Vision & Learning Seminar

Instance Segmentation - CVPR2022

Feng Zhu

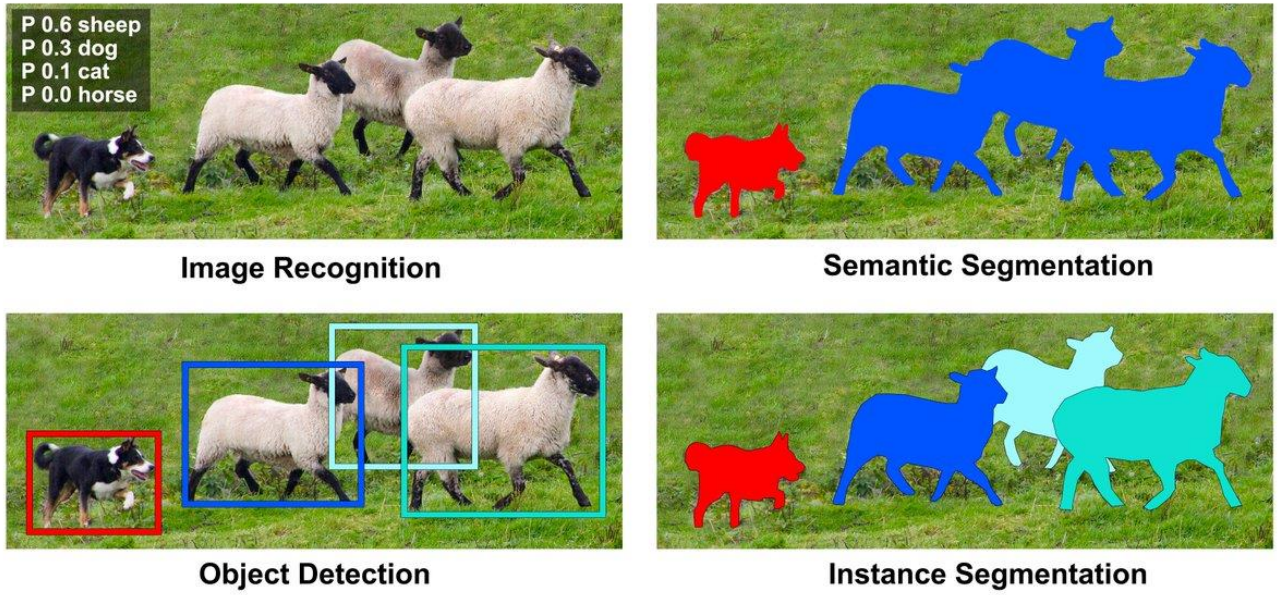
Instance Segmentation

Segmentation task: i**nstance segmentation, semantic segmentation** and **panoptic segmentation.**

Instance segmentation is the task of **detecting and delineating each distinct object** of interest appearing in an image.

Extend on **object detection**, need to predict masks of each object instead of bounding box.

Semantic segmentation, or image segmentation, is the task of clustering parts of an image together which belong to the same object class.



Instance Segmentation - CVPR2022

Trends:

1. Less fully-supervised work, more **weakly-supervised and self-supervised** work.
2. More **video segmentation** work.
3. More **panoptic segmentation** work and **3D instance segmentation** work.
4. Focused on **boundary refinement** and **speed-up**.
5. **Open world and open vocabulary** instance segmentation, with language (e.g. CLIP).

Instance Segmentation - CVPR2022

Paper sharing:

1. AdaMixer: A Fast-Converging Query-Based Object Detector. (oral, detailed)

2. Sparse Instance Activation for Real-Time Instance Segmentation. (detailed)

3. Mask Transfiner for High-Quality Instance Segmentation. (detailed)

4. Pointly-Supervised Instance Segmentation. (oral, brief)

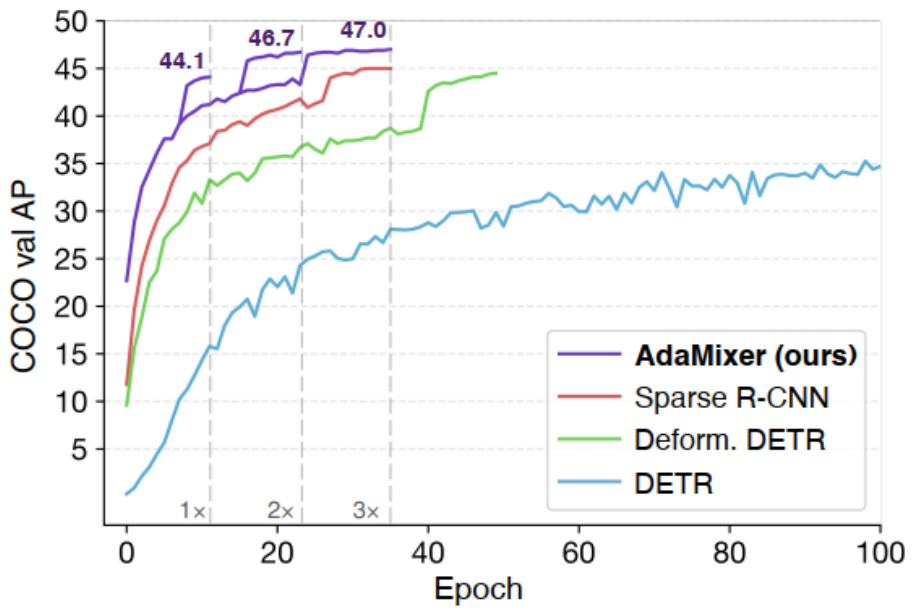
5. FreeSOLO: Learning to Segment Objects without Annotations. (brief)

AdaMixer: A Fast-Converging Query-Based Object Detector (oral)

Ziteng Gao[1] Limin Wang[1] Bing Han[2] Sheng Guo[2]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]MYbank, Ant Group, China

Motivation: The recent query-based object detectors still suffers from **slow convergence**, **limited performance**, and **design complexity of extra networks** between backbone and decoder.

The paper finds the key to these issues is the **adaptability of decoders** for casting queries to varying objects.



AdaMixer: A Fast-Converging Query-Based Object Detector (oral)

Revisit previous query-based method

1. DETR and Deformable DETR can **not be adaptive to decode** content, linear projection of features;

2. Sparse RCNN use adaptive bbox proposal and dynamic interactive head to decode content, but still restricted to **bbox and specific feature level** of FPN.

	adaptive to decode locations?	adaptive to decode content?	extra networks before the query decoder[1]?
DETR [4]	*yes*, multi-head attention aggregation	*no*, linear projection	TransformerEncoder
Deformable DETR [56]	*yes*, multi-scale multi-head adaptive sampling	*no*, linear projection[2]	Multi-scale DeformTransEncoder
Sparse R-CNN [39]	*restricted*, RoIAlign [17]	*partially yes*, adaptive point-wise conv.	FPN
AdaMixer (ours)	*yes*, adaptive 3D sampling	*yes*, adaptive channel and spatial mixing	linear projection to form 3D feature space

Ideas:

1. The decoder should adaptively decide which features to sample regarding the query, and adpative in both **spaital and scales dimension**.

