InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions

발표자: 김병현

이미지처리팀:
강인하, 김현진, 안종식, 이주영, 이희재, 현청천

InterImage

- State-of-the-Art (COCO test-dev) Backbone Network
  - Released by OpenGVLab

Rank	Model	box AP	AP50	AP75	APS	APM	APL	Params (M)	Extra Training Data	Paper	Code	Result	Year	Tags
1	**InternImage-DCNv3-H** (M3I Pre-training)	65.4							×	InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions			2022	DCN, Deformable Convolution, DINO, Giant
2	**M3I Pre-training** (InternImage-H)	65.4							×	Towards All-in-one Pre-training via Maximizing Multi-modal Mutual Information			2022	
3	**EVA**	64.7	81.9	71.7	48.5	67.7	77.9		×	EVA: Exploring the Limits of Masked Visual Representation Learning at Scale			2022	

Introduction

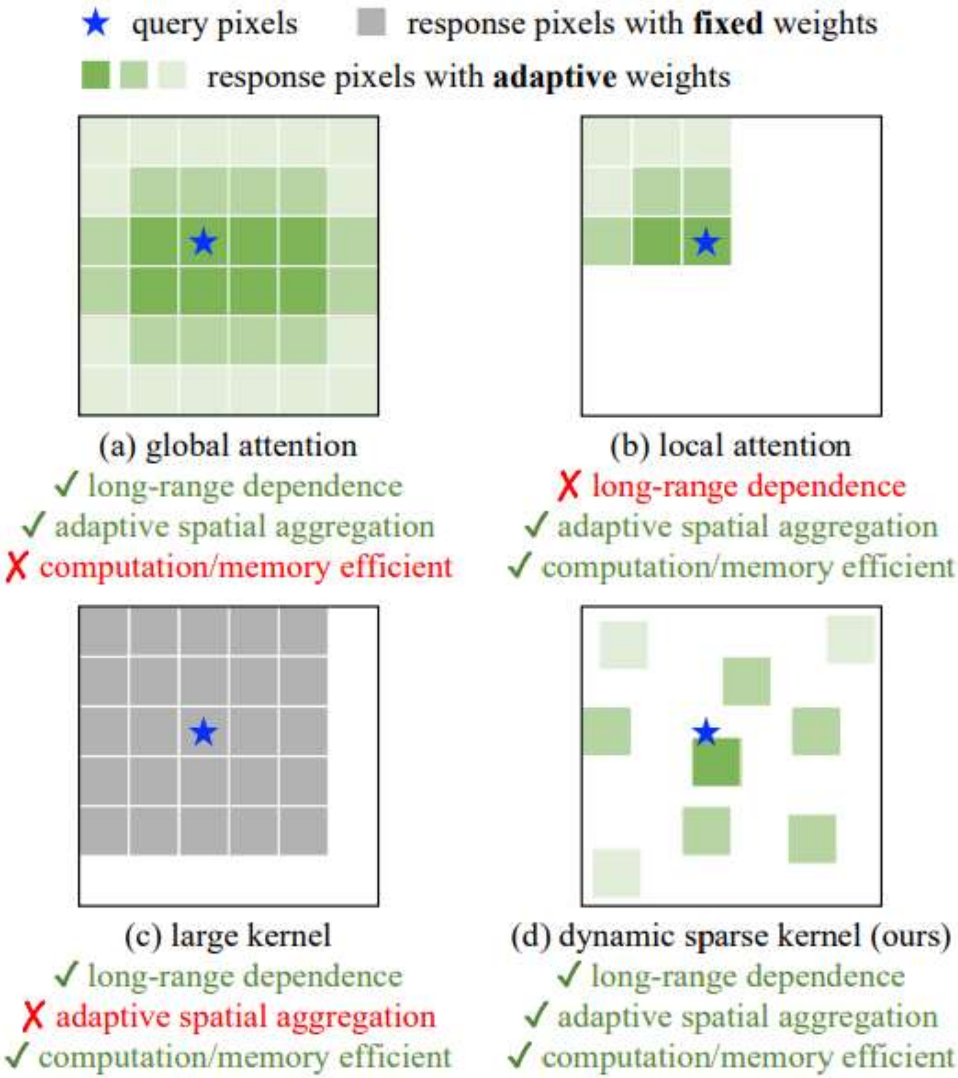
- Success of Transformers in Computer Vision Tasks
- CNN-based foundation models can also achieve comparable or even better performance than ViTs when equipped with similar operator-/architecture-level designs, scaling-up parameters, and massive data.

Introduction

- Gap between CNNs and ViTs
  - Operator Level
    - Long-range dependency
    - Adaptive spatial aggregation
  - Architecture View
    - Advanced components
      - Layer Normalization
      - Feed Forward Networks
      - GELU
  - Recent Long-Range CNNs
    - Very large kernels (31x31)
    - Gap with SOTA ViTs

