









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

Image Classification on ImageNet

15	InternImage-DCNv3-G (M3I Pre-training)	90.1%	3000M	×	InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions			2022
----	---	-------	-------	---	---	---	---	------

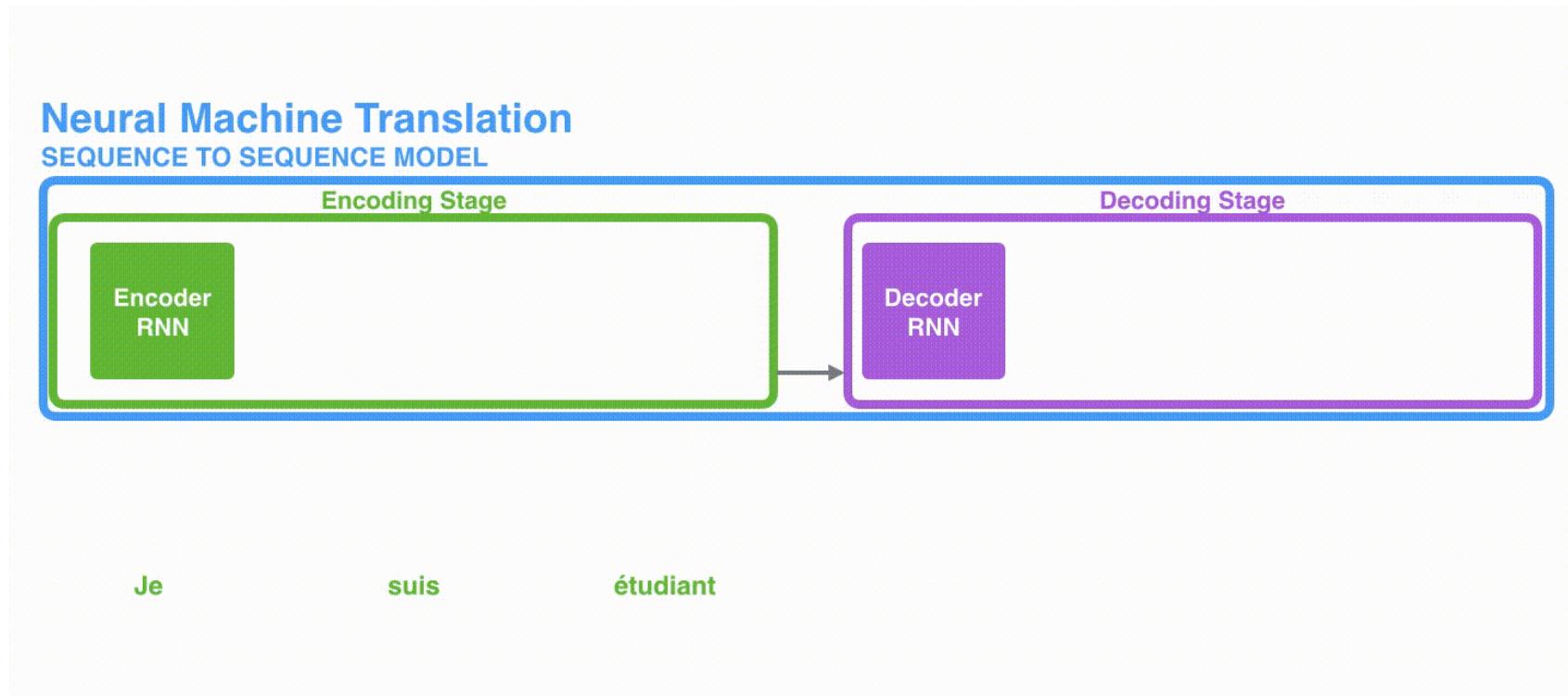
Object Detection on COCO minival test-dev

1	Co-DETR	66.0	348	×	DETRs with Collaborative Hybrid Assignments Training			2023	Large
2	InternImage-H	65.4	2180	×	InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions			2022	
3	M3I Pre-training (InternImage-H)	65.4		×	Towards All-in-one Pre-training via Maximizing Multi-modal Mutual Information			2022	

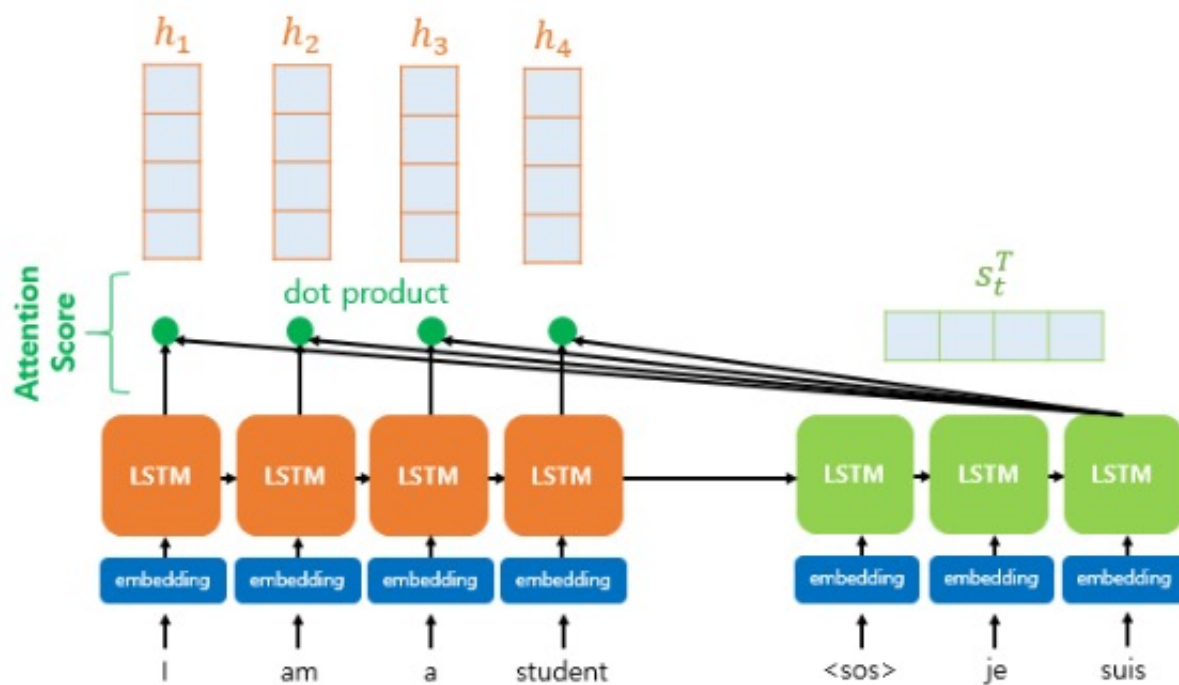
Highlights

- 👍 The strongest open-source visual universal backbone model with up to 3 billion parameters
- 🏆 Achieved **90.1% Top1** accuracy in ImageNet, the most accurate among open-source models
- 🏆 Achieved **65.5 mAP** on the COCO benchmark dataset for object detection, the only model that exceeded **65.0 mAP**

