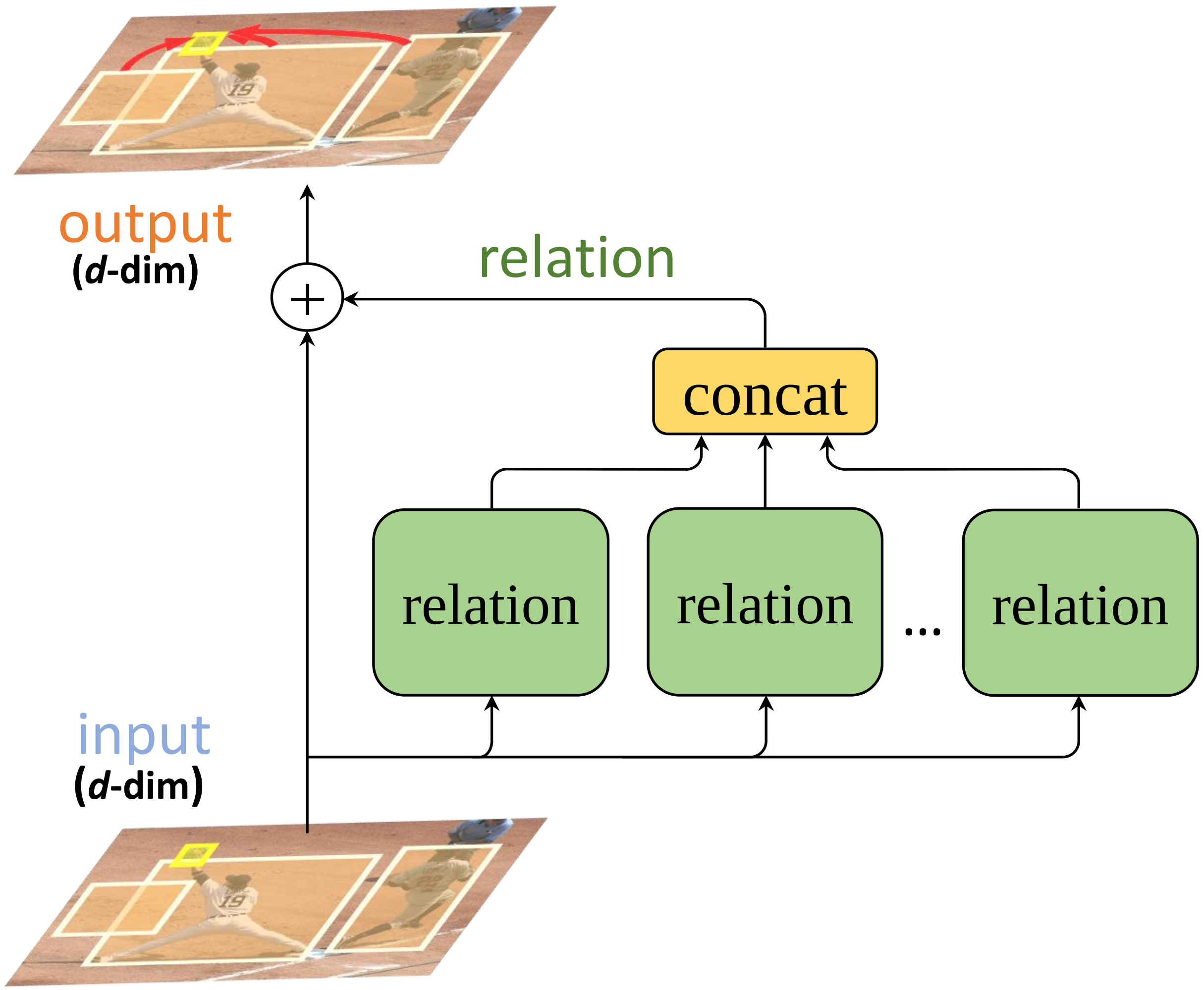
pixel-to-pixel object-to-pixel object-to-object

Relation Networks for Object Detection (CVPR'2018)



It is much easier to detect the ***glove*** if we know there is a ***baseball player***.

Relation Networks for Object Detection (CVPR'2018)



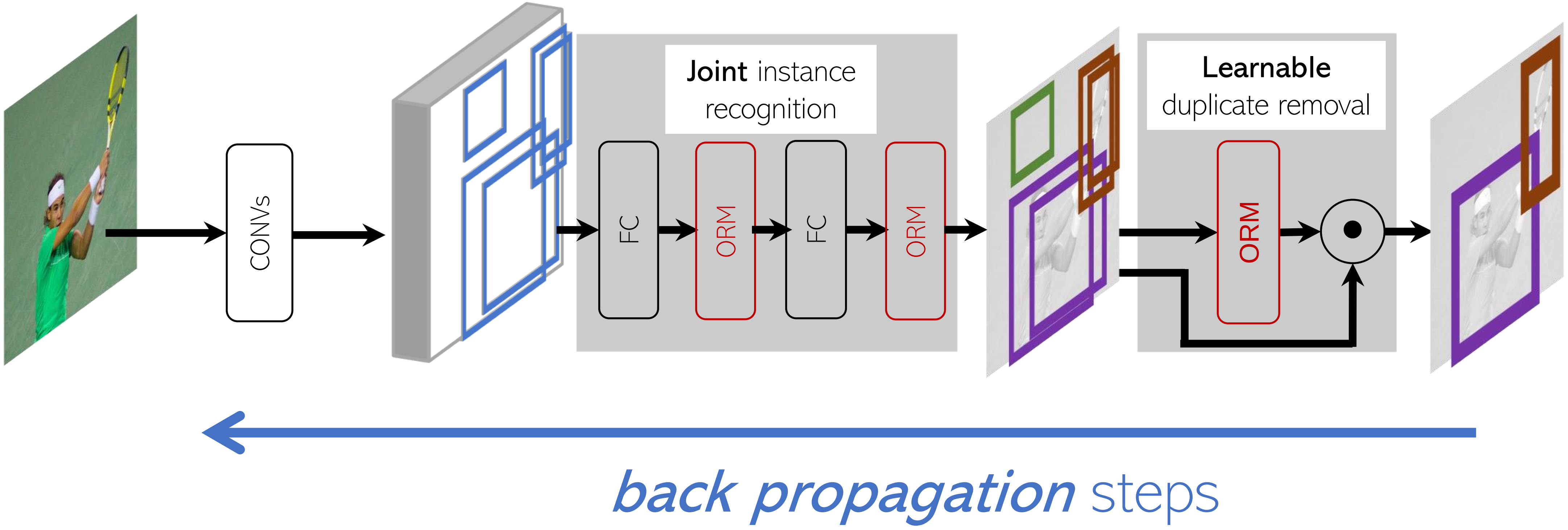
Key Feature

✓Relative position

Han Hu et al. *Relation Networks for Object Detection*. CVPR 2018

Relation Networks for Object Detection (CVPR'2018)

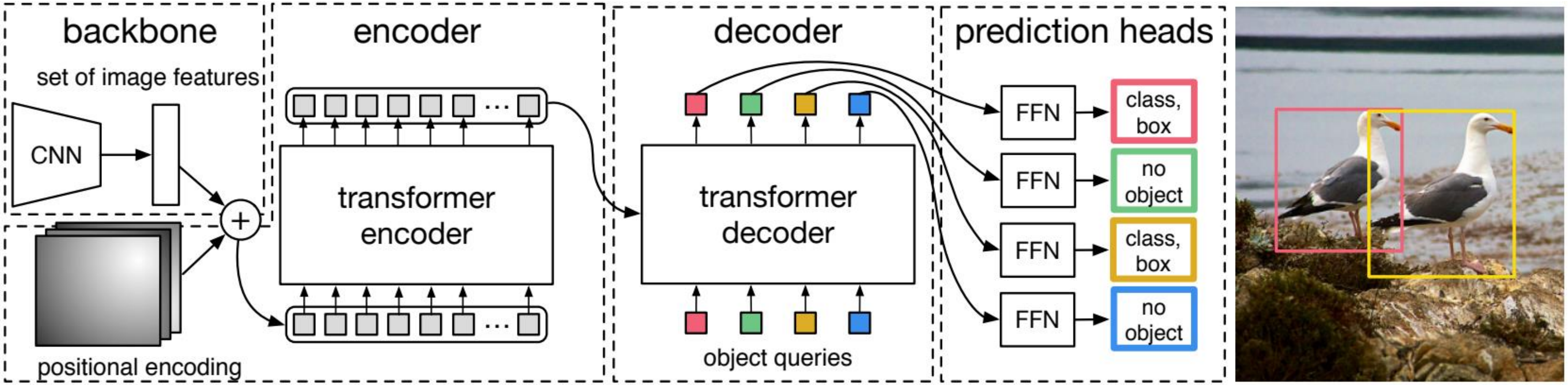
- The first fully end-to-end object detector



Han Hu et al. *Relation Networks for Object Detection*. CVPR 2018

DeTR (ECCV'2020)

- Another end-to-end object detector



Nicolas Carion et al. End-to-End Object Detection with Transformers. ECCV 2020

Reason II to use Transformer in computer vision

- Complement convolution
  - "Convolution is too local!"
  - Global (Transformer) vs. local (conv.)

