Exploring Plain Vision Transformer Backbones for Object Detection

2022 Mar 30

鄧仲恩
NTUST GAMELab

Outline

1. Introduction
2. Related Work
3. Method
4. Experiments
5. Conclusion

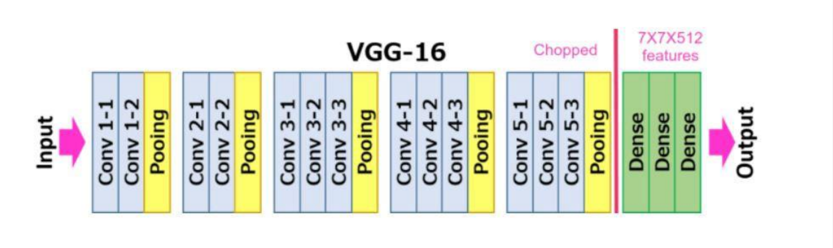
Introduction

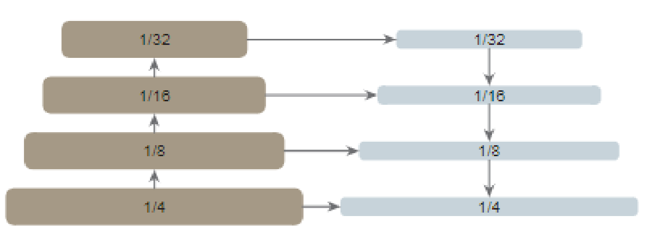
Introduction – Detection CNN base

Backbones → Neck → Head

1. VGG
2. ResNet
3. DarkNet

1. Region Proposal Networks (RPN)
2. Region-of-Interest (RoI)
3. Feature Pyramid Networks (FPN)

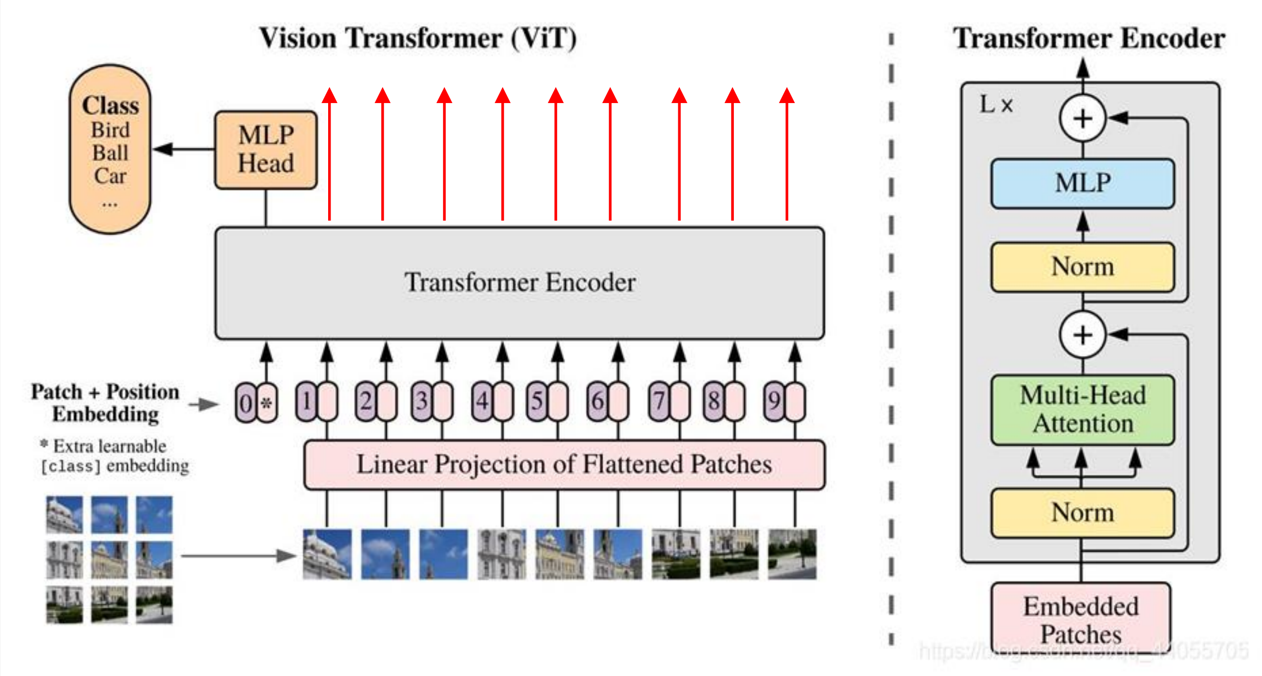




