NEURAL INFORMATION PROCESSING SYSTEMS

DCFS Project Page

Tencent YouTu Lab

Decoupling Classifier for Boosting Few-shot Object Detection and Instance Segmentation

Bin-Bin Gao[1], Xiaochen Chen[1], Zhongyi Huang[1], Congchong Nie[1], Jun Liu[1]

Jinxiang Lai[1], Guannan Jiang[2], Xi Wang[2] and Chengjie Wang[1]

[1]Tencent YouTu Lab, [2]CATL

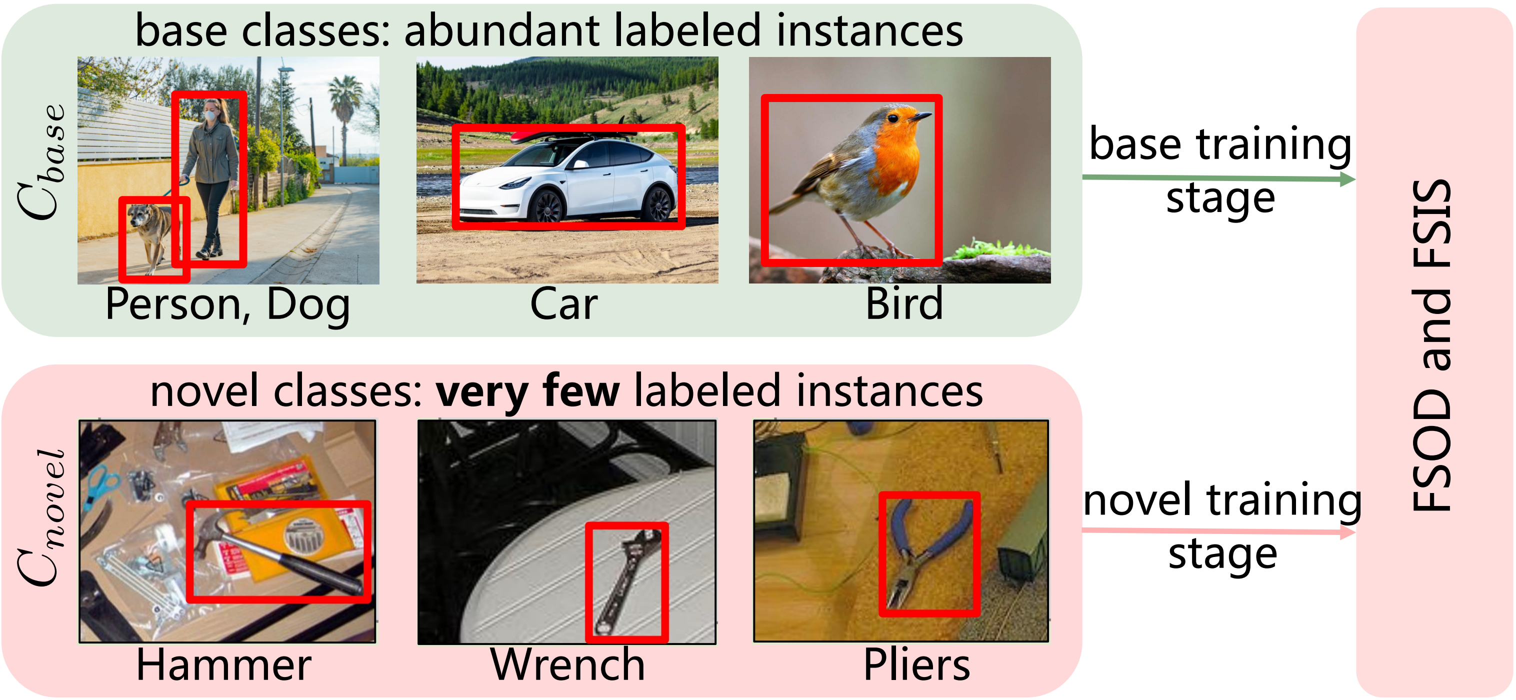
Nov. 28— Dec. 9, 2022

Introduction

Few-shot object detection/ instance segmentation

Few-shot object detection (FSOD) aims to detect novel objects with very few novel instances and abundant base instances.