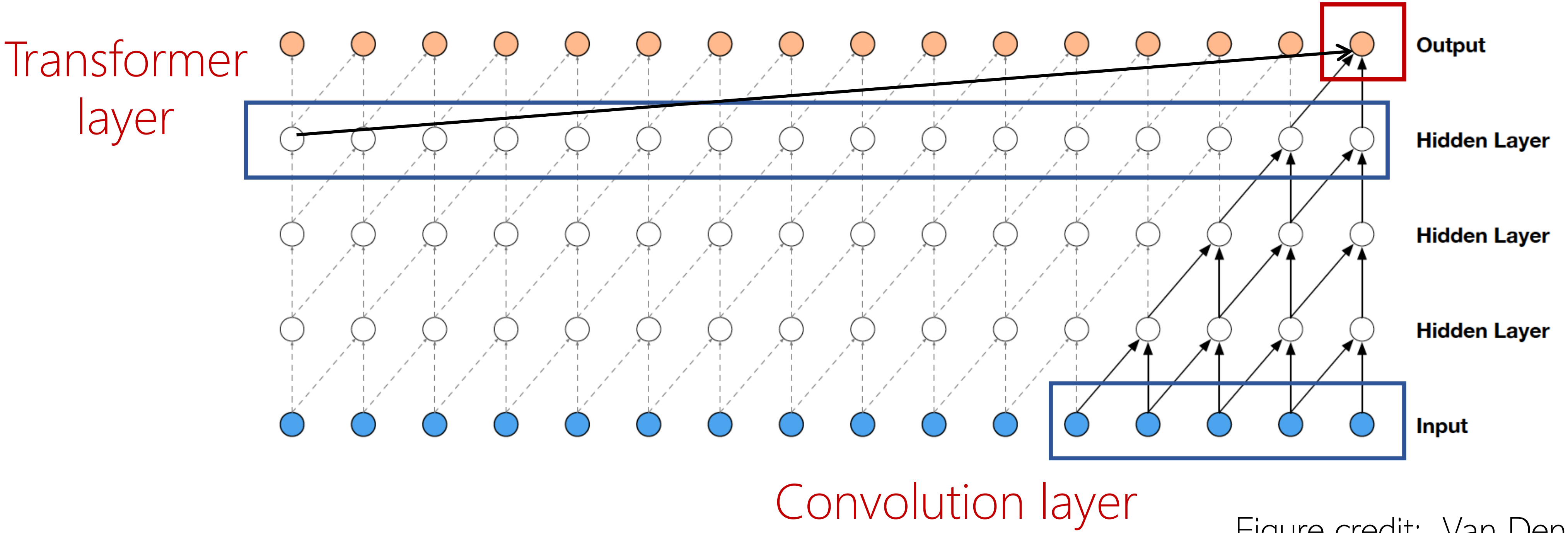
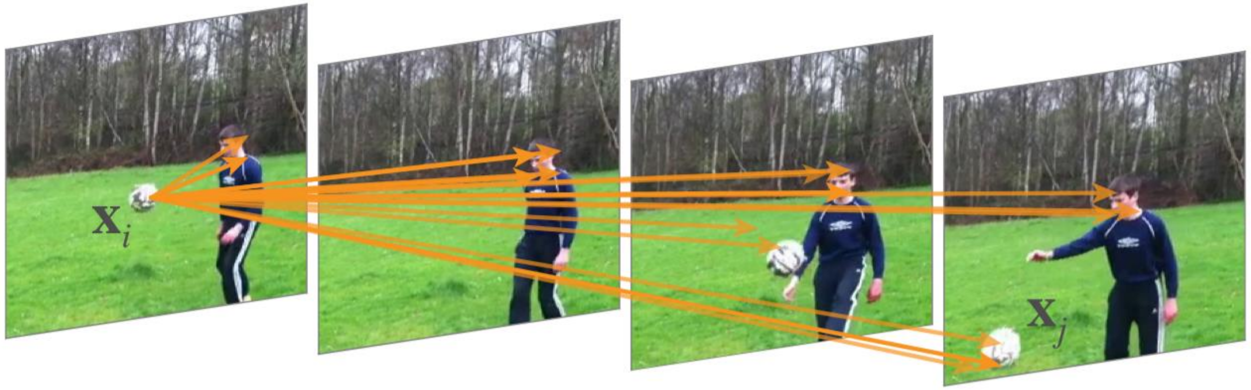
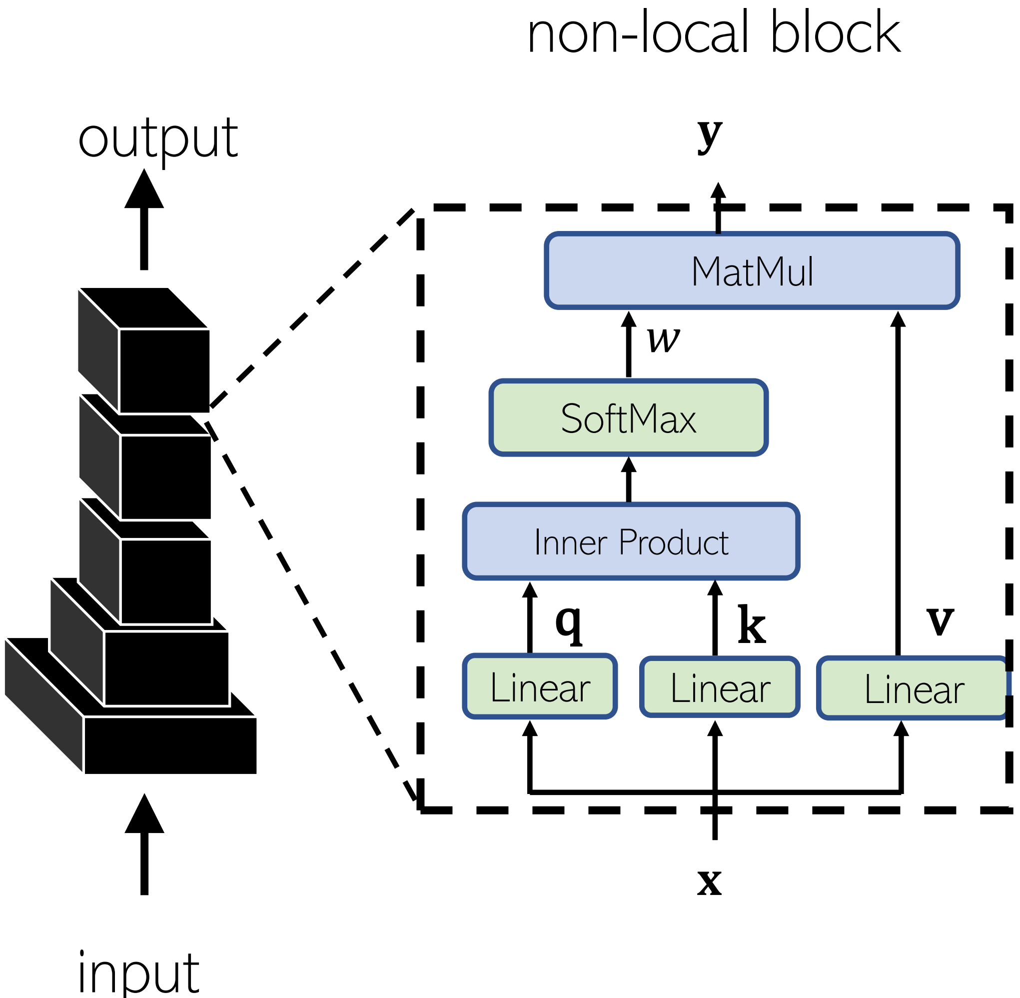


Figure credit: Van Den Oord et al.

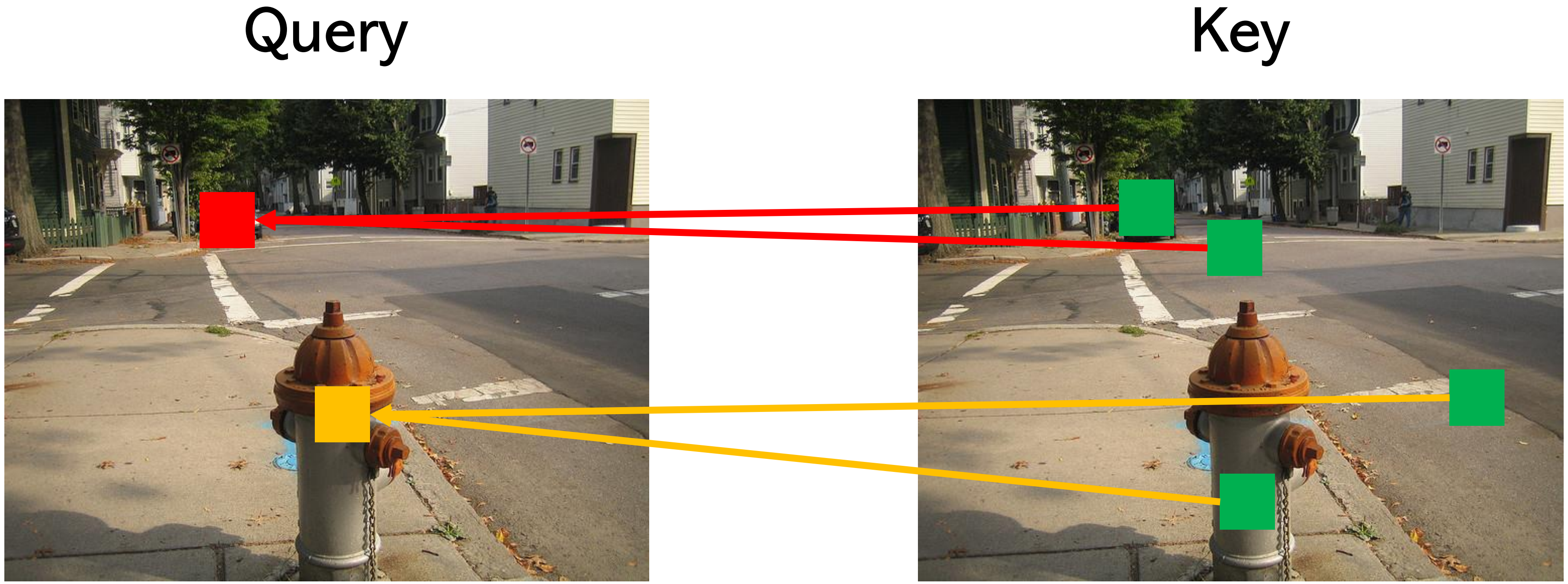
Non-local networks (CVPR'2018)





The Degeneration Problem of NLNet

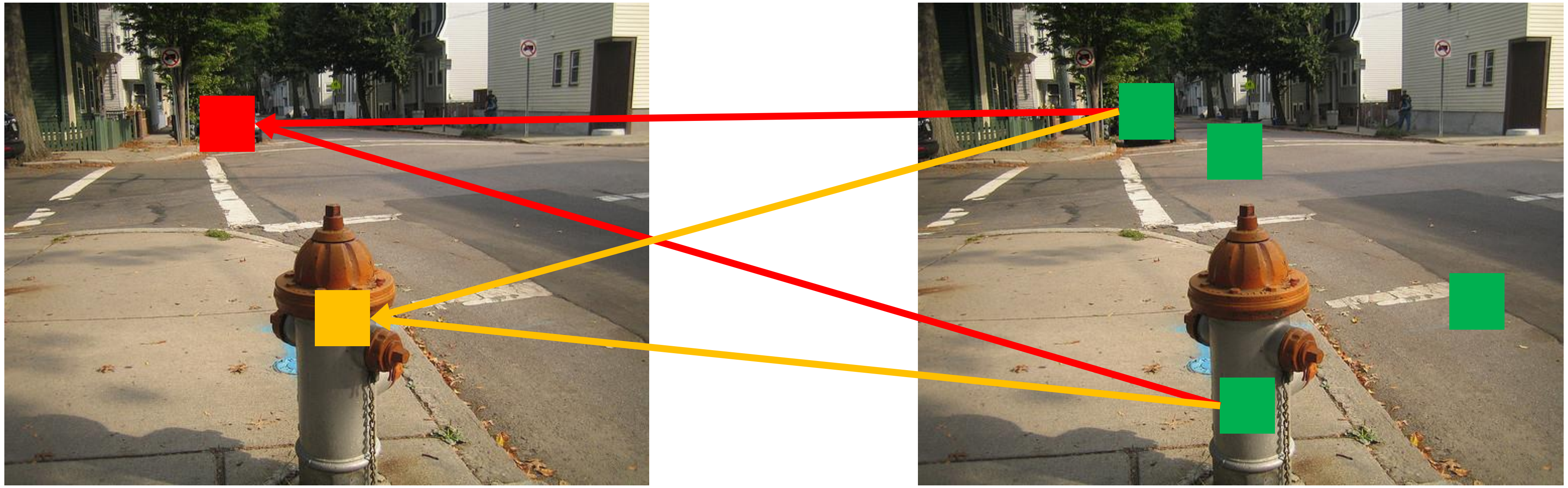
- Expectation of Ideally Learnt Relation
  - Different queries affected by **different** key