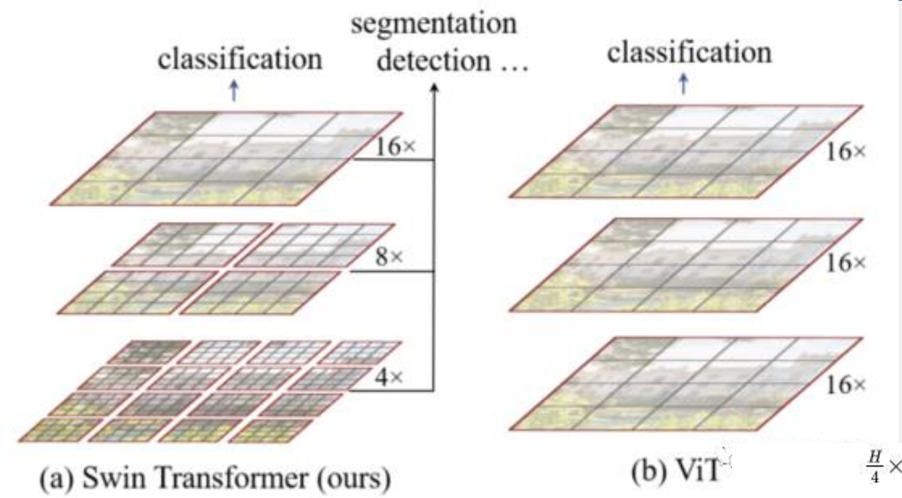
hierarchical backbone, w/ FPN

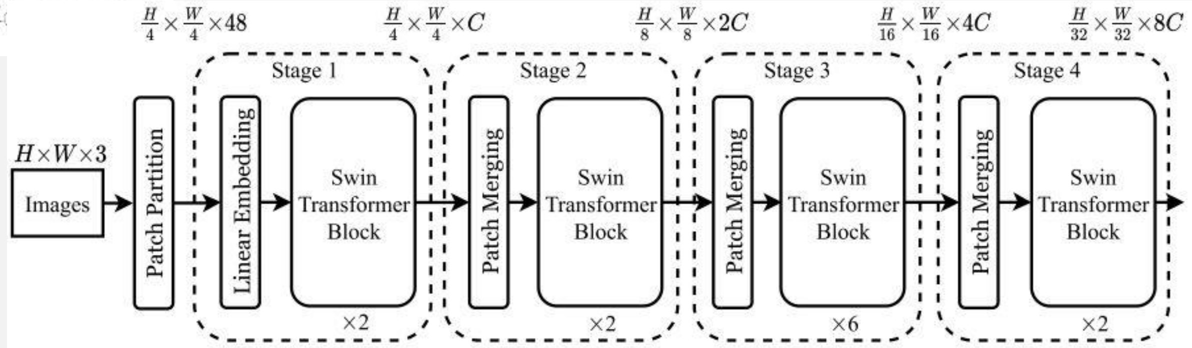
Introduction – Vision Transformers

1. Vision Transformers
2. Swin Transformers



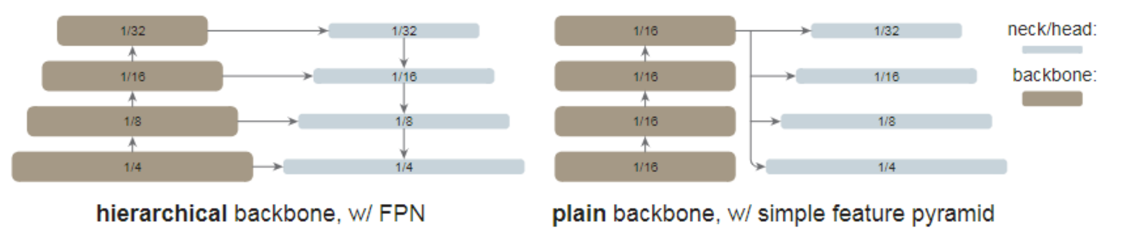
Introduction – Swin





Introduction – Target

1. Use plain, non-hierarchical backbones. (Vision Transformers)
2. Independence of upstream vs. downstream tasks.
3. Use a simple feature pyramid.
4. Use Masked Autoencoder (MAE) pretraining.
5. Compete with the hierarchical-backbone detectors. (Swin, MViT)