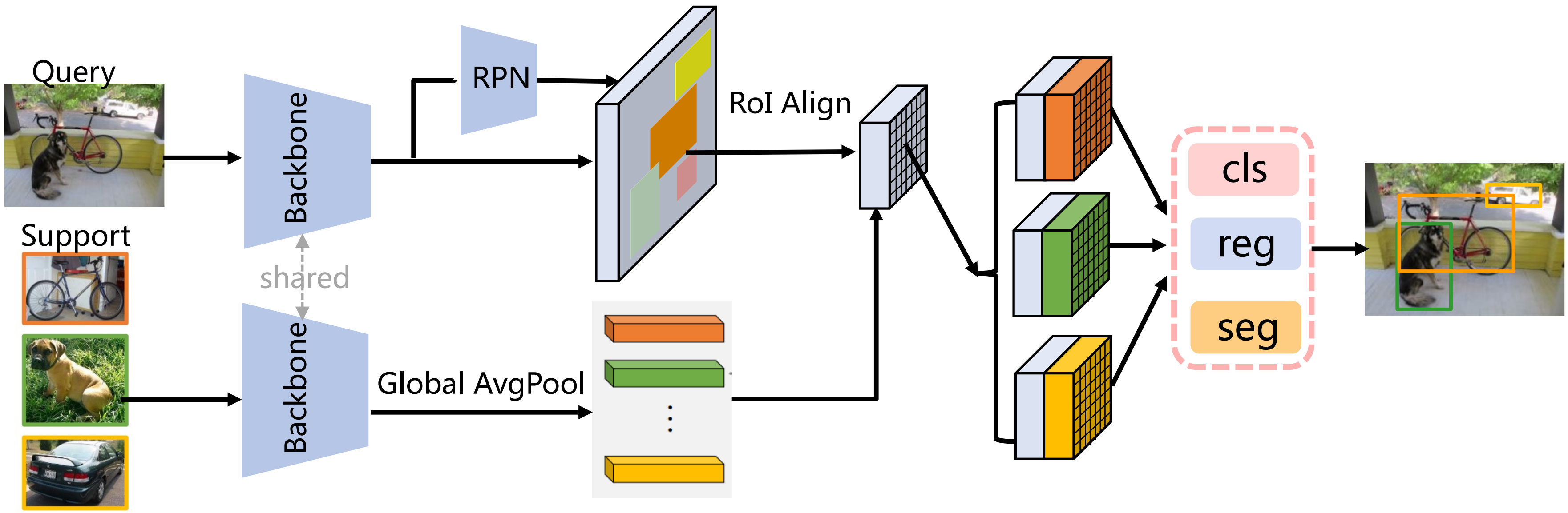
General idea: a few-shot model should be able to transfer previous knowledge about base classes to help future detection tasks on novel classes.



Introduction

Few-shot learning paradigm

Meta-learning paradigm aims to acquire task-level knowledge on base classes and generalize better to novel classes.



- ✓ Meta RCNN, *ICCV 19;*
- ✓ FSRW, *ICCV 19;*
- ✓ FSDetView, *ECCV 20;*
- ✓ TIP, *CVPR 21;*
- ✓ FCT, *CVPR 22;*

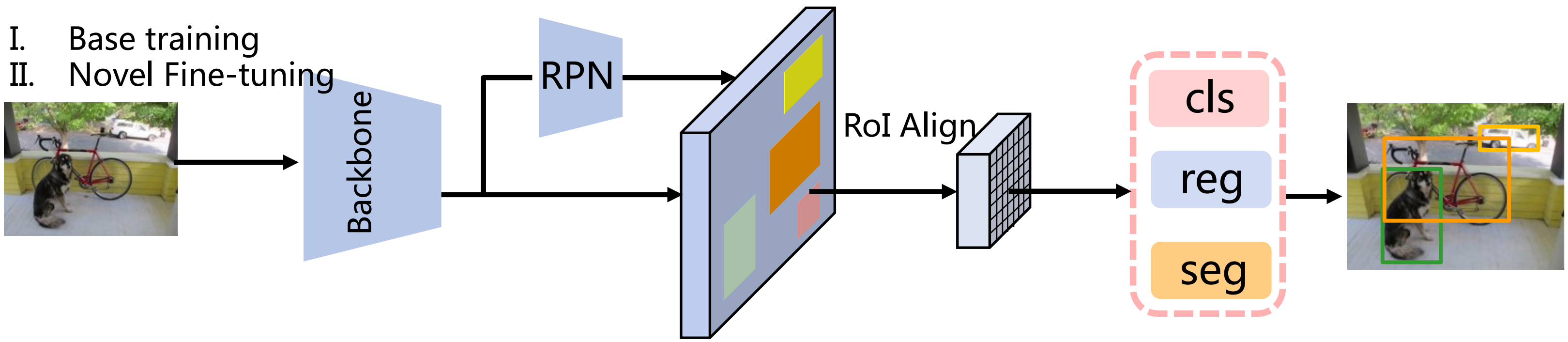
……

These methods suffer from a complicated training process (episodic training) and data organization (support-query pair).

