BoxInst: High-Performance Instance Segmentation with Box Annotations

Zhi Tian et al (CVPR 2021)

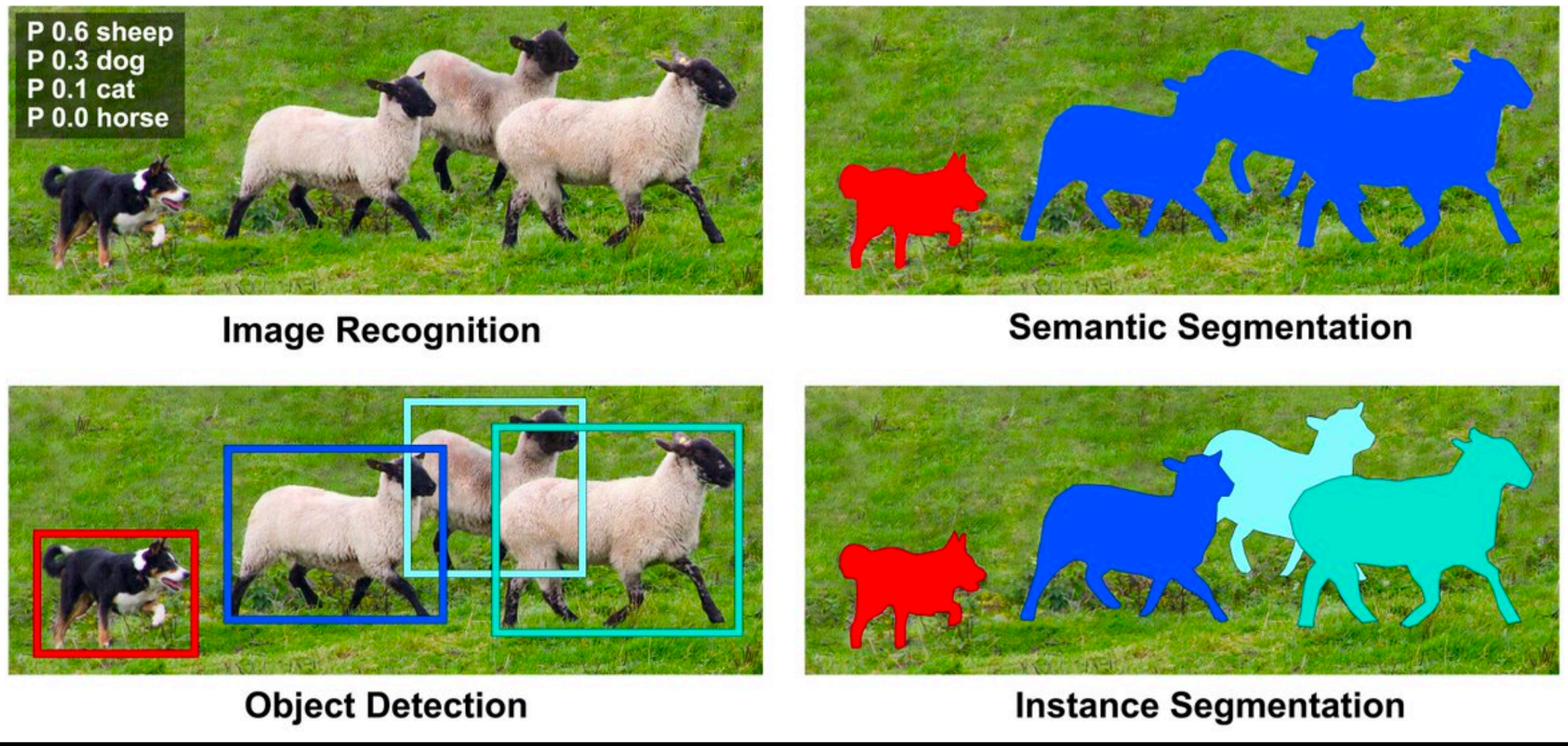
Choi Dongmin

Abstract

- **A high-performance mask-level instance segmentation**
  - with only bounding-box annotations for training
  - dramatic improvement; AP 21.1% → 31.6 on COCO
  - redesign the loss of learning masks
    - can supervise the mask training without relying on mask annotations
    - narrows the performance gap between weakly and fully supervision

Introduction

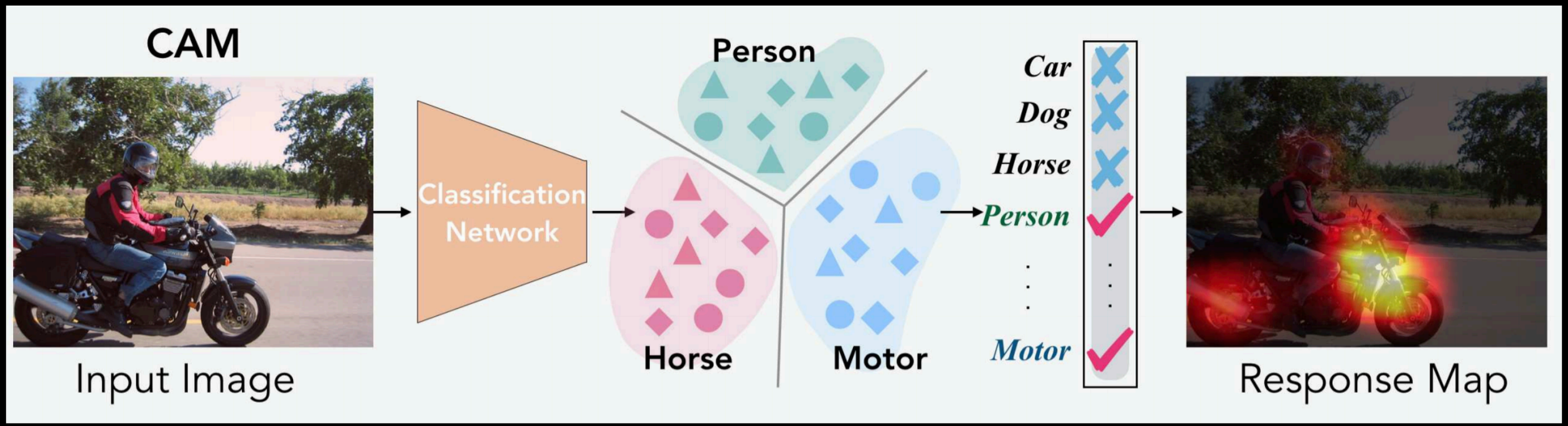
- **Instance Segmentation**



https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works

Introduction

- **Weakly-Supervised Instance Segmentation (WSIS)**
  - Image-level annotation (e.g., CAM)
  - Bounding-box annotation