2. How to adaptively decode the features? To capture correlation in **spatial and channel** dimension.

AdaMixer: A Fast-Converging Query-Based Object Detector (oral)

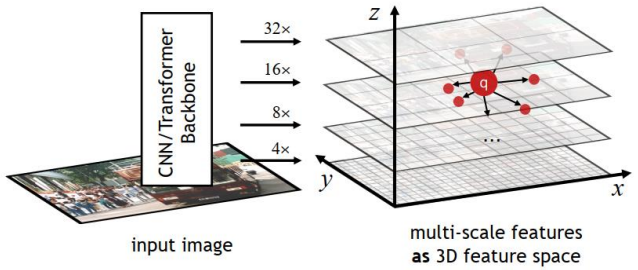


Figure 2. **3D feature sampling process.** A query first obtains sampling points in the 3D feature space and then perform 3D interpolation on these sampling points.

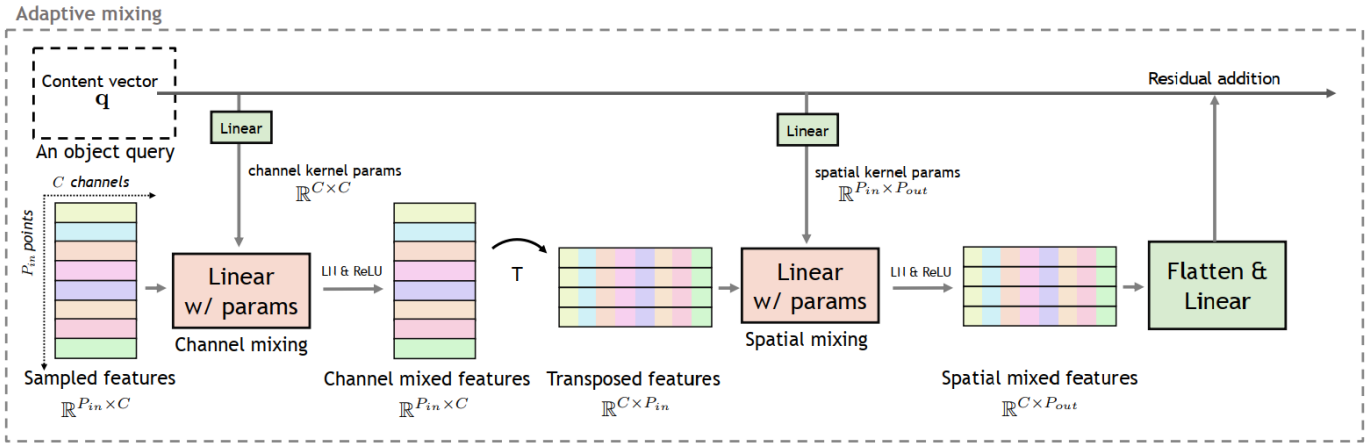


Figure 3. **Adaptive mixing procedure** between an object query and sampled features. The object query first generates adaptive mixing weights and then apply these weights to mix sampled features in the channel and spatial dimension. Note that for clarity, we demonstrate adaptive mixing for one sampling group.

Method:

1. 3D feature sampling: treat multi-scale feature maps as **3D feature space**, make query able to handle both location and scale variations. (group sampling)

2. Adaptive mixing: **channel and spatial mixing** to the sampled features with **dynamic kernels**.

AdaMixer: A Fast-Converging Query-Based Object Detector (oral)

Experiments

detector	epochs	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
FCOS [40]	12	38.7	57.4	41.8	22.9	42.5	50.1
Cascade R-CNN [3]	12	40.4	58.9	44.1	22.8	43.7	54.0
GFocalV2 [22]	12	41.1	58.8	44.9	23.5	44.9	53.3
BorderDet [33]	12	41.4	59.4	44.5	23.6	45.1	54.6
Dynamic Head [8]	12	42.6	60.1	**46.4**	**26.1**	**46.8**	56.0
DETR [4]	12	20.0	36.2	19.3	6.0	20.5	32.2
Deformable DETR [56]	12	35.1	53.6	37.7	18.2	38.5	48.7
Sparse R-CNN [39]	12	37.9	56.0	40.5	20.7	40.0	53.5
AdaMixer (N=100)	12	**42.7**	**61.5**	45.9	24.7	45.4	**59.2**
AdaMixer (N=300)	12	**44.1**	**63.4**	**47.4**	**27.0**	**46.9**	**59.5**
AdaMixer (N=500)	12	**45.0**	**64.2**	**48.6**	**27.9**	**47.8**	**61.1**

Table 3. **1× training scheme** performance on COCO `minival` set with different detectors and ResNet-50 as backbone.

Fast convergence: 1 X training scheme outperforms SOTA.

AdaMixer: A Fast-Converging Query-Based Object Detector (oral)

Experiments

Ablation Study

adaptive loc.	adaptive cont.	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
		35.7	55.2	37.8	20.1	38.1	48.8
✓		37.3	55.8	39.7	20.7	40.1	50.9
	✓	40.4	60.5	43.4	23.0	42.5	56.7
✓	✓	**42.7**	**61.5**	**45.9**	**24.7**	**45.4**	**59.2**

(a) **Adaptability of decoding** sampling locations and sampled content.

mixing	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
ACMACM	41.5	60.5	44.3	23.5	44.1	57.4
ASMASM	39.8	58.8	42.6	22.8	42.4	56.1
ACMASM	**42.7**	**61.5**	**45.9**	**24.7**	**45.4**	**59.2**
ASMACM	41.5	60.4	44.5	23.9	44.4	57.1

(b) **Design** in our adaptive mixing procedure.

pyramid	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
FPN [24]	42.1	61.0	45.0	24.1	44.8	58.7
PAFPN [27]	41.7	60.5	44.7	23.5	44.6	58.7
-	**42.7**	**61.5**	**45.9**	**24.7**	**45.4**	**59.2**

(c) **Extra pyramid networks** after the backbone?

P_{in}	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
8	41.2	60.3	44.1	24.0	43.9	57.2
16	41.8	60.9	44.5	24.5	44.6	58.4
32	**42.7**	**61.5**	45.9	24.7	45.4	59.2
64	**42.7**	**61.5**	**46.1**	**24.9**	**45.5**	**59.3**

(d) **Sampling points** P_{in} per group.

P_{out}	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
32	41.1	60.0	44.0	24.5	43.6	57.2
64	42.1	61.2	45.0	24.0	44.8	57.8
128	**42.7**	**61.5**	**45.9**	**24.7**	**45.4**	**59.2**
256	42.4	61.4	45.5	24.4	45.0	58.7

(e) **Spatial mixing out patterns** P_{out} per group.

pos. inf. sinus.	IoF	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
		41.2	59.6	44.2	23.6	43.5	57.9
✓		41.5	59.9	44.3	23.6	44.0	57.8
	✓	42.2	61.2	45.0	**24.8**	45.1	58.8
✓	✓	**42.7**	**61.5**	**45.9**	24.7	**45.4**	**59.2**

(f) **Position information** in self-attention between queries.