Seq2Seq 구조와 Attention



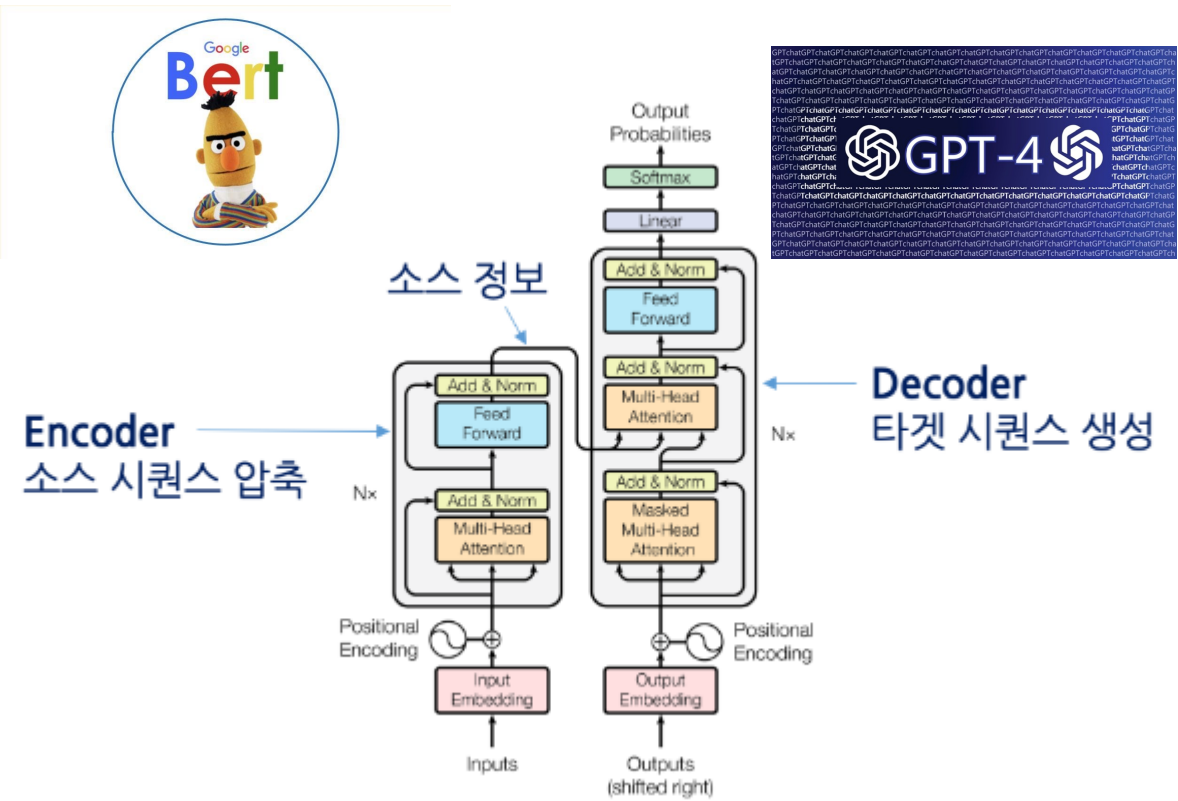
Seq2Seq 구조와 Attention



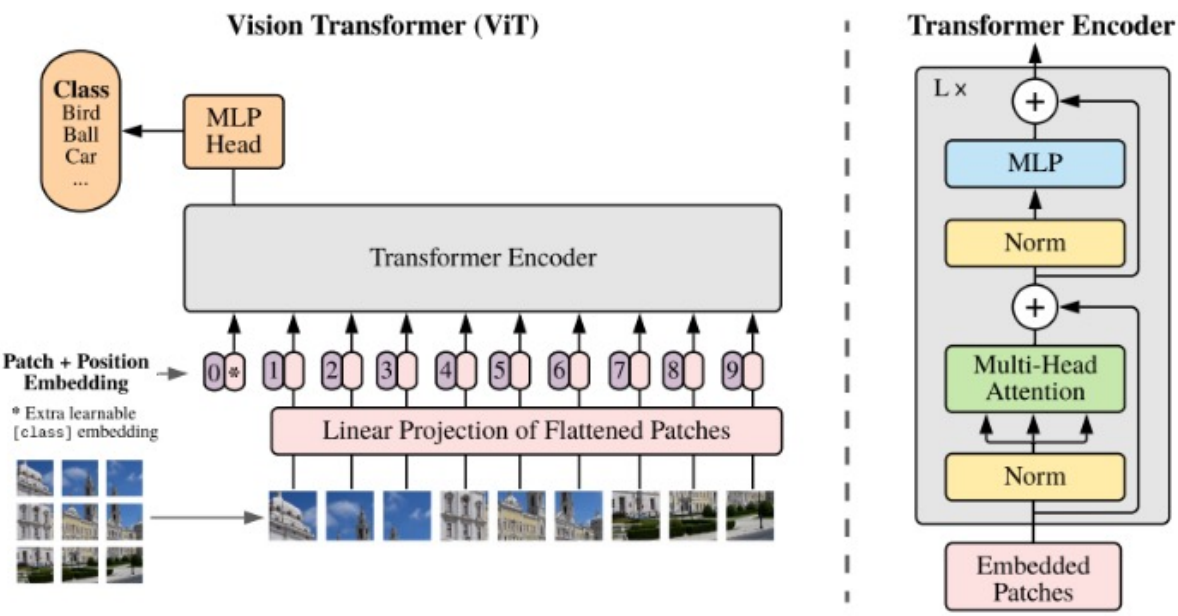
Transformer와 Vision Transformer

Attention Is All You Need

Transformer

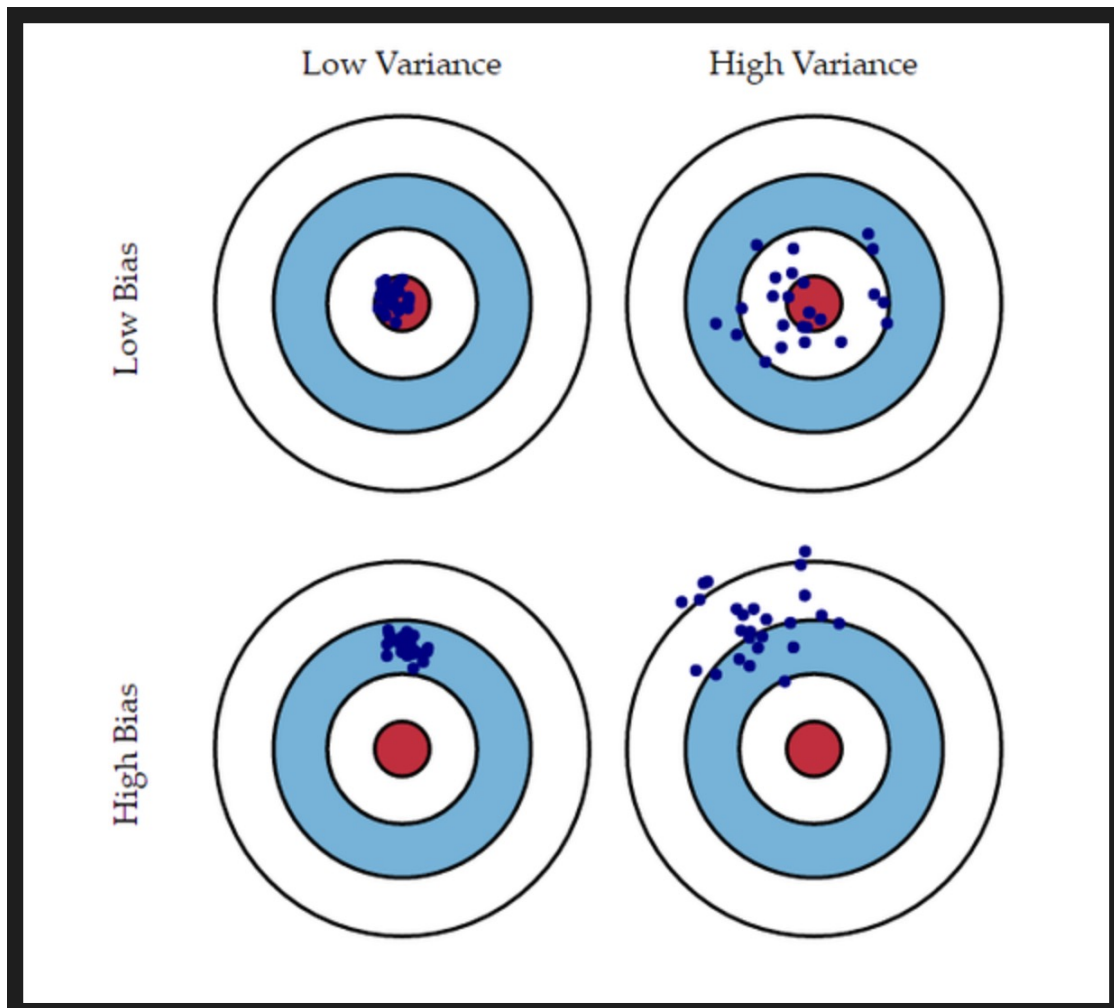


Vision Transformer



Transformer와 CNN

CNN의 inductive bias



- Inductive bias는 학습시에 만나지 않았던 상황에 대한 예측을 위한 추가적인 가정
- 결국 CNN은 이미지가 지역적으로 얻을 정보가 많다는 것을 가정한다.(추가적인 가정)
- 파라미터를 공유하기 때문에 input에 따라 가중치를 adaptive하게 변경할 수 없다.
- 강한 inductive bias는 데이터가 적을 때 일반화된 좋은 성능을 내는 모델을 만드는것에 유리하다. 수렴 또한 빠르다
- 하지만, Large scale의 global한 모델을 만들땐 inductive bias는 장애물이 된다.