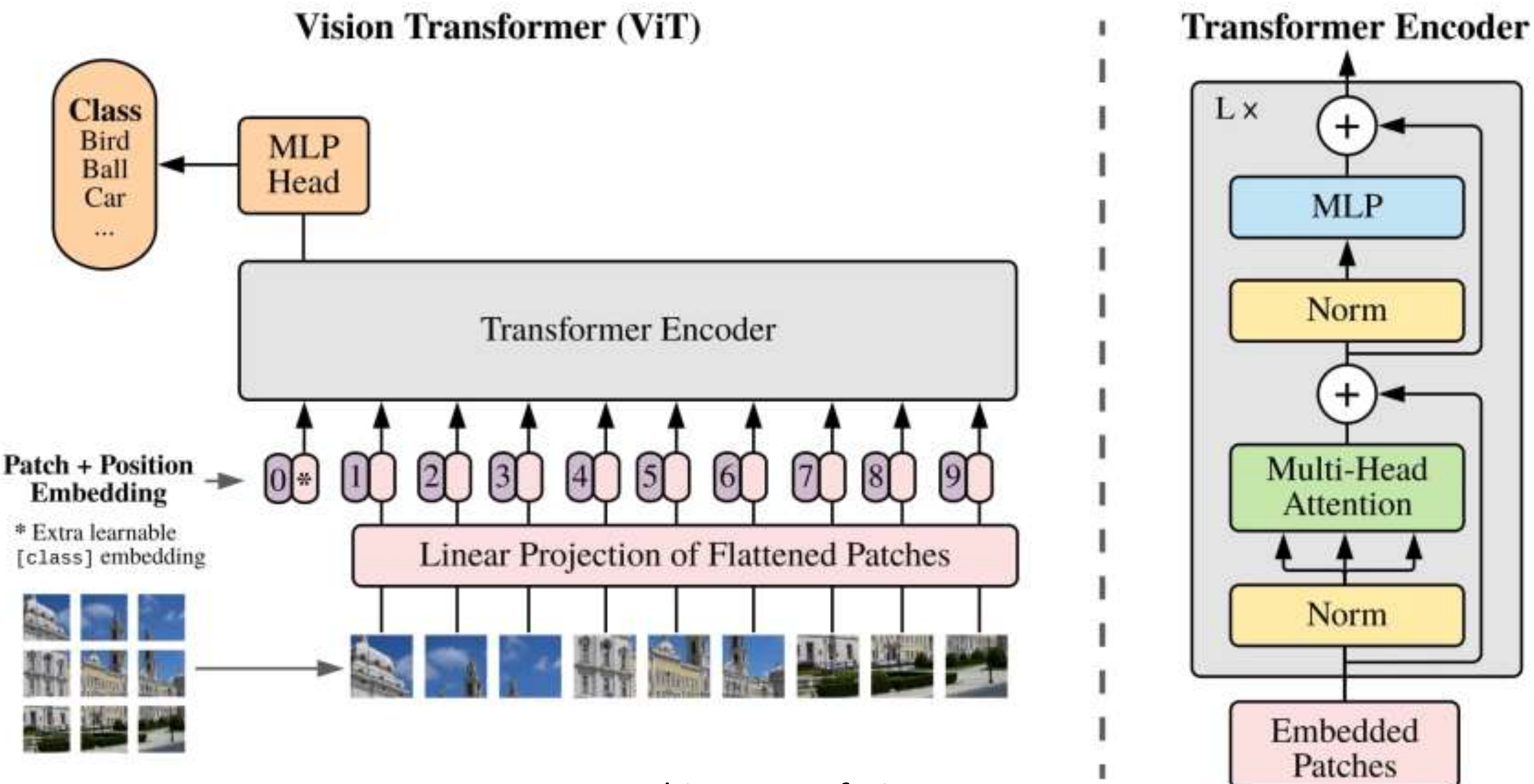
Introduction

❖ Comparison of different core operators



Introduction

❖ Global Attention: Vision Transformer

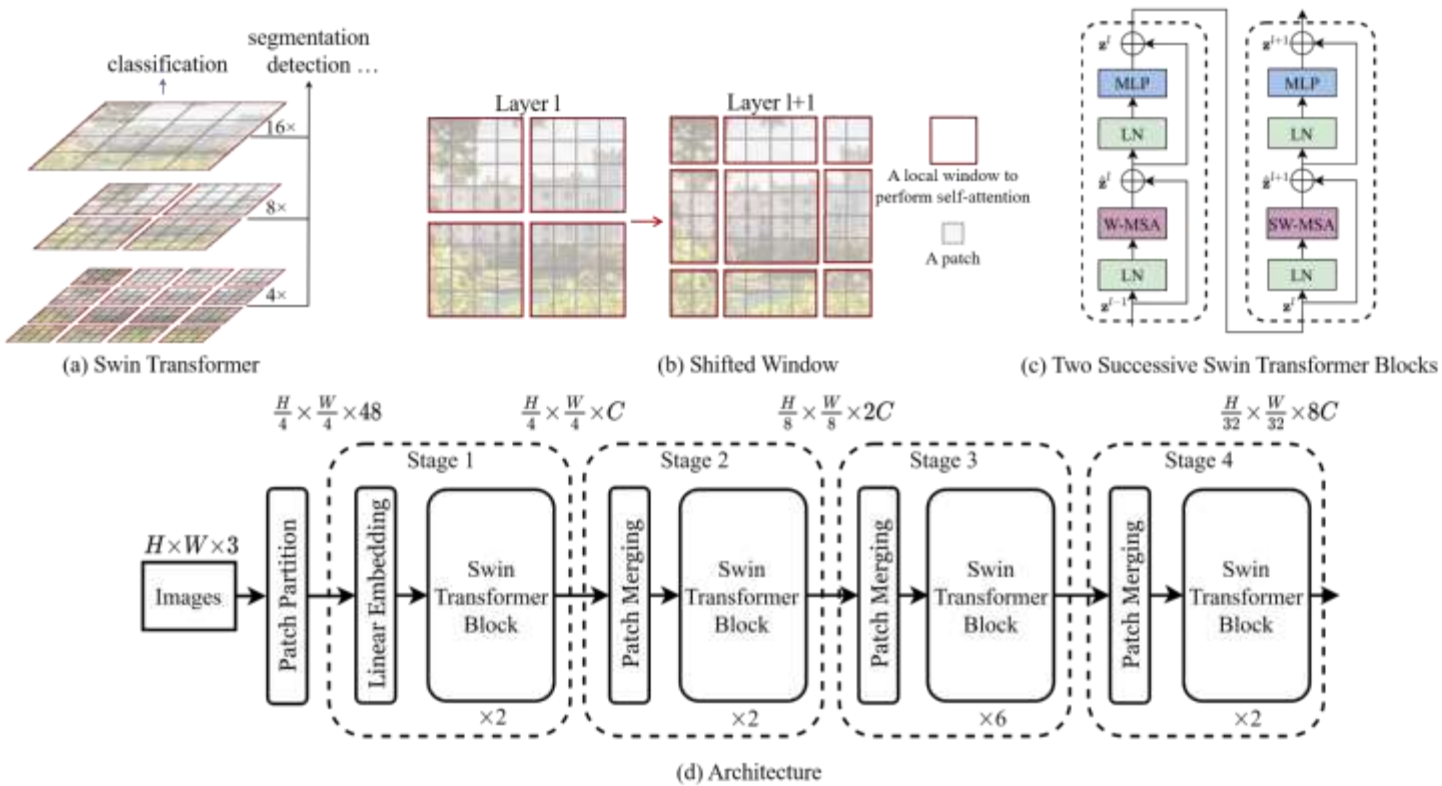


Architecture of ViT

Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

Introduction

❖ Local Attention: Swin Transformer



Architecture of Swin Transformer

Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

Introduction

❖ Large Kernel: SLaK

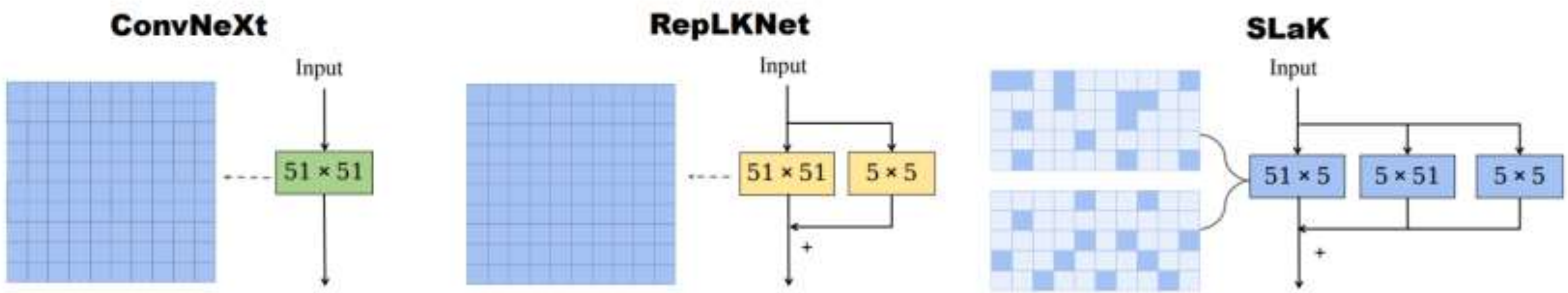


Figure 1: Large depth-wise kernel (e.g., 51×51) paradigms of ConvNeXt, RepLKNet, and SLaK. Dark blue squares refer to the dense weights in convolutional kernels. Light blue squares refer to the sparse weights in convolutional kernels.

Liu, Shiwei, et al. "More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity." arXiv preprint arXiv:2207.03620 (2022).

Introduction

- Concentration on CNN-based Model
  - *InterImage*
    - Brand-New CNN-based Backbone Network
    - Characteristics
      - Dynamic sparse convolutional layer
        - Only with 3x3 kernels
        - Adaptive spatial aggregation
        - Reduce inductive bias
        - Low computational cost compared to large convolutional layers
      - Overall Architecture of ViT

Introduction

- Contributions
  - 1st CNN-based backbone with more than 1 billion params.
  - Add long-rage dependencies and adaptive spatial aggregation with 3x3 DCN
  - SOTA accuracy in COCO dataset

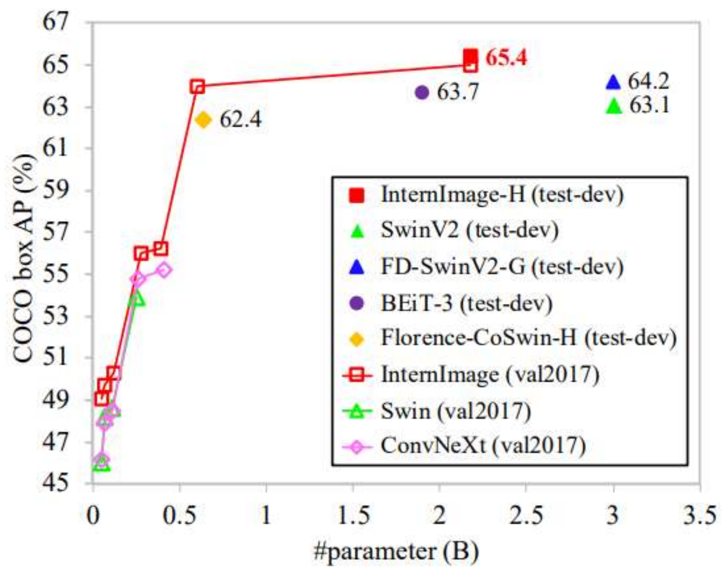
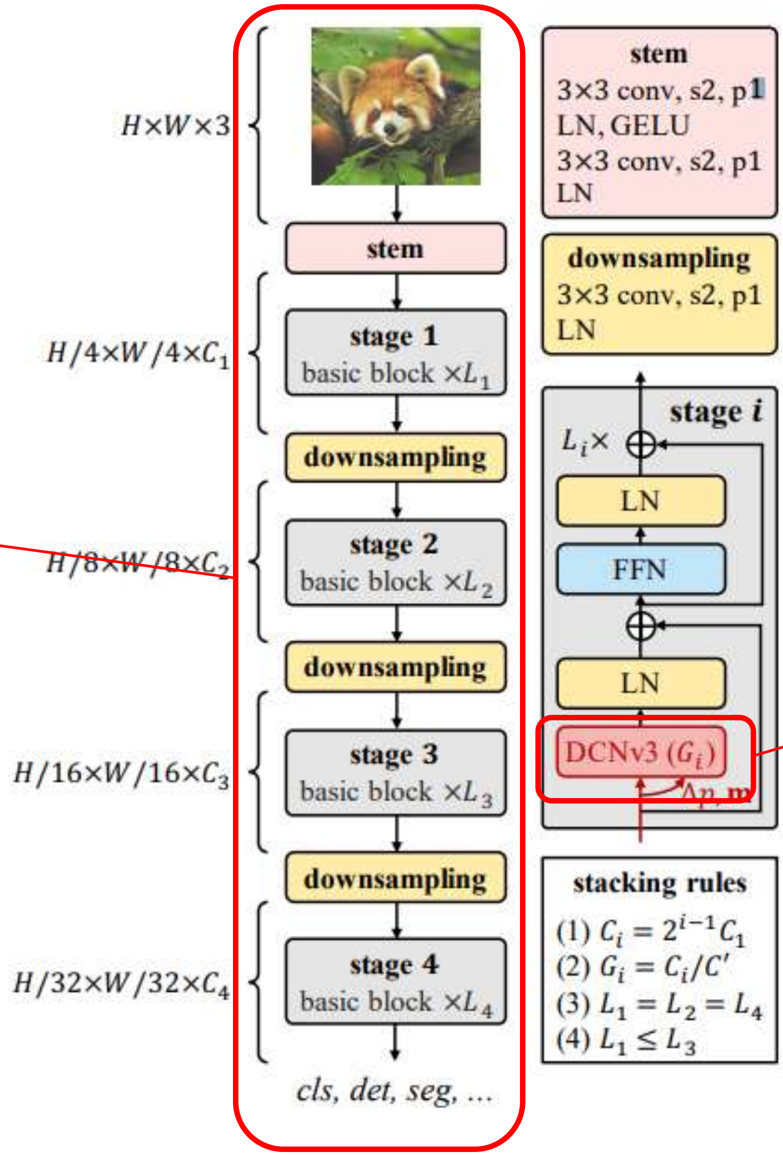


Figure 2. **Performance comparison on COCO of different backbones.** The proposed InternImage-H achieves a new record 65.4 box AP on COCO test-dev, significantly outperforming state-of-the-art CNNs and large-scale ViTs.