Table 4. **AdaMixer ablation experiments** with ResNet-50 on MS COCO `minival` set. Default choice for our model is colored gray

AdaMixer: A Fast-Converging Query-Based Object Detector (oral)

Experiments

SOTA comparison: bbox AP 51.3

detector	backbone	encoder/pyramid net	#epochs	GFLOPs	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
DETR [4]	ResNet-50-DC5	TransformerEnc	500	187	43.3	63.1	45.9	22.5	47.3	61.1
SMCA [13]	ResNet-50	TransformerEnc	50	152	43.7	63.6	47.2	24.2	47.0	60.4
Deformable DETR [56]	ResNet-50	DeformTransEnc	50	173	43.8	62.6	47.7	26.4	47.1	58.0
Sparse R-CNN [39]	ResNet-50	FPN	**36**	174	45.0	63.4	48.2	26.9	47.2	59.5
Efficient DETR [49]	ResNet-50	DeformTransEnc	**36**	210	45.1	63.1	49.1	28.3	48.4	59.0
Conditional DETR [31]	ResNet-50-DC5	TransformerEnc	108	195	45.1	65.4	48.5	25.3	49.0	**62.2**
Anchor DETR [46]	ResNet-50-DC5	DecoupTransEnc	50	151	44.2	64.7	47.5	24.7	48.2	60.6
AdaMixer (ours)	ResNet-50	-	**12**	132	44.1	63.1	47.8	29.5	47.0	58.8
AdaMixer (ours)	ResNet-50	-	**24**	132	46.7	65.9	50.5	29.7	49.7	61.5
AdaMixer (ours)	ResNet-50	-	**36**	132	**47.0**	**66.0**	**51.1**	**30.1**	**50.2**	61.8
DETR [4]	ResNet-101-DC5	TransformerEnc	500	253	44.9	64.7	47.7	23.7	49.5	62.3
SMCA [13]	ResNet-101	TransformerEnc	50	218	44.4	65.2	48.0	24.3	48.5	61.0
Sparse R-CNN [39]	ResNet-101	FPN	**36**	250	46.4	64.6	49.5	28.3	48.3	61.6
Efficient DETR [49]	ResNet-101	DeformTransEnc	**36**	289	45.7	64.1	49.5	28.2	49.1	60.2
Conditional DETR [31]	ResNet-101-DC5	TransformerEnc	108	262	45.9	66.8	49.5	27.2	50.3	63.3
AdaMixer (ours)	ResNet-101	-	**36**	208	**48.0**	**67.0**	**52.4**	**30.0**	**51.2**	**63.7**
AdaMixer (ours)	ResNeXt-101-DCN	-	**36**	214	**49.5**	**68.9**	**53.9**	**31.3**	**52.3**	**66.3**
AdaMixer (ours)	Swin-S	-	**36**	234	**51.3**	**71.2**	**55.7**	**34.2**	**54.6**	**67.3**

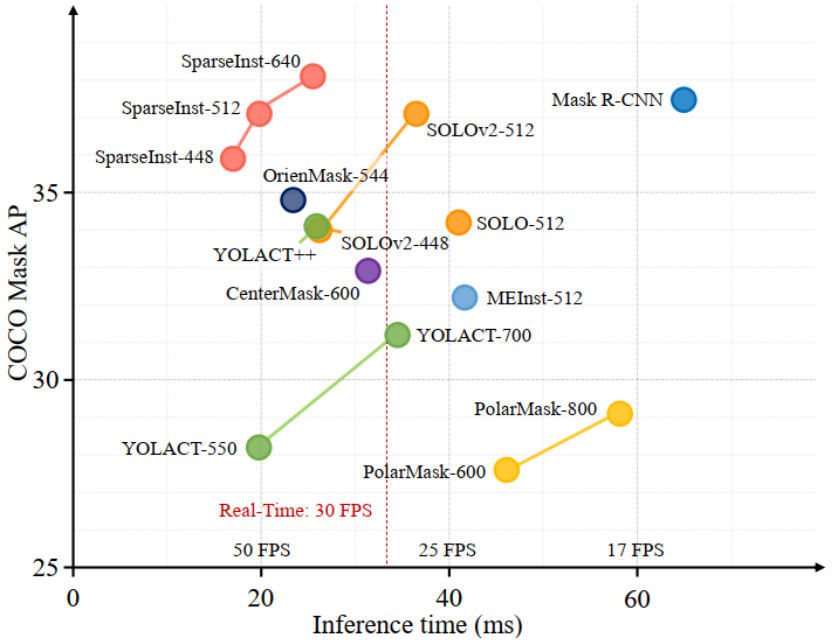
Table 5. **Different query-based detector performance** on COCO `minival` set with **longer training scheme** and single scale testing.

Sparse Instance Activation for Real-Time Instance Segmentation

Tianheng Cheng [1,2] Xinggang Wang [1†] Shaoyu Chen [1,2] Wenqiang Zhang [1]
Qian Zhang [2] Chang Huang [2] Zhaoxiang Zhang [3] Wenyu Liu [1]
[1] School of EIC, Huazhong University of Science & Technology [2] Horizon Robotics
[3] Institute of Automation, Chinese Academy of Sciences (CASIA)

Motivation: Previous instance segmentation methods heavily rely on object detection and perform mask prediction based on **bounding boxes or dense centers**, which is quite **time-consuming**. Moreover, the post-processing operation such as NMS takes much time.

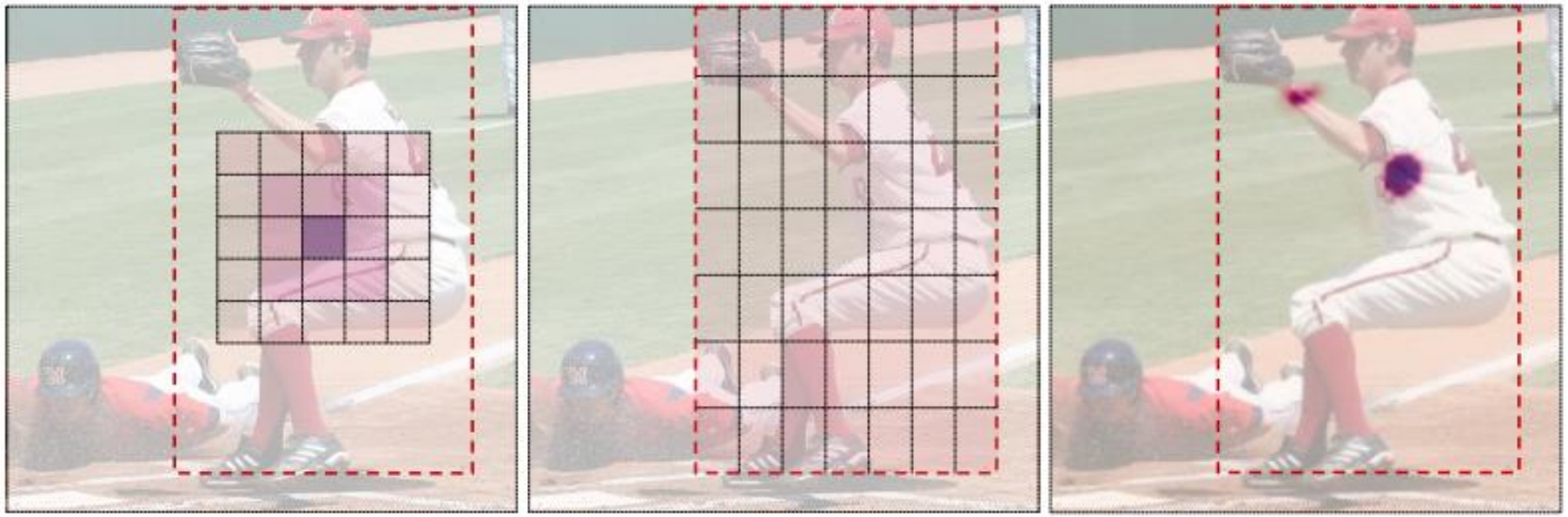
This paper aims to design a new segmentation paradigm for **real-time** instance segmentation



