Microsoft

CVPR

微软亚洲研究院创研论坛

CVPR 2020 论文分享会

Revisiting the Sibling Head in Object Detector

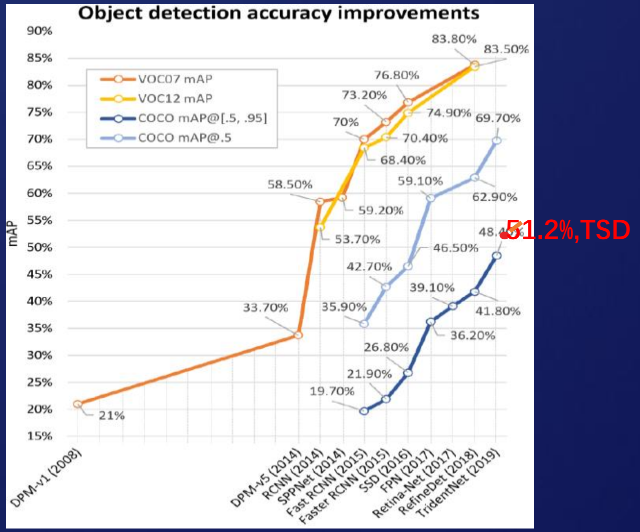
Guanglu Song[1], Yu Liu[2], Xiaogang Wang[2]
[1]SenseTime X-Lab
[2]The Chinese University of Hong Kong, Hong Kong
[1]songguanglu@sensetime.com, [2]{yuliu, xgwang}@ee.cuhk.edu.hk

Code is available: https://github.com/Sense-X/TSD

01 Revisiting the object detection task

- More complex scenes and large-scale object IDs

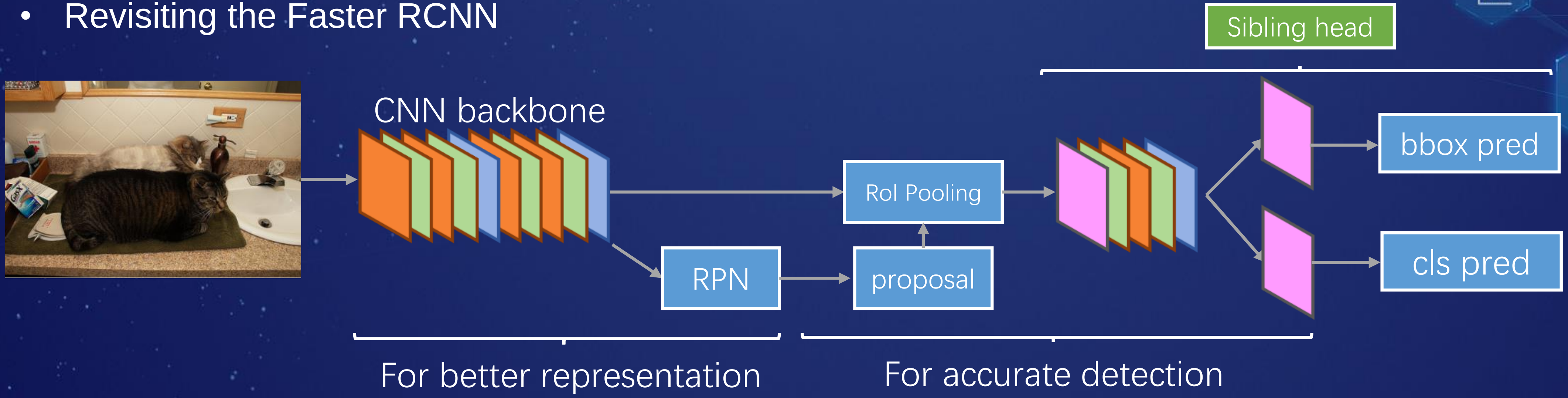
- Main benchmarks
    - Pascal VOC dataset
    - COCO dataset
    - ILSVRC
    - Object 365
    - ……
- Main challenges
    - Accurate cls and precise loc
    - Missed GT labels
    - Heavy occlusion
    - Dense instances
    - Noise annotations
    - ……
- Performance Based on COCO and VOC datasets



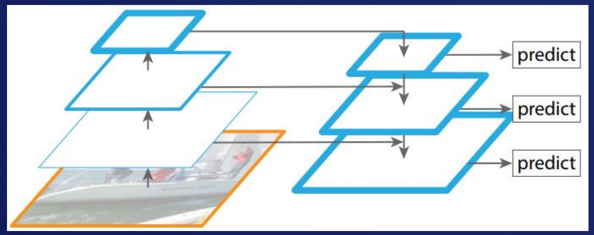
[1] Zou Z, Shi Z, Guo Y, et al. Object detection in 20 years: A survey[J]. arXiv preprint arXiv:1905.05055, 2019.
[2] https://storage.googleapis.com/openimages

01 Revisiting the SOTA framework

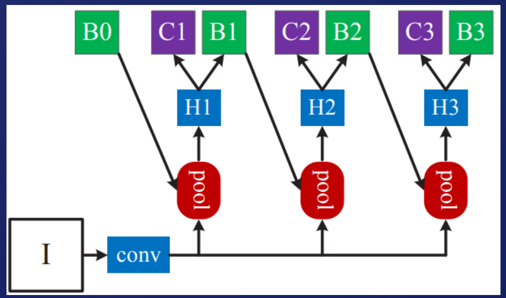
- Revisiting the Faster RCNN



Since then, many efficient detectors are proposed to solve the visual detection task.