The Degeneration Problem of NLNet

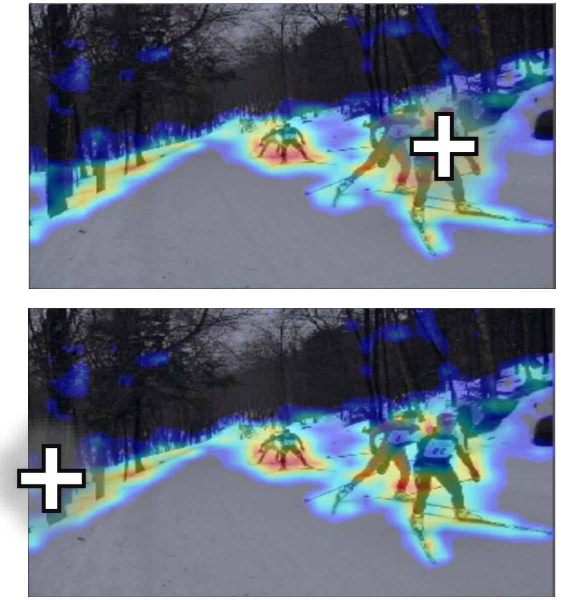
- What does the Self-Attention Learn?
  - Different queries affected by the **same** keys

Query **Key**

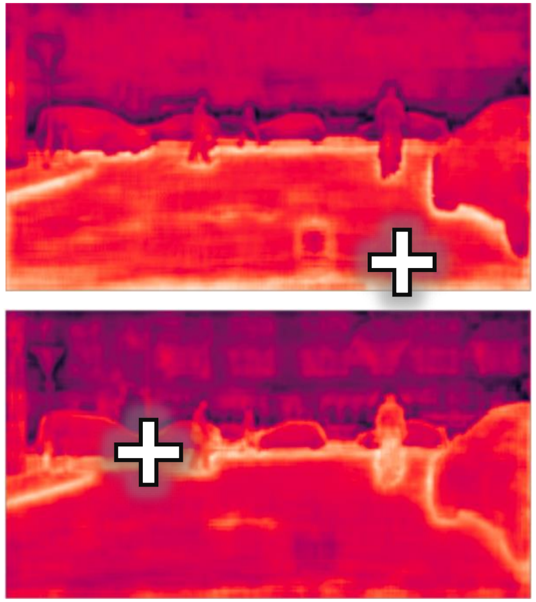


Visualizations on Real Tasks

- indicates the query point
- The activation map for different queries are similar
- The self-attention model degenerates to a unary model



Object Detection



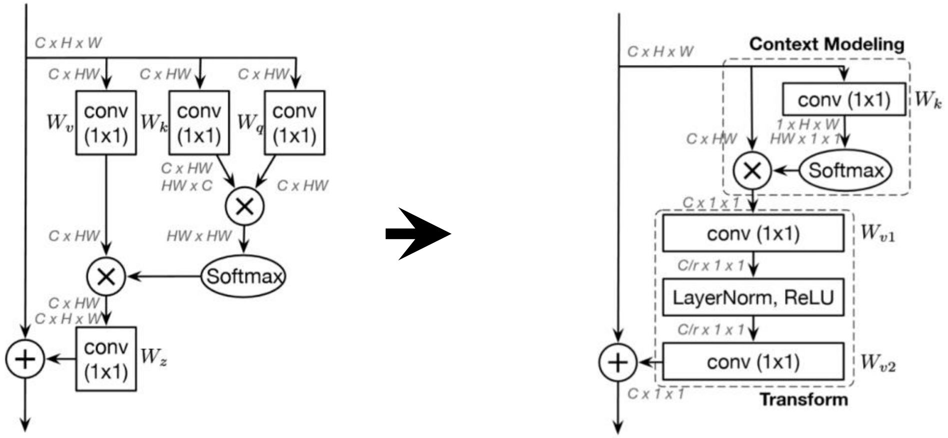
Semantic Segmentation

[GCNet, ICCVW'2019]

https://arxiv.org/pdf/1904.11492.pdf

GCNet (ICCVW'2019, PAMI'2021)

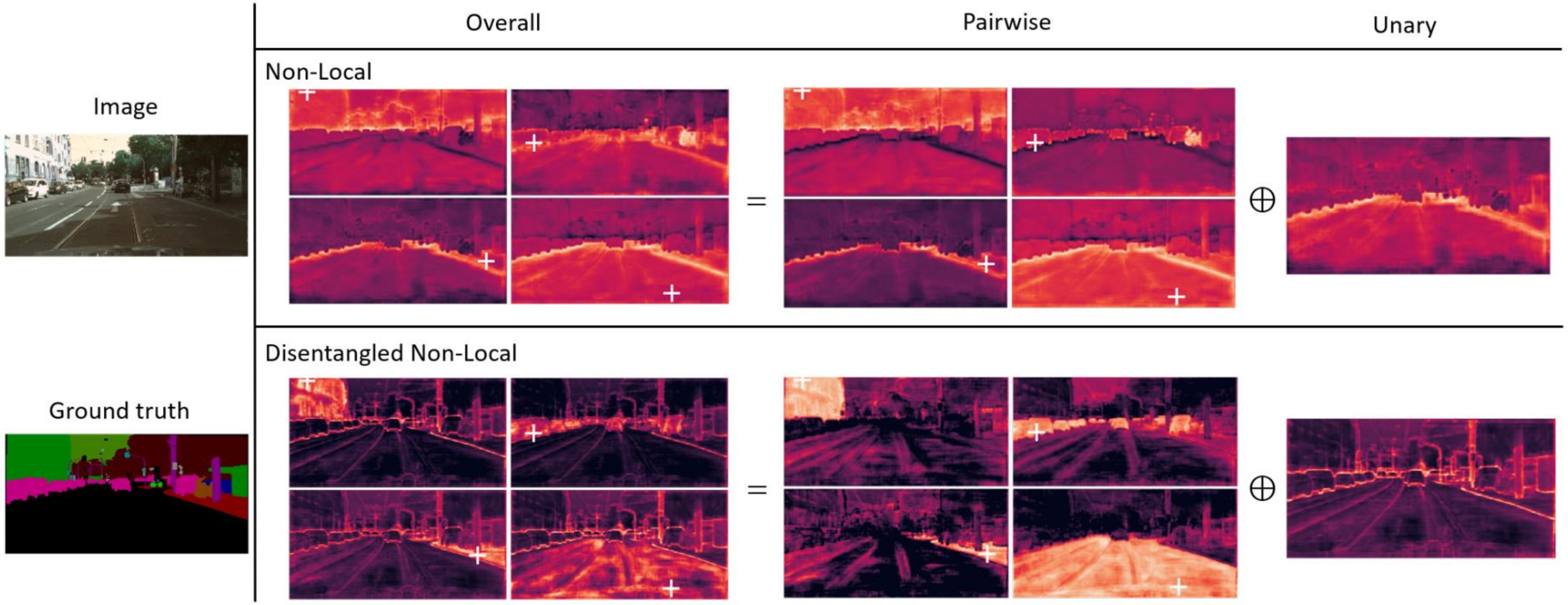
- Find the degeneration issue in computer vision
- Explicitly leverage degenerated formulation for better efficiency



	Non-Local Block	**Global Context Block**	**Reduction ratio**
FLOPs	9.3G	**4.0M**	**2,300x**
model size	2.1M	**0.1M**	**20x**
accuracy (mAP)	38.0	**38.1**	**unchanged**

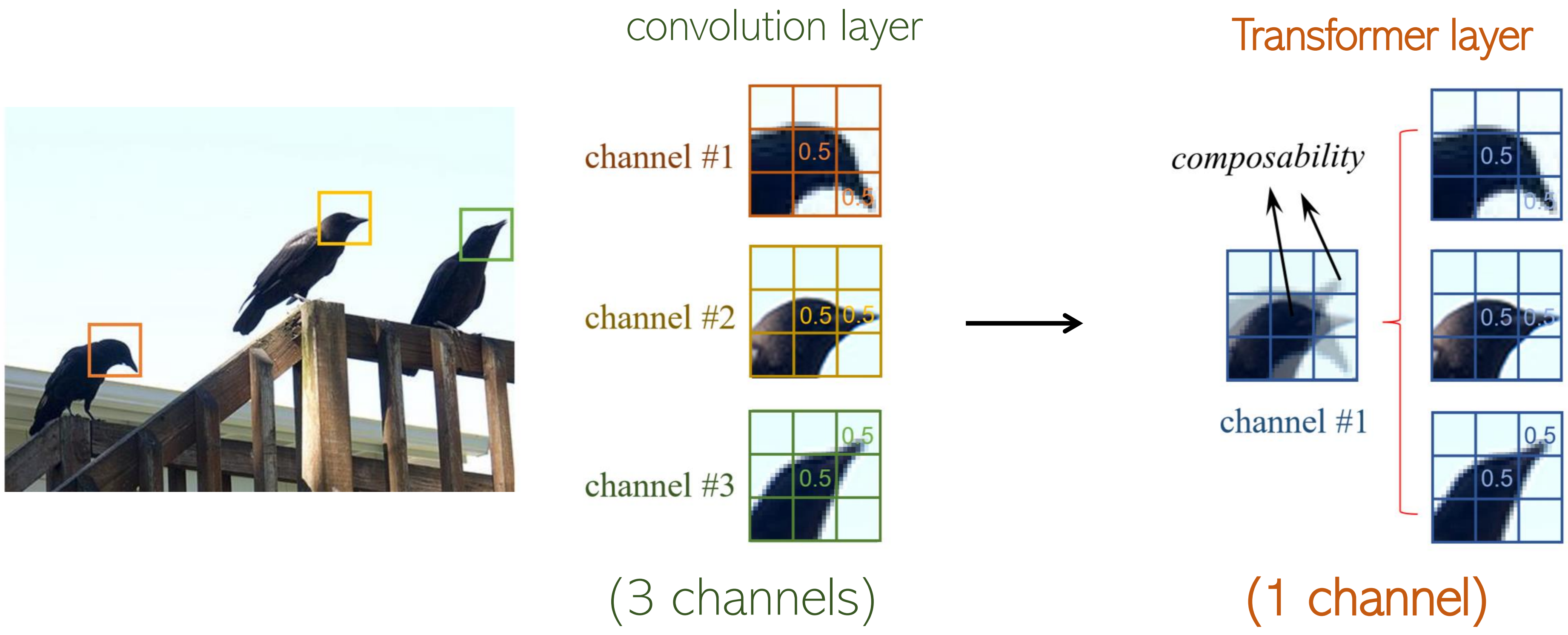
Disentangled non-local networks (ECCV'2020)