Q & A

Overall Architecture

2. Architecture Design



1. Deformable Convolutional Layer V3

Deformable Convolutional Layer v3

❖ Revisiting DCNv2

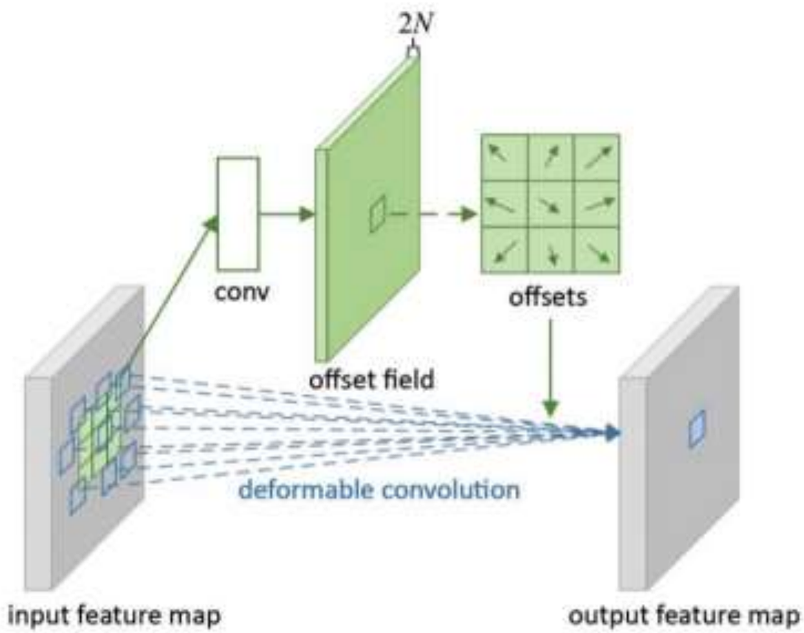
Revisiting DCNv2. A straightforward way to bridge the gap between convolution and MHSA is to introduce long-range dependencies and adaptive spatial aggregation into regular convolutions. Let us start with DCNv2 [28], which is a general variant of regular convolution. Given an input $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$ and current pixel p_0, DCNv2 can be formulated as:

$$\mathbf{y}(p_0) = \sum_{k=1}^{K} \mathbf{w}_k \mathbf{m}_k \mathbf{x}(p_0 + p_k + \Delta p_k), \qquad (1)$$

where K represents the total number of sampling points, and k enumerates the sampling point. $\mathbf{w}_k \in \mathbb{R}^{C \times C}$ denotes the projection weights of the k-th sampling point, and $\mathbf{m}_k \in \mathbb{R}$ represents the modulation scalar of the k-th sampling point, which is normalized by sigmoid function. p_k denotes the k-th location of the pre-defined grid sampling $\{(-1,-1), (-1,0), ..., (0,+1), ..., (+1,+1)\}$ as in regular convolutions, and Δp_k is the offset corresponding to the k-th grid sampling location. We see from the equation that (1) for long-range dependencies, the sampling offset Δp_k is flexible and able to interact with short- or

Deformable Convolutional Layer v3

Quick survey of Deformable ConvNets



Regular convolution

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n)$$

Deformable convolution

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n)$$

where $\Delta\mathbf{p}_n$ is generated by a sibling branch of regular convolution

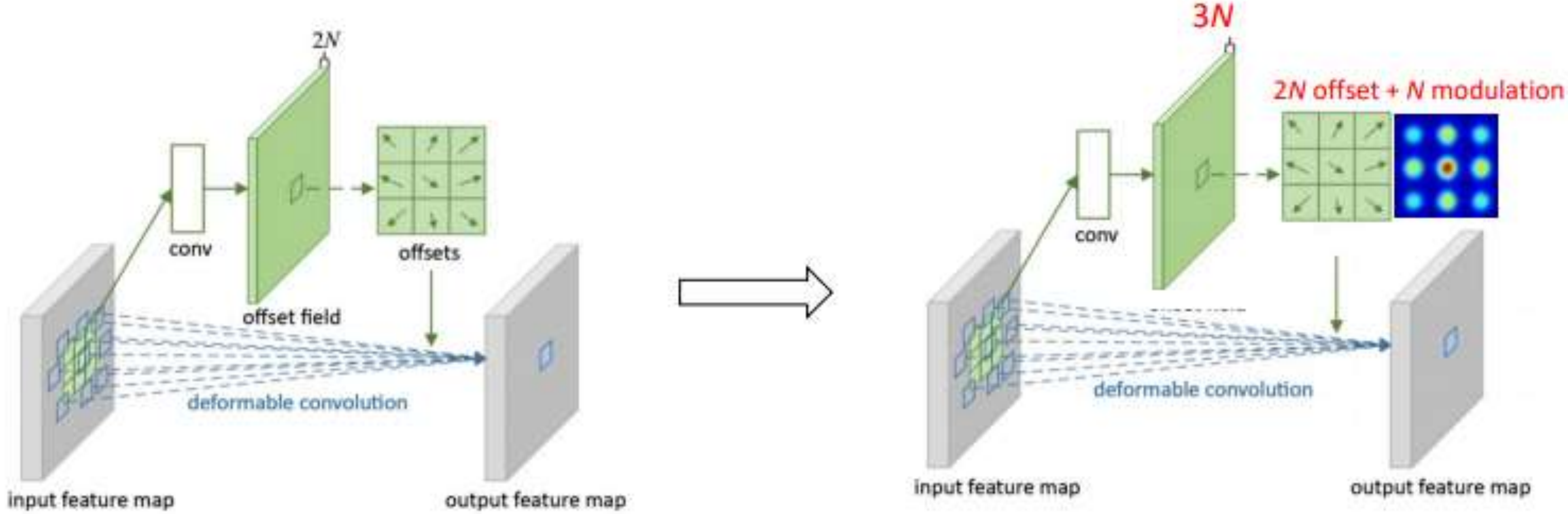
The grid sampling locations of standard convolution are each offset by displacements learned with respect to the preceding feature maps.

https://www.taokong.org/report/DCN_v2_paper_reading.pdf

Deformable Convolutional Layer v3

Modulated Deformable Modules:
DCN-v2

$$y(p) = \sum_{k=1}^{K} w_k \cdot x(p + p_k + \Delta p_k) \cdot \Delta m_k$$



With modulation, the Deformable ConvNets modules can not only adjust offsets in perceiving input features, but also modulate the input feature amplitudes/weights from different spatial locations.

18

https://www.taokong.org/report/DCN_v2_paper_reading.pdf

Deformable Convolutional Layer v3

1. Sharing weights among convolutional neurons.
   - Heavy computational cost of DCNv2
     - independent linear projection weights
     - memory complexity is linear with the total number of sampling points
   - To remedy this problem, we borrow the idea from the separable convolution and detach the original convolution weights into depth-wise and point-wise parts
2. Introducing multi-group mechanism
   - Split the spatial aggregation process into G groups
3. Normalizing modulation scalars along sampling points
   - Change element-wise sigmoid normalization to softmax normalization along sample points.

Deformable Convolutional Layer v3

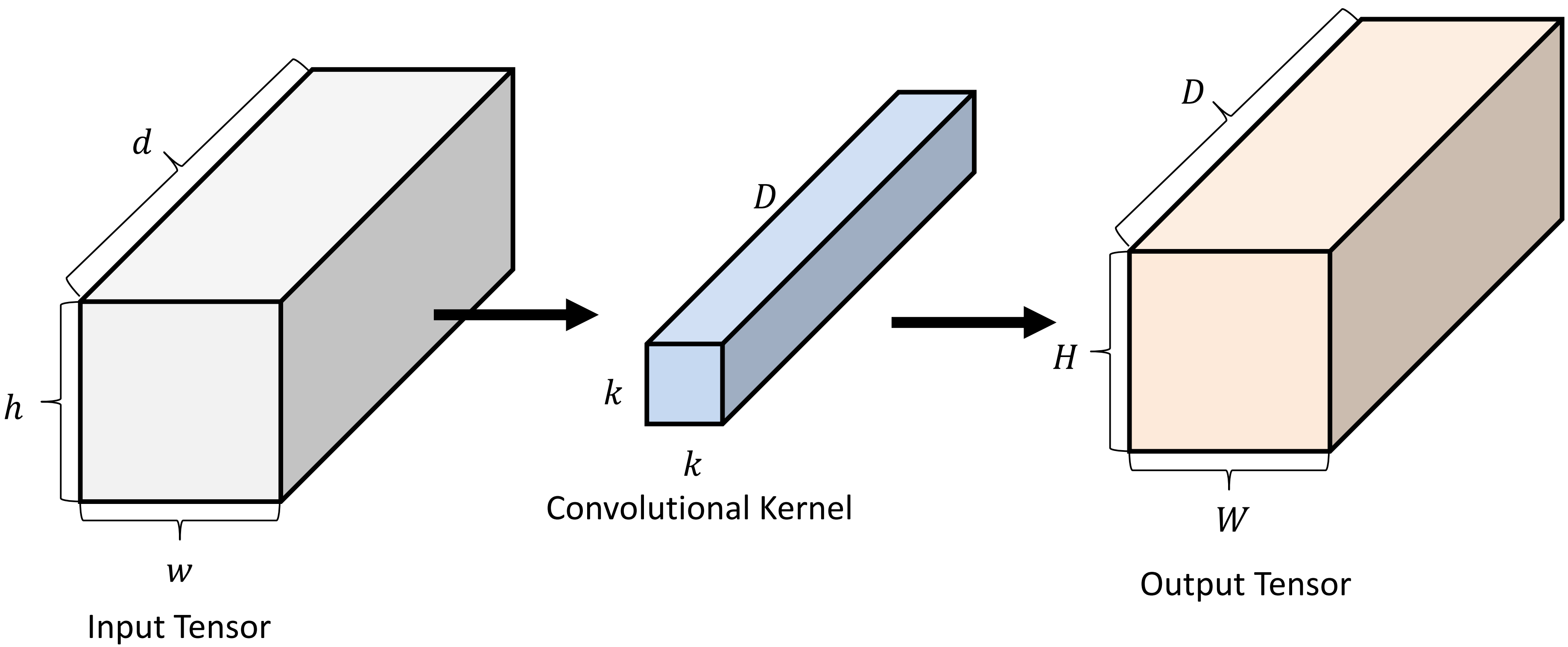
Combining the aforementioned modifications, the extended DCNv2, marked as DCNv3, can be formulated as Eqn. (2).

$$\mathbf{y}(p_0) = \sum_{g=1}^{G}\sum_{k=1}^{K} \mathbf{w}_g \mathbf{m}_{gk} \mathbf{x}_g (p_0 + p_k + \Delta p_{gk}), \qquad (2)$$

where G denotes the total number of aggregation groups. For the g-th group, $\mathbf{w}_g \in \mathbb{R}^{C \times C'}$, $\mathbf{m}_{gk} \in \mathbb{R}$ denote the location-irrelevant projection weights of the group, where $C' = C/G$ represents the group dimension. $\mathbf{m}_{gk} \in \mathbb{R}$ denotes the modulation scalar of the k-th sampling point in the g-th group, normalized by the softmax function along the dimension K. $\mathbf{x}_g \in \mathbb{R}^{C' \times H \times W}$ represents the sliced input feature map. Δp_{gk} is the offset corresponding to the grid sampling location p_k in the g-th group.