FPN



Cascade RCNN

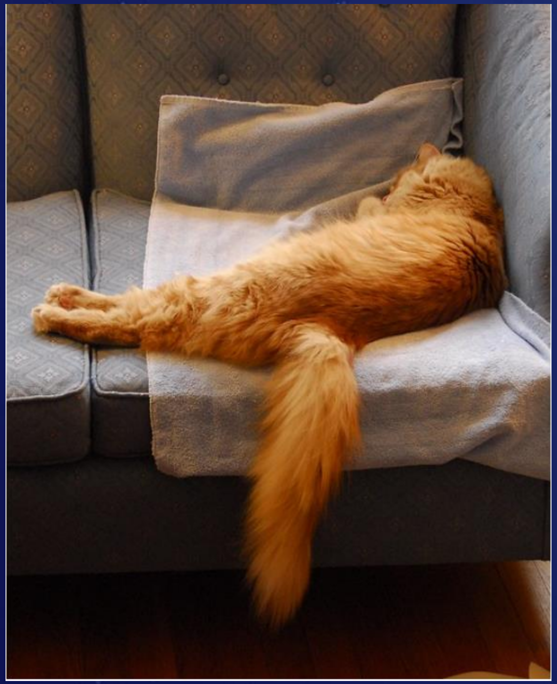
[3] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems. 2015: 91-99.

[4] Cai Z, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6154-6162.

01 The conflict in sibling head

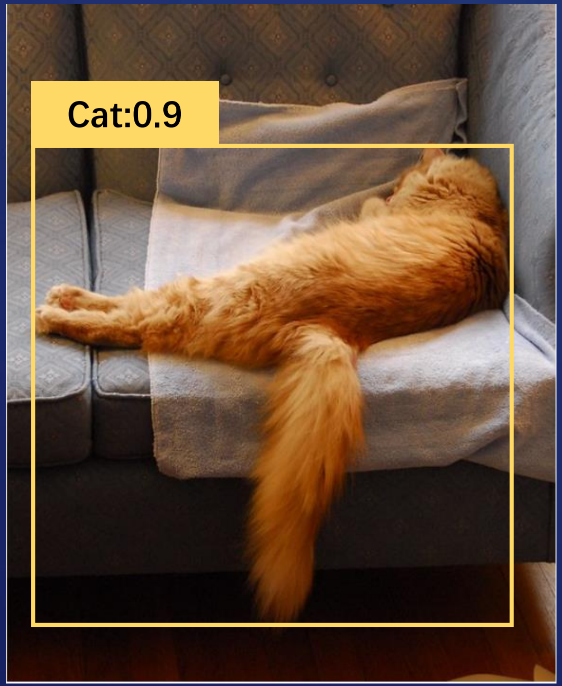
- On such a large scale object detection task, there is the potential conflict in sibling head.

Classification



We need to predict the class in it.

Detection



We need to predict the class and localization in it.