- Solve the degeneration problem



Reason III to use Transformer in computer vision

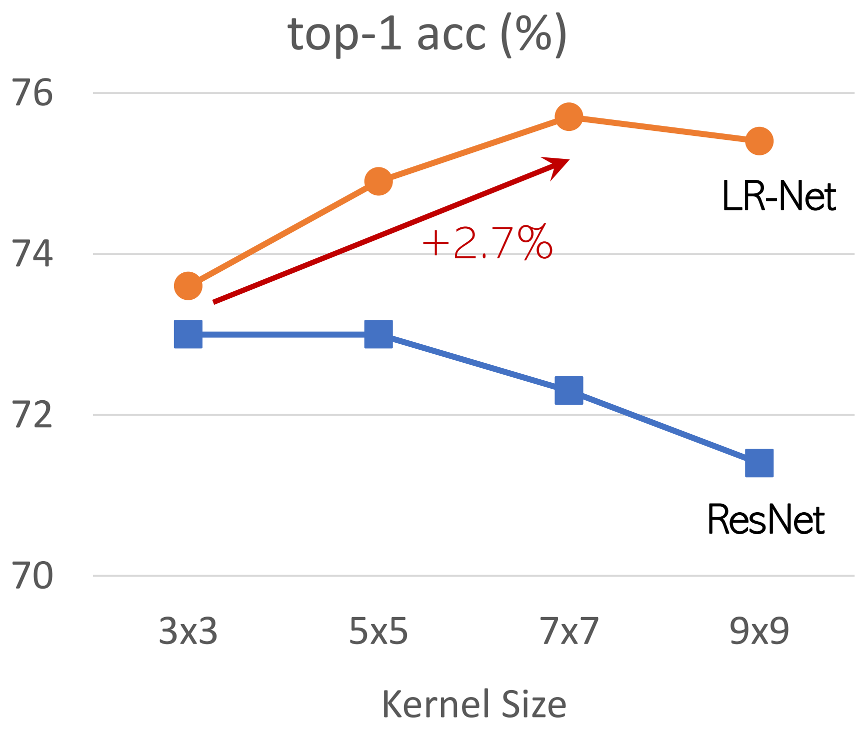
- Powerful due to adaptive computation
  - “Convolution is exponentially inefficient!”



Local relation networks (2019.4)

- Transformer as backbones

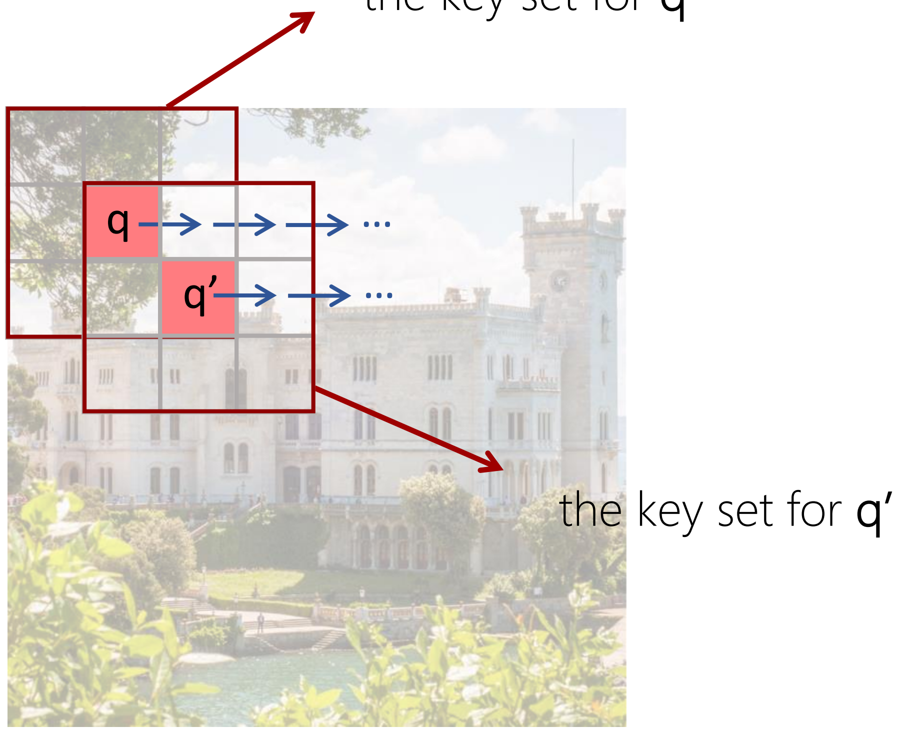
		ResNet	LR-Net
stage	output	ResNet-50	**LR-Net-50 (7×7, m=8)**
res1	112×112	7×7 conv, 64, stride 2	**1×1, 64** **7×7 LR, 64, stride 2**
res2	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 1\times1, 64 \\ 3\times3 \text{ conv}, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	3×3 max pool, stride 2 $\begin{bmatrix} 1\times1, 100 \\ \mathbf{7\times7 \text{ LR}, 100} \\ 1\times1, 256 \end{bmatrix} \times 3$
res3	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3 \text{ conv}, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 200 \\ \mathbf{7\times7 \text{ LR}, 200} \\ 1\times1, 512 \end{bmatrix} \times 4$
res4	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3 \text{ conv}, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 400 \\ \mathbf{7\times7 \text{ LR}, 400} \\ 1\times1, 1024 \end{bmatrix} \times 6$
res5	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3 \text{ conv}, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 800 \\ \mathbf{7\times7 \text{ LR}, 800} \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params		**25.5**×10^6	**23.3**×10^6
FLOPs		**4.3**×10^9	**4.3**×10^9



Han Hu et al. *Local Relation Networks for Visual Recognition. ICCV 2019*

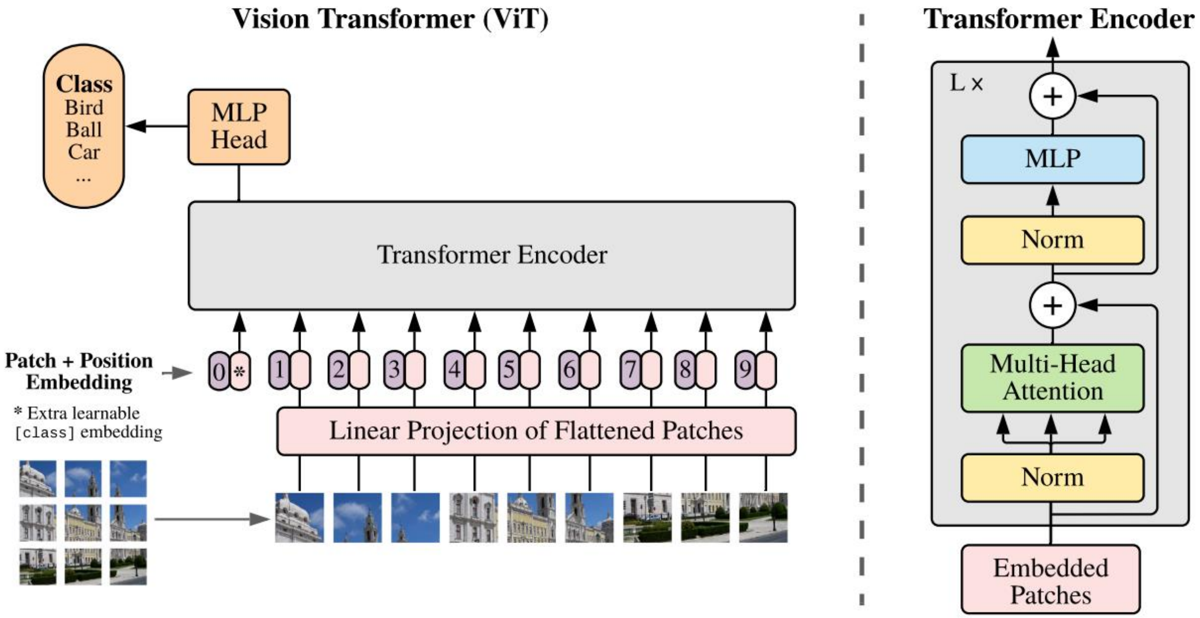
But … slow in real computation

- Because different queries use different key sets



Vision Transformer (ViT)

- by Google Brain (2020.10)



Computation speed

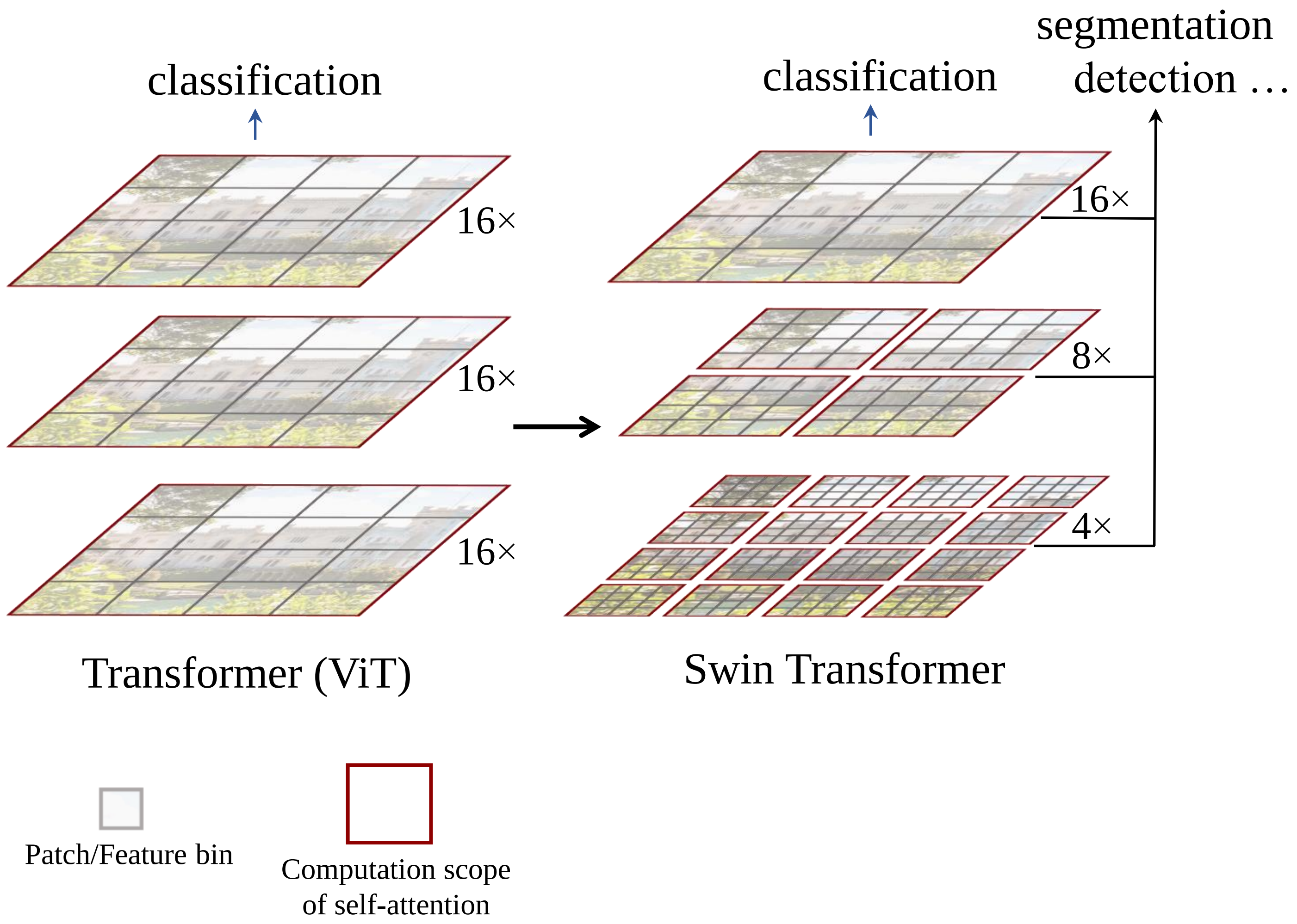
Model	FLOPs	Speed (TPU)
R50	4.3G	~2100 im/s
ViT-B/32	4.3G	~3000 im/s