Chang et al. Weakly-Supervised Semantic Segmentation via Sub-category Exploration. CVPR 2020

Introduction

- **BBTB: Previous S.O.T.A of WSIS**

  - still low performance (21.1 AP on COCO benchmark)

Hsu et al. Weakly Supervised Instance Segmentation using the Bounding Box Tightness Prior. NIPS 2019

Introduction

- **BoxInst**
  - only with bounding-box annotation
  - COCO AP: 21.1 (previous S.O.T.A BBTA) → **31.6**
  - with aug. and 3x scheduling, achieved **32.1** AP

NOTE: Mask R-CNN achieved 37.5 AP

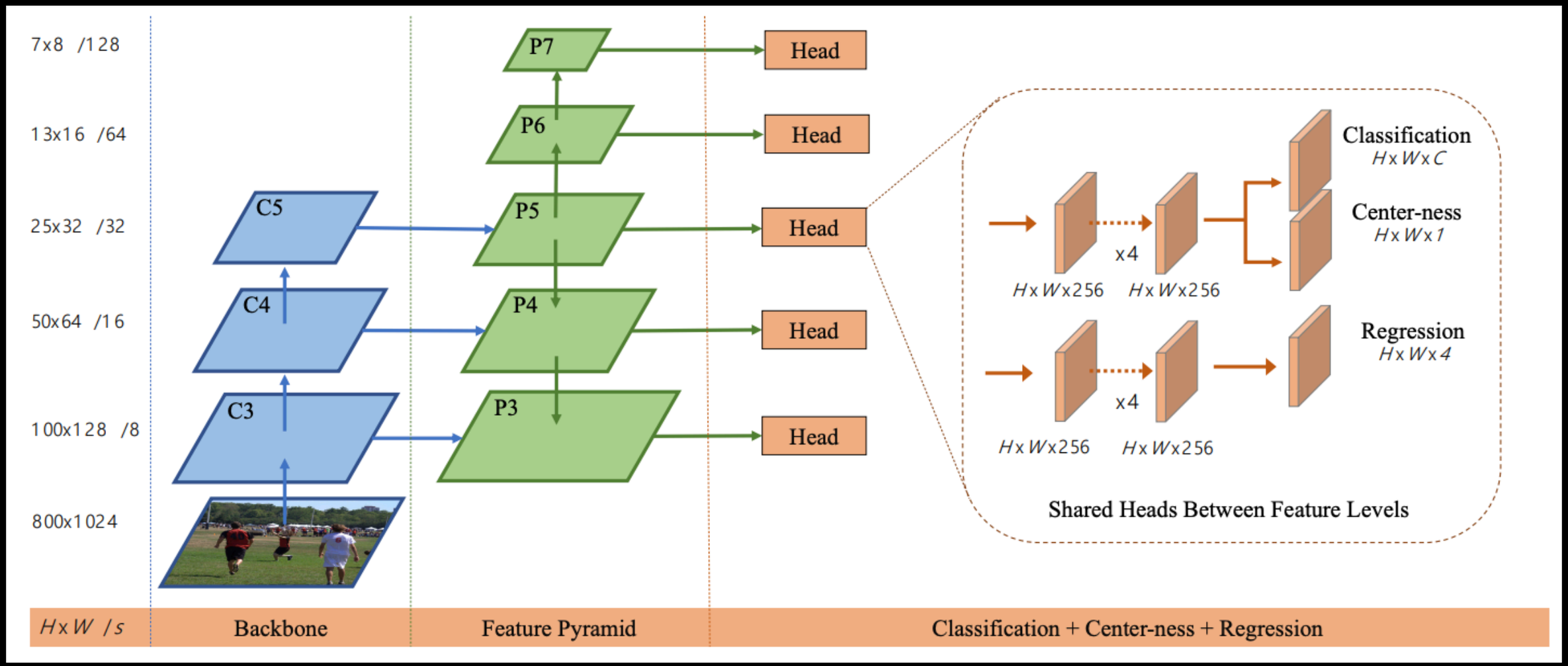
Introduction

- **BoxInst**
  - only with bounding-box annotation
  - COCO AP: 21.1 (previous S.O.T.A BBTA) → **31.6**
  - with aug. and 3x scheduling, achieved **32.1** AP