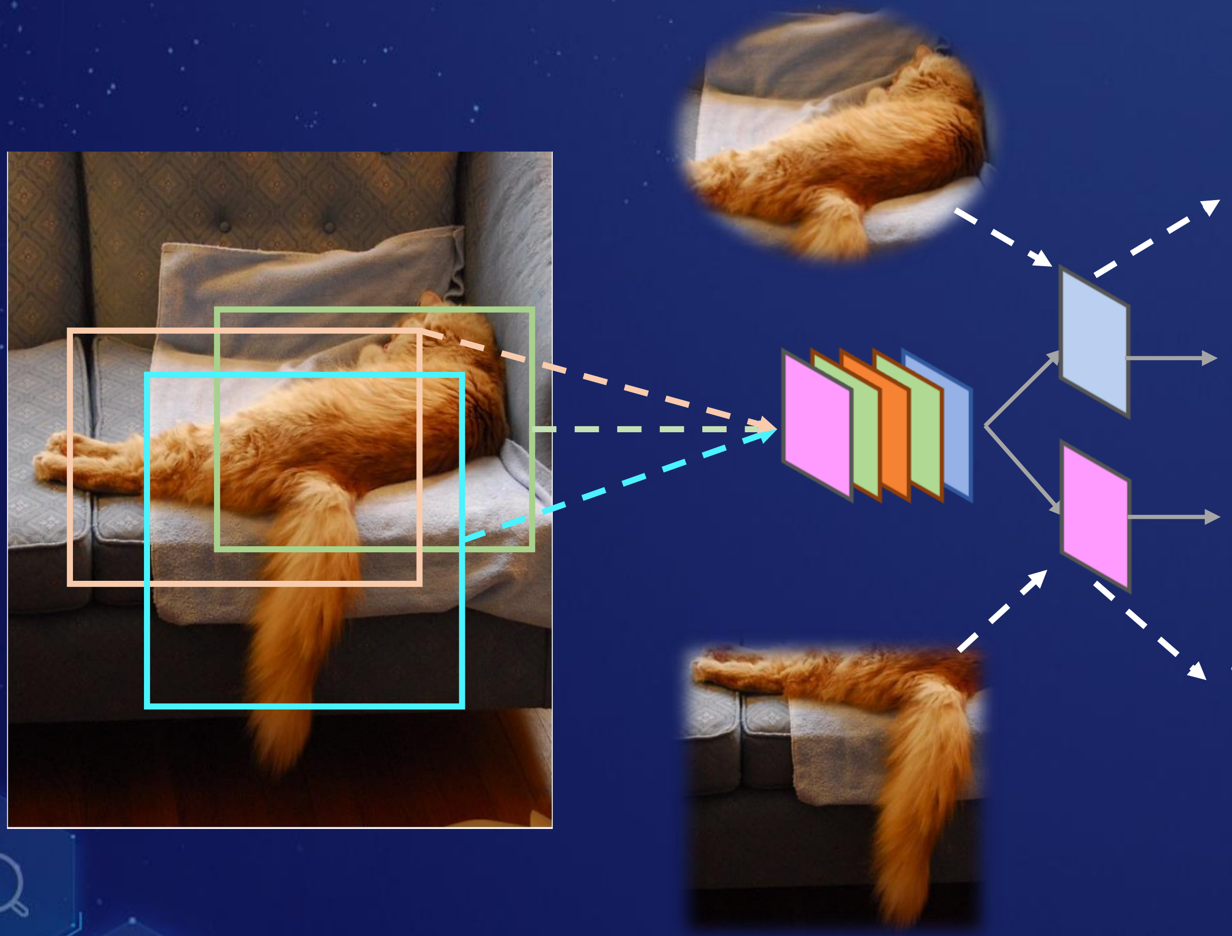
Faster RCNN ⟶ Multi-task Learning

Potential conflict:

$$\text{Classification: } \overbrace{\mathcal{C}(f(F_l,P)) = \mathcal{C}(f(F_l,P+\varepsilon)),}^{\text{translation-agnostic}}$$

$$\text{Localization: } \underbrace{\mathcal{R}(f(F_l,P)) \neq \mathcal{R}(f(F_l,P+\varepsilon))}_{\text{translation sensitivity}}$$

01 The conflict in sibling head



These clues here help us determine it's a cat.

For classification, the predicted confidence should be 1.

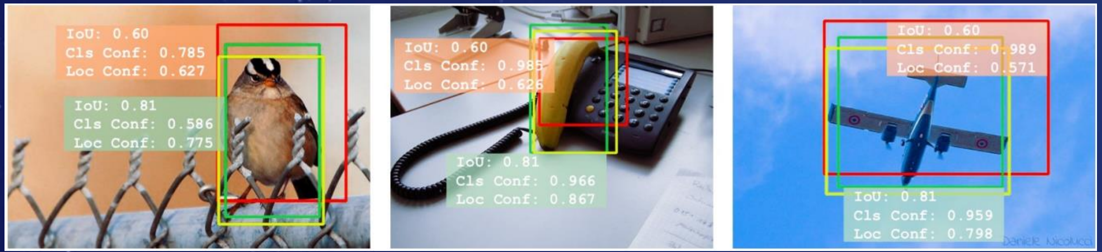
For localization, the $[\Delta x, \Delta y, \Delta w, \Delta h]$ should be different.

The clues here help us determine where its boundaries are.