+50% speed-up

Alexey Dosovitskiy et al. an Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR' 2021

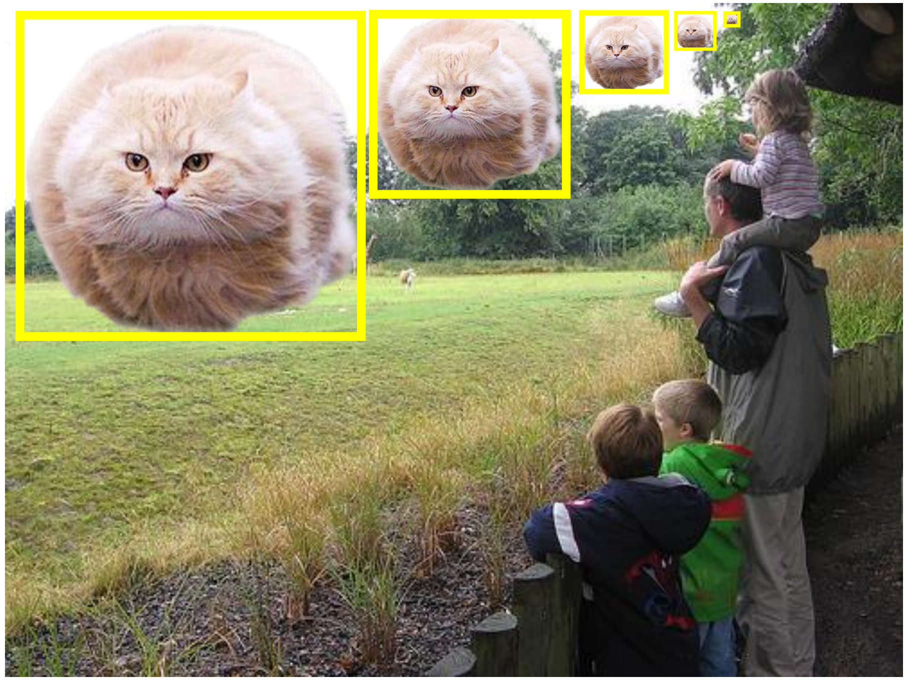
Swin Transformer =

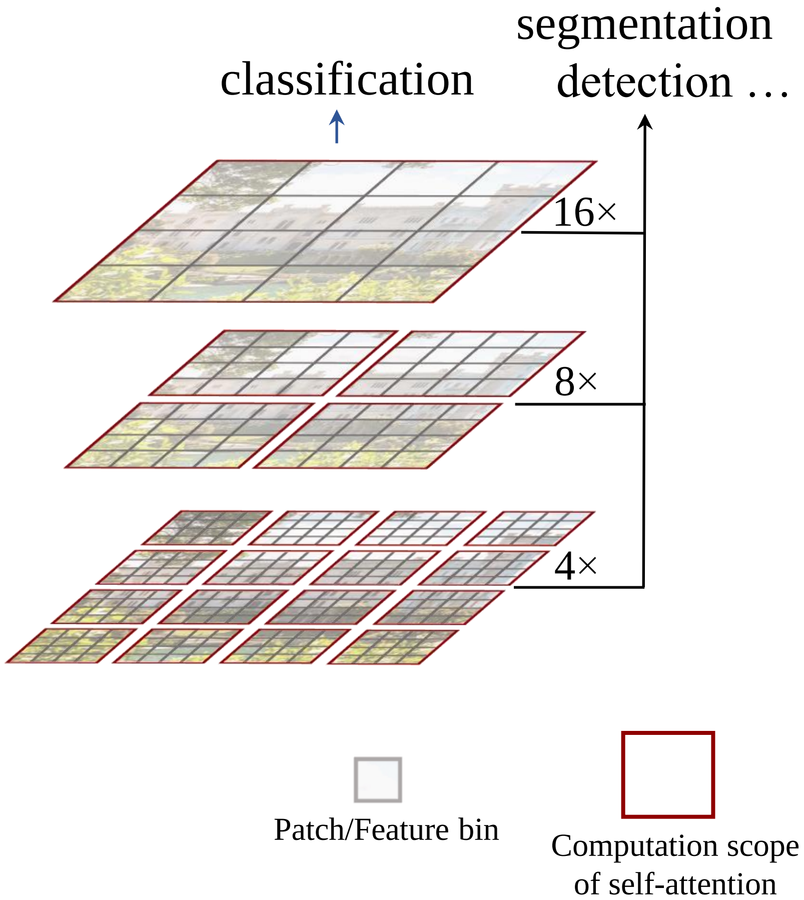
- Transformer
  - Strong modeling power
- + good priors for visual modeling
  - Hierarchy
  - Locality
  - Translational invariance



Hierarchy

- Processing objects of different scales

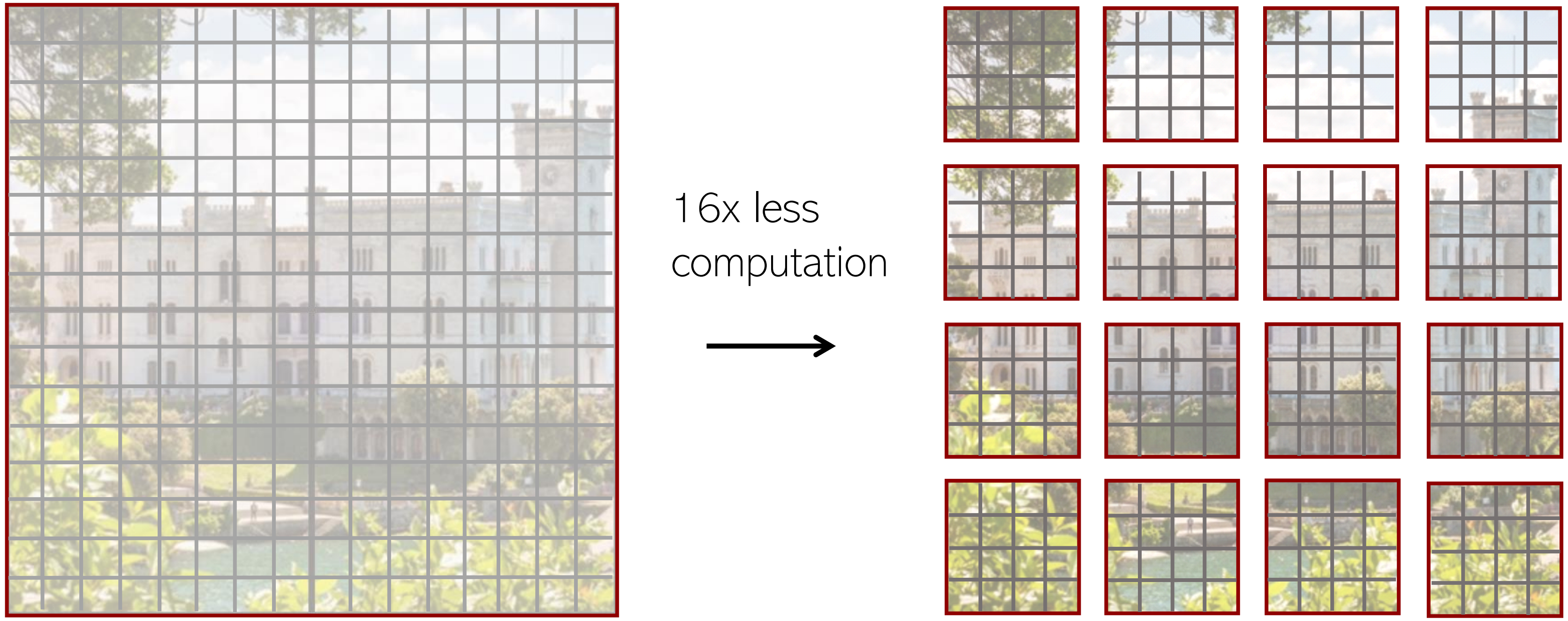




Left figure credit by Ross Girshick

Locality by non-overlapped windows

- Proves beneficial in modeling the high correlation in visual signals (Yann LeCun)
- Linear complexity with increasing image resolution: from $O(n^2)$ to $O(n)$

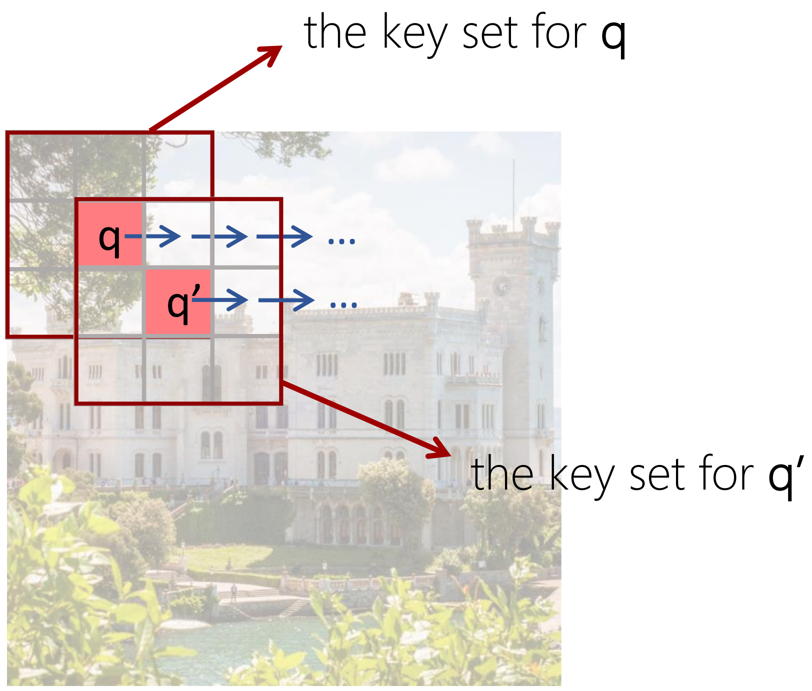


ViT: 256^2=65536 (Global)