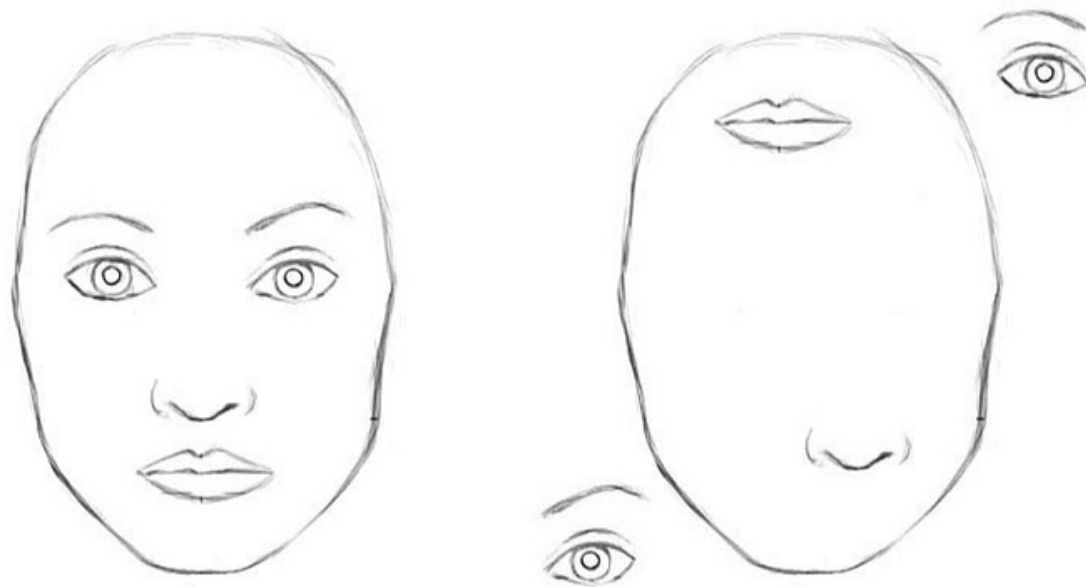
Transformer와 CNN

왜 Vision Transformer가 좋을까?

Big Transfer (BiT): General Visual Representation Learning

- 성능 향상을 위해서 새로운 Neural Network 구조 설계가 가장 중요한 것이 아니다.
- Large-scale dataset을 학습한 Pre-training 모델과 Large model size가 중요하다.

- CNN에서는 왼쪽 그림과 오른쪽 그림이 똑같이 사람의 얼굴이라고 판단할 가능성이 높다.
- 하지만 attention을 이용한다면 이미지 전체를 참고하기 때문에 사람의 얼굴이 아닌 것을 알 수 있다.



Abstract

Abstract

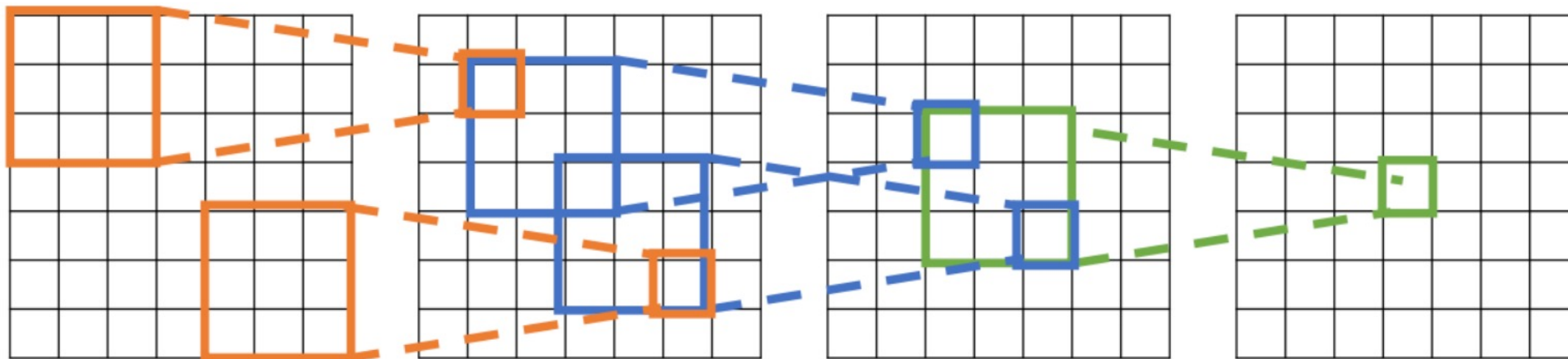
Compared to the great progress of large-scale vision transformers (ViTs) in recent years, large-scale models based on convolutional neural networks (CNNs) are still in an early state. This work presents a new large-scale CNN-based foundation model, termed InternImage, which can obtain the gain from increasing parameters and training data like ViTs. Different from the recent CNNs that focus on large dense kernels, InternImage takes deformable convolution as the core operator, so that our model not only has the large effective receptive field required for downstream tasks such as detection and segmentation, but also has the adaptive spatial aggregation conditioned by input and task information. As a result, the proposed InternImage reduces the strict inductive bias of traditional CNNs and makes it possible to learn stronger and more robust patterns with large-scale parameters from massive data like ViTs. The effectiveness of our model is proven on challenging benchmarks including ImageNet, COCO, and ADE20K. It is worth mentioning that InternImage-H achieved a new record 65.4 mAP on COCO test-dev and 62.9 mIoU on ADE20K, outperforming current leading CNNs and ViTs.

- CNN에 기반한 Large-scale 모델은 ViT와는 달리 아직 초기 단계이다.
- 현재의 CNN 연구는 대부분 **large effective receptive field**를 위한 large dense kernels(41x41 kernel 등)을 이용하는 것에 초점이 맞춰져 있다.
- InternImage는 parameters의 수와 training data를 ViT처럼 증가시켰다.
- InternImage는 현재 CNN의 연구처럼 **large effective receptive field**를 갖는 것 뿐만 아니라 **adaptive spatial aggregation**까지 갖는다.
- 이를 통해 CNN의 **strict inductive bias**를 줄여 강하고 robust한 패턴을 배운다.