NOTE: Mask R-CNN achieved 37.5 AP

Related Work

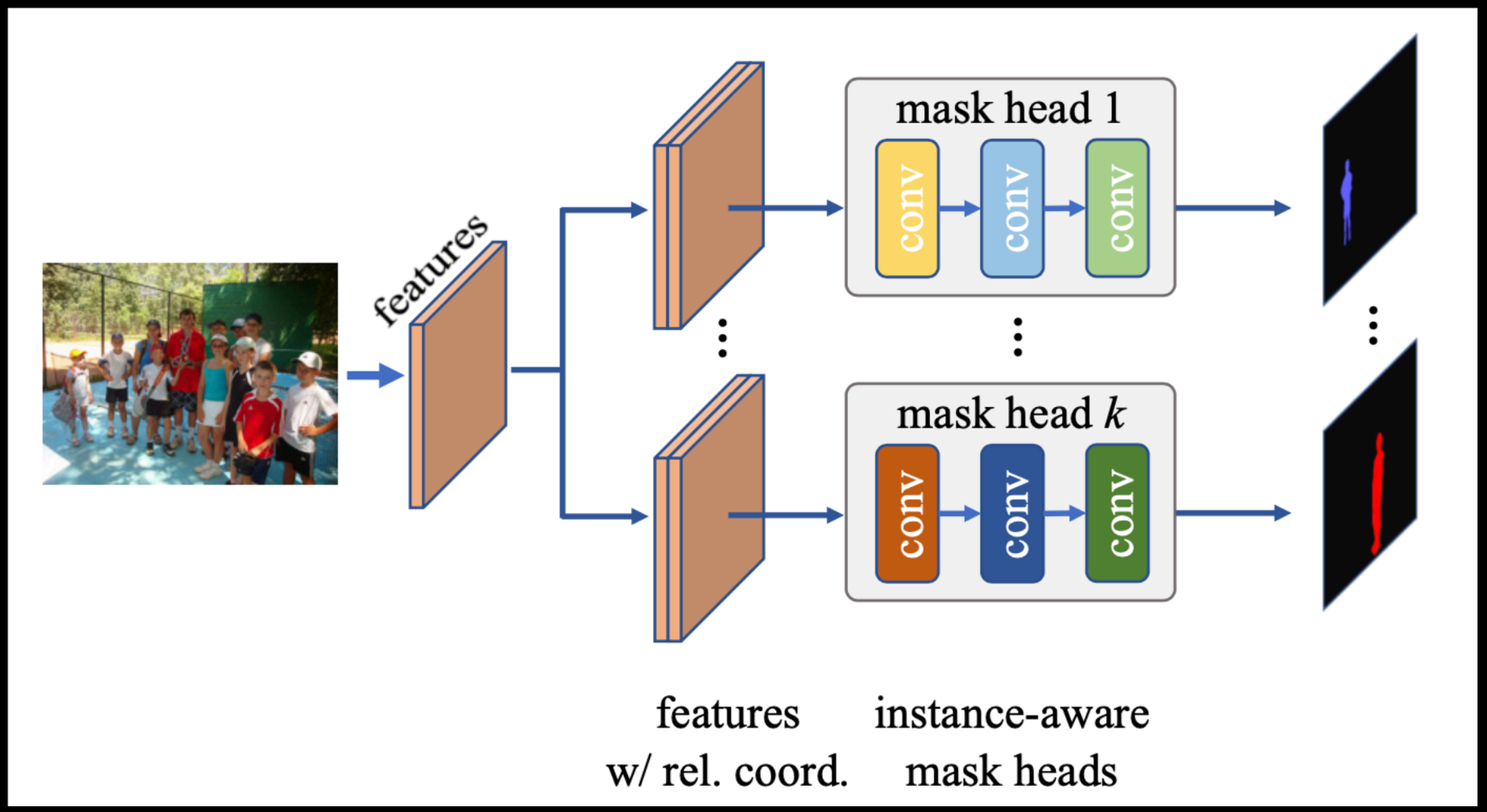
- **FCOS**
  - anchor-free one-stage object detector
  - FPN + Detection head



Tian et al. FCOS: Fully Convolutional One-Stage Object Detection. ICCV 2019

Related Work

- **CondInst**
  - solve instance segmentation in an RoI-free fully conv. way
  - employ dynamic filters instead of a fixed mask head
  - FCOS + Mask head