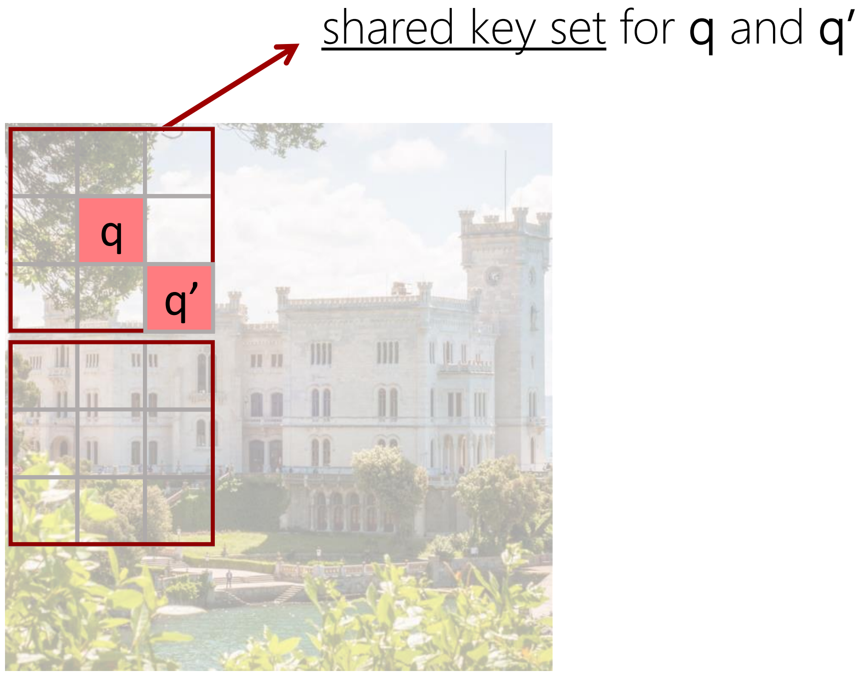
Swin Transformer: 16×16^2=4096 (Local)

Locality by non-overlapped windows

- Compared to sliding window (LR-Net)
  - Shared key set enables friendly memory access and is thus good for speed (larger than 3x)



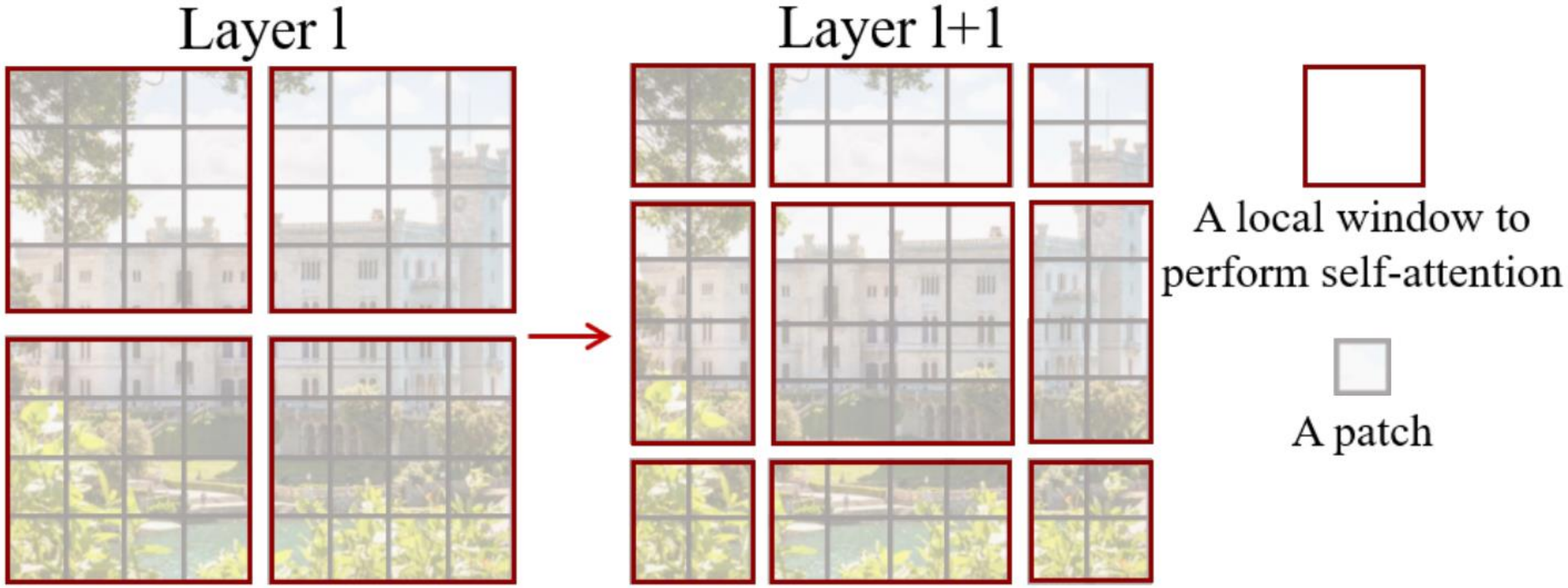
sliding window (LR-Net)



Non-overlapped window (Swin Transformer)

Shifted non-overlapped windows

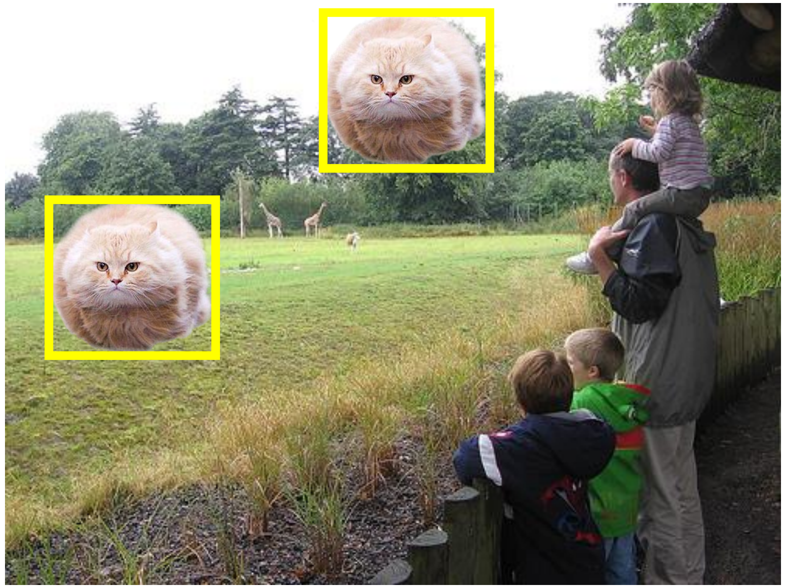
- Enable cross-window connection
  - Non-overlapped windows will result in no connection between windows
  - Performs as effective or even slightly better than the sliding window approach, due to regularization effects