Receptive Field

* 하나의 필터가 커버할 수 있는 영역

* 출력 레이어의 뉴런 하나에 영향을 미치는 입력 뉴런들의 공간 크기



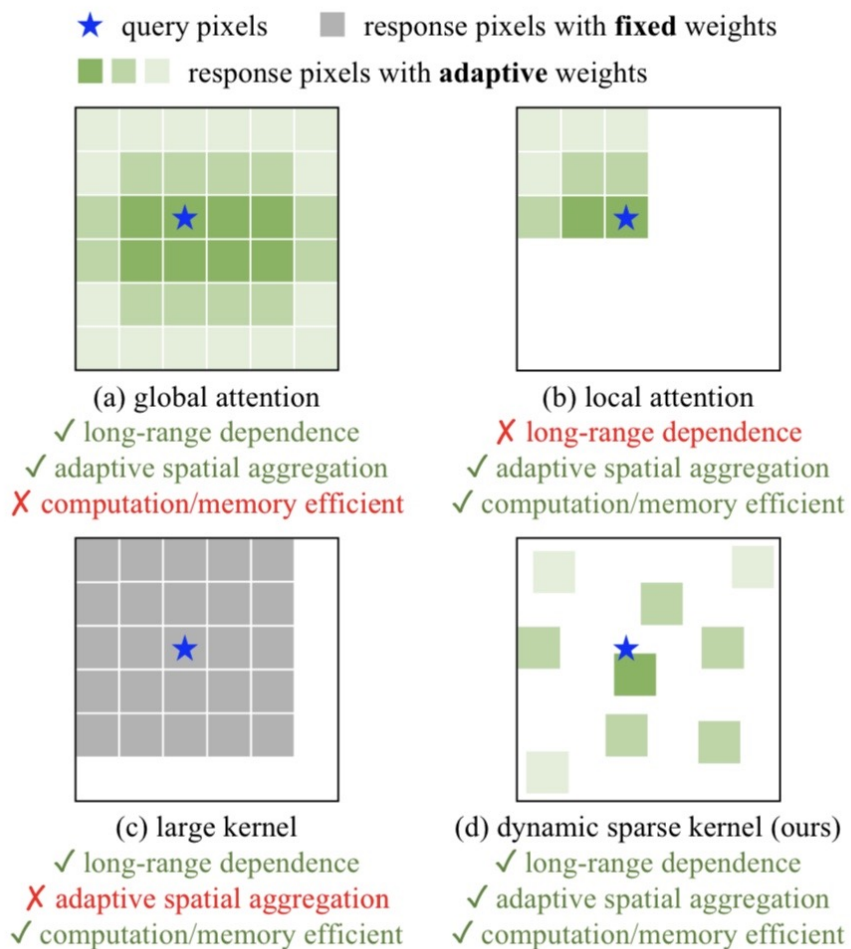
Input

Problem: For large images we need many layers
for each output to "see" the whole image

Output

Introduction

왜 Vision Transformer가 좋을까?



Vision Transformer가 좋은 이유는 크게 세가지로 볼 수 있다.

1. Long-range dependence
2. Adaptive spatial aggregation
3. Advanced components
 - Layer Normalization
 - Feed-forward network
 - GELU

1,2번은 결국 Multi-Head Self Attention(MHSA)를 통해 만들어진다.

결국 CNN을 이용해서 최대한 Vision Transformer와 비슷하게 만드는 것이 핵심이다. 특히 CNN이 MHSA의 효과를 내게 해야 한다!

Deformable Convolution

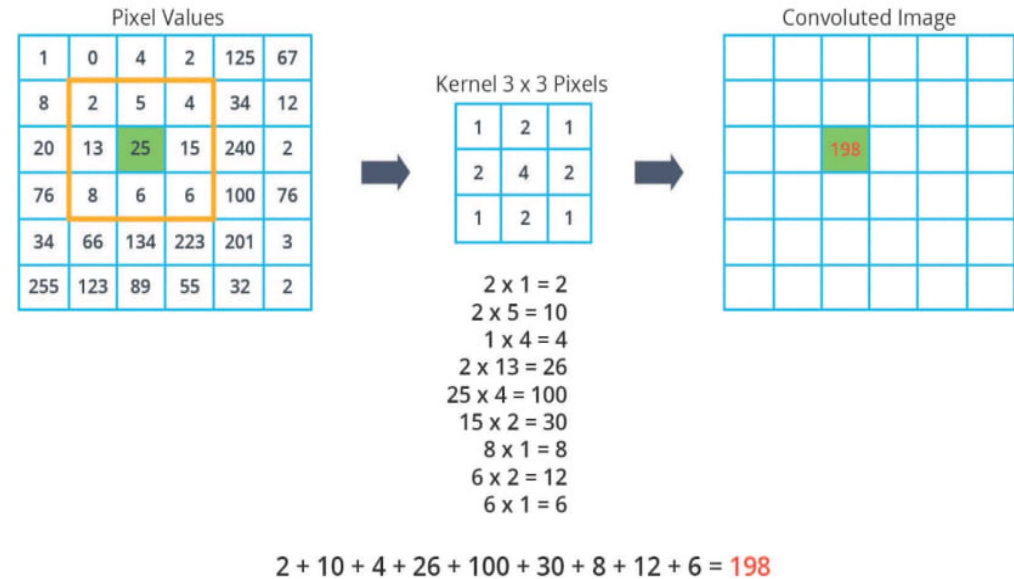
InternImage의 핵심 operator

1. CNN

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

$$\mathcal{R} = \{(-1, -1), (-1, 0), \dots, (0, 1), (1, 1)\}$$

- $w()$ = kernel
- $x()$ = image



Deformable Convolution

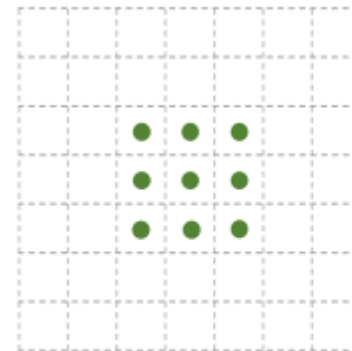
InternImage의 핵심 operator

1. Deformable Convolution

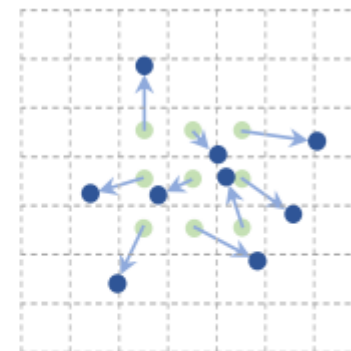
$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta \mathbf{p}_n).$$

$$\mathcal{R} = \{(-1, -1), (-1, 0), \dots, (0, 1), (1, 1)\}$$

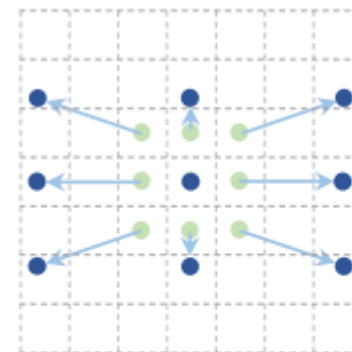
- $\mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n)$ 에서 $\Delta \mathbf{p}_n$ 만큼을 더 더해준다.
- 이를 통해 고정된 3x3 kernel이 아니라 원하는 point의 3x3 영역에서 convolution이 가능해진다.
- $\Delta \mathbf{p}_n$ 은 real value도 가능하다.
- -> Interpolation 이용



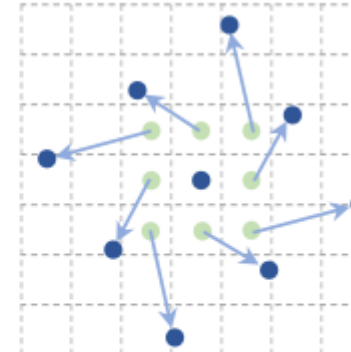
(a)



(b)



(c)