Introduction

Few-shot learning paradigm

Transfer-learning mainly follows a fully supervised framework.



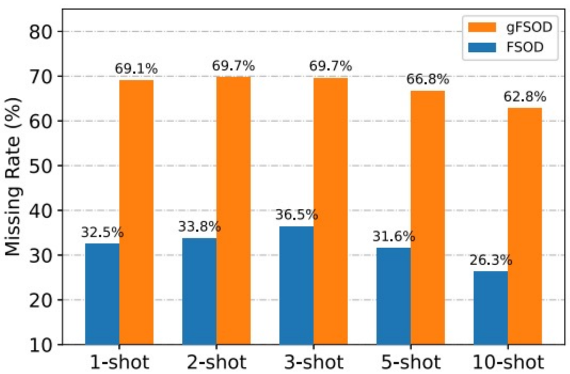
- ✓ TFA, *ICML 19;*
- ✓ MPSR, *ECCV 20;*
- ✓ FSCE, *CVPR 21;*
- ✓ SRR-FSD, *CVPR 21;*
- ✓ DeFRCN, *ICCV 21;*
- ✓ FADI, *NeurIPS 21;*

……

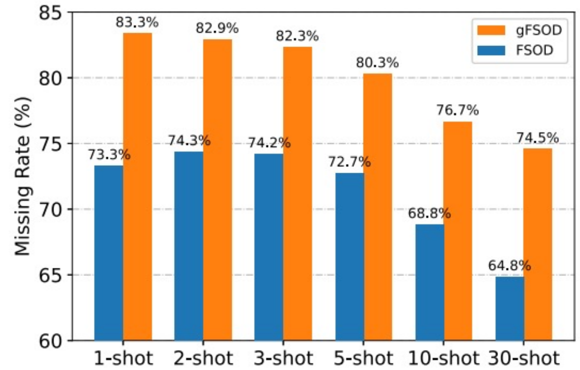
These transfer-learning methods is more simple and more efficient.

Motivation

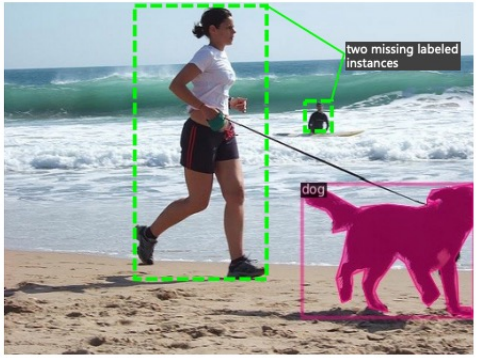
Missing label issue



(a) PASCAL VOC



(b) MS-COCO



(c) a one-shot labeled image

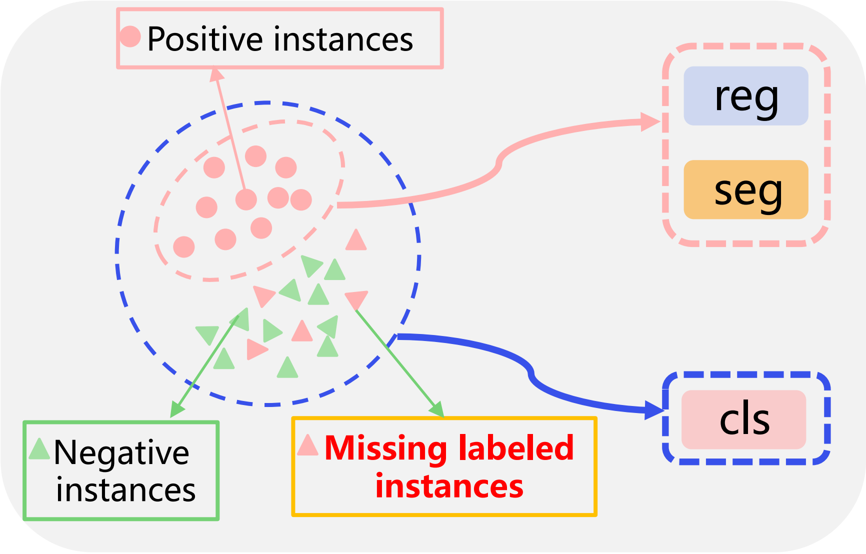
Instance-level few-shot setting: an instance as a shot for each class.
This easily meets missing label issue.