Related Work

Related Work - Object detector backbones

1. Pioneered by the work of R-CNN.
2. SSD is the first works that leverage the hierarchical nature of the ConvNet backbones (VGG).
3. FPN pushes this direction further by using all stages of a hierarchical backbone, approached by lateral and top-down connections.
4. ViT is a powerful alternative to standard ConvNets for image classification. (Swin , MViT , PVT , PiT)

Related Work - Plain-backbone detectors

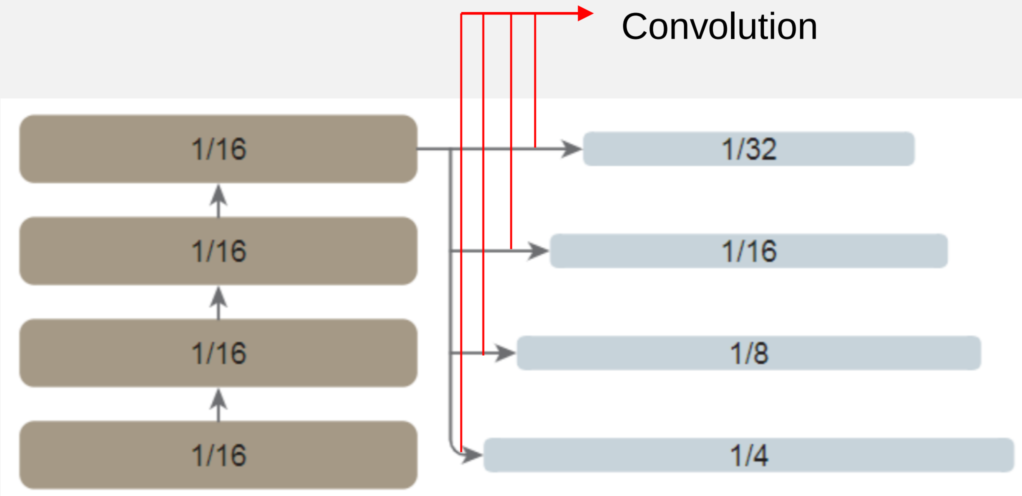
1. ViT has inspired people to push the frontier of plain backbones for object detection.
2. UViT is presented as a single-scale Transformer for object detection.
   1. depth, width, input resolution.
   2. window attention strategy

Related Work - Object detection methodologies

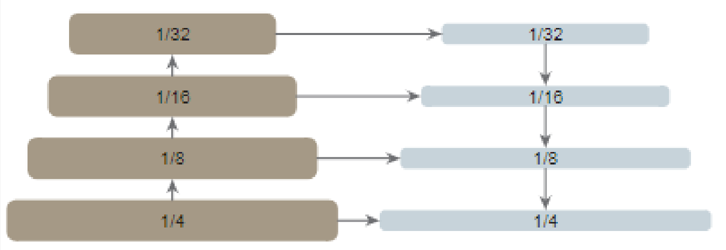
1. two-stage (R-CNN, Fast R-CNN, Faster R-CNN, SPP-Net) vs. one-stage (YOLO, SSD, RetinaNet)
2. anchor-based (Faster R-CNN) vs. anchor-free (FCOS, CenterNet, CornerNet)
3. region-based (R-CNN, Fast R-CNN, Faster R-CNN, SPP-Net) vs. query-based (DETR)
4. **Plain vs. Hierarchical**