02 Motivation

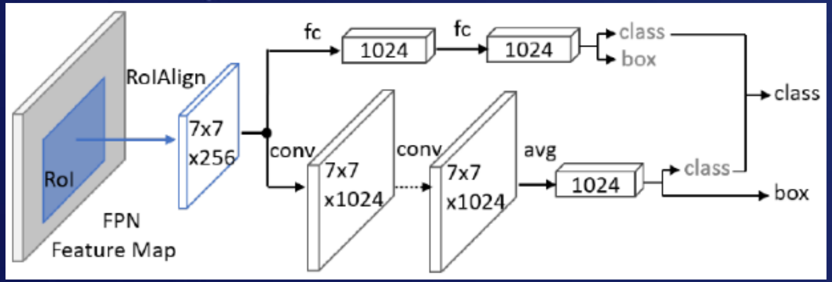
Some works have explored this conflict

IOUNet

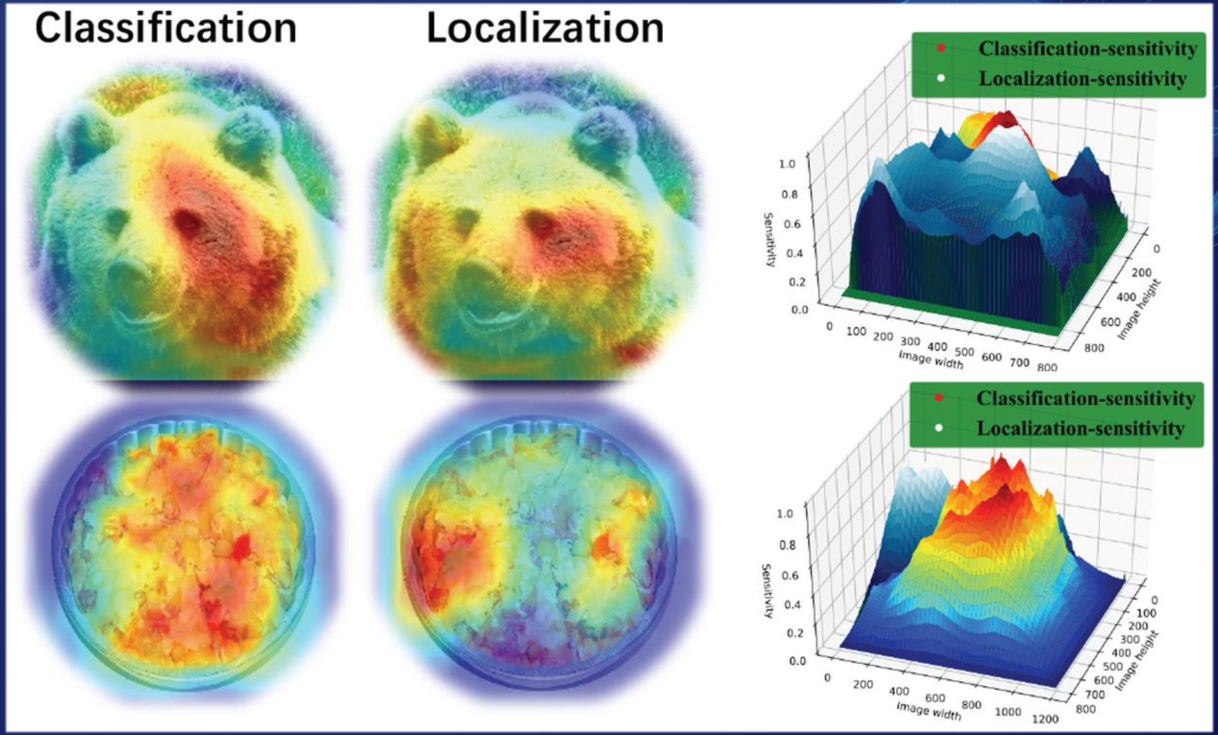


Misalignment between classification and localization.

Double-Head RCNN



Task spatial misalignment



[5] Jiang B, Luo R, Mao J, et al. Acquisition of localization confidence for accurate object detection[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 784-799.
[6] Wu Y, Chen Y, Yuan L, et al. Rethinking Classification and Localization in R-CNN[J]. arXiv preprint arXiv:1904.06493, 2019.

03 TSD (Task-aware spatial disentanglement)

Classical Faster RCNN

$$\mathcal{L} = \mathcal{L}_{cls}(\mathcal{H}_1(F_l, P), y) + \mathcal{L}_{loc}(\mathcal{H}_2(F_l, P), \mathcal{B})$$

Extracting feature

$$\mathcal{H}_1(\cdot) = \{\boxed{f(\cdot)}, \boxed{C(\cdot)}\}, \quad \mathcal{H}_2(\cdot) = \{f(\cdot), \boxed{R(\cdot)}\}$$

Classification

Localization

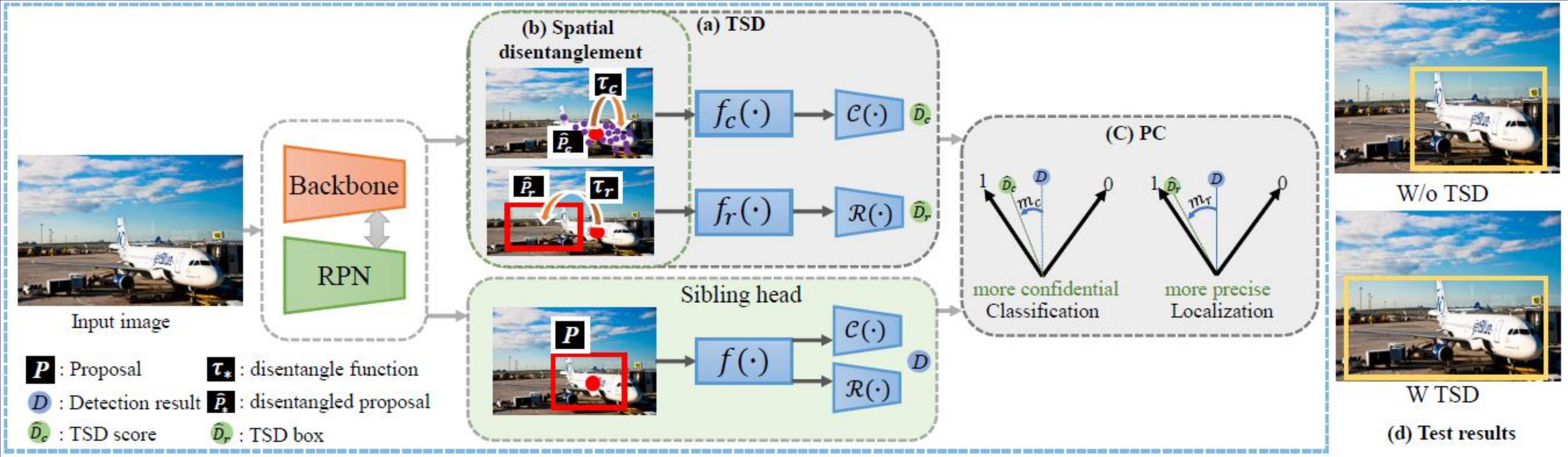
Disentangle them from both input and feature extractor.

$$\mathcal{L} = \mathcal{L}^D_{cls}(\mathcal{H}^D_1(F_l, \hat{P}_c), y) + \mathcal{L}^D_{loc}(\mathcal{H}^D_2(F_l, \hat{P}_r), \mathcal{B})$$

$$\underbrace{\mathcal{H}^D_1 = \{f_c(\cdot), C(\cdot)\} \quad \hat{P}_c = \tau_c(P, \Delta C),}_{\text{Be friendly to classification}} \qquad \underbrace{\mathcal{H}^D_2 = \{f_r(\cdot), R(\cdot)\} \quad \hat{P}_r = \tau_r(P, \Delta R)}_{\text{Be friendly to localization}}$$

This naturally leads to the pipeline of TSD.