Does a few-shot model learn well under missing label conditions?

Motivation

Missing labele issue

Biased classification



The box regression and mask segmentation heads only accept clear positive instances and thus no negative effects.

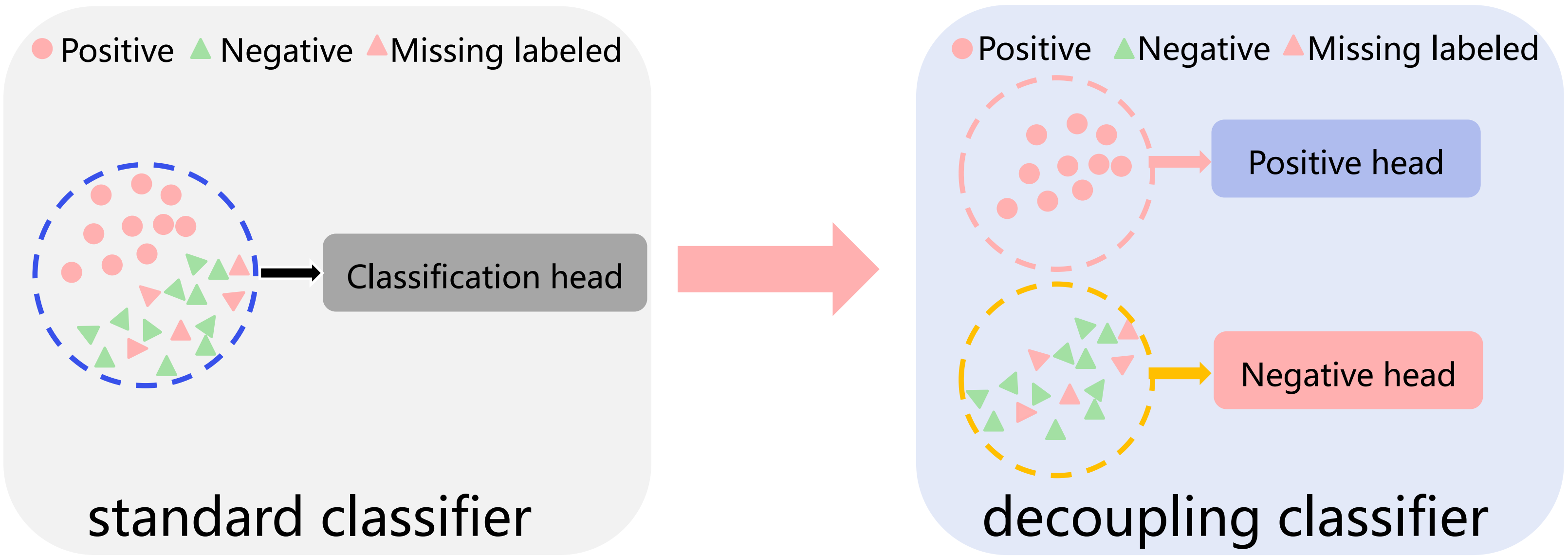
However, the classification head may be confused by missing labeled instances and thus results in a biased classification towards incorrectly recognizing foreground objects as background.

Can we design a method to mitigate the biased classification?

Proposed Method

Tencent YouTu Lab

Decoupling Classifier

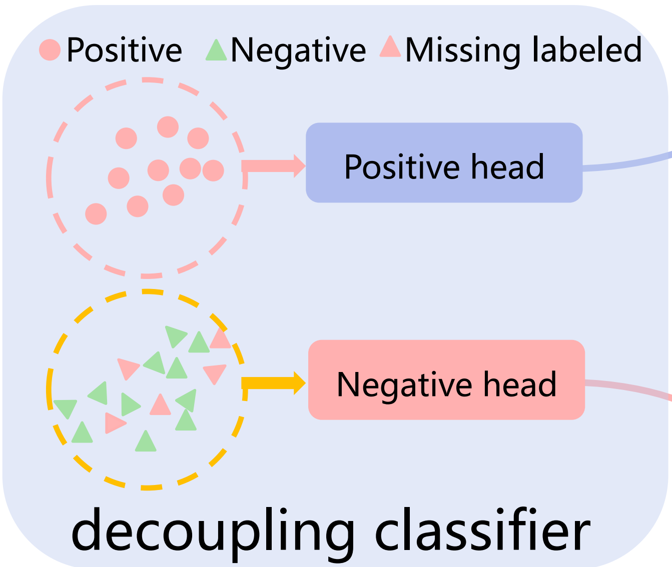


We propose a simple but effective method that decouples the standard classifier into two parallel heads to process clear positive instances and negative instances with missing labels.