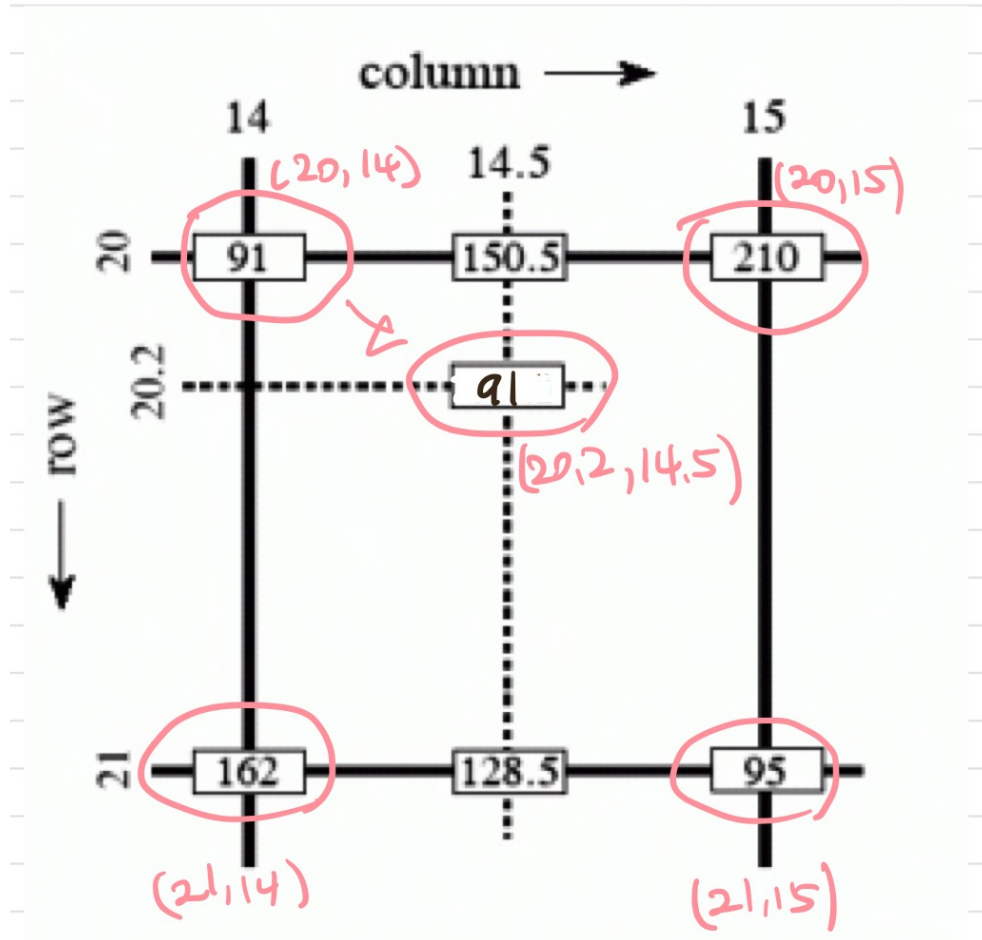


(d)

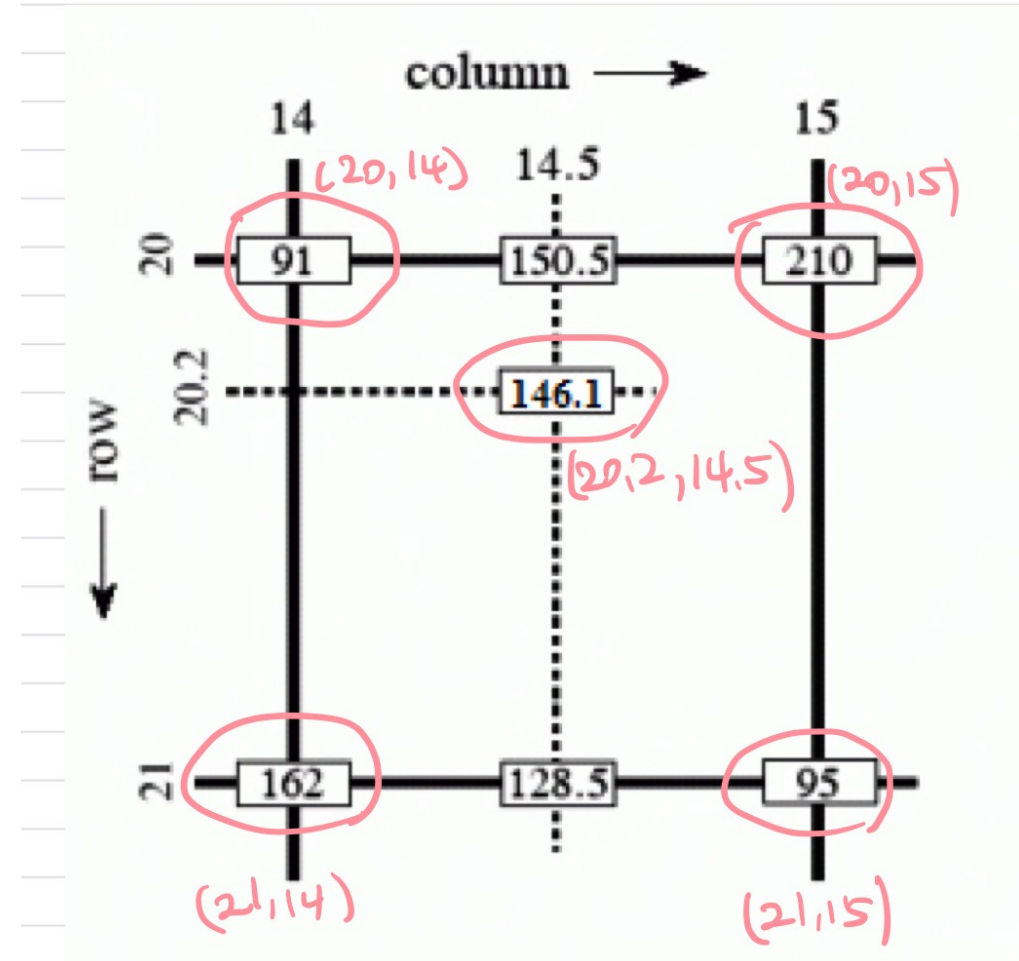
Deformable Convolution

Interpolation

Nearest Interpolation



Bilinear Interpolation



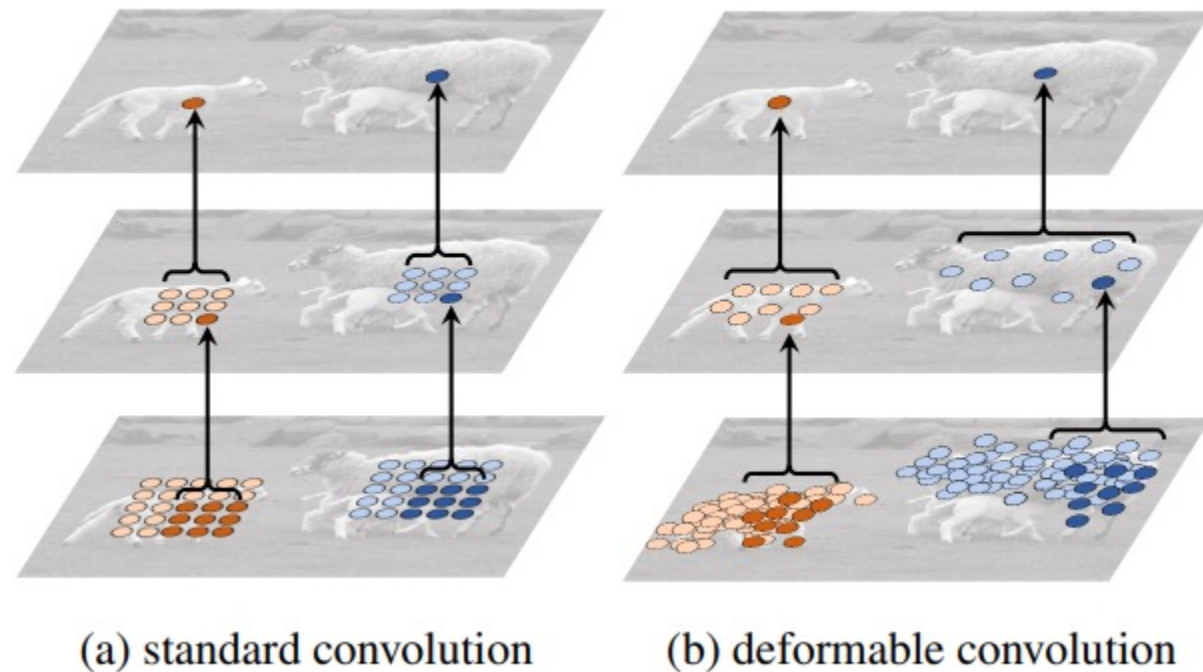
Deformable Convolution

InternImage의 핵심 operator

2. Deformable Convolution V2

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

- Deformable convolution에서 amplitude modulation을 수행한다.
- Δm_k 는 convolution으로 나온 amplitude를 modulation하는 기능을 한다.
- 필요 없는 영역의 amplitude를 0으로 만들 수 있다.