03 TSD (Task-aware spatial disentanglement)



For classification and localization in **TSD**

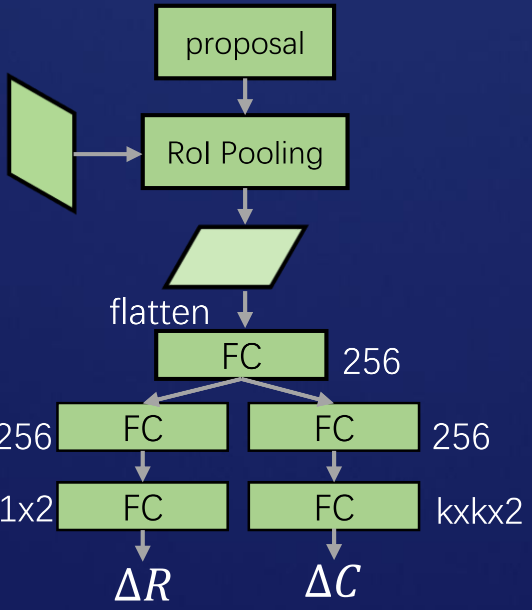
$$\Delta R = \gamma \mathcal{F}_r(F;\theta_r) \cdot (w,h)$$

$$\Delta C = \gamma \mathcal{F}_c(F;\theta_c) \cdot (w,h)$$

Bilinear interpolation is used

$$\hat{F}_c(x,y) = \sum_{p\in G(x,y)} \frac{\mathcal{F}_B(p_0 + \Delta C(x,y,1), p_1 + \Delta C(x,y,2))}{|G(x,y)|}$$

$$\hat{F}_r(x,y) = \sum_{p\in G(x,y)} \frac{\mathcal{F}_B(p_0 + \Delta R(1,1,1), p_1 + \Delta R(1,1,2))}{|G(x,y)|}$$



Progressive constraint （PC）

For classification

$$\mathcal{M}_{cls} = |\mathcal{H}_1(y|F_l, P) - \mathcal{H}_1^D(y|F_l, \tau_c(P,\Delta C)) + m_c|_+$$

For localization

$$\mathcal{M}_{loc} = |IOU(\hat{\mathcal{B}},\mathcal{B}) - IOU(\hat{\mathcal{B}}_D,\mathcal{B}) + m_r|_+$$

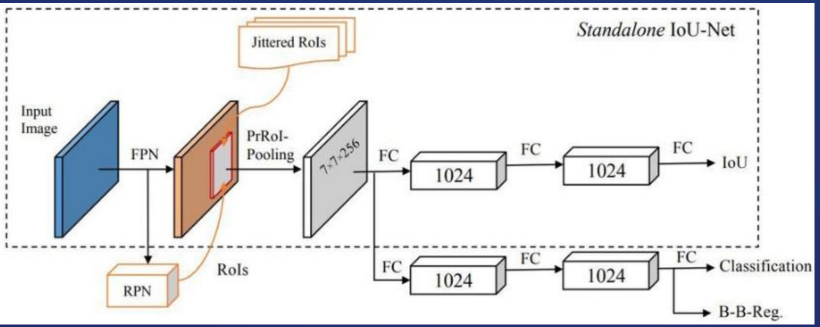
Total optimization

$$\mathcal{L} = \underbrace{\mathcal{L}_{rpn} + \mathcal{L}_{cls} + \mathcal{L}_{loc}}_{\text{classical loss}} + \underbrace{\mathcal{L}_{cls}^D + \mathcal{L}_{loc}^D + \mathcal{M}_{cls} + \mathcal{M}_{loc}}_{\text{TSD loss}}$$

03 TSD (Task-aware spatial disentanglement)

Different from other related works

IOUNet

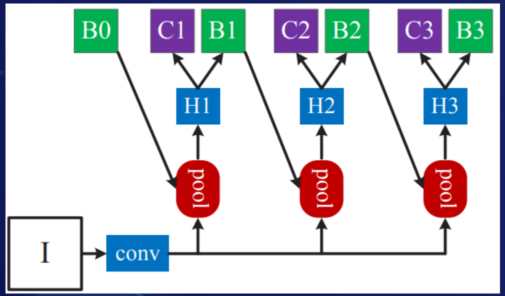


Learning IoU for bbox to alleviate the conflict between cls and loc.

Double-Head RCNN

Disentangling them from feature extractors.

Cascade RCNN

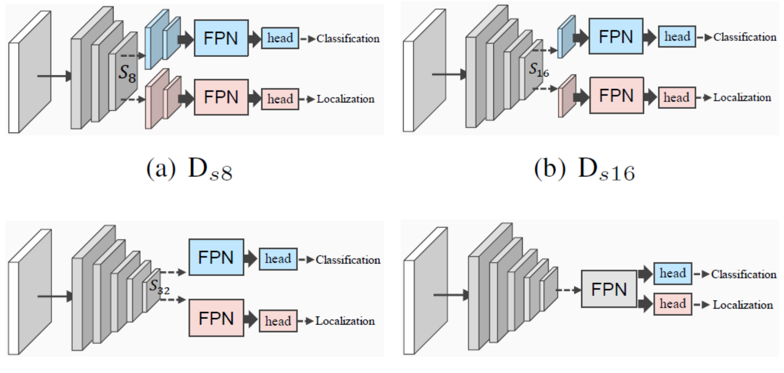


Proposals (or Bboxs) are also shared between classification and localization.