Proposed Method

Tencent YouTu Lab

Decoupling Classifier



$$\hat{p}_i = \frac{\exp(x_i)}{\sum_t \exp(x_t)}$$

$$\mathcal{L}^{\mathrm{fg}}_{\mathrm{CLS}} = -\sum_{i=0}^{C} y_i \log(\hat{p}_i).$$

$$\boxed{\vec{m}} = [m^0, m^1, \cdots, m^{C-1}, m^C]^T.$$

$$\bar{p}_i = \frac{\exp(\boxed{m_i} x_i)}{\sum_t \exp(m_t x_t)}$$

$$\mathcal{L}^{\mathrm{bg}}_{\mathrm{CLS}} = -\sum_{i=0}^{C} y_i^{\mathrm{bg}} \log(\bar{p}_i).$$

Positive head:
standard softmax function and cross-entropy loss.

Negative head:
The only change is to introduce an image-level label vector $\vec{m}$ into the softmax function.

Proposed Method

The core code for decoupling classifier

The core implementation only uses one line of code but leads to consistent improvements.

Algorithm 1 PyTorch-like Style Code for Decoupling Classifier.

```
def dc_loss(x, y, m):
    """
    Compute loss for the decoupling classifier.
    Return scalar Tensor for single image.

    Args:
        x: predicted class scores in [-inf, +inf], x's size: N x (1+C), where N is the
           number of region proposals of one image.
        y: ground-truth classification labels in [0, C-1], y's size: N x 1, where [0,C-1]
           represent foreground classes and C-1 represents the background class.
        m: image-level label vector and its element is 0 or 1, m's size: 1 x (1+C)

    Returns:
        loss
    """

    # background class index
    N = x.shape[0]
    bg_label = x.shape[1]-1

    # positive head
    pos_ind = y!=bg_label
    pos_logit = x[pos_ind,:]
    pos_score = F.softmax(pos_logit, dim=1) # Eq. 4
    pos_loss = F.nll_loss(pos_score.log(), y[pos_ind], reduction="sum") #Eq. 5

    # negative head
    neg_ind = y==bg_label
    neg_logit = x[neg_ind,:]
    neg_score = F.softmax(m.expand_as(neg_logit)*neg_logit, dim=1) #Eq. 8
    neg_loss = F.nll_loss(neg_score.log(), y[neg_ind], reduction="sum")  #Eq. 9

    # total loss
    loss = (pos_loss + neg_loss)/N #Eq. 6

    return loss
```

Experiments

Comparisons with state-of-the-arts

Table 1: Comparisons with SOTA FSOD methods on PASCAL-VOC.