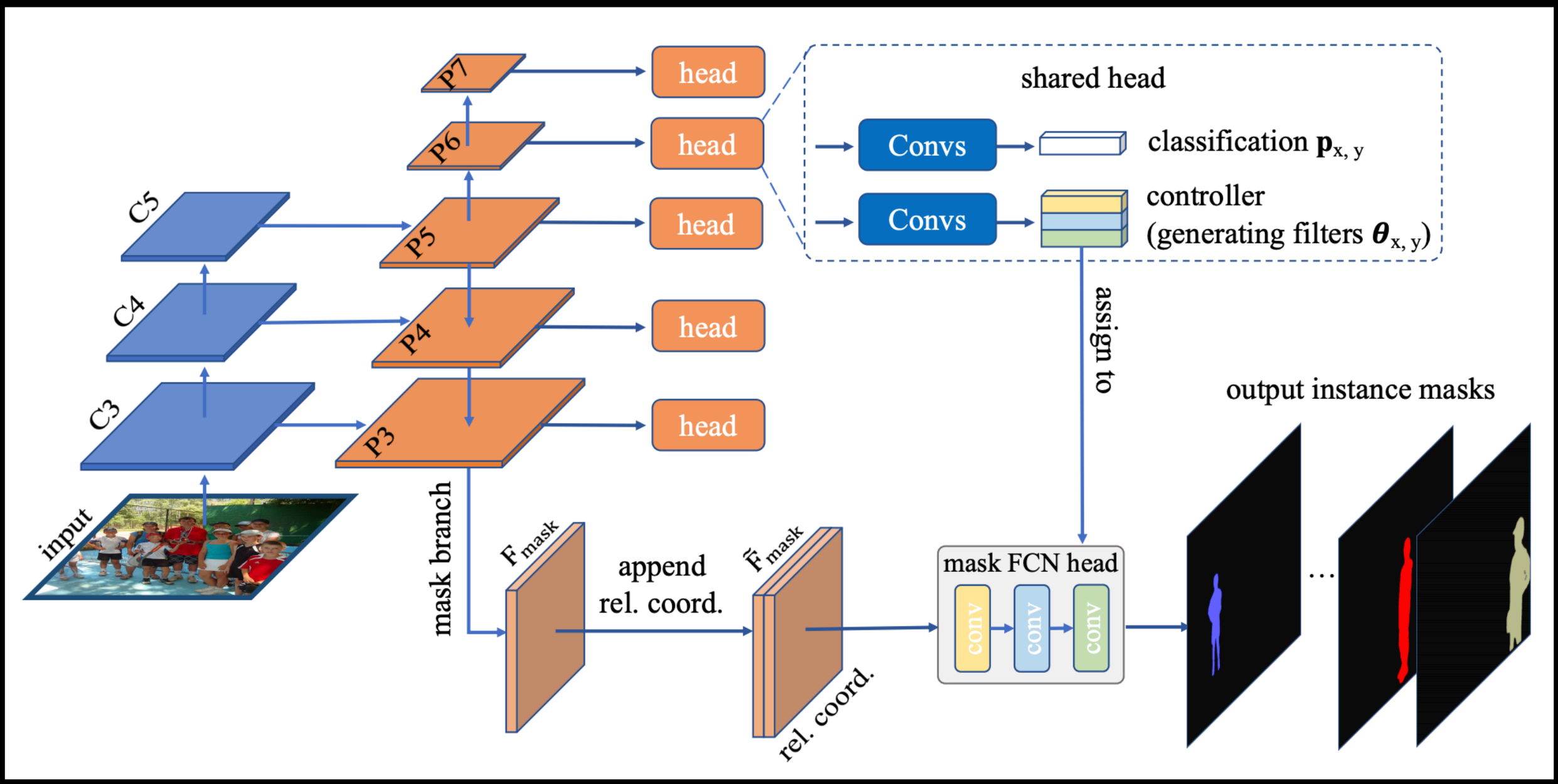


Tian et al. Conditional Convolutions for Instance Segmentation. ECCV 2020

Related Work

- **CondInst**

  - FCOS + Mask head

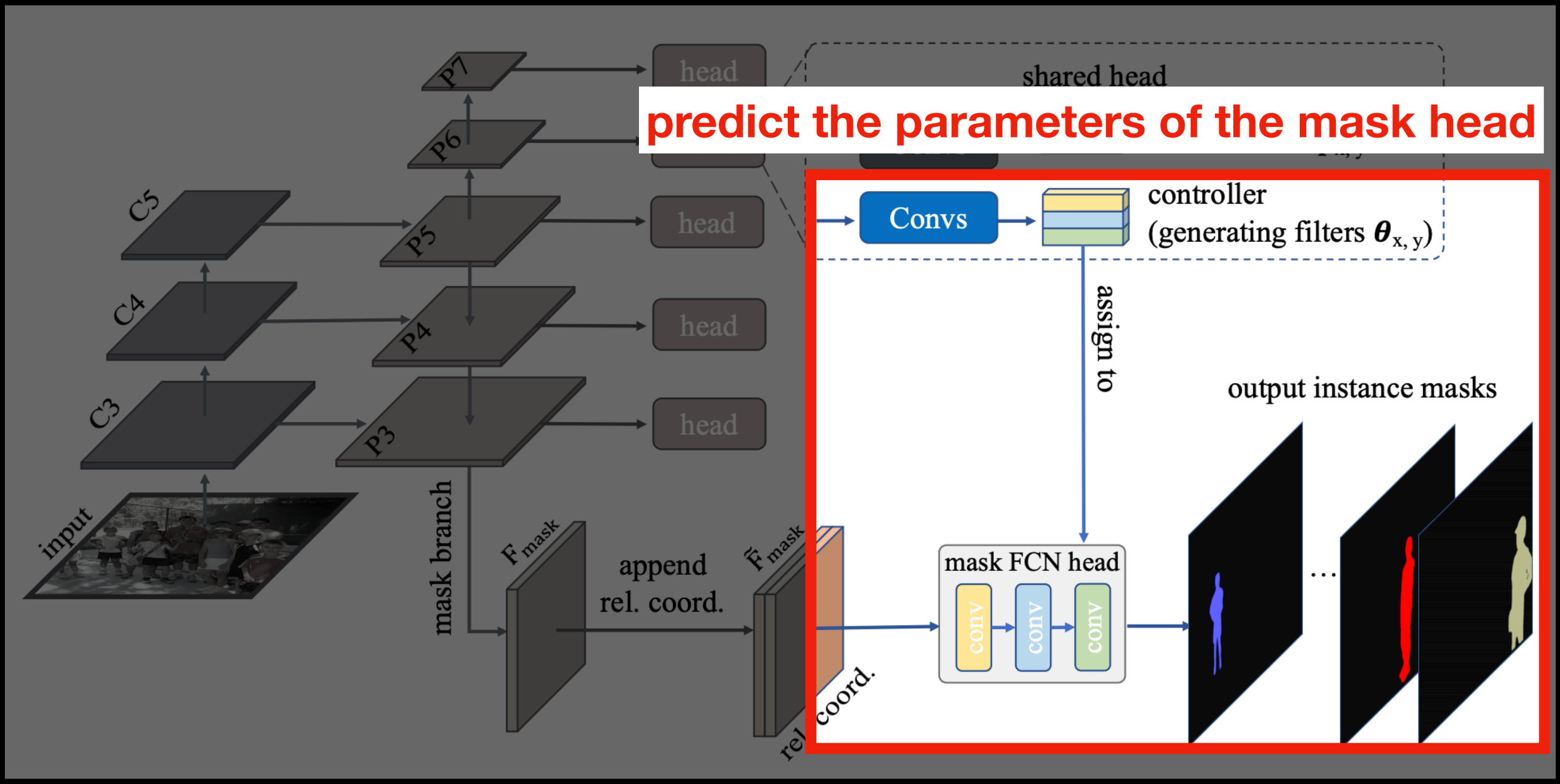


Tian et al. Conditional Convolutions for Instance Segmentation. ECCV 2020

Related Work

- **CondInst**

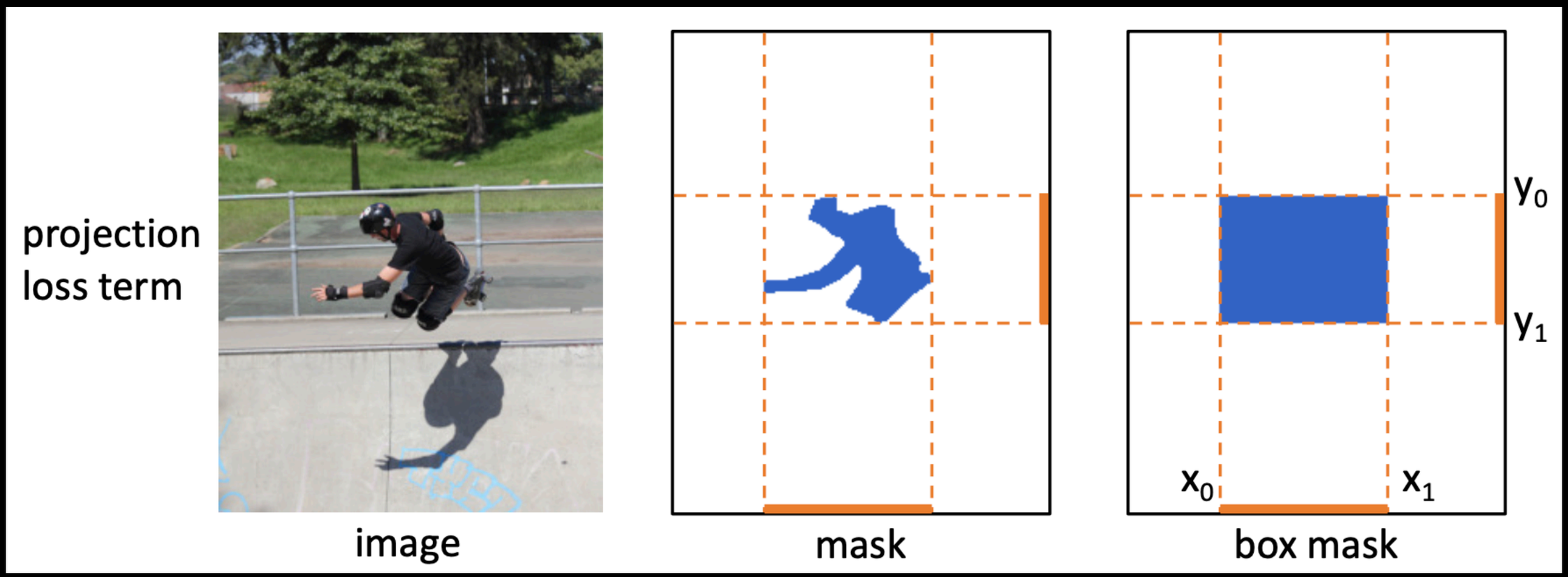
  - FCOS + Mask head



Tian et al. Conditional Convolutions for Instance Segmentation. ECCV 2020

Approach

- **Two proposed loss term: 1. Projection loss term**
  - projections of the mask and box should be the same



$$L_{proj} = L(\text{Proj}_x(\tilde{m}), \text{Proj}_x(b)) + L(\text{Proj}_y(\tilde{m}), \text{Proj}_y(b))$$

$L(\cdot, \cdot)$: Dice Loss

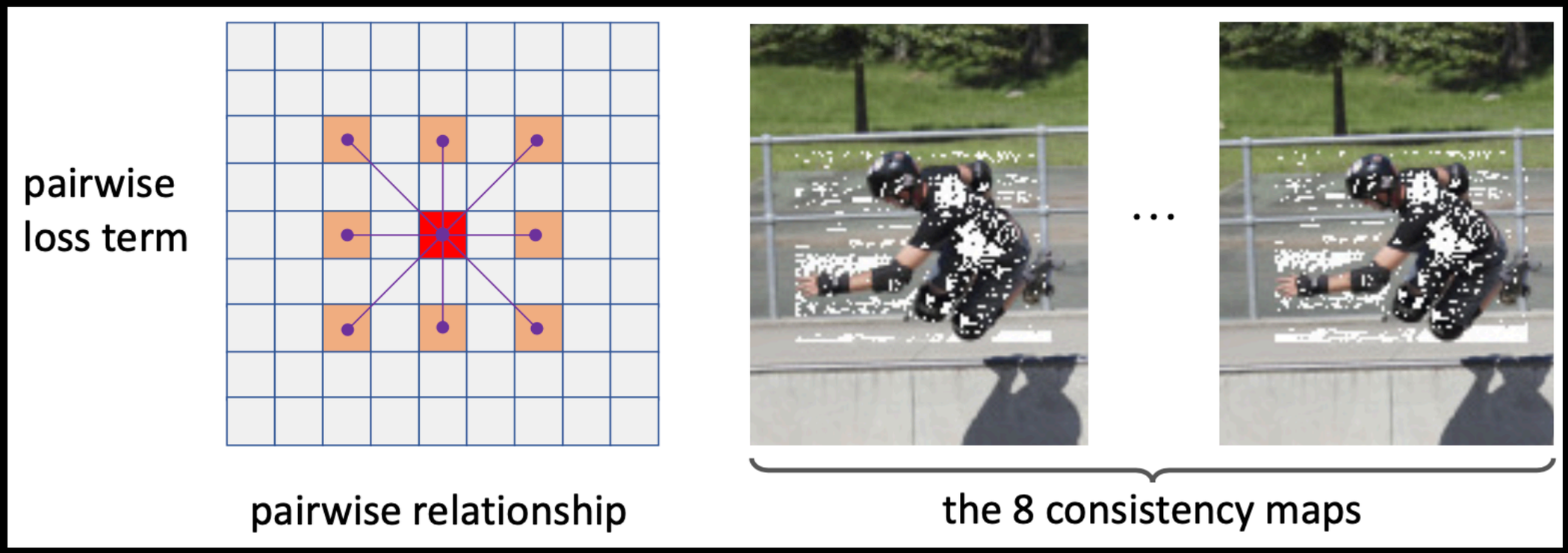
$\tilde{m} \in (0,1)^{H \times W}$: the network predictions for the instance mask

b : ground-truth bounding box

Approach

- **Two proposed loss term: 2. Pairwise affinity loss term**

  - supervise the mask in a *pairwise* way