Sparse Instance Activation for Real-Time Instance Segmentation

Method:

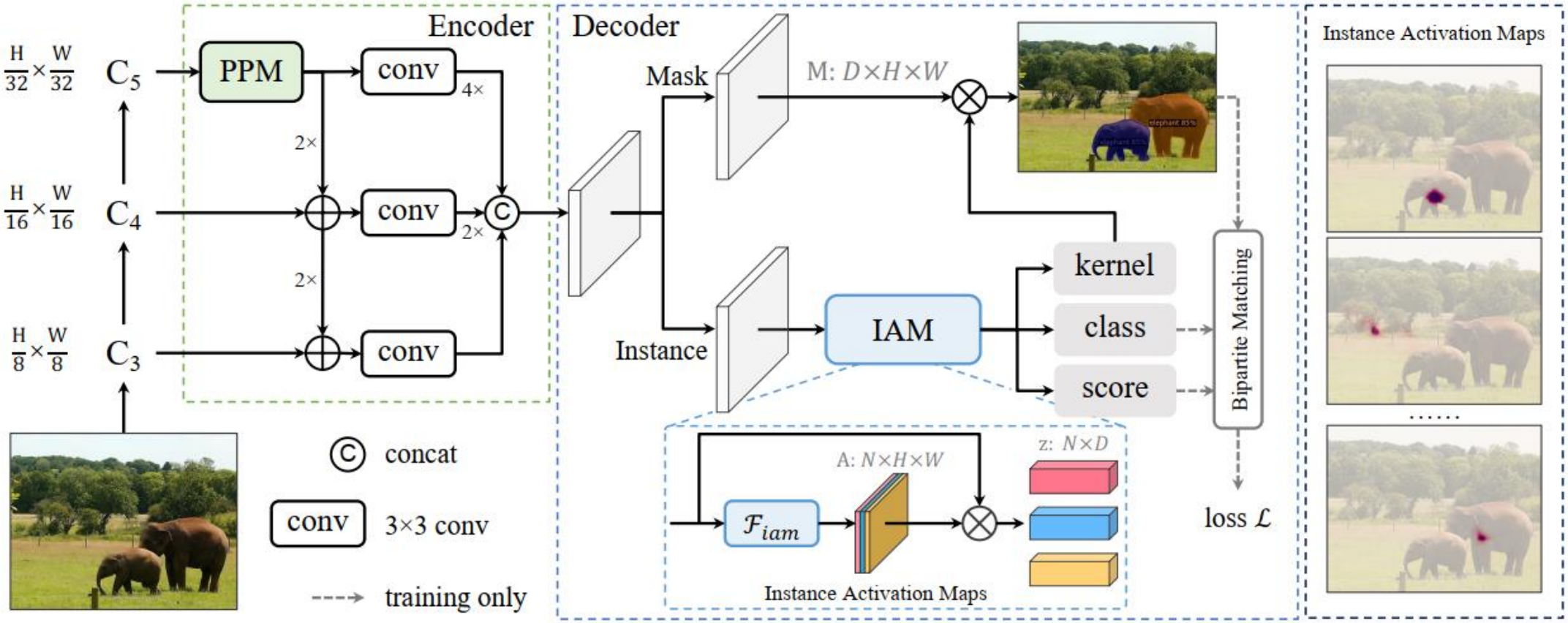
The paper exploits a sparse set of **instance activation maps (IAM)** to highlight informative object regions, which is motivated by CAM widely used in weakly-supervised object localization.



(a) center-based (b) region-based (c) instance activation map

Sparse Instance Activation for Real-Time Instance Segmentation

Method:



Sparse Instance Activation for Real-Time Instance Segmentation

Method:

Advantages of using IAM:

(1) it highlights **discriminative instance pixels**, suppresses **obstructive pixels**, and conceptually avoids the incorrect instance feature localization problems in center-/region-based methods;

(2) it aggregates instance features from the **whole image** and offers **more contexts**;

(3) computing instance features with activation maps is rather **simple without extra operation** like RoI-Align.

Additional methods:

(1) Location-Sensitive Features: similar to CoordConv;

(2) IOU-aware Objectness.

Sparse Instance Activation for Real-Time Instance Segmentation

Experiments

Comprision with SOTA on COCO

method	backbone	size	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MEInst [46]	R-50-FPN	512	24.0	32.2	53.9	33.0	13.9	34.4	48.7
CenterMask [20]	R-50-FPN	600	31.9	32.9	-	-	12.9	34.7	48.7
CondInst [36]	R-50-FPN	800	20.4†	35.4	56.4	37.6	18.4	37.9	46.9
SOLO [40]	R-50-FPN	512	24.4	34.2	55.9	36.0	-	-	-
SOLOv2-Lite [40]	R-50-FPN	448	38.2	34.0	54.0	36.1	10.3	36.3	54.4
SOLOv2-Lite [40]	R-50-DCN-FPN	512	28.2	37.1	57.7	39.7	12.9	**40.0**	**57.4**
PolarMask [43]	R-50-FPN	600	21.7†	27.6	47.5	28.3	9.8	30.1	43.1
PolarMask [43]	R-50-FPN	800	17.2†	29.1	49.5	29.7	12.6	31.8	42.3
YOLACT [1]	R-50-FPN	550	50.6	28.2	46.6	29.2	9.2	29.3	44.8
YOLACT [1]	R-101-FPN	700	29.0	31.2	50.6	32.8	12.1	33.3	47.1
YOLACT++ [1]	R-50-DCN-FPN	550	38.6	34.1	53.3	36.2	11.7	36.1	53.6
OrienMask [11]	D-53-FPN	544	42.7	34.8	56.7	36.4	**16.0**	38.2	47.8
SparseInst	R-50	608	44.6	34.7	55.3	36.6	14.3	36.2	50.7
SparseInst	R-50-DCN	608	41.6	36.8	57.6	38.9	15.0	38.2	55.2
SparseInst	R-50-d	608	42.8	36.1	57.0	38.2	15.0	37.7	53.1
SparseInst	R-50-d-DCN	608	40.0	**37.9**	**59.2**	**40.2**	15.7	39.4	56.9