Method

Method - Simple feature pyramid



(c) simple feature pyramid

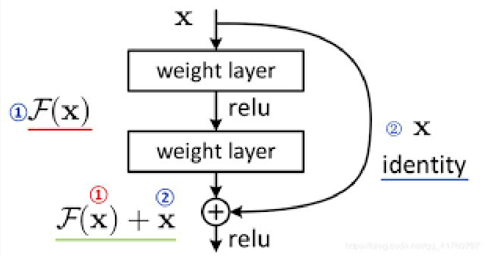


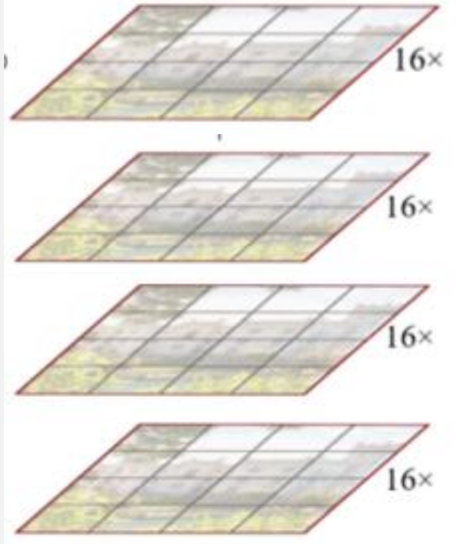
hierarchical backbone, w/ FPN

Method - Backbone adaptation

1. Global propagation

2. Convolutional propagation





propagation strategy → Simple feature pyramid

propagation strategy → Simple feature pyramid

propagation strategy → Simple feature pyramid

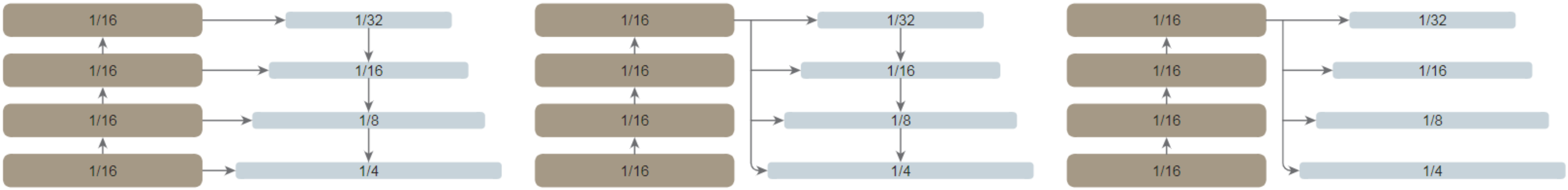
propagation strategy → Simple feature pyramid

Method - Implementation

1. Pretraining backbones : ViT-B, ViT-L, ViT-H with MAE
2. Patch size : 16
3. Detector heads : Mask R-CNN or Cascade Mask R-CNN
4. Input image : 1024 X 1024
5. Augmented : large-scale jittering
6. Dataset : COCO train2017/val2017
7. Optimizer : AdamW

Experiments

Experiments - Ablation Study and Analysis



(a) FPN, 4-stages (b) FPN, last map (c) simple feature pyramid

pyramid design	ViT-B AP^{box}	ViT-B AP^{mask}	ViT-L AP^{box}	ViT-L AP^{mask}
no feature pyramid	47.8	42.5	51.2	45.4
(a) FPN, 4-stage	50.3 (+2.5)	44.9 (+2.4)	54.4 (+3.2)	48.4 (+3.0)
(b) FPN, last-map	50.9 (+3.1)	45.3 (+2.8)	**54.6** (+3.4)	48.5 (+3.1)
(c) simple feature pyramid	**51.2** (+3.4)	**45.5** (+3.0)	**54.6** (+3.4)	**48.6** (+3.2)

Experiments - Ablation Study and Analysis

prop. strategy	AP^{box}	AP^{mask}
none	52.9	47.2
4 global blocks	54.6 (+1.7)	48.6 (+1.4)
4 conv blocks	**54.8** (+1.9)	**48.8** (+1.6)
shifted win.	54.0 (+1.1)	47.9 (+0.7)

(a) Window attention with various cross-window propagation strategies.

prop. conv	AP^{box}	AP^{mask}
none	52.9	47.2
naïve	54.3 (+1.4)	48.3 (+1.1)
basic	**54.8** (+1.9)	**48.8** (+1.6)
bottleneck	54.6 (+1.7)	48.6 (+1.4)

(b) Convolutional propagation with different residual block types (4 blocks).

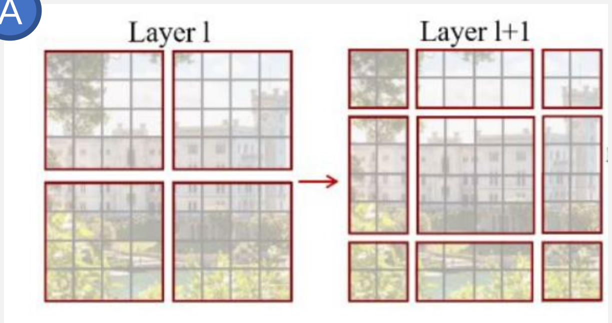
prop. locations	AP^{box}	AP^{mask}
none	52.9	47.2
first 4 blocks	52.9 (+0.0)	47.1 (–0.1)
last 4 blocks	54.3 (+1.4)	48.3 (+1.1)
evenly 4 blocks	**54.6** (+1.7)	**48.6** (+1.4)

(c) Locations of cross-window global propagation blocks.

prop. blks	AP^{box}	AP^{mask}
none	52.9	47.2
2	54.4 (+1.5)	48.5 (+1.3)
4	54.6 (+1.7)	48.6 (+1.4)
$24^{\dagger}$	**55.1** (+2.2)	**48.9** (+1.7)

(d) Number of global propagation blocks. †: Memory optimization required.