Methods / Shots		w/g	Novel Set 1: 1	2	3	5	10	Novel Set 2: 1	2	3	5	10	Novel Set 3: 1	2	3	5	10
FRCN-ft [39]	*ICCV 19*	✗	13.8	19.6	32.8	41.5	45.6	7.9	15.3	26.2	31.6	39.1	9.8	11.3	19.1	35.0	45.1
FSRW [17]	*ICCV 19*	✗	14.8	15.5	26.7	33.9	47.2	15.7	15.2	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9
MetaDet [34]	*ICCV 19*	✗	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1
MetaRCNN [39]	*ICCV 19*	✗	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
TFA [33]	*ICML 20*	✗	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
MPSR [35]	*ECCV 20*	✗	41.7	-	51.4	55.2	61.8	24.4	-	39.2	39.9	47.8	35.6	-	42.3	48.0	49.7
TIP [19]	*CVPR 21*	✗	27.7	36.5	43.3	50.2	59.6	22.7	30.1	33.8	40.9	46.9	21.7	30.6	38.1	44.5	50.9
DCNet [16]	*CVPR 21*	✗	33.9	37.4	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	34.9	39.7	42.6	50.7
CME [20]	*CVPR 21*	✗	41.5	47.5	50.4	58.2	60.9	27.2	30.2	41.4	42.5	46.8	34.3	39.6	45.1	48.3	51.5
FSCE [31]	*CVPR 21*	✗	44.2	43.8	51.4	61.9	63.4	27.3	29.5	43.5	44.2	50.2	37.2	41.9	47.5	54.6	58.5
SRR-FSD [43]	*CVPR 21*	✗	47.8	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4
FADI [1]	*NeurIPS 21*	✗	**50.3**	54.8	54.2	59.3	63.2	30.6	35.0	40.3	42.8	48.0	45.7	49.7	49.1	55.0	59.6
FCT [13]	*CVPR 22*	✗	38.5	49.6	53.5	59.8	64.3	25.9	34.2	40.1	44.9	47.4	34.7	43.9	49.3	53.1	56.3
DeFRCN† [28]	*ICCV 21*	✗	46.2	56.4	59.3	62.4	63.7	32.6	39.9	**44.5**	**48.3**	**51.8**	39.8	49.9	52.6	56.1	59.7
Ours		✗	46.2	**57.4**	**59.9**	**62.9**	**64.5**	**32.6**	**39.9**	43.4	47.9	51.3	**40.3**	**50.5**	**53.8**	**56.9**	**60.7**
DeFRCN * [28]	*ICCV 21*	✗	53.6	57.5	61.5	64.1	60.8	**30.1**	38.1	**47.0**	**53.3**	47.9	**48.4**	50.9	52.3	54.9	57.4
Ours *		✗	**56.6**	**59.6**	**62.9**	**65.6**	**62.5**	29.7	**38.7**	46.2	48.9	**48.1**	47.9	**51.9**	**53.3**	**56.1**	**59.4**

Table 2: Comparisons on MS-COCO

Methods / Shots		1	2	3	5	10	30
FRCN-ft [39]	*ICCV 19*	1.0	1.8	2.8	4.0	6.5	11.1
FSRW [17]	*ICCV 19*	-	-	-	-	5.6	9.1
MetaDet [34]	*ICCV 19*	-	-	-	-	7.1	11.3
MetaRCNN [39]	*ICCV 19*	-	-	-	-	8.7	12.4
TFA [33]	*ICML 20*	4.4	5.4	6.0	7.7	10.0	13.7
MPSR [35]	*ECCV 20*	5.1	6.7	7.4	8.7	9.8	14.1
FSDetView [38]	*ICCV 20*	4.5	6.6	7.2	10.7	12.5	14.7
TIP [19]	*CVPR 21*	-	-	-	-	16.3	18.3
DCNet [16]	*CVPR 21*	-	-	-	-	12.8	18.6
CME [20]	*CVPR 21*	-	-	-	-	15.1	16.9
FSCE [31]	*CVPR 21*	-	-	-	-	11.1	15.3
SRR-FSD [43]	*CVPR 21*	-	-	-	-	11.3	14.7
FADI [1]	*NeurIPS 21*	5.7	7.0	8.6	10.1	12.2	16.1
FCT [13]	*CVPR 22*	5.1	7.2	9.8	12.0	15.3	20.2
DeFRCN† [28]	*ICCV 21*	7.7	11.4	13.3	15.5	18.5	22.5
Ours		**8.1**	**12.1**	**14.4**	**16.6**	**19.5**	**22.7**

Table 3: Comparisons with SOTA gFSOD methods on PASCAL-VOC.

Methods / Shots		w/g	Novel Set 1: 1	2	3	5	10	Novel Set 2: 1	2	3	5	10	Novel Set 3: 1	2	3	5	10
FRCN-ft [39]	*ICCV 19*	✓	9.9	15.6	21.6	28.0	52.0	9.4	13.8	17.4	21.9	39.7	8.1	13.9	19.0	23.9	44.6
FSRW [17]	*ICCV 19*	✓	14.2	23.6	29.8	36.5	35.6	12.3	19.6	25.1	31.4	29.8	12.5	21.3	26.8	33.8	31.0
TFA [33]	*ICML 20*	✓	25.3	36.4	42.1	47.9	52.8	18.3	27.5	30.9	34.1	39.5	17.9	27.2	34.3	40.8	45.6
FSDetView [38]	*ECCV 20*	✓	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6
DeFRCN [28]	*ICCV 21*	✓	40.2	53.6	58.2	63.6	66.5	29.5	39.7	43.4	48.1	52.8	35.0	38.3	52.9	57.7	60.8
Ours		✓	**45.8**	**59.1**	**62.1**	**66.8**	**68.0**	**31.8**	**41.7**	**46.6**	**50.3**	**53.7**	**39.6**	**52.1**	**56.3**	**60.3**	**63.3**

Table 4: Comparisons with SOTA gFSOD methods on MS-COCO.