Sparse Instance Activation for Real-Time Instance Segmentation

Experiments

Ablation study

Different network of F_{iam}

$\mathcal{F}_{iam}$	act.	AP	AP_{50}	AP_{75}	t (ms)
3×3 conv	sigmoid	32.0	51.9	33.5	22.9
3×3 conv	softmax	31.6	51.4	32.9	22.9
1×1 conv	sigmoid	30.8	50.7	32.0	**22.4**
3×3 conv, ReLU, 3×3 conv	sigmoid	31.9	52.2	33.0	23.6
Group 3×3 conv (2 groups)	sigmoid	32.2	52.3	33.5	23.1
Group 3×3 conv (4 groups)	sigmoid	**32.7**	**53.1**	**34.0**	23.3

$\mathcal{F}_{iam}$	AP	AP_{50}	AP_{75}	t (ms)
1×1 conv	30.8	50.7	32.0	**22.4**
3×3 conv	32.0	51.9	33.5	22.9
Group 3×3 conv	**32.7**	**53.1**	**34.0**	23.3
Cross Attention	31.8	51.7	33.1	23.4

Sparse Instance Activation for Real-Time Instance Segmentation

Experiments

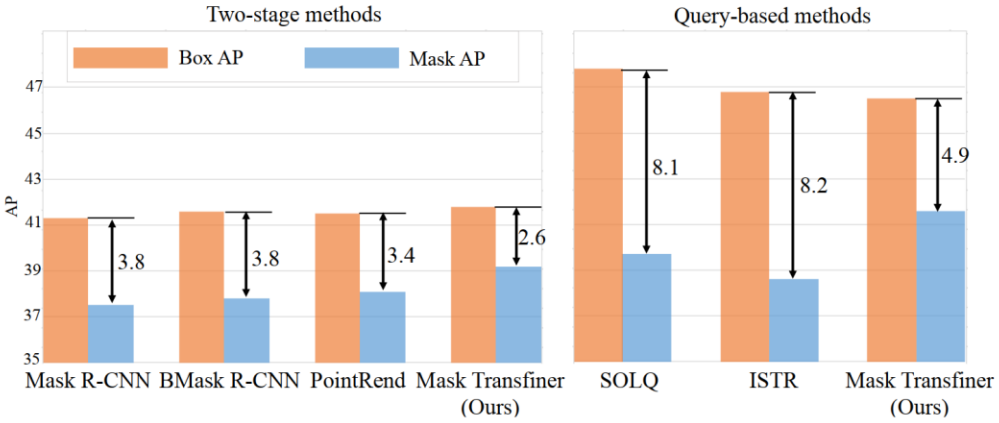
Visualization of IAM

Mask Transfiner for High-Quality Instance Segmentation

Lei Ke[1,2] Martin Danelljan[1] Xia Li[1] Yu-Wing Tai[3] Chi-Keung Tang[2] Fisher Yu[1]
[1]ETH Zurich ¨ [2]HKUST [3]Kuaishou Technology

Motivation: Two-stage and query-based instance segmentation methods have achieved remarkable results. However, their segmented masks are still very **coarse**. **Large gap between bbox AP and Mask AP**.

Accurate mask prediction is highly challenging, due to the need for **high-resolution** deep features, which demands large computational and memory costs.

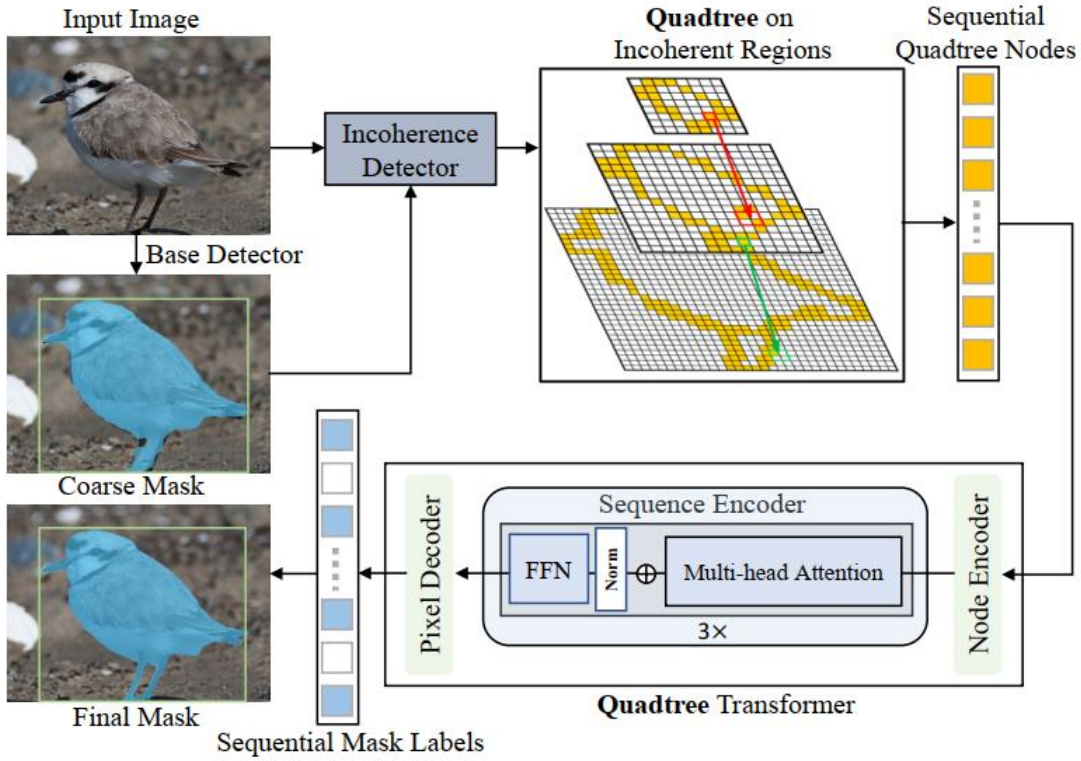


Mask Transfiner for High-Quality Instance Segmentation

Method:

1. Identify **incoherent regions (error-prone regions)**, defined by the loss of information when downsampling mask. This paper builds a hierarchical **quadtree** to represent and process the incoherent image pixels at multiple scales.