an undirected graph $G = (V, E)$; V: the set of pixels, E: the set of edges

the label for the edge $y_e \in \{0,1\}$

$y_e = 1$ if the two pixels have the same G.T. label or $y_e = 0$ if not

$$P(y_e = 1) = \tilde{m}_{i,j} \cdot \tilde{m}_{k,l} + (1 - \tilde{m}_{i,j}) \cdot (1 - \tilde{m}_{k,l})$$

Approach

- **Two proposed loss term: 2. Pairwise affinity loss term**

  - supervise the mask in a *pairwise* way

an undirected graph $G = (V, E)$; V: the set of pixels, E: the set of edges

the label for the edge $y_e \in \{0,1\}$
$y_e = 1$ if the two pixels have the same G.T. label or $y_e = 0$ if not

$$P(y_e = 1) = \tilde{m}_{i,j} \cdot \tilde{m}_{k,l} + (1 - \tilde{m}_{i,j}) \cdot (1 - \tilde{m}_{k,l})$$

Approach

- **Two proposed loss term: 2. Pairwise affinity loss term**

  - supervise the mask in a *pairwise* way

$$P(y_e = 1) = \tilde{m}_{i,j} \cdot \tilde{m}_{k,l} + (1 - \tilde{m}_{i,j}) \cdot (1 - \tilde{m}_{k,l})$$
$$P(y_e = 0) = 1 - P(y_e = 1)$$