If the B_i generated by the last stage is also dominated by the classification, it may be still failed to regress the GT in this stage.

04 Experiments

Task-aware disentanglement.



(a) D_{s8} (b) D_{s16} (c) D_{s32} (d) D_{head}

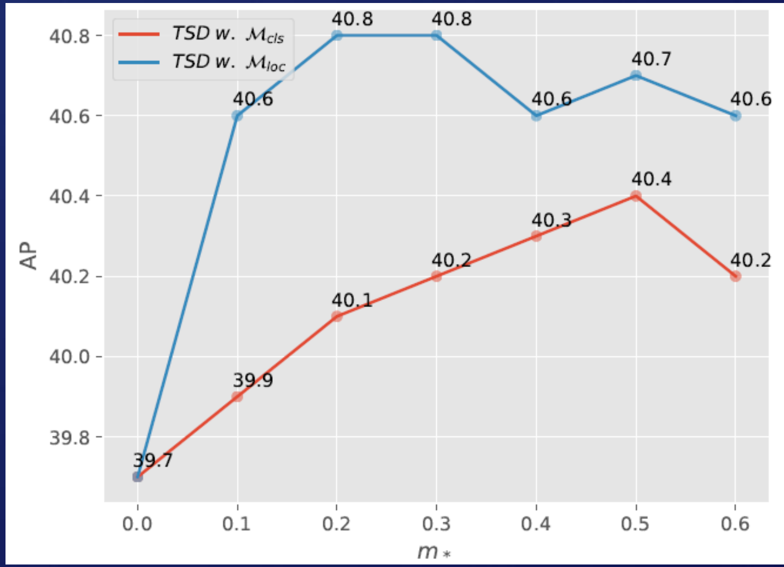
Disentanglement	#param	AP	$AP_{.5}$	$AP_{.75}$
ResNet-50	41.8M	36.1	58.0	38.8
ResNet-50+D_{s8}	81.1M	22.3	46.3	16.7
ResNet-50+ D_{s16}	74.0M	22.0	46.2	16.3
ResNet-50+ D_{s32}	59M	20.3	44.7	13.2
ResNet-50+ D_{head}	55.7M	37.3	59.4	40.2
TSD w/o PC	58.9M	**38.2**	**60.5**	**41.1**

Joint training with sibling head

Method	AP	$AP_{.5}$	$AP_{.75}$
TSD w/o PC	38.2	60.5	41.1
+ Joint training with sibling head $\mathcal{H}_*$	39.7	61.7	42.8

Effectiveness of PC

Method	TSD	PC		AP	$AP_{.5}$	$AP_{.75}$
		$\mathcal{M}_{cls}$	$\mathcal{M}_{loc}$			
ResNet-50	✓			39.7	61.7	42.8
ResNet-50	✓	✓		40.1	61.7	43.2
ResNet-50	✓		✓	40.8	61.7	43.8
ResNet-50	✓	✓	✓	41.0	61.7	44.3



04 Experiments

Applicable to variant backbones

Method	Ours	AP	$AP_{.5}$	$AP_{.75}$	runtime
ResNet-50		36.1	58.0	38.8	159.4 ms
ResNet-50	✓	**41.0**	**61.7**	**44.3**	174.9 ms
ResNet-101		38.6	60.6	41.8	172.4ms
ResNet-101	✓	**42.4**	**63.1**	**46.0**	189.0ms
ResNet-101-DCN		40.8	63.2	44.6	179.3ms
ResNet-101-DCN	✓	**43.5**	**64.4**	**47.0**	200.8ms
ResNet-152		40.7	62.6	44.6	191.3ms
ResNet-152	✓	**43.9**	**64.5**	**47.7**	213.2ms
ResNeXt-101 [36]		40.5	62.6	44.2	187.5ms
ResNeXt-101 [36]	✓	**43.5**	**64.5**	**46.9**	206.6ms