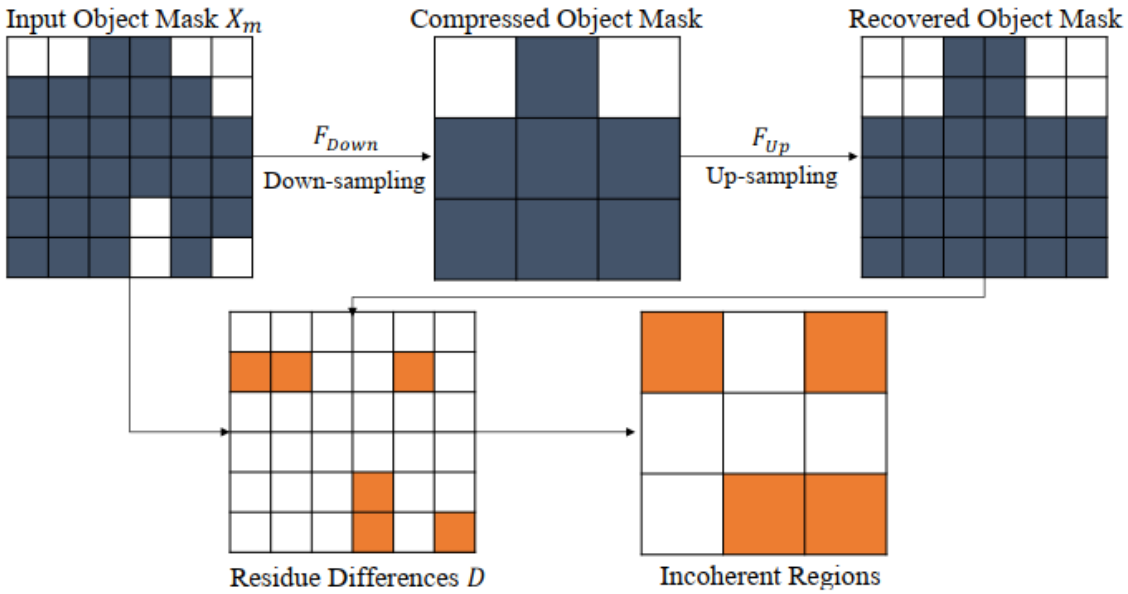
2. To refine the mask labels of the incoherent quadtree nodes, they design a **refinement network based on the transformer** instead of convolutional networks because they require operating on uniform grids.



Mask Transfiner for High-Quality Instance Segmentation

Method:

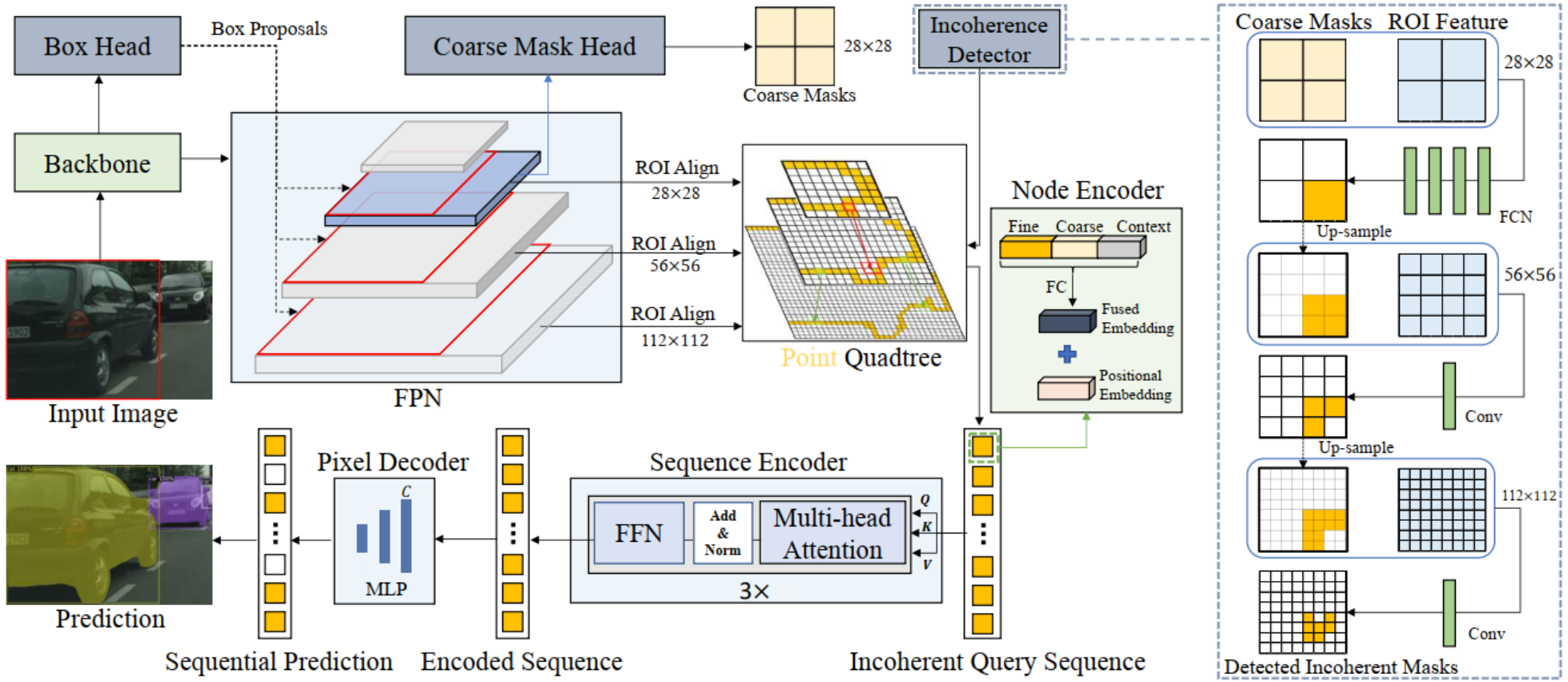
Properties of Incoherent Regions: occupying 43% of all wrongly predicted pixels, while only taking 14% to the corresponding bounding box areas.



Percent	$\text{Recall}_{\text{Err}}$	Acc	AP_{GT}	$\text{AP}_{\text{Coarse}}$
14%	43%	56%	51.0	35.5

Mask Transfiner for High-Quality Instance Segmentation

Method:



Mask Transfiner for High-Quality Instance Segmentation

Experiments

Ablation study

Table 2. Effect of the incoherent regions on COCO *val* set. AP^B is evaluated Boundary IoU [10] while $AP^\star$ uses LVIS annotations.

Region Type	AP	AP^B	$AP^\star$	$AP^\star_{50}$
Full RoIs (28×28)	35.5	21.4	38.3	59.5
Boundary regions	36.6	23.8	40.1	60.2
Incoherent regions	**37.3**	**24.2**	**40.5**	**60.7**
Incoherent regions (**w/o** L_1)	36.5	23.5	39.8	59.7
Incoherent regions (**w/o** L_2)	36.8	23.8	40.2	60.1
Incoherent regions (**w/o** L_3)	36.7	23.6	40.0	59.9

Table 5. Analysis of the quadtree depth on the COCO *val* using R50-FPN as backbone.

Depth	Output size	AP	$AP^\star$	AP_L	AP_M	AP_S	FPS
0	28×28	35.2	37.6	50.3	37.7	17.2	12.3
1	28×28	35.5	38.4	50.9	38.1	17.2	10.6
2	56×56	36.2	39.1	51.9	38.7	17.3	8.9
3	112×112	**37.3**	40.5	52.9	**39.5**	**17.5**	7.1
4	224×224	37.1	**40.7**	**53.1**	39.3	17.4	5.2

Mask Transfiner for High-Quality Instance Segmentation

Experiments

Ablation study

Table 6. Mask Transfiner vs. MLP and CNN on COCO *val* set using ResNet-50-FPN.