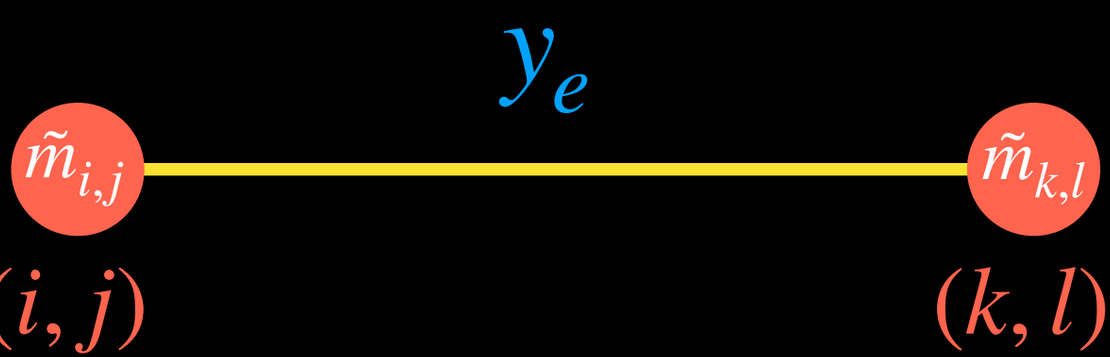
$$L_{pairwise} = -\frac{1}{N} \sum_{e \in E_{in}} y_e \log P(y_e = 1) + (1 - y_e) \log P(y_e = 0)$$

Approach

- **Two proposed loss term: 2. Pairwise affinity loss term**

  - supervise the mask in a *pairwise* way

However, we only have bounding box annotations…



$$P(y_e = 1) = \tilde{m}_{i,j} \cdot \tilde{m}_{k,l} + (1 - \tilde{m}_{i,j}) \cdot (1 - \tilde{m}_{k,l})$$
$$P(y_e = 0) = 1 - P(y_e = 1)$$

$$L_{pairwise} = -\frac{1}{N} \sum_{e \in E_{in}} y_e \log P(y_e = 1) + (1 - y_e) \log P(y_e = 0)$$

Approach

- **Two proposed loss term: 2. Pairwise affinity loss term**

  - Learning without mask annotations: color similarity

  - idea: if two pixels have similar colors, they are likely to have the same labels as well

$$S_e = S(c_{i,j}, c_{l,k}) = \exp(-\frac{||c_{i,j} - c_{j,k}||}{\theta})$$

assign $y_e = 1$ if $S_e > \tau$ (a color similarity threshold)

S_e : color similarity of the edge e

$c_{i,j}$: the color vector of the pixel (i, j) (LAB color space)

$\theta = 2$: hyper-parameter

$$L_{pairwise} = -\frac{1}{N} \sum_{e \in E_n} 1_{\{S_e > \tau\}} \log P(y_e = 1)$$

only for the positive edges

Approach

- **Two proposed loss term**
  - **Projection loss term + Pairwise affinity loss term**

$$L_{mask} = L_{proj} + L_{pairwise}$$

$$L_{proj} = L(\text{Proj}_x(\tilde{m}), \text{Proj}_x(b)) + L(\text{Proj}_y(\tilde{m}), \text{Proj}_y(b))$$

$$L_{pairwise} = -\frac{1}{N} \sum_{e \in E_n} 1_{\{S_e > \tau\}} \log P(y_e = 1)$$

Experiments

- **Projection and Pairwise Affinity Loss for Mask Learning**
  - the redesigned mask loss can have similar performance to the original pixel mask loss (e.g., Dice loss)

mask loss	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Dice loss	35.6	56.3	37.8	16.9	38.9	51.0
proposed	35.4	55.9	37.6	17.0	38.8	50.7

Table 2: The projection and pairwise affinity mask loss vs. the original pixelwise one in the fully-supervised settings. As we can see here, they attain very similar mask AP on the COCO split `val2017`.

Experiments

- **Box-supervised Instance Segmentation**
  - Varying the threshold of color similarity

proportion of true positive edges

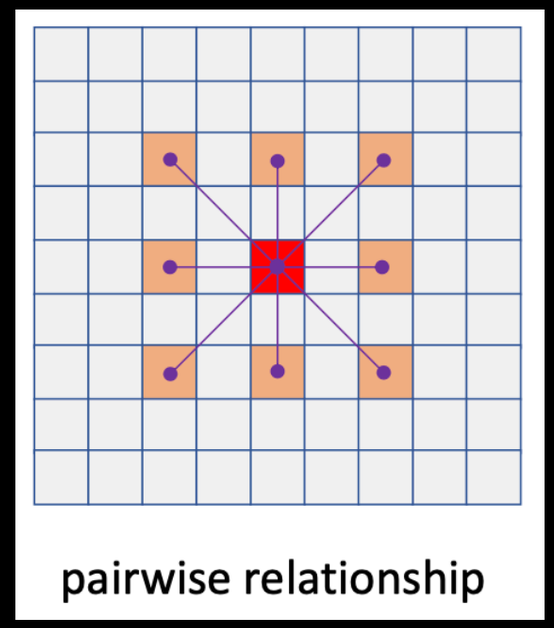
	prop.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
fully-sup.	-	35.4	55.9	37.6	17.0	38.8	50.7
$\tau = 0$	94.1%	9.4	30.3	3.3	7.6	10.3	11.4
$\tau = 0.1$	98.3%	**30.7**	52.2	**31.1**	13.8	**33.1**	**45.7**
$\tau = 0.2$	**98.4%**	30.6	**52.6**	30.9	**13.9**	32.8	45.5

only with box annotations

(a) Varying the color similarity threshold τ.

Experiments

- **Box-supervised Instance Segmentation**
  - Varying the neighborhood of the pixels



size	dilation	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
3	1	29.7	52.0	29.6	13.4	32.3	44.4
3	2	**30.7**	52.2	**31.1**	**13.8**	**33.1**	**45.7**
5	1	30.5	**52.3**	30.7	13.7	33.0	**45.7**
5	2	29.9	51.9	30.0	13.8	32.1	45.0

(b) Varying the size and dilation of the local patches (with $\tau = 0.1$).