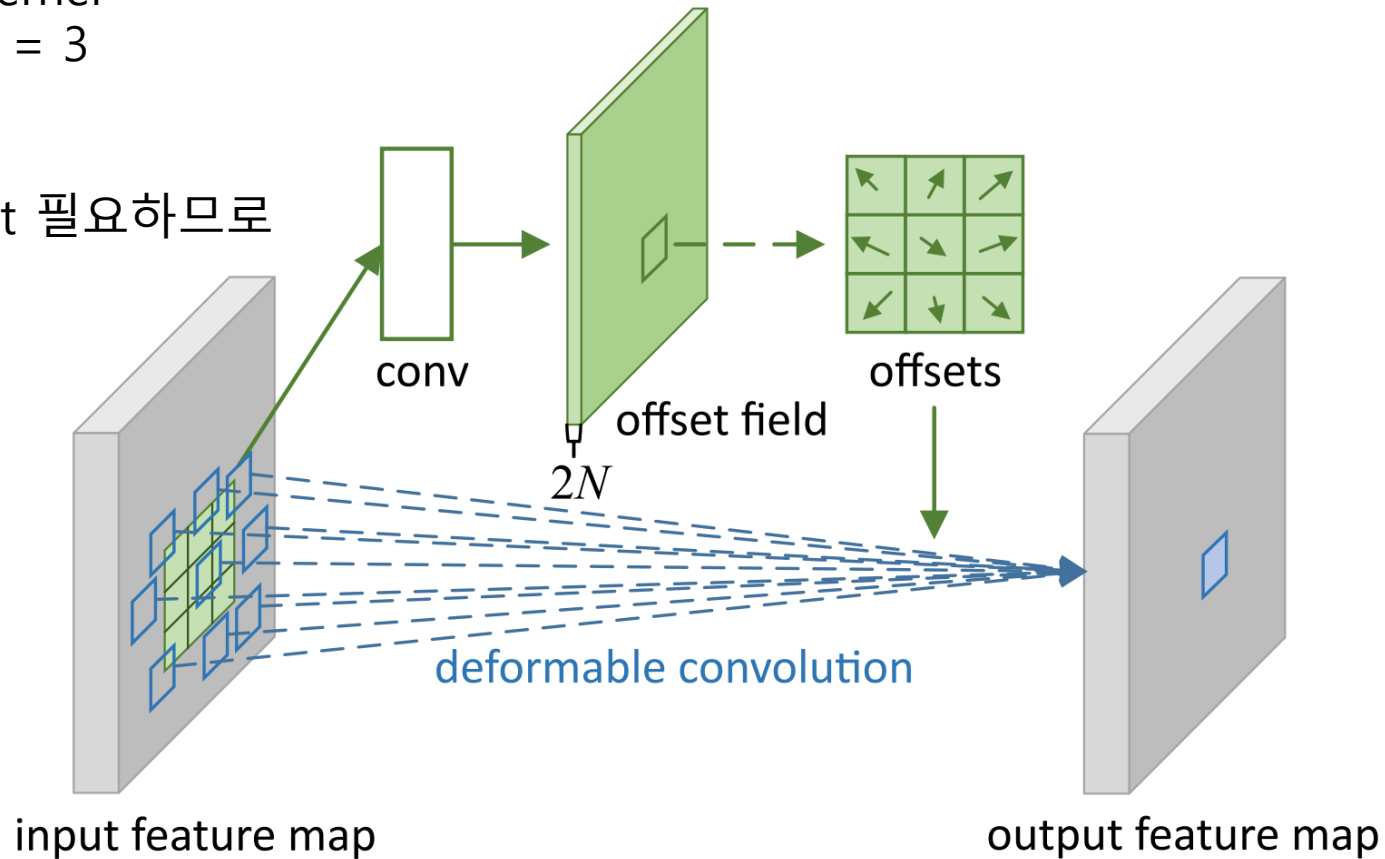


Deformable Convolution

InternImage의 핵심 operator

K = row, col of kernel
ex) 3x3 kernel $K = 3$
 $N = K^2$

x,y축으로의 offset 필요하므로
 $2N$ 개의 채널



Q&A

Deformable Convolution

InternImage의 핵심 operator

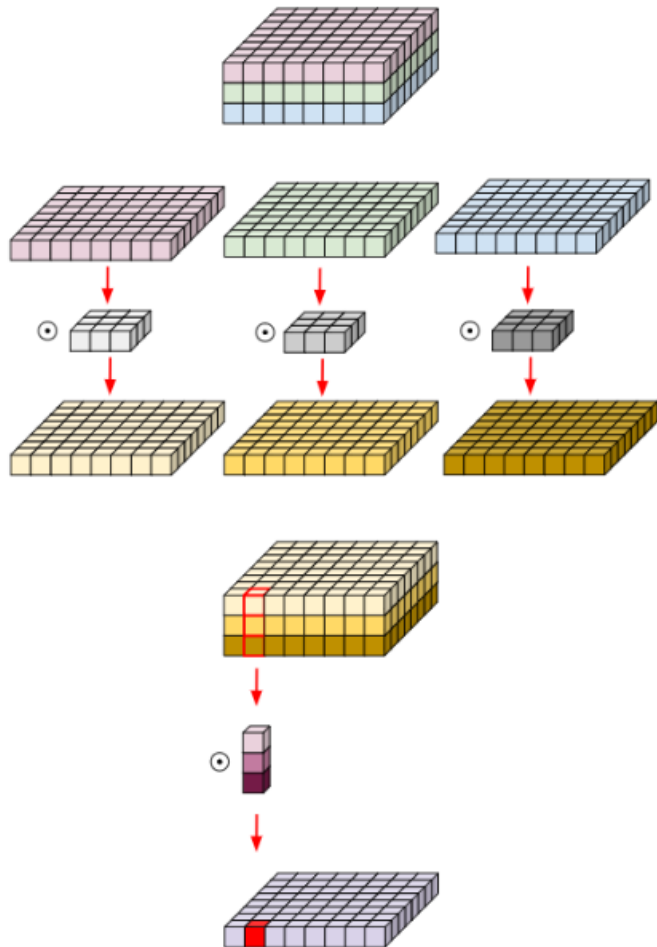
$$\mathbf{y}(p_0) = \sum_{g=1}^G \sum_{k=1}^K \mathbf{w}_g \mathbf{m}_{gk} \mathbf{x}_g(p_0 + p_k + \Delta p_{gk}),$$

1. Deformable Convolution V3

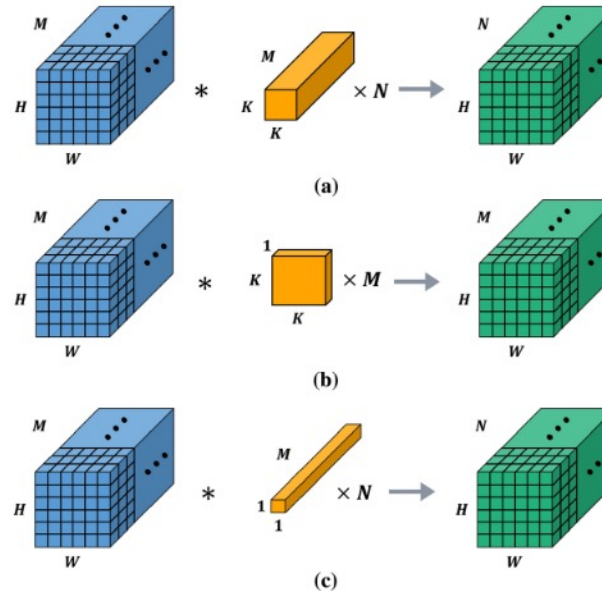
- 1. Sharing weights among convolutional neurons.
 - Computational cost를 줄이기 위해 depth-wise와 point-wise convolution으로 나눴다.
 - Depthwise Part는 modulation scalar M_k 를 이용하고 Pointwise Part는 shared projection weight w 를 이용했다.
- 2. Introducing multi-group mechanism
 - MHSA와 같이 Group convolution을 이용해서 풍부한 representation sub-space의 정보를 효과적으로 학습한다.
 - 이를 통해 MHSA와 같이 서로 다른 Spatial Aggregation pattern을 가질 수 있으므로 Downstream task에서 유용하다
- 3. Normalizing modulation scalars along sampling point.
 - Modulation 계산에 Sigmoid가 아닌 softmax함수를 이용했다.