Model	AP	AP^B	$AP^\star$	$AP^\star_{50}$
CNN (full regions, 56 × 56)	35.7	21.8	38.7	58.8
MLP (full regions, 56 × 56)	36.1	23.4	39.2	59.2
MLP (PointRend [28], 112 × 112)	36.2	23.1	39.1	59.0
MLP (incoherent regions)	36.4	23.7	39.7	59.8
Mask Transfiner (D = 3, H = 4)	37.3	24.2	40.5	60.7
Mask Transfiner (D = 3, H = 8)	37.1	24.1	40.2	60.8
Mask Transfiner (D = 6, H = 4)	37.4	24.4	40.6	60.9

Table 7. Efficacy of Transfiner compared to standard attention models on COCO *val*. NLA denotes non-local attention [39].

Model	AP	FLOPs (G)	Memory (M)	FPS
NLA [39] (112×112)	36.3	24.6	8347	4.6
NLA [39] (224×224)	36.6	80.2	18091	2.4
Transformer [4] (28×28)	36.1	37.2	4368	6.9
Transformer [4] (56×56)	36.5	68.3	17359	2.1
Mask Transfiner (112×112)	**37.3**	**16.8**	**2316**	**7.1**
Mask Transfiner (224×224)	37.1	38.1	4871	5.2

Mask Transfiner for High-Quality Instance Segmentation

Experiments

Compare with SOTA on COCO

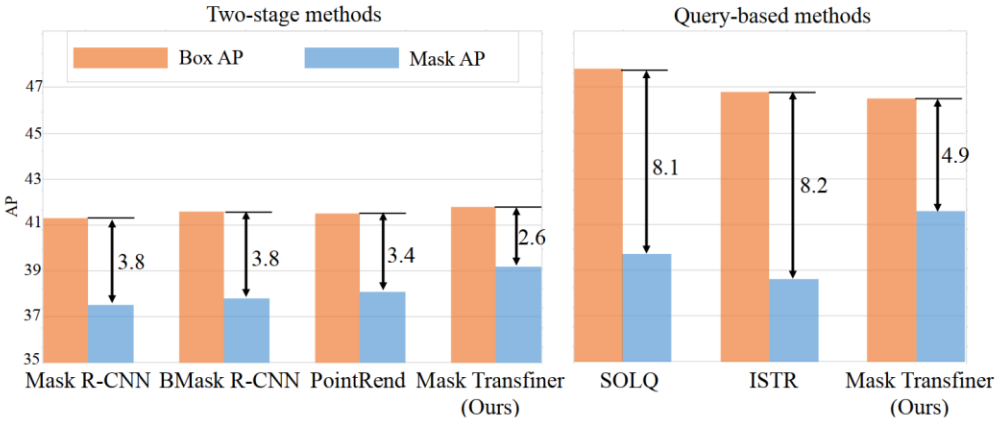
Method	Backbone	Type	AP	$AP^{\star}_{val}$	AP^{B}_{val}	AP^{Box}	AP_S	AP_M	AP_L
Mask R-CNN [21]	R50-FPN	T	37.5	38.2	21.2	41.3	21.1	39.6	48.3
PointRend [28]	R50-FPN	T	38.1	39.7	23.5	41.5	18.8	40.2	49.4
B-MRCNN [12]	R50-FPN	T	37.8	39.8	23.5	41.6	19.7	40.3	49.6
BPR [36]	R50-FPN	T	38.4	40.2	24.3	41.3	20.2	40.5	49.7
Mask Transfiner	R50-FPN	T	39.4	42.3	26.0	41.8	22.3	41.2	50.2
Mask Transfiner†	R50-FPN	T	**40.5**	**43.1**	**26.8**	**43.2**	**22.8**	**42.3**	**52.5**
Mask R-CNN [21]	R101-FPN	T	38.8	39.3	23.1	43.1	21.8	41.4	50.5
PointRend [28]	R101-FPN	T	39.6	41.4	25.3	43.3	19.8	42.6	53.7
MS R-CNN† [24]	R101-FPN	T	39.6	41.1	25.0	44.1	18.9	42.7	55.1
HTC [6]	R101-FPN	T	39.7	42.5	25.4	**45.9**	21.0	42.2	53.5
RefineMask [47]	R101-FPN	T	39.4	42.3	26.8	43.8	21.6	42.0	53.1
BCNet [26]	R101-FPN	T	39.8	41.9	26.1	43.5	22.7	42.4	51.1
Mask Transfiner	R101-FPN	T	40.7	43.6	27.3	43.9	23.1	42.8	53.8
Mask Transfiner†	R101-FPN	T	**42.2**	**45.0**	**28.6**	45.8	**24.1**	**44.8**	**55.4**
ISTR [23]	R50-FPN	Q	38.6	39.5	23.0	46.8	22.1	40.4	50.6
QueryInst [17]	R50-FPN	Q	39.9	42.1	25.1	44.5	22.9	41.7	51.9
SOLQ [15]	R50-FPN	Q	39.7	39.8	23.3	**47.8**	21.5	42.5	53.1
Mask Transfiner	R50-FPN	Q	**41.6**	**45.4**	**28.2**	46.5	**24.2**	**44.6**	**55.2**

Mask Transfiner for High-Quality Instance Segmentation

Lei Ke[1,2] Martin Danelljan[1] Xia Li[1] Yu-Wing Tai[3] Chi-Keung Tang[2] Fisher Yu[1]
[1]ETH Zurich ¨ [2]HKUST [3]Kuaishou Technology

Motivation: Two-stage and query-based instance segmentation methods have achieved remarkable results. However, their segmented masks are still very **coarse**. **Large gap between bbox AP and Mask AP**.

Accurate mask prediction is highly challenging, due to the need for **high-resolution** deep features, which demands large computational and memory costs.



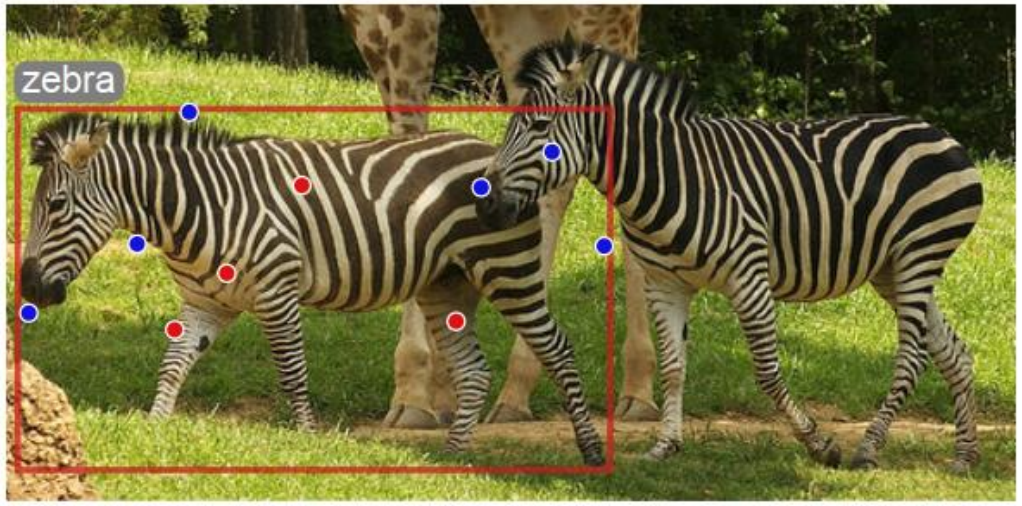
Pointly-Supervised Instance Segmentation (oral)

Bowen Chen[1] Omkar Parkhi[2] Alexander Kirillov[2]

[1]UIUC [2]Facebook AI

Motivation: Manual annotation of object masks for training is very **complex and time-consuming**. For example, it takes on **average 79.2 seconds** to create an object mask in COCO, whereas a bounding box can be annotated **~11 times faster in only 7 seconds**.