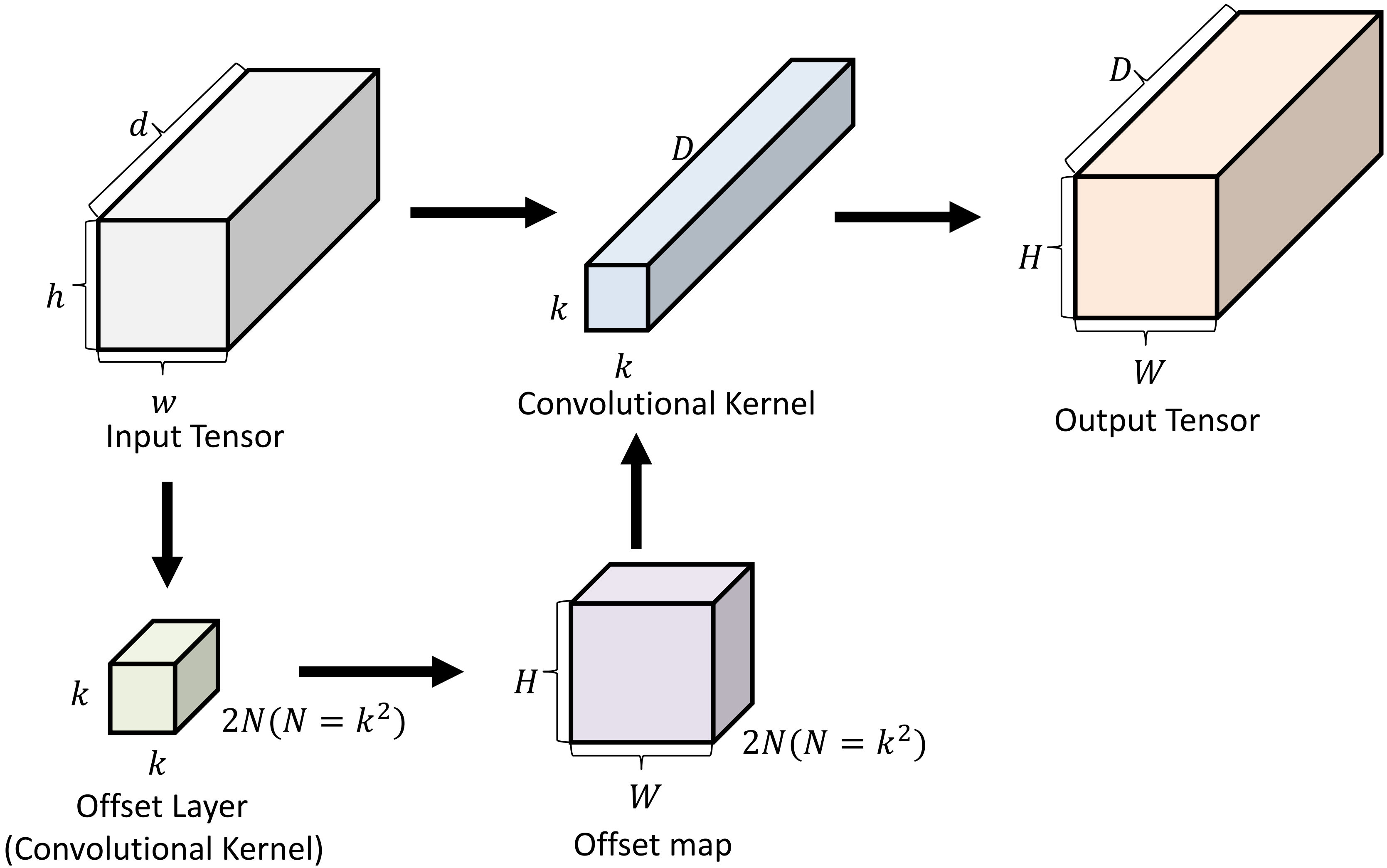
Deformable Convolutional Layer v3

❖ Normal Convolutional Layer



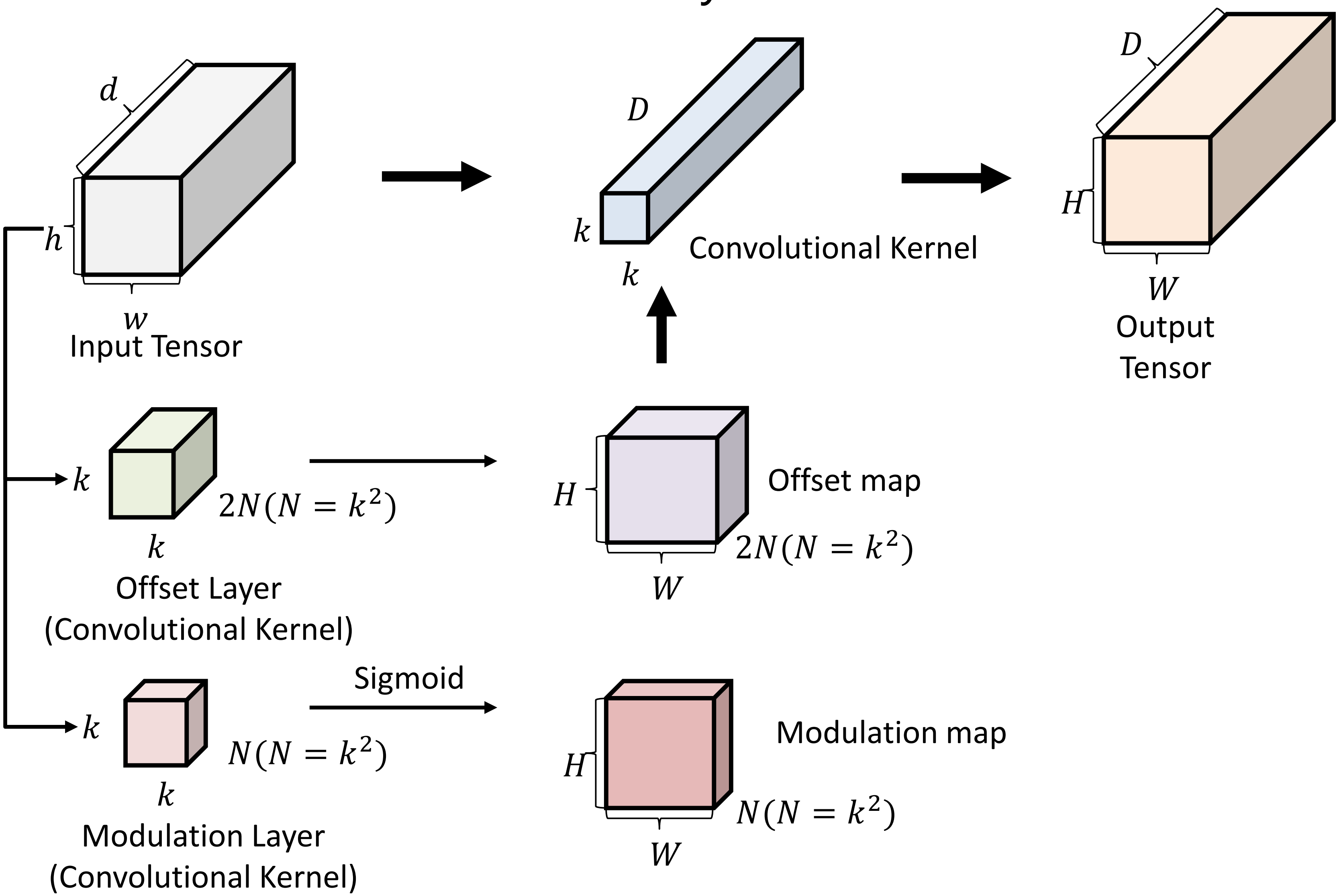
Deformable Convolutional Layer v3

❖ Deformable Convolutional Layer v1



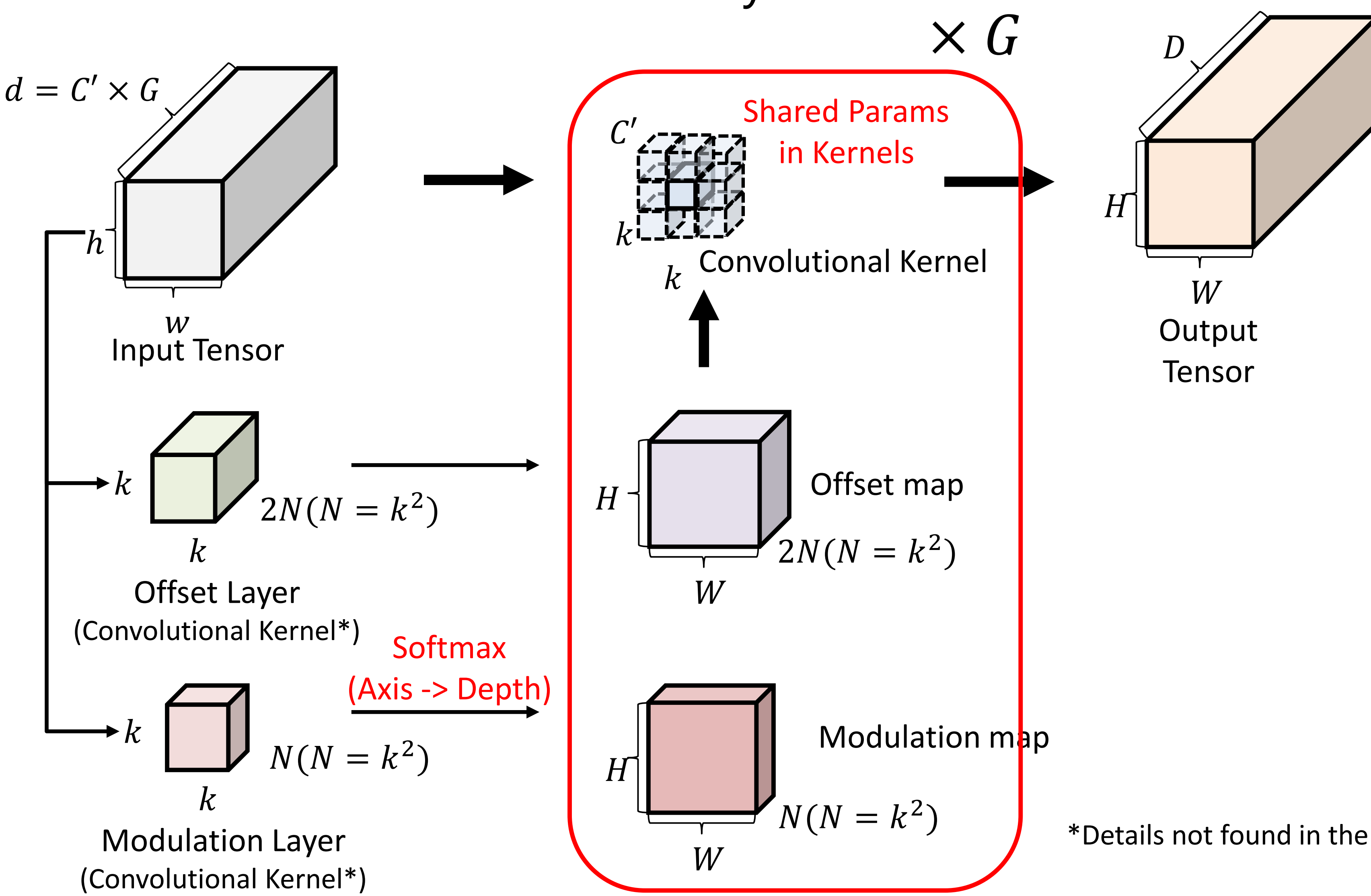
Deformable Convolutional Layer v3

❖ Deformable Convolutional Layer v2



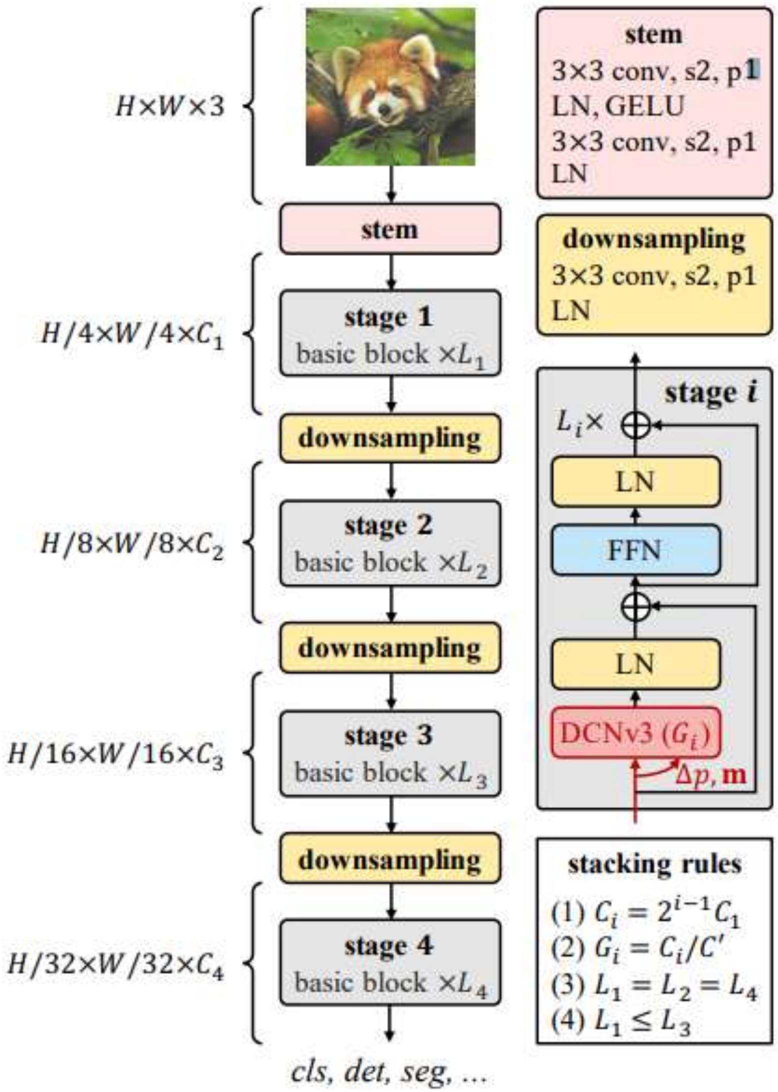
Deformable Convolutional Layer v3

❖ Deformable Convolutional Layer v3

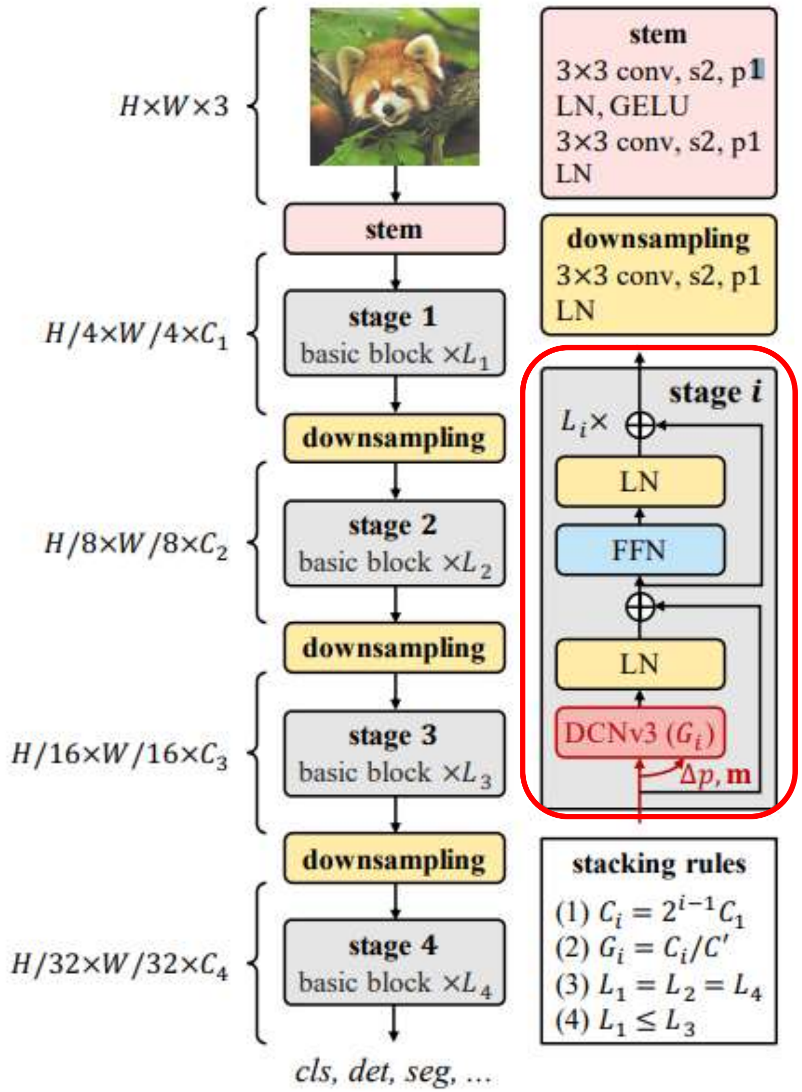


*Details not found in the paper

InterImage Model

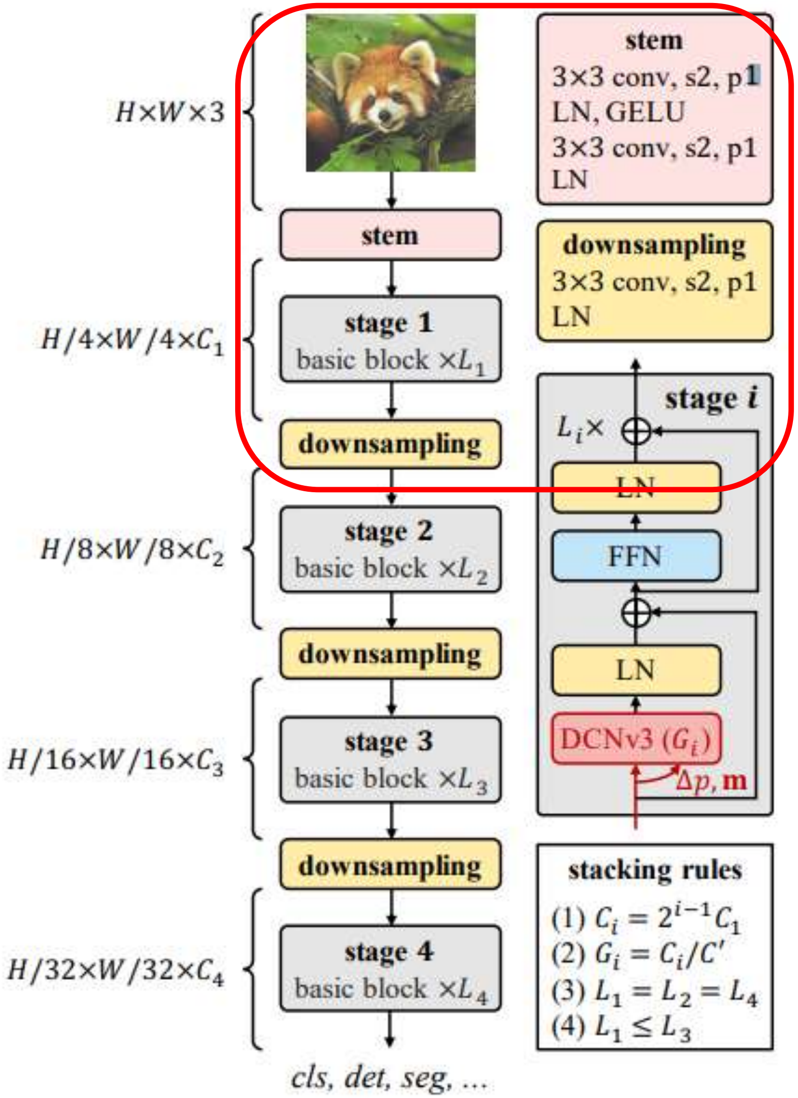


InterImage Model



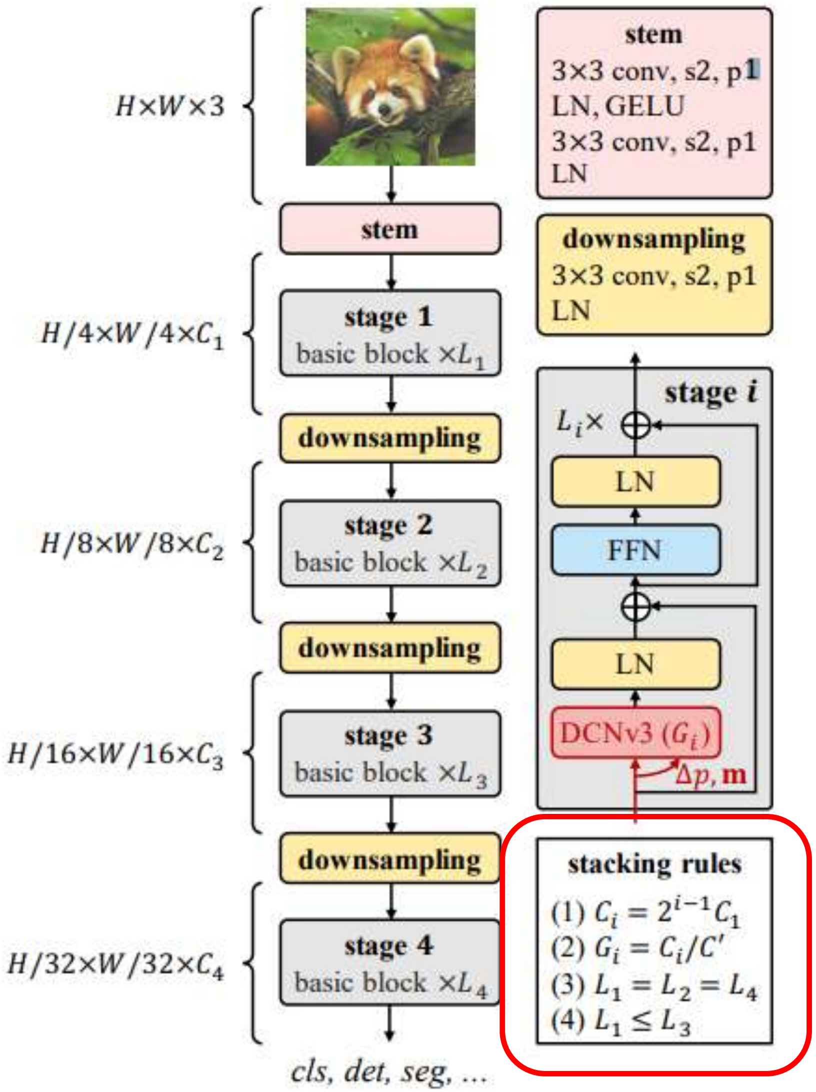
1. Basic Block

InterImage Model



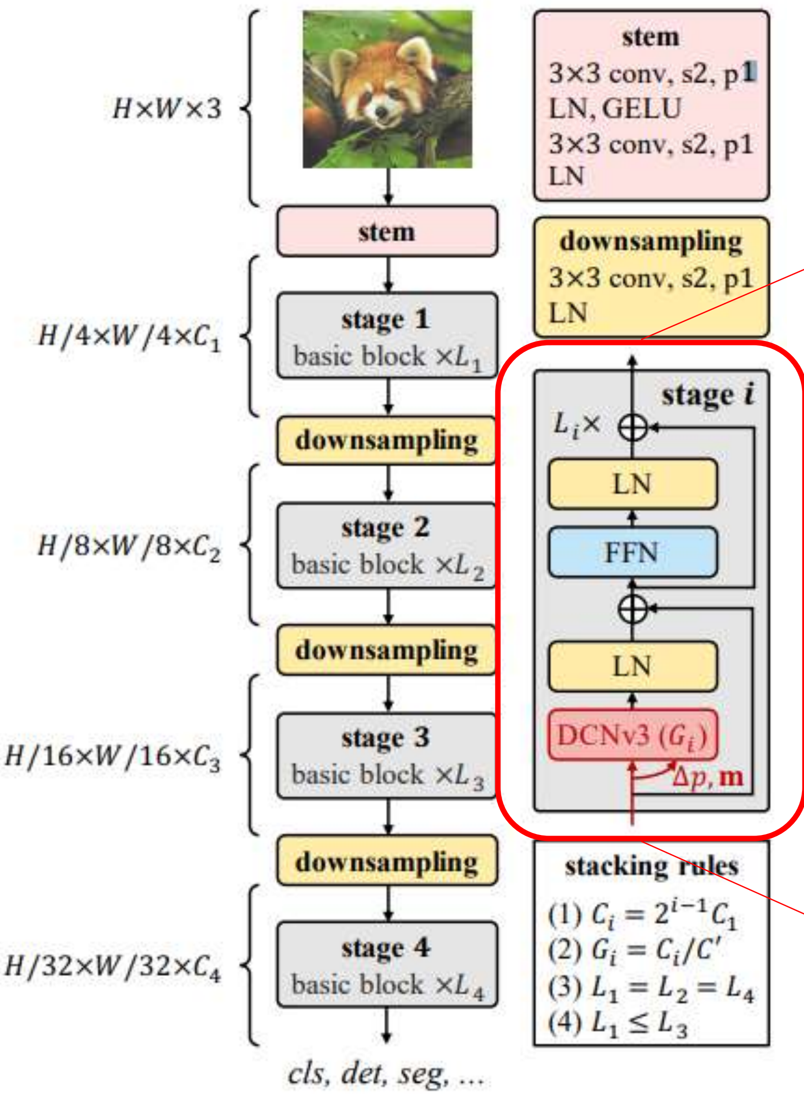