Generalization on large scale

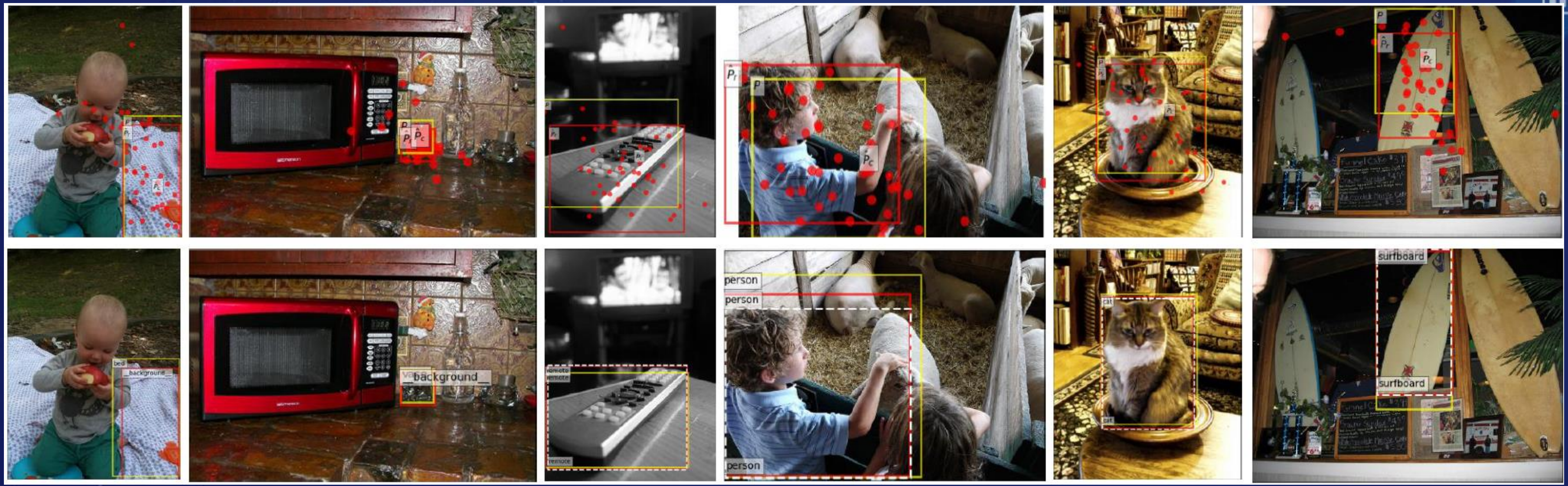
Method	TSD	$AP_{.5}$ (Val)	$AP_{.5}$ (LB)
ResNet-50		64.64	49.79
ResNet-50	✓	**68.18**	**52.55**
Cascade-DCN-SENet154		69.27	55.979
Cascade-DCN-SENet154	✓	**71.17**	**58.34**
DCN-ResNeXt101*		68.70	55.05
DCN-ResNeXt101*	✓	**71.71**	**58.59**
DCN-SENet154*		70	57.771
DCN-SENet154*	✓	**72.19**	**60.5**

Dataset	train		validation		trainval		test	
	images	objects	images	objects	images	objects	images	objects
VOC-2007	2,501	6,301	2,510	6,307	5,011	12,608	4,952	14,976
VOC-2012	5,717	13,609	5,823	13,841	11,540	27,450	10,991	-
ILSVRC-2014	456,567	478,807	20,121	55,502	476,688	534,309	40,152	-
ILSVRC-2017	456,567	478,807	20,121	55,502	476,688	534,309	65,500	-
MS-COCO-2015	82,783	604,907	40,504	291,875	123,287	896,782	81,434	-
MS-COCO-2018	118,287	860,001	5,000	36,781	123,287	896,782	40,670	-
OID-2018	1,743,042	14,610,229	41,620	204,621	1,784,662	14,814,850	125,436	625,282

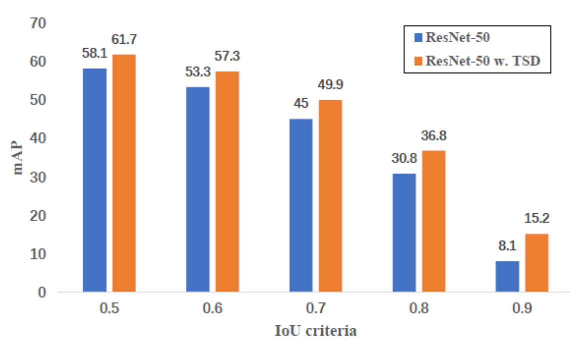
Applicable to Mask RCNN

Method	Ours	AP^{bb}	$AP^{bb}_{.5}$	$AP^{bb}_{.75}$	AP^{mask}	$AP^{mask}_{.5}$	$AP^{mask}_{.75}$
ResNet-50 w. FPN		37.2	58.8	40.2	33.6	55.3	35.4
ResNet-50 w. FPN	✓	**41.5**	**62.1**	**44.8**	**35.8**	**58.3**	**37.7**
ResNet-101 w. FPN		39.5	61.2	43.0	35.7	57.9	38.0
ResNet-101 w. FPN	✓	**43.0**	**63.6**	**46.8**	**37.2**	**59.9**	**39.5**

04 Experiments



Performance in different IoU criteria.



Performance in different scale criteria.

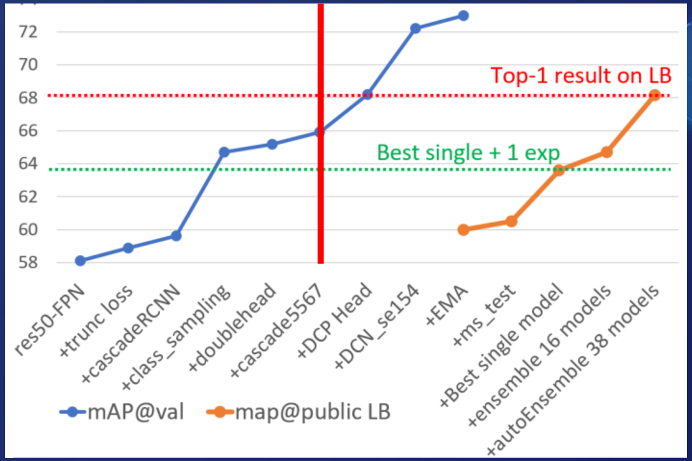
Criteria	TSD	$AP_{.5}$	$AP_{.6}$	$AP_{.7}$	$AP_{.8}$	$AP_{.9}$
AP_{small}		38.4	33.7	26.7	16.2	3.6
AP_{small}	✓	**40.0**	**35.6**	**28.8**	**17.7**	**5.3**
AP_{medium}		62.9	58.4	49.7	33.6	8.7
AP_{medium}	✓	**67.7**	**62.4**	**54.9**	**40.2**	**15.4**
AP_{large}		69.5	65.5	56.8	43.2	14.8
AP_{large}	✓	**74.8**	**71.6**	**65.0**	**53.2**	**27.9**