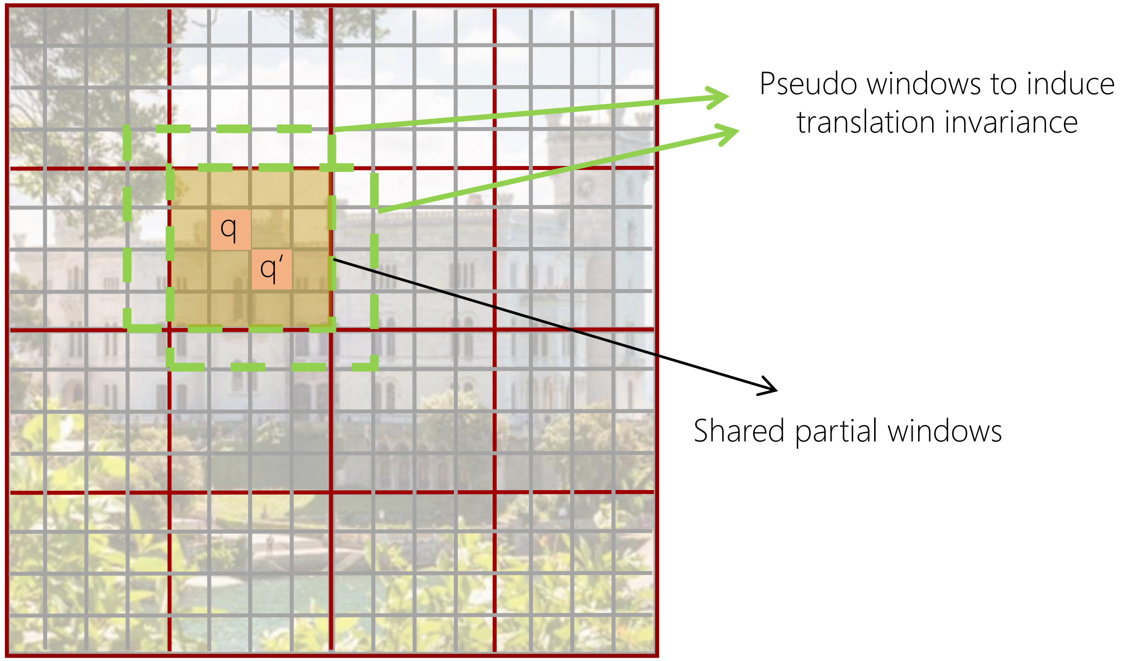


Translational semi-invariance

- Relative position bias plays a more important role in vision than in NLP

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + \boxed{B})V,$$

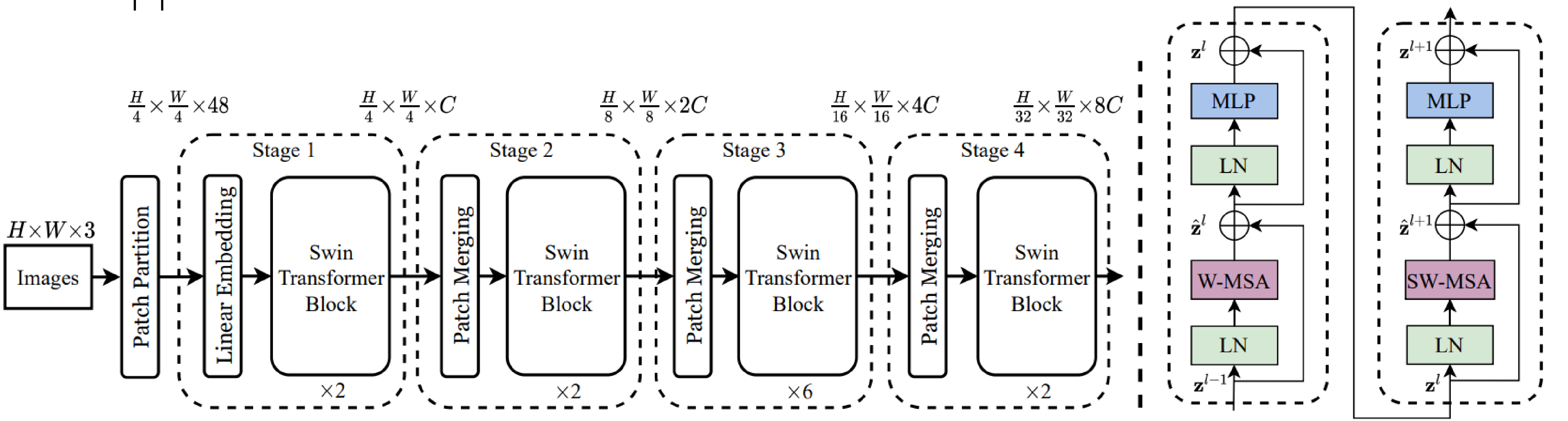




semi-invariance is as effective as full-invariance in our experiments

Architecture instantiations

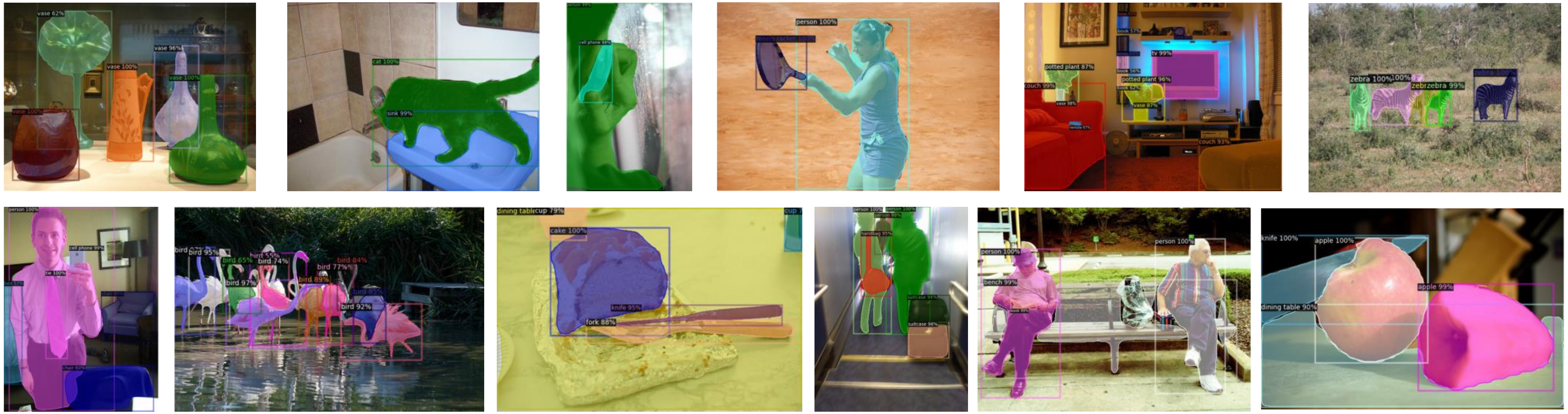
- Resolution of each stage is set similar as ResNet, to facilitate application to down-stream tasks



(a) Architecture

(b) Two Successive Swin Transformer Blocks

Application: object detection



- COCO object detection: #1 #2 #3 for single model (60.6 mAP)
  - Significantly surpass all previous CNN models (+3.5 mAP)
- COCO instance segmentation: #1 for single model (52.4 mAP)
  - Significantly surpass all previous CNN models (+3.3 mAP)

Application: object detection

- Performs consistently better than CNN on various object detectors and various model sizes (+3~4.5 mAP)

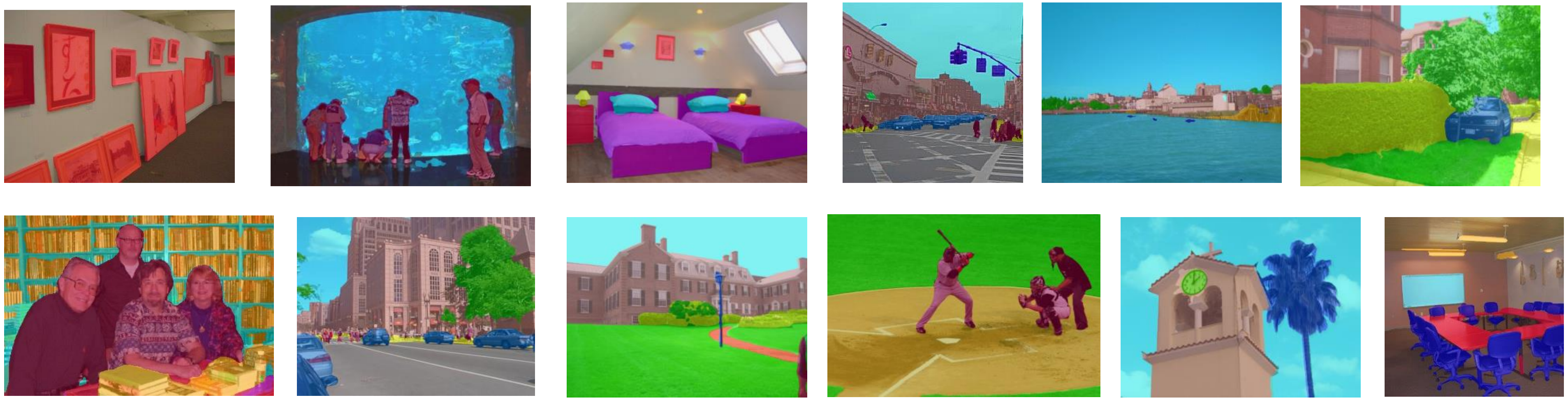
(a) Various frameworks

Method	Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	#param.	FLOPs	FPS	
Cascade	R-50	46.3	64.3	50.5	82M	739G	18.0	
Mask R-CNN	Swin-T	**50.5**	**69.3**	**54.9**	86M	745G	15.3	+4.2
ATSS	R-50	43.5	61.9	47.0	32M	205G	28.3	
	Swin-T	**47.2**	**66.5**	**51.3**	36M	215G	22.3	+3.7
RepPointsV2	R-50	46.5	64.6	50.3	42M	274G	13.6	
	Swin-T	**50.0**	**68.5**	**54.2**	45M	283G	12.0	+3.5
Sparse	R-50	44.5	63.4	48.2	106M	166G	21.0	
R-CNN	Swin-T	**47.9**	**67.3**	**52.3**	110M	172G	18.4	+3.4

(b) Various backbones w. Cascade Mask R-CNN

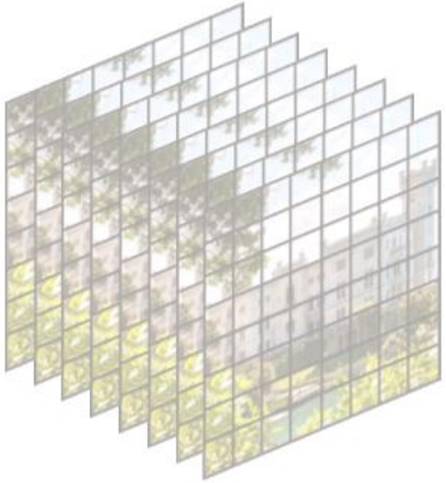
	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	param	FLOPs	FPS	
DeiT-S†	48.0	67.2	51.7	41.4	64.2	44.3	80M	889G	10.4	
R50	46.3	64.3	50.5	40.1	61.7	43.4	82M	739G	18.0	
Swin-T	**50.5**	**69.3**	**54.9**	**43.7**	**66.6**	**47.1**	86M	745G	15.3	+4.2
X101-32	48.1	66.5	52.4	41.6	63.9	45.2	101M	819G	12.8	
Swin-S	**51.8**	**70.4**	**56.3**	**44.7**	**67.9**	**48.5**	107M	838G	12.0	+3.7
X101-64	48.3	66.4	52.3	41.7	64.0	45.1	140M	972G	10.4	
Swin-B	**51.9**	**70.9**	**56.5**	**45.0**	**68.4**	**48.7**	145M	982G	11.6	+3.6

Application: semantic segmentation



- ADE20K semantic segmentation: #1 for single model (53.9 mIoU)
  - The largest and most difficult semantic segmentation benchmark
    - 20,000 training images, 150 categories
  - Significantly surpass all previous CNN models (+5.5 mIoU vs. the previous best CNN model)

Application: video recognition (coming soon)



3D tokens: T'×H'×W' = 8×8×8
Window size: P×M×M = 4×4×4

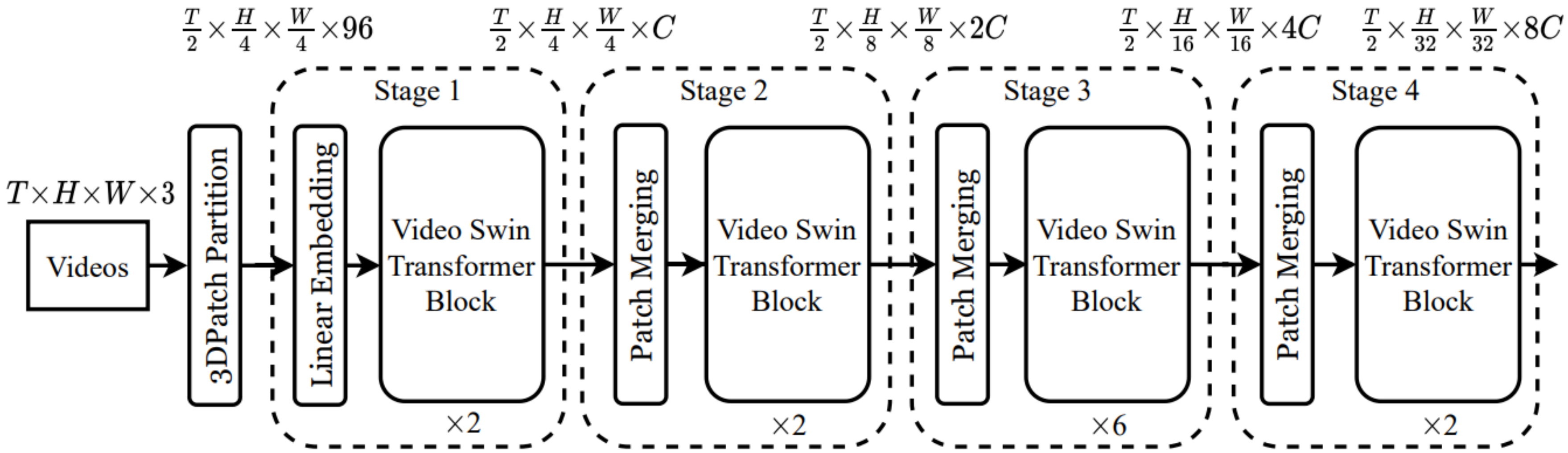


Figure 2: Overall architecture of Video Swin Transformer (tiny version, referred to as Swin-T).