Is **object mask** training data **necessary** to get closer to the fully supervised performance? And is there an easier to collect **annotation form** for the instance segmentation task?



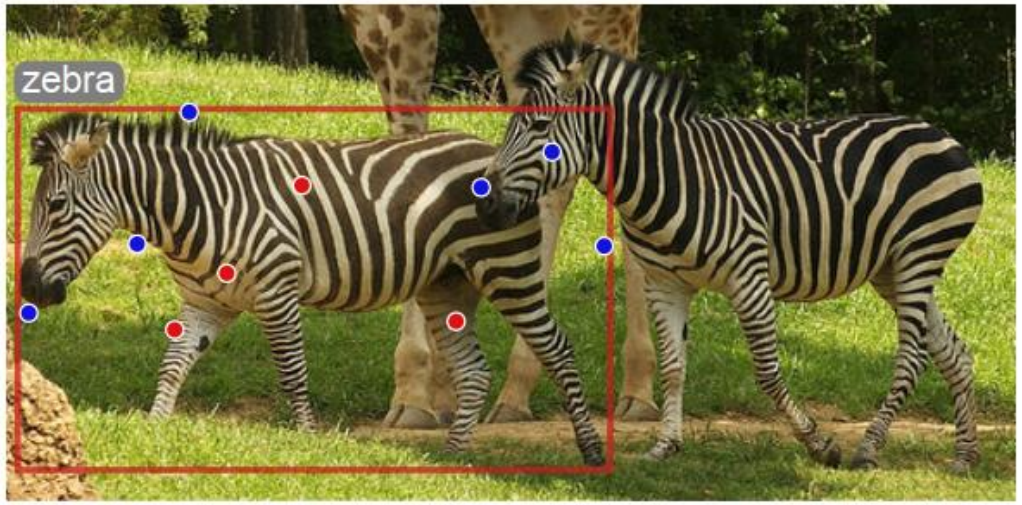
Pointly-Supervised Instance Segmentation (oral)

Bowen Chen[1] Omkar Parkhi[2] Alexander Kirillov[2]

[1]UIUC [2]Facebook AI

Method: randomly sample 10 points inside bbox.

Performance: Mask R-CNN trained on COCO, PASCAL VOC, Cityscapes, and LVIS achieves **94%–98%** of its fully-supervised performance.



FreeSOLO: Learning to Segment Objects without Annotations

Xinlong Wang[1], Zhiding Yu[2], Shalini De Mello[2], Jan Kautz[2], Anima Anandkumar[2,3], Chunhua Shen[4],Jose M. Alvarez[2]

[1]The unversity of Adelaide [2]NVIDIA [3]Caltech [4]Zhejiang University

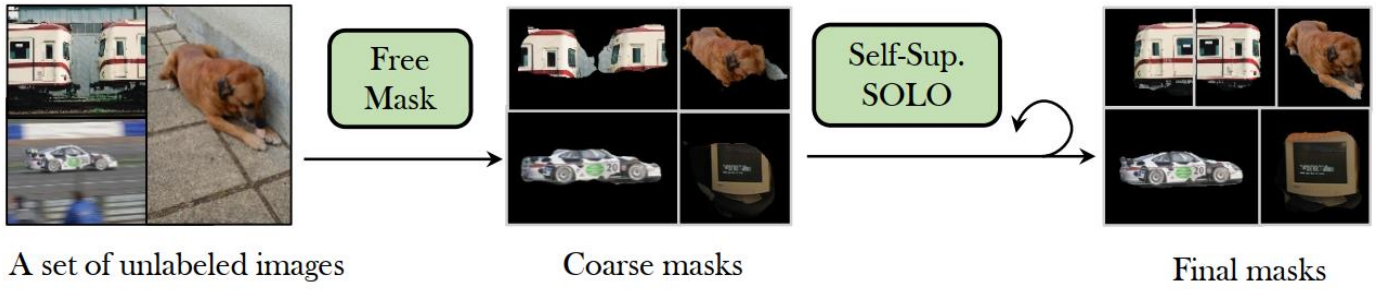
Motivation: Instance segmentation requires **costly annotations** such as bounding boxes and segmentation masks for learning.

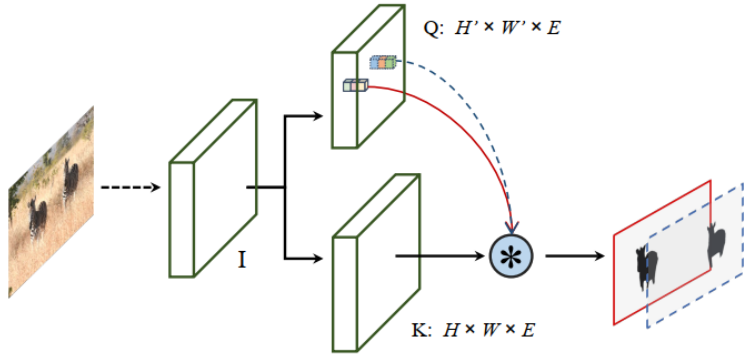
This paper presents FreeSOLO to explore learning class-agnostic instance segmentation without any annotations.

FreeSOLO: Learning to Segment Objects without Annotations

Method:

1.propose the **Free Mask** approach, which leverages the specific design of SOLO to effectively extract coarse object masks and semantic embeddings in an unsupervised manner.
2. further propose Self-Supervised SOLO, which takes the **coarse masks and semantic embeddings from Free Mask** and trains the SOLO model, with several novel design elements to overcome label noise in the coarse masks.





FreeSOLO: Learning to Segment Objects without Annotations

Performance:

FreeSOLO achieves **9.8% AP50** on the challenging COCO dataset, which even outperforms several segmentation proposal methods that use manual annotations.

method	AP_{50}	AP_{75}	AP	AR_1	AR_{10}	AR_{100}
w/ anns:						
MCG [61]	4.6	0.8	1.6	1.9	7.4	18.2
COB [62]	8.8	1.9	3.3	2.9	10.1	22.7
w/o anns:						
FreeSOLO	9.8	2.9	4.0	4.1	10.5	12.7

Table 1. Class-agnostic instance segmentation on MS COCO `val2017`. Both MCG and COB require annotations more or less.