Deformable Convolution

Depth-wise Separable Convolution



Depthwise Separable Convolution



(a) Normal Convolution

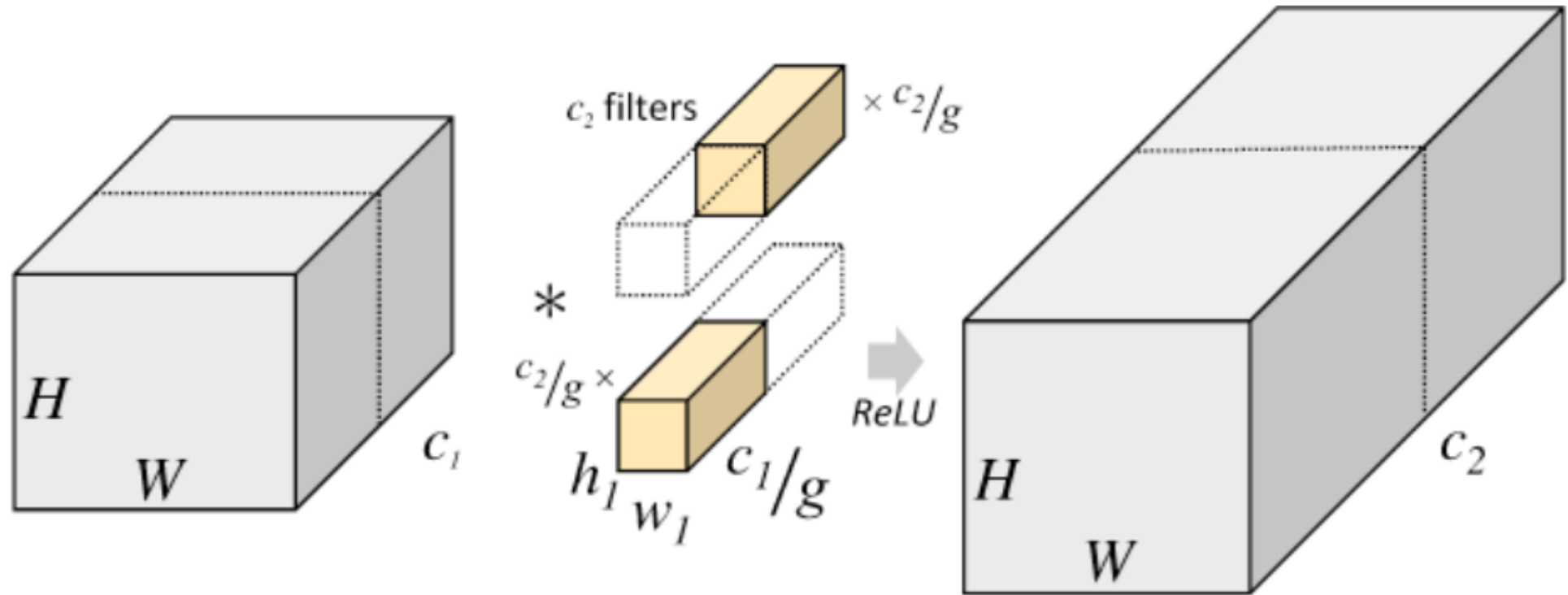
(b) Depthwise Convolution

(c) Pointwise Convolution

Normal Convolution보다 Depthwise Convolution + Pointwise Convolution의 연산량이 더 적다.

Deformable Convolution

Group Convolution



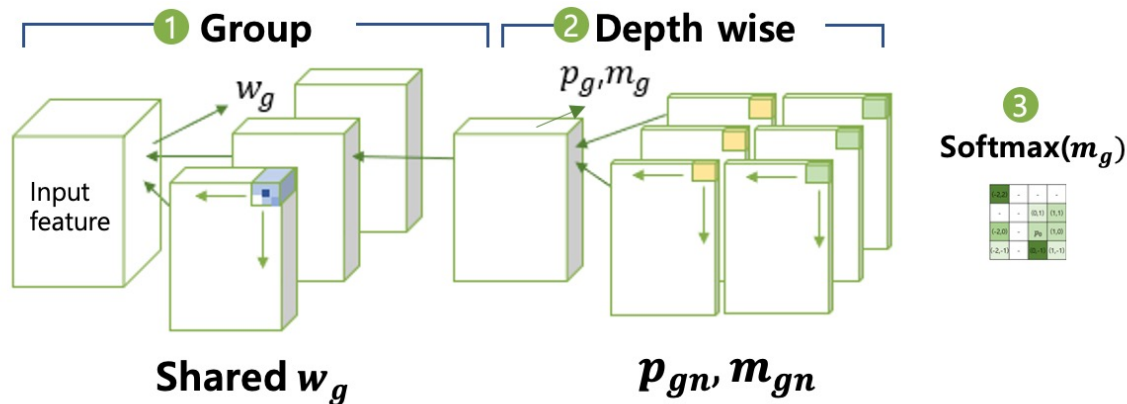
Deformable Convolution

정리

본 논문

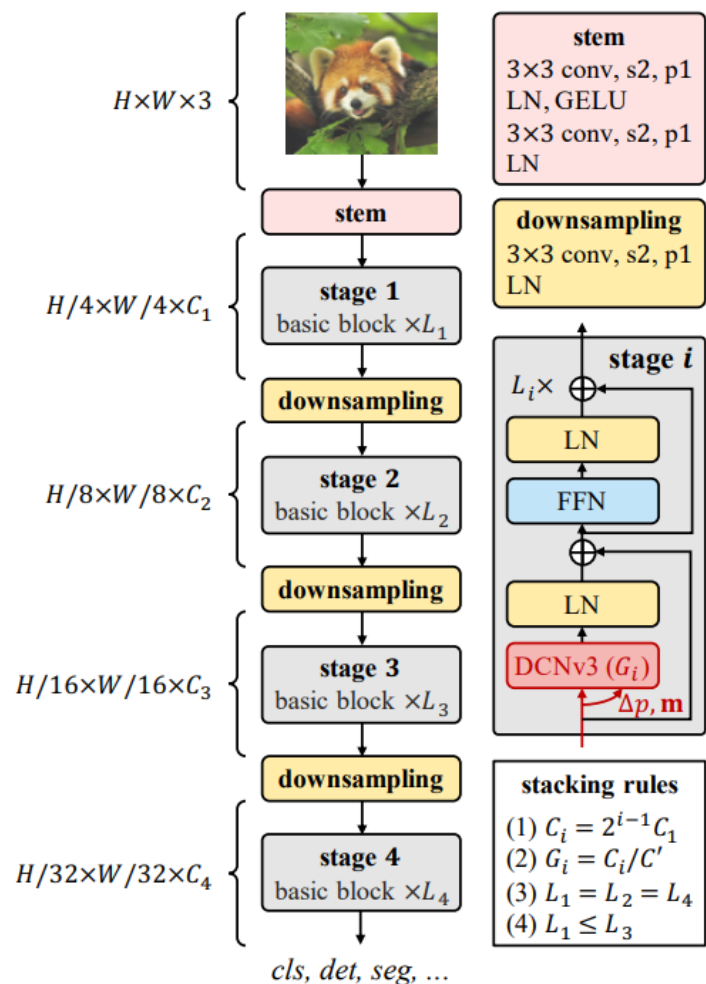
$$y(p_0) = \sum_{g=G} w_g \cdot \sum_{p_n \in R} x_g(p_0 + p_n + \Delta \mathbf{p}_{gn}) \cdot \Delta \mathbf{m}_{gn}$$