A



B

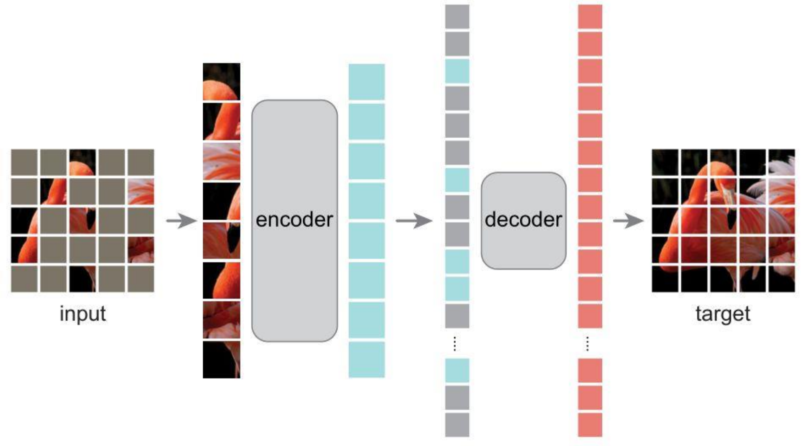
Naïve : 3x3 conv
Basic : two 3×3 conv
Bottleneck : 1x1 -> 3x3 -> 1x1

Experiments - Ablation Study and Analysis

prop. strategy	AP^{box}	# params	train mem	test time
none	52.9	1.00× (331M)	1.00× (14.6G)	1.00× (88ms)
4 conv (bottleneck)	54.6 (+1.7)	1.04×	1.05×	1.04×
4 global	54.6 (+1.7)	1.00×	1.39×	1.16×
24 global	55.1 (+2.2)	1.00×	3.34×†	1.86×

Experiments - Ablation Study and Analysis

pre-train	ViT-B AP^{box}	ViT-B AP^{mask}	ViT-L AP^{box}	ViT-L AP^{mask}
none (random init.)	48.1	42.6	50.0	44.2
IN-1K, supervised	47.6 (−0.5)	42.4 (−0.2)	49.6 (−0.4)	43.8 (−0.4)
IN-21K, supervised	47.8 (−0.3)	42.6 (+0.0)	50.6 (+0.6)	44.8 (+0.6)
IN-1K, MAE	**51.2** (+3.1)	**45.5** (+2.9)	**54.6** (+4.6)	**48.6** (+4.4)