Method / Shots	1: O	B	N	2: O	B	N	3: O	B	N	5: O	B	N	10: O	B	N	30: O	B	N
FRCN-ft [39]	16.2	21.0	1.7	15.8	20.0	3.1	15.0	18.8	3.7	14.4	17.6	4.6	13.4	16.1	5.5	13.5	15.6	7.4
TFA [33]	24.4	31.9	1.9	24.9	31.9	3.9	25.3	32.0	5.1	25.9	41.2	7.0	26.6	32.4	9.1	28.7	34.2	12.1
FSDetView [38]			3.2			4.9			6.7			8.1			10.7			15.9
DeFRCN [28]	24.4	30.4	4.8	25.7	31.4	8.5	26.6	32.1	10.7	27.8	32.6	13.6	29.7	34.0	16.8	31.4	34.8	21.2
Ours	**27.4**	**34.4**	**6.2**	**28.6**	**34.7**	**10.4**	**29.4**	**34.9**	**12.9**	**30.2**	**35.0**	**15.7**	**31.4**	**35.7**	**18.3**	**32.3**	**35.8**	**21.9**

gFSOD:
Our method significantly outperforms the SOTA by a large margin;

FSOD:
Ours method is also better than the SOTA under most cases.

Experiments

Comparisons with state-of-the-arts

Table 5: Comparisons with SOTA ***FSIS*** methods on MS-COCO.

Methods		Tasks	1		2		3		5		10		30	
			AP	AP50	AP	AP50	AP	AP50	AP	AP50	AP	AP50	AP	AP50
Meta R-CNN [39]	*ICCV 19*	Det	-	-	-	-	-	-	3.5	9.9	5.6	14.2	-	-
MTFA [10]	*CVPR 21*	Det	2.47	4.85	-	-	-	-	6.61	12.32	8.52	15.53	-	-
iMTFA [10]	*CVPR 21*	Det	3.28	6.01	-	-	-	-	6.22	11.28	7.14	12.91	-	-
Mask-DeFRCN† [28]	*ICCV 21*	Det	7.54	14.46	11.01	20.20	13.07	23.28	15.39	27.29	18.72	32.80	22.63	38.95
Ours		Det	**8.09**	**15.85**	**11.90**	**22.39**	**14.04**	**25.74**	**16.39**	**29.96**	**19.33**	**34.78**	**22.73**	**40.24**
Meta R-CNN [39]	*ICCV 19*	Seg	-	-	-	-	-	-	2.8	6.9	4.4	10.6	-	-
MTFA [10]	*CVPR 21*	Seg	2.66	4.56	-	-	-	-	6.62	11.58	8.39	14.64	-	-
iMTFA [10]	*CVPR 21*	Seg	2.83	4.75	-	-	-	-	5.24	8.73	5.94	9.96	-	-
Mask-DeFRCN† [28]	*ICCV 21*	Seg	6.69	13.24	9.51	18.58	11.01	21.27	12.66	24.58	15.39	29.71	18.28	35.20
Ours		Seg	**7.18**	**14.33**	**10.31**	**20.43**	**11.85**	**23.24**	**13.48**	**26.67**	**15.85**	**31.33**	**18.34**	**35.99**

Table 6: Comparisons with SOTA ***gFSIS*** methods on MS-COCO.