Experiments

- **Box-supervised Instance Segmentation**

  - The contribution of each loss term

L_{proj}	$L_{pairwise}$	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
box mask		10.6	32.2	4.6	5.7	11.3	15.6
✓		21.2	45.2	17.7	10.0	21.4	32.5
✓	✓	**30.7**	**52.2**	**31.1**	**13.8**	**33.1**	**45.7**

Table 3: The mask AP on COCO `val2017` by applying the different loss terms. "box mask": using the masks generated by boxes. If both terms are not used, the model can only provide the box-level localization precision (10.6% mask AP).

Experiments

- **Box-supervised Instance Segmentation**
  - Failure cases

Figure 5: Some failure cases on COCO. The incorrect parts are in the red boxes.

Experiments

- **Comparison with State-of-the-art on COCO**

method	backbone	aug.	sched.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
fully supervised methods:									
Mask R-CNN [11]	ResNet-50-FPN	✓	3×	37.5	59.3	40.2	21.1	39.6	48.3
CondInst [28]	ResNet-50-FPN	✓	3×	37.8	59.1	40.5	21.0	40.3	48.7
Mask R-CNN	ResNet-101-FPN	✓	3×	38.8	60.9	41.9	21.8	41.4	50.5
YOLACT-700 [4]	ResNet-101-FPN	✓	4.5×	31.2	50.6	32.8	12.1	33.3	47.1
PolarMask [32]	ResNet-101-FPN	✓	2×	32.1	53.7	33.1	14.7	33.8	45.3
CondInst	ResNet-101-FPN	✓	3×	39.1	60.9	42.0	21.5	41.7	50.9
box-supervised methods:									
BBTP† [13] (prev. best)	ResNet-101-FPN		1×	21.1	45.5	17.2	11.2	22.0	29.8
BoxInst†	ResNet-101-FPN		1×	31.6	54.0	31.9	13.9	34.2	48.2
BoxInst	ResNet-50-FPN	✓	3×	32.1	55.1	32.4	15.6	34.3	43.5
BoxInst	ResNet-101-FPN	✓	1×	32.5	55.3	33.0	15.6	35.1	44.1
BoxInst	ResNet-101-FPN	✓	3×	33.2	56.5	33.6	16.2	35.3	45.1
BoxInst	ResNet-101-BiFPN [27]	✓	3×	33.9	57.7	34.5	16.5	36.1	46.6
BoxInst	ResNet-DCN-101-BiFPN [34]	✓	3×	**35.0**	**59.3**	**35.6**	**17.1**	**37.2**	**48.9**

Table 4: Comparisons with state-of-the-art methods on the COCO `test-dev` split. "†" means that the results are on the COCO `val2017` split. BBTP only reported the results on the `val2017` split. Our BoxInst outperforms the previous best reported mask AP by over absolute 10% mask AP. Ours even outperforms two recent fully supervised methods, YOLACT and PolarMask, and is close to state-of-the-art fully-supervised results. 'DCN': deformable convolutions [34]. '1×' means 90K iterations.

Experiments

- **Comparison with State-of-the-art on COCO**

method	backbone	aug.	sched.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
fully supervised methods:									
Mask R-CNN [11]	ResNet-50-FPN	✓	3×	37.5	59.3	40.2	21.1	39.6	48.3
CondInst [28]	ResNet-50-FPN	✓	3×	37.8	59.1	40.5	21.0	40.3	48.7
Mask R-CNN	ResNet-101-FPN	✓	3×	38.8	60.9	41.9	21.8	41.4	50.5
YOLACT-700 [4]	ResNet-101-FPN	✓	4.5×	31.2	50.6	32.8	12.1	33.3	47.1
PolarMask [32]	ResNet-101-FPN	✓	2×	32.1	53.7	33.1	14.7	33.8	45.3
CondInst	ResNet-101-FPN	✓	3×	39.1	60.9	42.0	21.5	41.7	50.9
box-supervised methods:									
BBTP† [13] (prev. best)	ResNet-101-FPN		1×	21.1	45.5	17.2	11.2	22.0	29.8
BoxInst†	ResNet-101-FPN		1×	31.6	54.0	31.9	13.9	34.2	48.2
BoxInst	ResNet-50-FPN	✓	3×	32.1	55.1	32.4	15.6	34.3	43.5
BoxInst	ResNet-101-FPN	✓	1×	32.5	55.3	33.0	15.6	35.1	44.1
BoxInst	ResNet-101-FPN	✓	3×	33.2	56.5	33.6	16.2	35.3	45.1
BoxInst	ResNet-101-BiFPN [27]	✓	3×	33.9	57.7	34.5	16.5	36.1	46.6
BoxInst	ResNet-DCN-101-BiFPN [34]	✓	3×	**35.0**	**59.3**	**35.6**	**17.1**	**37.2**	**48.9**

BoxInst outperforms the previous S.O.T.A BBTP with a large margin

Table 4: Comparisons with state-of-the-art methods on the COCO `test-dev` split. "†" means that the results are on the COCO `val2017` split. BBTP only reported the results on the `val2017` split. Our BoxInst outperforms the previous best reported mask AP by over absolute 10% mask AP. Ours even outperforms two recent fully supervised methods, YOLACT and PolarMask, and is close to state-of-the-art fully-supervised results. 'DCN': deformable convolutions [34]. '1×' means 90K iterations.