2. Stem layer & Downsampling

InterImage Model

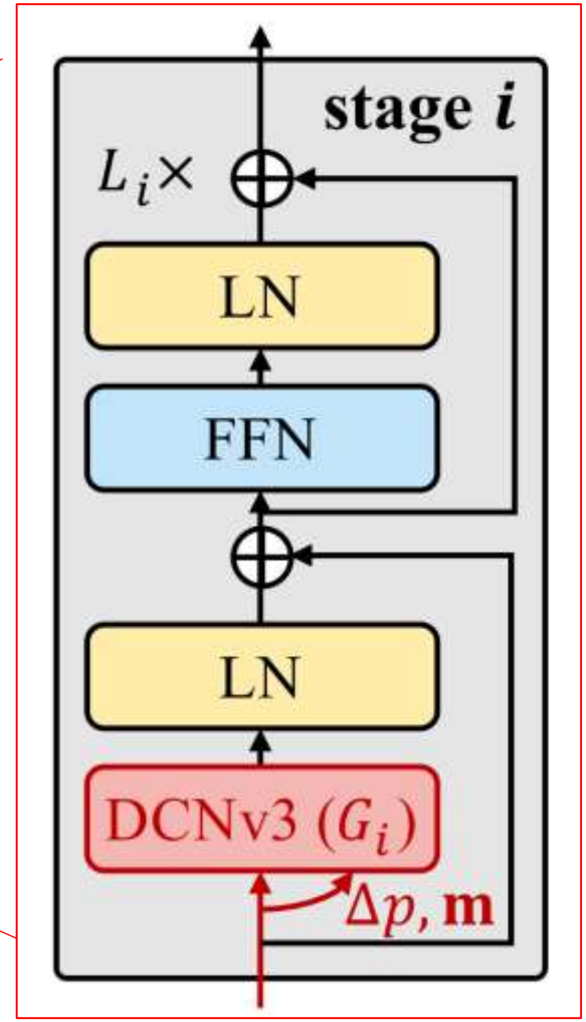


3. Stacking Rules

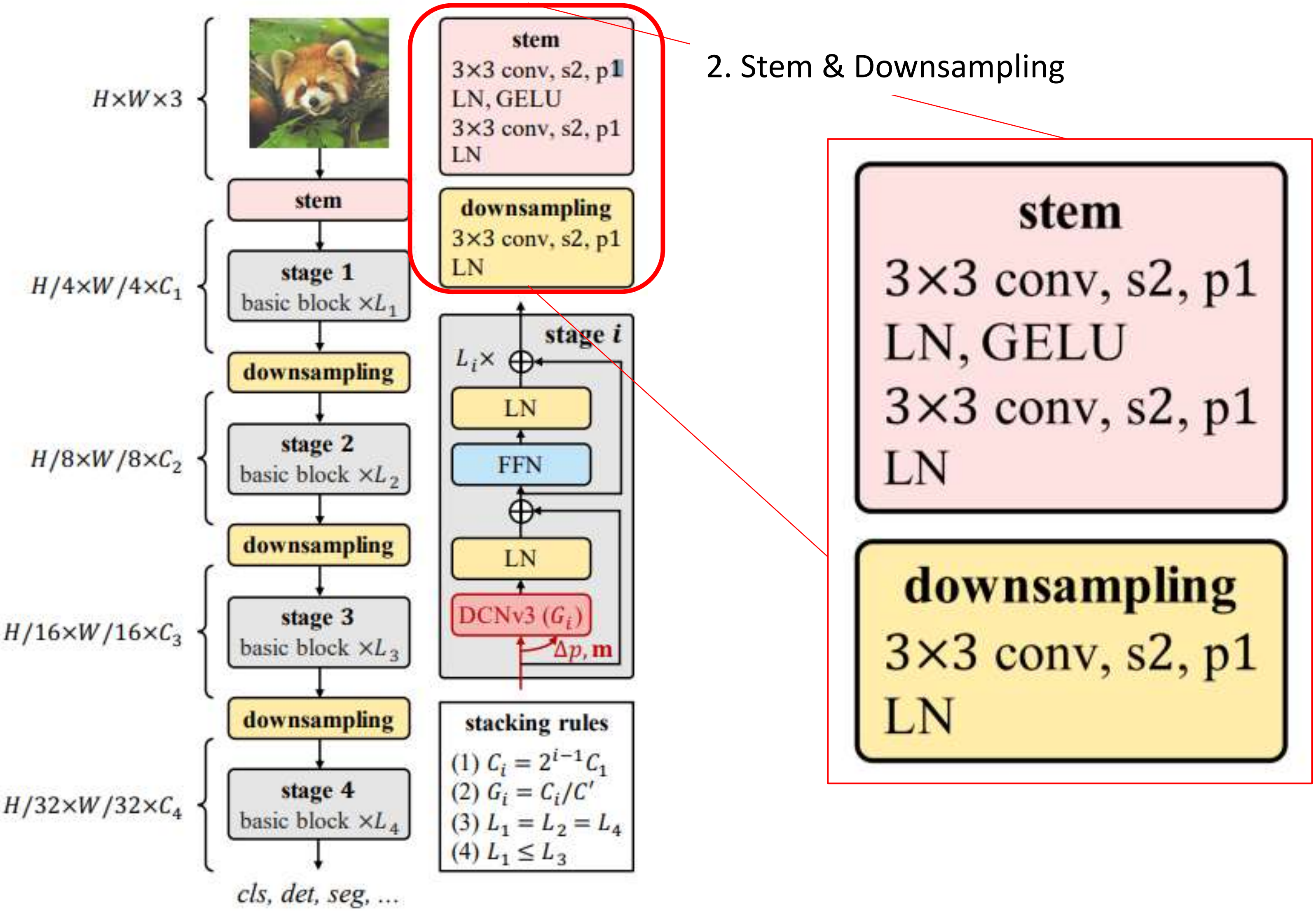
InterImage Model



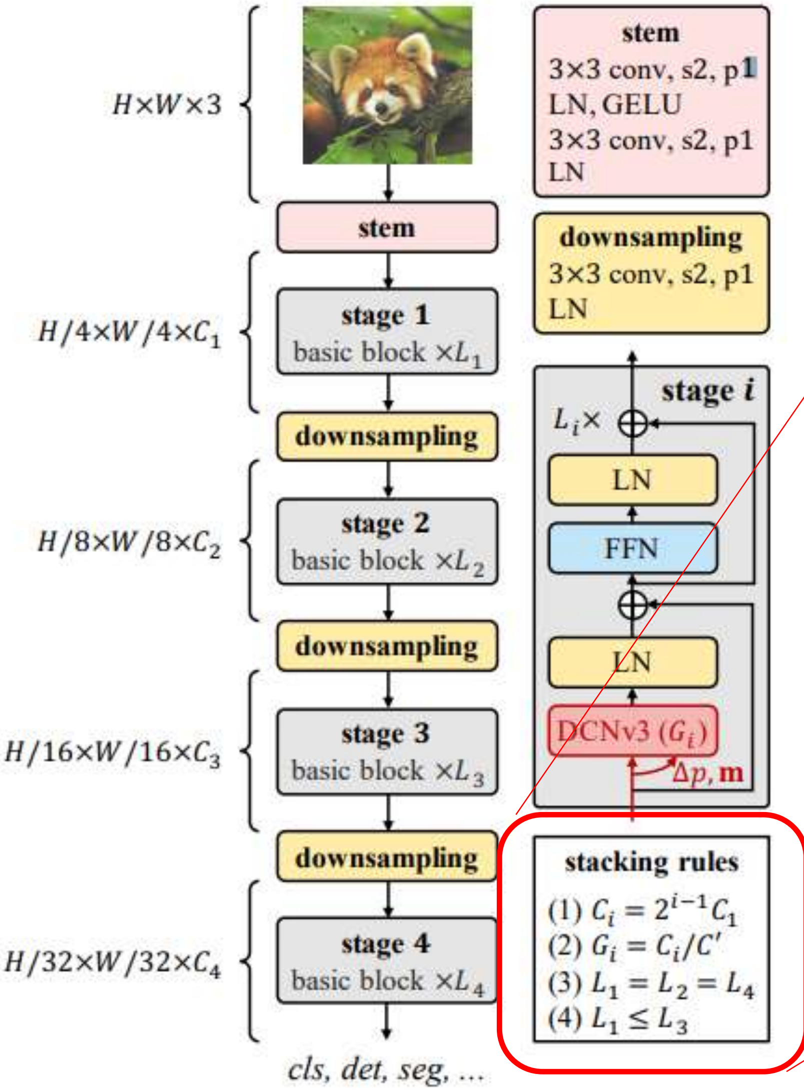
1. Basic Block



InterImage Model



InterImage Model



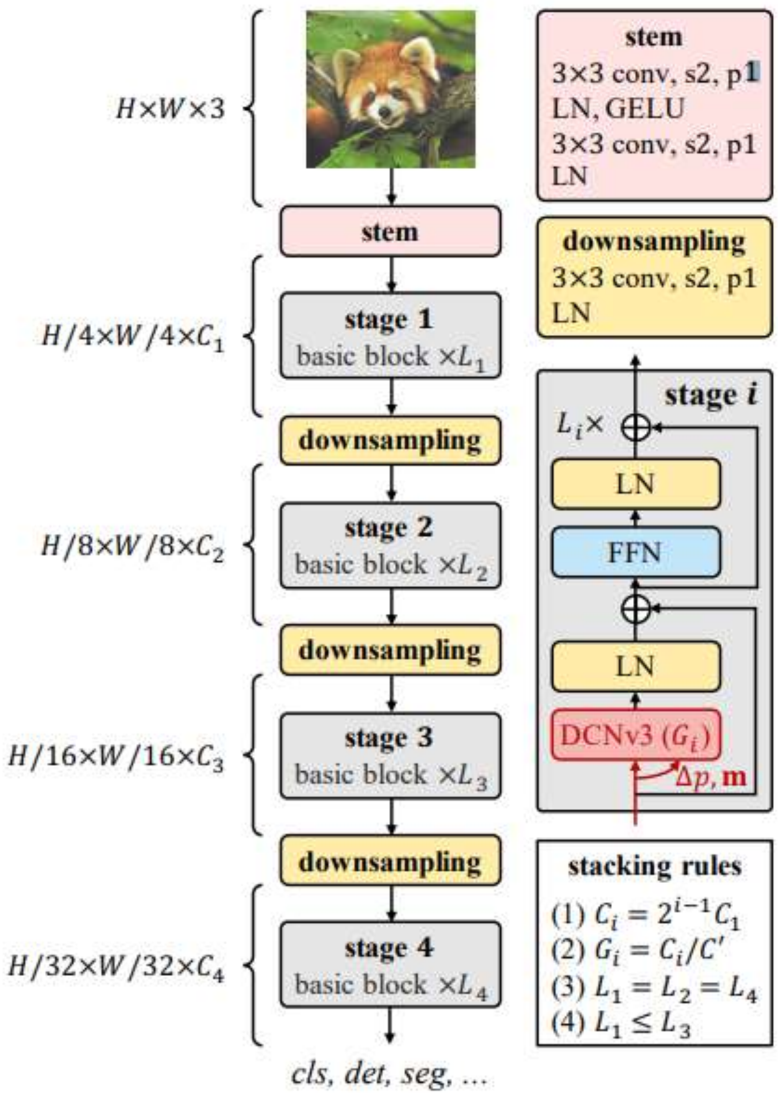
3. Stacking Rules

stacking rules

(1) $C_i = 2^{i-1} C_1$
(2) $G_i = C_i / C'$
(3) $L_1 = L_2 = L_4$
(4) $L_1 \leq L_3$

C_i: the channel number of the i-th stage;
G_i: the group number of the DCNv3 in the i-th stage;
L_i: the number of basic blocks in the i-th stage.

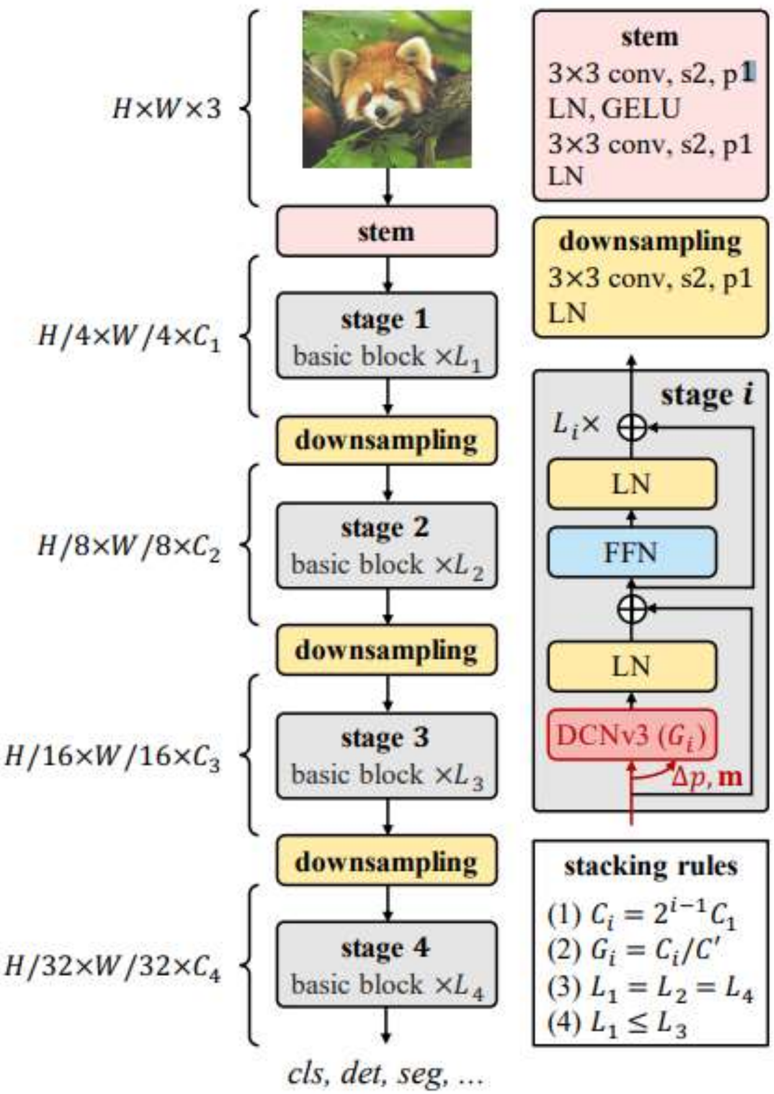
InterImage Model



4. Scaling Rules