Application: video recognition

- Swin Transformer achieves SOTA on major video benchmarks with 20x less pre-training data and 3x smaller model size

Table 1: Comparison to state-of-the-art on Kinetics-400. "384↑" signifies that the model uses a larger spatial resolution of 384×384. "Views" indicates # temporal clip × # spatial crop. The magnitudes are Giga (10^9) and Mega (10^6) for FLOPs and Param respectively.

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
R(2+1)D [37]	-	72.0	90.0	10 × 1	75	61.8
I3D [6]	ImageNet-1K	72.1	90.3	-	108	25.0
NL I3D-101 [40]	ImageNet-1K	77.7	93.3	10 × 3	359	61.8
ip-CSN-152 [36]	-	77.8	92.8	10 × 3	109	32.8
CorrNet-101 [39]	-	79.2	-	10 × 3	224	-
SlowFast R101+NL [13]	-	79.8	93.9	10 × 3	234	59.9
X3D-XXL [12]	-	80.4	94.6	10 × 3	144	20.3
MViT-B, 32×3 [10]	-	80.2	94.4	1 × 5	170	36.6
MViT-B, 64×3 [10]	-	81.2	95.1	3 × 3	455	36.6
TimeSformer-L [3]	ImageNet-21K	80.7	94.7	1 × 3	2380	121.4
ViT-B-VTN [29]	ImageNet-21K	78.6	93.7	1 × 1	4218	11.04
ViViT-L/16x2 [1]	ImageNet-21K	80.6	94.7	4 × 3	1446	310.8
ViViT-L/16x2 320 [1]	ImageNet-21K	$\boxed{81.3}$	94.7	4 × 3	3992	310.8
ip-CSN-152 [36]	IG-65M	82.5	95.3	10 × 3	109	32.8
ViViT-L/16x2 [1]	JFT-300M	82.8	95.5	4 × 3	1446	310.8
ViViT-L/16x2 320 [1]	JFT-300M	83.5	95.5	4 × 3	3992	310.8
ViViT-H/16x2 [1]	JFT-300M	84.8	95.8	4 × 3	8316	647.5
Swin-T	ImageNet-1K	78.8	93.6	4 × 3	88	28.2
Swin-S	ImageNet-1K	80.6	94.5	4 × 3	166	49.8
Swin-B	ImageNet-1K	80.6	94.6	4 × 3	282	88.1
Swin-B	ImageNet-21K	82.7	95.5	4 × 3	282	88.1
Swin-L	ImageNet-21K	83.1	95.9	4 × 3	604	197.0
Swin-L (384↑)	ImageNet-21K	$\boxed{\mathbf{84.9}}$	**96.6**	10 × 5	2107	200.0

+3.6% using the same pre-training data

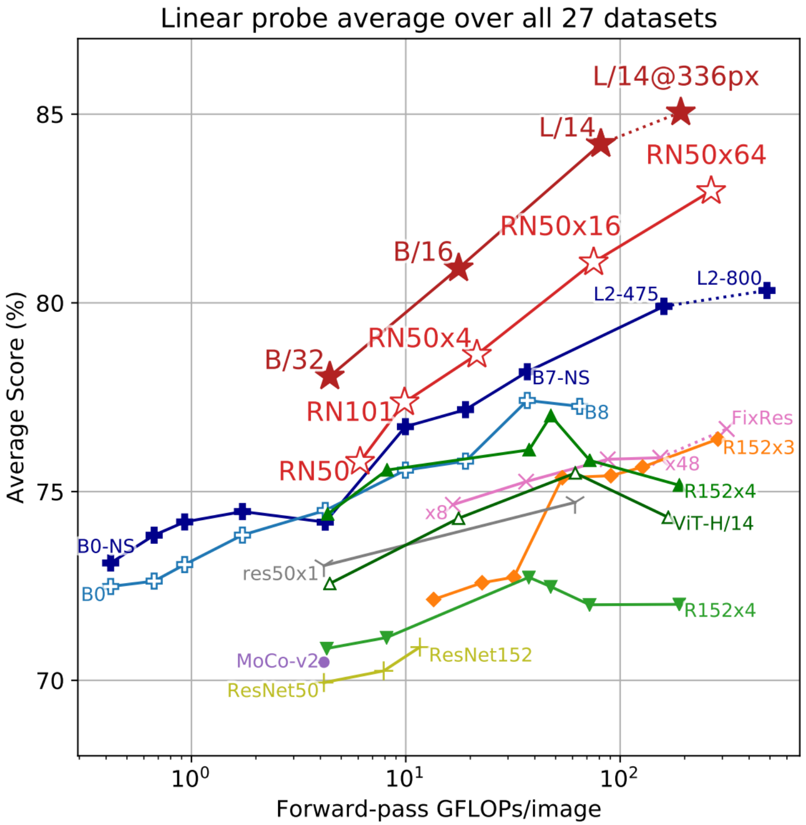
Table 2: Comparison to state-of-the-art on Kinetics-600.

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
SlowFast R101+NL [13]	-	81.8	95.1	10 × 3	234	59.9
X3D-XL [12]	-	81.9	95.5	10 × 3	48	11.0
MViT-B-24, 32×3 [9]	-	83.8	96.3	5 × 1	236	52.9
TimeSformer-HR [3]	ImageNet-21K	82.4	96	1 × 3	1703	121.4
ViViT-L/16x2 320 [1]	ImageNet-21K	$\boxed{83.0}$	95.7	4 × 3	3992	310.8
ViViT-H/16x2 [9]	JFT-300M	85.8	96.5	4 × 3	8316	647.5
Swin-B	ImageNet-21K	83.8	96.4	4 × 3	282	88.1
Swin-L (384↑)	ImageNet-21K	$\boxed{\mathbf{85.9}}$	**97.1**	4 × 3	2107	200.0

+2.9% using the same pre-training data

Reason IV to use Transformer in computer vision

- Better connect vision and language: unified modeling

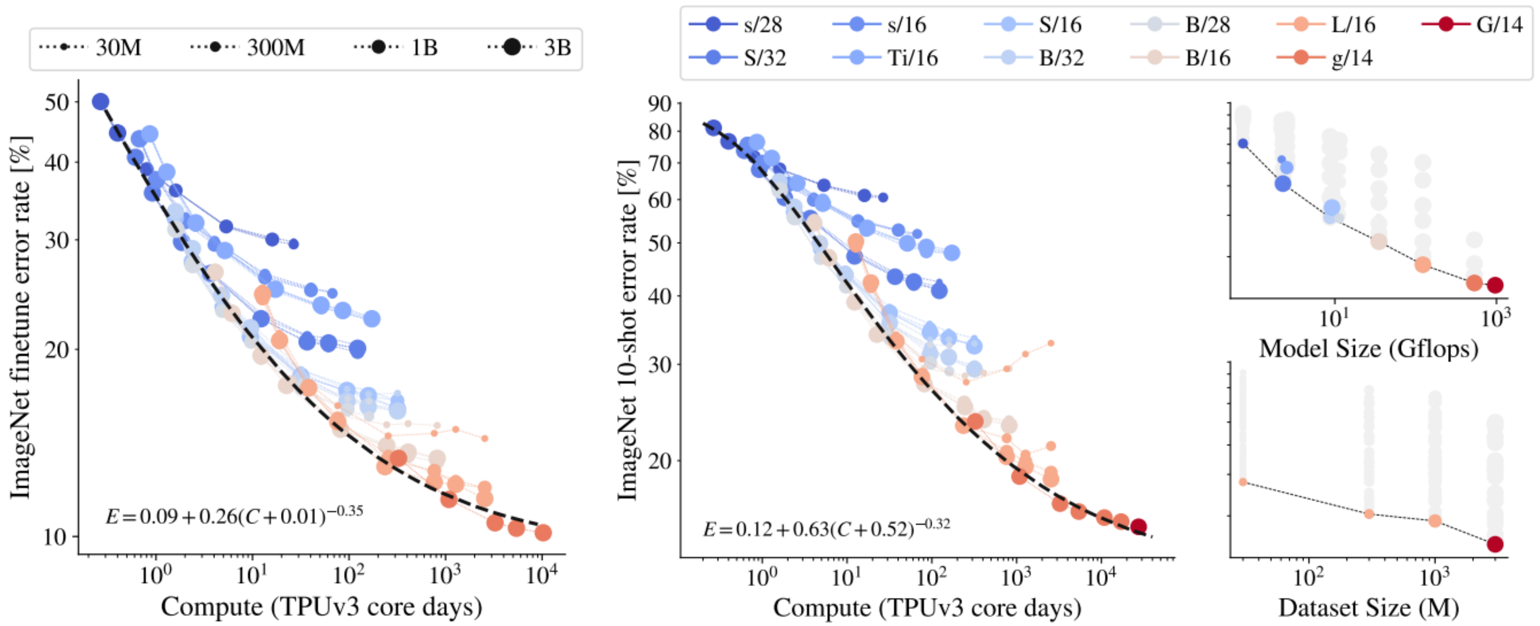


CLIP

https://openai.com/blog/clip/

Reason V to use Transformer in computer vision

- Scalable to large model and large data



- ViT G/14
  - 1000 G Flops
  - 2B parameters
  - 3B images

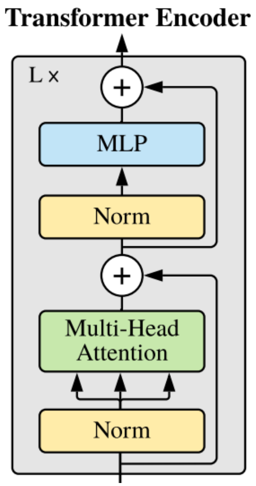
Scaling Vision Transformers: https://arxiv.org/pdf/2106.04560.pdf

Summary: 4 years unleash the power of Transformer in CV

Transformer (2017.6)

refresh & dominate

refresh & dominate



RelationNet (CVPR'2018)

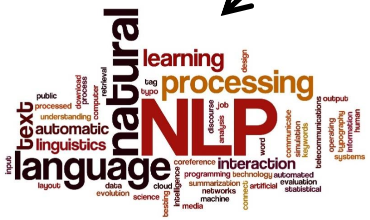
LearnRegionFeat (ECCV'2018)

STRN (ICCV'2020)

MEGA (CVPR'2020)

DeTr (ECCV'2020)

RelationNet++ (NeruIPS'2020)



ViT-G (Arxiv'2021)

Reason I: General modeling capability

2019.4

2021.1

Reason V: Scalability

2017.06

2017.11

Reason III: Strong modeling power

Reason IV: Better connect vision and language

2021.6

Reason II: Complement convolution

NLNet (CVPR'2018)

GCNet (ICCVW'2019)

DNL (ECCV'2020)

LRNet (ICCV'2019)

ViT (ICLR'2021)

Swin Transformer (Arxiv'2021)

CLIP (ICML'2021)