V3
V1
V2



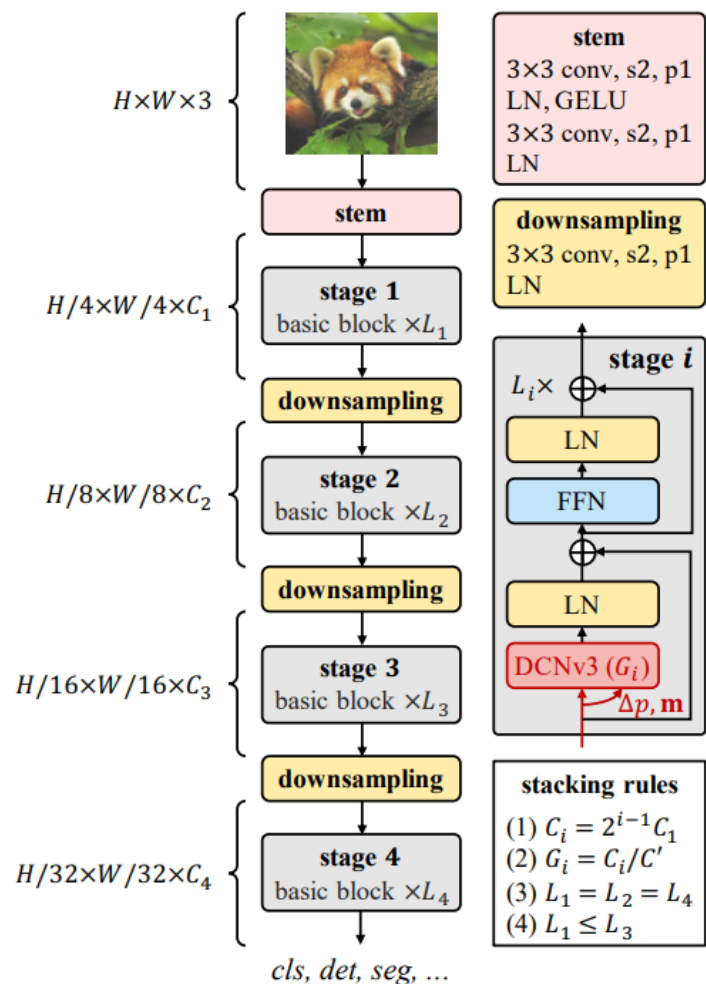
InternImage Architecture

InternImage



InternImage Architecture

InternImage



- Scaling rules
 - Efficient net에서 영감을 받아 더 큰 Model을 만들 때 사용할 rules를 만들었다.
 - Depth = (i.e., $3L_1 + L_3$) and Width C_1 이라는 두개의 Scaling Dimension을 고려했다.
 - 그리고 α, β and Composite Factor ϕ 를 가지고 scale을 진행했다.

$$D' = \alpha^\phi D \text{ and } C'_1 = \beta^\phi C_1,$$

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	32	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.

Experiment

Classification Imagenet Dataset

method	type	scale	#params	#FLOPs	acc (%)
DeiT-S [58]	T	224 ²	22M	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 [64]	T	224 ²	22M	5G	81.4
SwinV2-T/8 [16]	T	256 ²	28M	6G	81.8
Focal-T [65]	T	224 ²	29M	5G	82.2
ConvNeXt-T [21]	C	224 ²	29M	5G	82.1
ConvNeXt-T-dcls [66]	C	224 ²	29M	5G	82.5
SLaK-T [29]	C	224 ²	30M	5G	82.5
HorNet-T [43]	C	224 ²	23M	4G	83.0
InternImage-T (ours)	C	224 ²	30M	5G	83.5
PVT-L [10]	T	224 ²	61M	10G	81.7
Swin-S [2]	T	224 ²	50M	9G	83.0
CoAtNet-1 [20]	T	224 ²	42M	8G	83.3
PVTv2-B4 [11]	T	224 ²	63M	10G	83.6
SwinV2-S/8 [16]	T	256 ²	50M	12G	83.7
ConvNeXt-S [21]	C	224 ²	50M	9G	83.1
ConvNeXt-S-dcls [66]	C	224 ²	50M	10G	83.7
SLaK-S [29]	C	224 ²	55M	10G	83.8
HorNet-S [43]	C	224 ²	50M	9G	84.0
InternImage-S (ours)	C	224 ²	50M	8G	84.2

DeiT-B [58]	T	224 ²	87M	18G	83.1
Swin-B [2]	T	224 ²	88M	15G	83.5
CoAtNet-2 [20]	T	224 ²	75M	16G	84.1
PVTv2-B5 [11]	T	224 ²	82M	12G	83.8
DeiT III-B [64]	T	224 ²	87M	18G	83.8
SwinV2-B/8 [16]	T	256 ²	88M	20G	84.2
RepLkNet-31B [22]	C	224 ²	79M	15G	83.5
ConvNeXt-B [21]	C	224 ²	88M	15G	83.8
ConvNeXt-B-dcls [66]	C	224 ²	89M	17G	84.1
SLaK-B [29]	C	224 ²	95M	17G	84.0
HorNet-B [43]	C	224 ²	88M	16G	84.3
InternImage-B (ours)	C	224 ²	97M	16G	84.9
Swin-L [†] [2]	T	384 ²	197M	104G	87.3
CoAtNet-3 [†] [20]	T	384 ²	168M	107G	87.6
CoAtNet-4 [†] [20]	T	384 ²	275M	190G	87.9
DeiT III-L [†] [64]	T	384 ²	304M	191G	87.7
SwinV2-L/24 [†] [16]	T	384 ²	197M	115G	87.6
RepLkNet-31L [†] [22]	C	384 ²	172M	96G	86.6
HorNet-L [†] [43]	C	384 ²	202M	102G	87.7
ConvNeXt-L [†] [21]	C	384 ²	198M	101G	87.5
ConvNeXt-XL [†] [21]	C	384 ²	350M	179G	87.8
InternImage-L [†] (ours)	C	384 ²	223M	108G	87.7
InternImage-XL [†] (ours)	C	384 ²	335M	163G	88.0
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 [#] [67]	C	480 ²	928M	—	87.5
InternImage-H [#] (ours)	C	224 ²	1.08B	188G	88.9
InternImage-H [#] (ours)	C	640 ²	1.08B	1478G	89.6

Experiment

Object Detection COCO val12017

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-Adapter-S [69]	48M	403G	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	70.4	54.1	43.7	67.3	47.3
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	69.8	52.8	43.3	67.1	46.7	49.7	71.1	54.5	44.5	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-Adapter-B [69]	120M	832G	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	70.9	54.0	44.0	67.8	47.4	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					
			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-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 [†] [43]	259M	1358G	—	—	—	—	—	—	56.0	—	—	48.6	—	—
InternImage-L [†] (ours)	277M	1399G	54.9	74.0	59.8	47.7	71.4	52.1	56.1	74.8	60.7	48.5	72.4	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	74.4	60.1	48.1	71.9	52.4	56.2	75.0	61.2	48.8	72.5	53.4

Table 3. **Object detection and instance segmentation performance on COCO val12017.** 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.

method	detector	#params	AP ^v	
			val2017	test-dev
Swin-L [2]	DyHead [72]	213M	56.2	58.4
Swin-L [†] [2]	HTC++ [2]	284M	58.0	58.7
Swin-L [†] [2]	Soft-Teacher [73]	284M	60.7	61.3
Florence-CoSwin-H [#] [59]	DyHead [72]	637M	62.0	62.4
ViT-L [†] [9]	ViT-Adapter [69]	401M	62.6	62.6
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
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 val12017 and test-dev.**

Experiment

Semantic segmentation ADE20K

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	3.00B	—	—	61.4
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.

Experiment

Visualization sampling location



Figure 5. **Visualization of sampling locations for different groups at different stages.** The blue star indicates the query point (on the left sheep), and the dots with different colors indicate the sampling locations of different groups.

Conclusion & Limitation

- CNN 또한 large-scale vision foundation model research에서 생각해볼만한 선택지라는것을 시사하는 연구였다.
- 하지만 Deformable convolution operator를 high-speed가 필요한 task에 적용하는 것에는 연구가 필요하다. (high-speed task에는 아직 적용할 수 없다)
- Large-scale CNN은 아직 초기 연구단계이기 때문에 InternImage가 좋은 시작점이 됐으면 좋겠다.
- 개인적으로 이 논문이 '단순히 트랜스포머여서 좋다' 라는 생각을 바꿔주는 논문이었습니다. 결국 long range dependency와 adaptive weight가 large scale model의 중요한 요소였고 트랜스포머는 단순히 그 속성을 CNN보다 더 만족시키는 구조였을 뿐이라는 생각이 들었기 때문입니다. 따라서 트랜스포머 보다 해당 속성을 더 빠르고 충분히 만족시킬 수 있는 모델 또한 새로 등장할 수 있겠다는 생각이 들었습니다.

Q&A