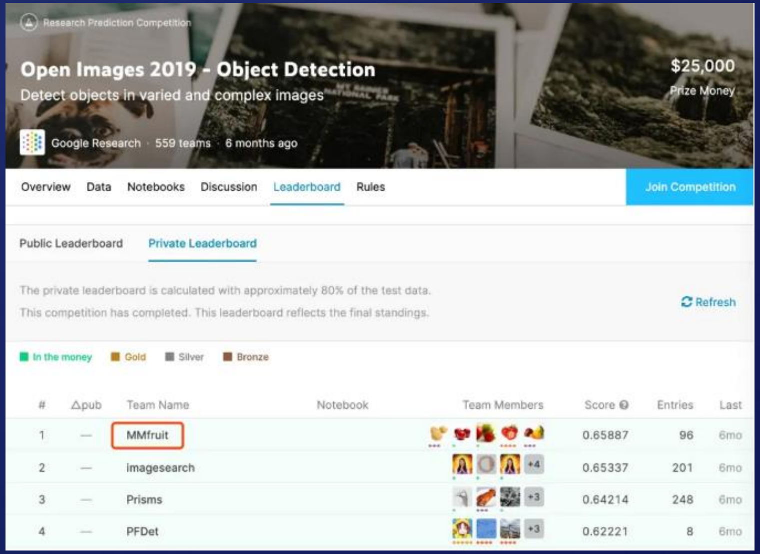
04 Experiments

Comparison with state-of-the-arts

Method	backbone	$b\&w$	AP	$AP_{.5}$	$AP_{.75}$	AP_s	AP_m	AP_l
RefineDet512 [41]	ResNet-101		36.4	57.5	39.5	16.6	39.9	51.4
RetinaNet800 [22]	ResNet-101		39.1	59.1	42.3	21.8	42.7	50.2
CornerNet [17]	Hourglass-104 [28]		40.5	56.5	43.1	19.4	42.7	53.9
ExtremeNet [42]	Hourglass-104 [28]		40.1	55.3	43.2	20.3	43.2	53.1
FCOS [34]	ResNet-101		41.5	60.7	45.0	24.4	44.8	51.6
RPDet [39]	ResNet-101-DCN	✓	46.5	67.4	50.9	30.3	49.7	57.1
CenterNet511 [6]	Hourglass-104	✓	47.0	64.5	50.7	28.9	49.9	58.9
TridentNet [20]	ResNet-101-DCN	✓	48.4	69.7	53.5	31.8	51.3	60.3
NAS-FPN [8]	AmoebaNet (7 @ 384)	✓	48.3	-	-	-	-	-
Faster R-CNN w FPN [21]	ResNet-101		36.2	59.1	39.0	18.2	39.0	48.2
Auto-FPN† [38]	ResNet-101		42.5	-	-	-	-	-
Regionlets [37]	ResNet-101		39.3	59.8	-	21.7	43.7	50.9
Grid R-CNN [27]	ResNet-101		41.5	60.9	44.5	23.3	44.9	54.1
Cascade R-CNN [2]	ResNet-101		42.8	62.1	46.3	23.7	45.5	55.2
DCR [4]	ResNet-101		40.7	64.4	44.6	24.3	43.7	51.9
IoU-Net† [15]	ResNet-101		40.6	59.0	-	-	-	-
Double-Head-Ext† [35]	ResNet-101		41.9	62.4	45.9	23.9	45.2	55.8
SNIPER [32]	ResNet-101-DCN	✓	46.1	67.0	51.6	29.6	48.9	58.1
DCNV2 [43]	ResNet-101	✓	46.0	67.9	50.8	27.8	49.1	59.5
PANet [24]	ResNet-101	✓	47.4	67.2	51.8	30.1	51.7	60.0
GCNet [3]	ResNet-101-DCN	✓	48.4	67.6	52.7	-	-	-
TSD†	ResNet-101		**43.1**	**63.6**	**46.7**	**24.9**	**46.8**	**57.5**
TSD	ResNet-101		**43.2**	64.0	**46.9**	**24.0**	46.3	55.8
TSD*	ResNet-101-DCN	✓	**49.4**	69.6	**54.4**	**32.7**	**52.5**	**61.0**
TSD*	SENet154-DCN [14]	✓	**51.2**	**71.9**	**56.0**	**33.8**	**54.8**	**64.2**

1st Place Solutions for OpenImage2019-





05 Conclusion

- We delve into the essential barriers behind the tangled tasks in RoI-based detectors and reveal the bottlenecks that limit the upper bound of detection performance.
- We propose a simple but effective operator called task-aware spatial disentanglement (TSD) to deal with the tangled tasks conflict.
- We further propose a progressive constraint (PC) to enlarge the performance margin between TSD and the classical sibling head.
- We validate the effectiveness of our approach on the standard COCO benchmark and large-scale OpenImageV5 dataset with thorough ablation studies. It can steadily improve performance with different backbones.

谢谢观看

THANK YOU

Q&A

Code is available: https://github.com/Sense-X/TSD