Experiments - Comparisons with Hierarchical Backbones

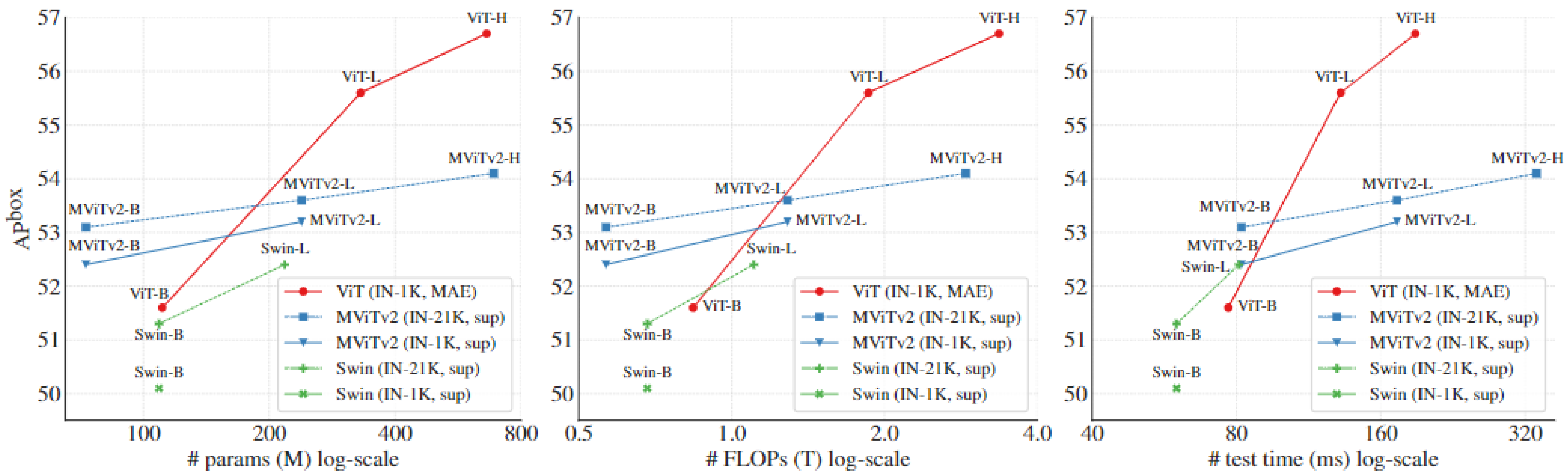
backbone	pre-train	Mask R-CNN AP^{box}	Mask R-CNN AP^{mask}	Cascade Mask R-CNN AP^{box}	Cascade Mask R-CNN AP^{mask}
hierarchical-backbone detectors:					
Swin-B	21K, sup	51.4	45.4	54.0	46.5
Swin-L	21K, sup	52.4	46.2	54.8	47.3
MViTv2-B	21K, sup	53.1	47.4	55.6	48.1
MViTv2-L	21K, sup	53.6	47.5	55.7	48.3
MViTv2-H	21K, sup	54.1	47.7	55.8	48.3
our plain-backbone detectors:					
ViT-B	1K, MAE	51.6	45.9	54.0	46.7
ViT-L	1K, MAE	55.6	49.2	57.6	49.8
ViT-H	1K, MAE	**56.7**	**50.1**	**58.7**	**50.9**

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

$$\Downarrow$$

$$Attention(Q,K,V) = Softmax(QK^T + B)V$$

Experiments - Comparisons with Hierarchical Backbones



Experiments - Comparisons with Hierarchical Backbones

method	framework	pre-train	single-scale test AP^{box}	single-scale test AP^{mask}	multi-scale test AP^{box}	multi-scale test AP^{mask}
hierarchical-backbone detectors:						
Swin-L [40]	HTC++	21K, sup	57.1	49.5	58.0	50.4
MViTv2-L [32]	Cascade	21K, sup	56.9	48.6	58.7	50.5
MViTv2-H [32]	Cascade	21K, sup	57.1	48.8	58.4	50.1
CBNetV2 [34]†	HTC	21K, sup	59.1	51.0	59.6	51.8
SwinV2-L [39]	HTC++	21K, sup	58.9	51.2	60.2	52.1
plain-backbone detectors:						
UViT-S [8]	Cascade	1K, sup	51.9	44.5	-	-
UViT-B [8]	Cascade	1K, sup	52.5	44.8	-	-
ViTDet, ViT-B	Cascade	1K, MAE	56.0	48.0	57.3	49.4
ViTDet, ViT-L	Cascade	1K, MAE	59.6	51.1	60.4	52.2
ViTDet, ViT-H	Cascade	1K, MAE	**60.4**	**52.0**	61.3	53.1

Comparisons on COCO

1. input size 1024->1080
2. adopt soft-nms

Experiments - Comparisons with Hierarchical Backbones

method	pre-train	AP^{mask}	AP^{mask}_{rare}	AP^{box}
hierarchical-backbone detectors:				
Copy-Paste [18]	unknown	38.1	32.1	41.6
Detic [56]	21K, sup; CLIP	41.7	41.7	-
competition winner 2021 [17]†, baseline	21K, sup	43.1	34.3	-
competition winner 2021 [17]†, full	21K, sup	**49.2**	**45.4**	-
plain-backbone detectors:				
ViTDet, ViT-L	1K, MAE	46.0	34.3	51.2
ViTDet, ViT-H	1K, MAE	48.1	36.9	53.4

LVIS
1.1203 classes
2. long-tailed object distribution

Comparisons on LVIS
1. federated loss
2. repeat factor sampling

HTC+CBNetV2+2*Swim-L

Conclusion

Conclusion

1. Plain-backbone detection is a promising research direction.
2. Decoupling pretraining from fine-tuning will generally benefit the community.
3. Plain-backbone detector has benefited from pretrained models from MAE.

We hope this methodology will also help bring the fields of computer vision and NLP closer.

報告完畢 THE END

謝謝 Thank You