Scaling rules. Based on the optimal origin model under the aforementioned constraints, we further explore the parameter scaling rules inspired by [38]. Specifically, we consider two scaling dimensions: depth D (*i.e.*, $3L_1 + L_3$) and width C_1, and scale the two dimensions using α, β and a composite factor ϕ. The scaling rules can be written as: $D' = \alpha^{\phi} D$ and $C'_1 = \beta^{\phi} C_1$, where $\alpha \geq 1$, $\beta \geq 1$, and $\alpha\beta^{1.99} \approx 2$. Here, 1.99 is specific for InternImage and calculated by doubling the model width and keeping the depth constant. We experimentally find out that the best scaling setting is $\alpha = 1.09$ and $\beta = 1.36$, and then we base on it to construct InternImage variants with different parameter scales, namely InternImage-T/S/B/L/XL, whose complexity is similar to those of ConvNeXt [21]. To further test the capability, we built a larger InternImage-H with 1 billion

InterImage Model



5. Hyper-parameters for models of different scales

model name	C_1	C'	$L_{1,2,3,4}$	#params
InternImage-T (origin)	64	16	4, 4, 18, 4	30M
InternImage-S	80	16	4, 4, 21, 4	50M
InternImage-B	112	16	4, 4, 21, 4	97M
InternImage-L	160	16	5, 5, 22, 5	223M
InternImage-XL	192	16	5, 5, 24, 5	335M
InternImage-H	320	16	6, 6, 32, 6	1.08B

Table 1. **Hyper-parameters for models of different scales.** InternImage-T is the origin model, and -S/B/L/XL/H are scaled up from -T. "#params" denotes the number of parameters.