Experiments

- **Comparison with State-of-the-art on COCO**

method	backbone	aug.	sched.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
fully supervised methods:									
Mask R-CNN [11]	ResNet-50-FPN	✓	3×	37.5	59.3	40.2	21.1	39.6	48.3
CondInst [28]	ResNet-50-FPN	✓	3×	37.8	59.1	40.5	21.0	40.3	48.7
Mask R-CNN	ResNet-101-FPN	✓	3×	38.8	60.9	41.9	21.8	41.4	50.5
YOLACT-700 [4]	ResNet-101-FPN	✓	4.5×	31.2	50.6	32.8	12.1	33.3	47.1
PolarMask [32]	ResNet-101-FPN	✓	2×	32.1	53.7	33.1	14.7	33.8	45.3
box-supervised methods:									
BBTP† [13] (prev. best)	ResNet-101-FPN		1×	21.1	45.5	17.2	11.2	22.0	29.8
BoxInst	ResNet-101-FPN	✓	1×	31.6	54.0	31.9	15.9	34.2	46.2
BoxInst	ResNet-50-FPN	✓	3×	32.1	55.1	32.4	15.6	34.3	43.5
BoxInst	ResNet-101-FPN	✓	1×	32.5	55.3	33.0	15.6	35.1	44.1
BoxInst	ResNet-101-FPN	✓	3×	33.2	56.5	33.6	16.2	35.3	45.1
BoxInst	ResNet-101-BiFPN [27]	✓	3×	33.9	57.7	34.5	16.5	36.1	46.6
BoxInst	ResNet-DCN-101-BiFPN [34]	✓	3×	**35.0**	**59.3**	**35.6**	**17.1**	**37.2**	**48.9**

BoxInst even outperforms the fully supervised YOLACT and PolarMask

Table 4: Comparisons with state-of-the-art methods on the COCO `test-dev` split. "†" means that the results are on the COCO `val2017` split. BBTP only reported the results on the `val2017` split. Our BoxInst outperforms the previous best reported mask AP by over absolute 10% mask AP. Ours even outperforms two recent fully supervised methods, YOLACT and PolarMask, and is close to state-of-the-art fully-supervised results. 'DCN': deformable convolutions [34]. '1×' means 90K iterations.

Experiments

- **Comparison with State-of-the-art on COCO**

method	backbone	aug.	sched.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
fully supervised methods:									
Mask R-CNN [11]	ResNet-50-FPN	✓	3×	37.5	59.3	40.2	21.1	39.6	48.3
CondInst [28]	ResNet-50-FPN	✓	3×	37.8	59.1	40.5	21.0	40.3	48.7
Mask R-CNN	ResNet-101-FPN	✓	3×	38.8	60.9	41.9	21.8	41.4	50.5
YOLACT-700 [4]	ResNet-101-FPN	✓	4.5×	31.2	50.6	32.8	12.1	33.3	47.1
PolarMask [32]	ResNet-101-FPN	✓	2×	32.1	53.7	33.1	14.7	33.8	45.3
box-supervised methods:									
BBTP† [13] (prev. best)	ResNet-101-FPN		1×	21.1	45.5	17.2	11.2	22.0	29.8
BoxInst†	ResNet-101-FPN		1×	31.6	54.0	31.9	13.9	34.2	48.2
BoxInst	ResNet-50-FPN	✓	3×	32.1	55.1	32.4	15.6	34.3	43.5
BoxInst	ResNet-101-FPN	✓	1×	32.5	55.3	33.0	15.6	35.1	44.1
BoxInst	ResNet-101-FPN	✓	3×	33.2	56.5	33.6	16.2	35.3	45.1
BoxInst	ResNet-101-BiFPN [27]	✓	3×	33.9	57.7	34.5	16.5	36.1	46.6
BoxInst	ResNet-DCN-101-BiFPN [34]	✓	3×	**35.0**	**59.3**	**35.6**	**17.1**	**37.2**	**48.9**

BoxInst vs. Mask R-CNN and CondInst

Table 4: Comparisons with state-of-the-art methods on the COCO `test-dev` split. "†" means that the results are on the COCO `val2017` split. BBTP only reported the results on the `val2017` split. Our BoxInst outperforms the previous best reported mask AP by over absolute 10% mask AP. Ours even outperforms two recent fully supervised methods, YOLACT and PolarMask, and is close to state-of-the-art fully-supervised results. 'DCN': deformable convolutions [34]. '1×' means 90K iterations.

Experiments

- **Experiments on PASCAL VOC**

method	backbone	AP	AP_{50}	AP_{75}
GrabCut [25]	ResNet-101	17.8	37.8	15.5
SDI [17]	VGG-16	-	44.8	16.3
BBTP [13]	ResNet-101	23.1	54.1	17.1
BBTP w/ CRF	ResNet-101	27.5	59.1	21.9
BBTP*	ResNet-101	20.5	51.1	14.3
BBTP* w/ CRF	ResNet-101	25.0	56.9	18.9
BoxInst	ResNet-50	32.2	58.1	31.0
BoxInst	ResNet-101	**34.4**	**60.1**	**34.6**

Table 5: Results on Pascal VOC `val2012`. Here, BBTP* denotes the results after we fix the issue [1] in its Matlab evaluation code. Clearly, BoxInst achieves significantly improved mask AP, outperforming previous best by about 10%. Here, the GrabCut obtains the instance masks by taking as input the boxes generated by BoxInst. Thus, the only difference between the GrabCut and BoxInst is the way to obtain the masks.

Experiments

- **Semi-supervised Instance Segmentation**

L_{proj}	$L_{pairwise}$	all 80 classes AP	AP$_{50}$	AP$_{75}$	60 unseen classes AP	AP$_{50}$	AP$_{75}$
		24.7	44.6	24.2	19.9	38.3	18.5
✓		31.8	52.5	33.2	29.7	49.3	31.0
✓	✓	**32.5**	**53.0**	**34.0**	**30.9**	**50.1**	**32.4**
box supervised		30.7	52.2	31.1	29.6	49.7	30.4

Table 6: BoxInst for semi-supervised instance segmentation. These models are trained with the
and the other 60 classes (*i.e.*, unsee
annotations.

L_{proj}	$L_{pairwise}$	all 80 classes AP	AP$_{50}$	AP$_{75}$	20 unseen classes AP	AP$_{50}$	AP$_{75}$
		32.1	51.6	33.9	25.5	45.5	25.1
✓		33.1	53.8	34.3	31.6	57.4	30.0
✓	✓	**33.8**	**54.3**	**35.7**	**35.9**	**60.9**	**36.3**
box supervised		30.7	52.2	31.1	29.6	49.7	30.4

Table 7: BoxInst for semi-supervised instance segmentation. The models are trained with the 60 classes mask annotations and other 20 classes (*i.e.*, unseen classes) are only with box annotations.

Abstract

- **BoxInst**

  - achieved high-quality instance segmentation w/ only box annotation

  - core idea: the projection and pair-wise affinity mask loss

  - excellent instance segmentation performance w/o mask on COCO and PASCAL VOC

Thank you