4 Stages Models are prevalent since DETR

- Swin Transformer
- MetaFormer

Q & A

Q & A

Experiments

❖ Image Classification (Tiny Model)

method	type	scale	#params	#FLOPs	acc (%)
DeiT-S [58]	T	224^2	22G	5G	79.9
PVT-S [10]	T	224^2	25M	4G	79.8
Swin-T [2]	T	224^2	29M	5G	81.3
CoAtNet-0 [20]	T	224^2	25M	4G	81.6
CSwin-T [12]	T	224^2	23M	4G	82.7
PVTv2-B2 [11]	T	224^2	25M	4G	82.0
DeiT III-S [65]	T	224^2	22M	5G	81.4
SwinV2-T/8 [16]	T	256^2	28M	6G	81.8
Focal-T [66]	T	224^2	29M	5G	82.2
ConvNeXt-T [21]	C	224^2	29M	5G	82.1
SLaK-T [29]	C	224^2	30M	5G	82.5
HorNet-T [44]	C	224^2	23M	4G	83.0
InternImage-T (ours)	C	224^2	30M	5G	83.5

Experiments

❖ Image Classification (Large Model)

method	type	scale	#params	#FLOPs	acc (%)
ViT-G/14# [30]	T	518^2	1.84B	5160G	90.5
CoAtNet-6# [20]	T	512^2	1.47B	1521G	90.5
CoAtNet-7# [20]	T	512^2	2.44B	2586G	90.9
Florence-CoSwin-H# [59]	T	–	893M	–	90.0
SwinV2-G# [16]	T	640^2	3.00B	–	90.2
RepLKNet-XL# [22]	C	384^2	335M	129G	87.8
BiT-L-ResNet152x4# [64]	C	480^2	928M	–	87.5
InternImage-H# (ours)	C	224^2	1.08B	188G	88.5
InternImage-H# (ours)	C	640^2	1.08B	1478G	89.2

Table 2. **Image classification performance on the ImageNet validation set**. "type" refers to model type, where "T" and "C" denote transformer and CNN, respectively. "scale" is the input scale. "acc" is the top-1 accuracy. "‡" indicates the model is pre-trained on ImageNet-22K [31]. "#" indicates pretraining on extra large-scale private dataset such as JFT-300M [67], FLD-900M [59], or the joint public dataset in this work.

Experiments

❖ Object Detection & Instance Segmentation

method	#params	#FLOPs	Mask R-CNN 1× schedule						Mask R-CNN 3×+MS schedule					
			AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Swin-T [2]	48M	267G	42.7	65.2	46.8	39.3	62.2	42.2	46.0	68.1	50.3	41.6	65.1	44.9
ConvNeXt-T [21]	48M	262G	44.2	66.6	48.3	40.1	63.3	42.8	46.2	67.9	50.8	41.7	65.0	44.9
PVTv2-B2 [11]	45M	309G	45.3	67.1	49.6	41.2	64.2	44.4	47.8	69.7	52.6	43.1	66.8	46.7
ViT-S [9,68]	48M	353G	44.7	65.8	48.3	39.9	62.5	42.8	48.2	69.7	52.5	42.8	66.4	45.9
InternImage-T (ours)	49M	270G	47.2	69.0	52.1	42.5	66.1	45.8	49.1	70.3	54.0	43.7	67.3	47.1
Swin-S [2]	69M	354G	44.8	66.6	48.9	40.9	63.4	44.2	48.2	69.8	52.8	43.2	67.0	46.1
ConvNeXt-S [21]	70M	348G	45.4	67.9	50.0	41.8	65.2	45.1	47.9	70.0	52.7	42.9	66.9	46.2
PVTv2-B3 [11]	65M	397G	47.0	68.1	51.7	42.5	65.7	45.7	48.4	69.8	53.3	43.2	66.9	46.7
InternImage-S (ours)	69M	340G	47.8	69.9	52.8	43.3	67.1	46.7	49.7	71.1	54.5	44.4	68.5	47.8
Swin-B [2]	107M	496G	46.9	–	–	42.3	–	–	48.6	70.0	53.4	43.3	67.1	46.7
ConvNeXt-B [21]	108M	486G	47.0	69.4	51.7	42.7	66.3	46.0	48.5	70.1	53.3	43.5	67.1	46.7
PVTv2-B5 [11]	102M	557G	47.4	68.6	51.9	42.5	65.7	46.0	48.4	69.2	52.9	42.9	66.6	46.2
ViT-B [9,68]	120M	781G	47.0	68.2	51.4	41.8	65.1	44.9	49.6	70.6	54.0	43.6	67.7	46.9
InternImage-B (ours)	115M	501G	48.8	71.0	53.9	44.0	67.8	47.5	50.3	71.4	55.3	44.8	68.7	48.0

method	#param	#FLOPs	Cascade Mask R-CNN 1× schedule						Cascade Mask R-CNN 3×+MS schedule					
Swin-L‡ [2]	253M	1382G	51.8	71.0	56.2	44.9	68.4	48.9	53.9	72.4	58.8	46.7	70.1	50.8
ConvNeXt-L‡ [21]	255M	1354G	53.5	72.8	58.3	46.4	70.2	50.2	54.8	73.8	59.8	47.6	71.3	51.7
RepLKNet-31L‡ [22]	229M	1321G	–	–	–	–	–	–	53.9	72.5	58.6	46.5	70.0	50.6
HorNet-L‡ [44]	259M	1358G	–	–	–	–	–	–	56.0	–	–	48.6	–	–
InternImage-L‡ (ours)	277M	1399G	54.9	73.8	59.6	47.7	71.3	52.4	56.0	74.7	61.3	48.4	72.2	53.0
ConvNeXt-XL‡ [21]	407M	1898G	53.6	72.9	58.5	46.5	70.3	50.5	55.2	74.2	59.9	47.7	71.6	52.2
InternImage-XL‡ (ours)	387M	1782G	55.3	74.5	60.2	48.0	72.0	52.4	56.2	74.9	61.7	48.8	72.6	53.8

Table 3. **Object detection and instance segmentation performance on COCO `val2017`.** The FLOPs are measured with 1280×800 inputs. AP^b and AP^m represent box AP and mask AP, respectively. "MS" means multi-scale training.

Experiments

❖ Object Detection & Instance Segmentation

method	detector	#params	AP^b val2017	AP^b test-dev
Swin-L‡ [2]	HTC++ [2]	284M	58.0	58.7
Swin-L [2]	DyHead [72]	213M	56.2	58.4
ViT-L‡ [9]	ViT-Adapter [68]	401M	60.5	60.9
Swin-L‡ [2]	Soft-Teacher [73]	284M	60.7	61.3
Swin-L‡ [2]	DINO [74]	218M	63.2	63.3
FocalNet-H‡ [75]	DINO [74]	746M	64.2	64.3
ViT-Huge [76]	Group-DETRv2 [76]	629M	–	64.5
Florence-CoSwin-H# [59]	DyHead [72]	637M	62.0	62.4
SwinV2-G# [16]	HTC++ [2]	3.00B	62.5	63.1
BEiT-3# [17]	ViTDet [77]	1.90B	–	63.7
FD-SwinV2-G# [26]	HTC++ [2]	3.00B	–	64.2
InternImage-XL‡ (ours)	DINO [74]	602M	64.2	64.3
InternImage-H# (ours)	DINO [74]	2.18B	65.0	65.4

Table 4. **Comparison of the state-of-the-art detectors on COCO `val2017` and test-dev.**

Experiments

❖ Semantic Segmentation

method	crop size	#params	#FLOPs	mIoU (SS)	mIoU (MS)
Swin-T [2]	512^2	60M	945G	44.5	45.8
ConvNeXt-T [21]	512^2	60M	939G	46.0	46.7
SLaK-T [29]	512^2	65M	936G	47.6	–
InternImage-T (ours)	512^2	59M	944G	47.9	48.1
Swin-S [2]	512^2	81M	1038G	47.6	49.5
ConvNeXt-S [21]	512^2	82M	1027G	48.7	49.6
SLaK-S [29]	512^2	91M	1028G	49.4	–
InternImage-S (ours)	512^2	80M	1017G	50.1	50.9
Swin-B [2]	512^2	121M	1188G	48.1	49.7
ConvNeXt-B [21]	512^2	122M	1170G	49.1	49.9
RepLKNet-31B [22]	512^2	112M	1170G	49.9	50.6
SLaK-B [29]	512^2	135M	1172G	50.2	–
InternImage-B (ours)	512^2	128M	1185G	50.8	51.3
Swin-L‡ [2]	640^2	234M	2468G	52.1	53.5
RepLKNet-31L‡ [22]	640^2	207M	2404G	52.4	52.7
ConvNeXt-L‡ [21]	640^2	235M	2458G	53.2	53.7
ConvNeXt-XL‡ [21]	640^2	391M	3335G	53.6	54.0
InternImage-L‡ (ours)	640^2	256M	2526G	53.9	54.1
InternImage-XL‡ (ours)	640^2	368M	3142G	55.0	55.3
SwinV2-G# [16]	896^2	3.00B	–	–	59.9
InternImage-H# (ours)	896^2	1.12B	3566G	59.9	60.3
BEiT-3# [17]	896^2	1.90B	–	–	62.8
FD-SwinV2-G# [26]	896^2	3000	–	–	61.3
InternImage-H# (ours) + Mask2Former [80]	896^2	1.31B	4635G	62.5	62.9

Table 5. **Semantic segmentation performance on the ADE20K validation set.** The FLOPs are measured with 512×2048, 640×2560, or 896×896 inputs according to the crop size. "SS" and "MS" means single-scale and multi-scale testing, respectively.

이미지처리팀 리뷰 의견

- ❖ Deformable Conv V3에 대한 분석이 부재
  - ⌗ Ablation Study 부재
    - ResNet 등 기존 Conv 기반 모델에서 성능향상을 가지는지 확인해줬으면 정말 좋았을 듯
  - ⌗ Deformable Conv V3의 장점에 대한 정성적인 분석 부재
    - Convolution을 Group으로 나누면서 생기는 단일 레이어의 다양한 Offset Map의 장점에 대한 시각화 자료가 있었으면 좋았을 듯
  - ⌗ Inductive Bias를 줄일 수 있었다는 주장에 대한 근거자료 부재

- ❖ 코드가 아직 공개되지 않아 정확한 검증은 어려움

Q & A

Q & A