Shots	Methods	Object Detection						Instance Segmentation					
		Overall		Base		Novel		Overall		Base		Novel	
		AP	AP50	AP	AP50	AP	AP50	AP	AP50	AP	AP50	AP	AP50
	Base-Only			39.86	59.25					32.58	55.12		
1	iMTFA [10]	21.67	31.55	27.81	40.11	3.23	5.89	20.13	30.64	25.90	39.28	2.81	4.72
1	Mask-DeFRCN† [28]	23.82	35.70	30.11	44.42	4.95	9.55	19.58	33.38	24.63	41.57	4.45	8.81
1	**Ours**	**27.35**	**42.55**	**34.35**	**52.46**	**6.34**	**12.79**	**22.45**	**39.33**	**28.03**	**48.60**	**5.72**	**11.53**
2	Mask-DeFRCN† [28]	25.42	38.31	31.06	45.82	8.52	15.79	21.09	35.92	25.61	43.03	7.54	14.59
2	**Ours**	**28.63**	**44.74**	**34.67**	**52.82**	**10.52**	**20.49**	**23.73**	**41.49**	**28.52**	**49.12**	**9.38**	**18.62**
3	Mask-DeFRCN† [28]	26.54	40.01	31.77	46.83	10.87	19.55	22.04	37.48	26.22	43.95	9.48	18.06
3	**Ours**	**29.59**	**46.21**	**35.07**	**53.30**	**13.15**	**24.95**	**24.55**	**42.81**	**28.91**	**49.61**	**11.46**	**22.43**
5	iMTFA [10]	19.62	28.06	24.13	33.69	6.07	11.15	18.22	27.10	22.56	33.25	5.19	8.65
5	Mask-DeFRCN† [28]	27.82	42.12	32.54	48.03	13.69	24.41	23.03	39.37	26.84	45.04	11.60	22.36
5	**Ours**	**30.48**	**47.75**	**35.30**	**53.65**	**16.02**	**30.05**	**25.20**	**44.12**	**29.10**	**49.87**	**13.50**	**26.86**
10	iMTFA [10]	19.26	27.49	23.36	32.41	6.97	12.72	17.87	26.46	21.87	32.01	5.88	9.81
10	Mask-DeFRCN† [28]	29.88	45.25	34.17	50.48	17.02	29.58	24.75	42.32	28.23	47.33	14.32	27.29
10	**Ours**	**31.77**	**49.77**	**36.14**	**54.85**	**18.67**	**34.55**	**26.36**	**46.13**	**29.91**	**51.11**	**15.71**	**31.19**
30	Mask-DeFRCN† [28]	31.66	48.11	35.10	52.01	21.33	36.44	26.23	44.97	29.12	48.82	17.57	33.42
30	**Ours**	**32.92**	**51.37**	**36.45**	**55.05**	**22.30**	**40.31**	**27.31**	**47.61**	**30.32**	**51.41**	**18.29**	**36.22**

Our method **outperforms** the SOAT on MS-COCO in either *FSIS* or *gFSIS* setting.

Experiments

Tencent YouTu Lab

Ablation study

Table 7: The effects of DC and PCB for ***FSIS*** performance on MS-COCO.

Shots	M-Rate	DC	PCB	Complexity		Detection				Segmentation			
				#Params.	GFLOPs	Base		Novel		Base		Novel	
						AP	AP50	AP	AP50	AP	AP50	AP	AP50
1	83.3%	✗	✗	54.9M	334.54	30.09	44.45	3.89	7.43	24.62	41.58	3.52	6.88
		✓	✗	54.9M	334.54	34.35	52.46	5.04	10.03	28.03	48.60	4.59	9.12
		✗	✓	99.4M	377.88	30.11	44.42	4.95	9.55	24.63	41.57	4.45	8.81
		✓	✓	99.4M	377.88	**34.35**	**52.46**	**6.34**	**12.79**	**28.03**	**48.60**	**5.72**	**11.53**
5	80.3%	✗	✗	54.9M	334.54	32.54	48.03	11.94	21.16	26.84	45.04	10.10	19.37
		✓	✗	54.9M	334.54	35.30	53.65	14.01	26.17	29.10	49.87	11.80	23.38
		✗	✓	99.4M	377.88	32.54	48.03	13.69	24.41	26.84	45.04	11.60	22.36
		✓	✓	99.4M	377.88	**35.30**	**53.65**	**16.02**	**30.05**	**29.10**	**49.87**	**13.50**	**26.86**
10	76.7%	✗	✗	54.9M	334.54	34.05	50.21	14.96	25.70	28.12	47.10	12.60	23.81
		✓	✗	54.9M	334.54	36.13	54.81	16.66	30.79	29.90	51.07	13.98	27.72
		✗	✓	99.4M	377.88	34.17	50.48	17.02	29.58	28.23	47.33	14.32	27.29
		✓	✓	99.4M	377.88	**36.14**	**54.85**	**18.67**	**34.55**	**29.91**	**51.11**	**15.71**	**31.19**

Effectiveness:

The decoupling classifier is effective not only on novel classes but also on base classes;

Efficiency:

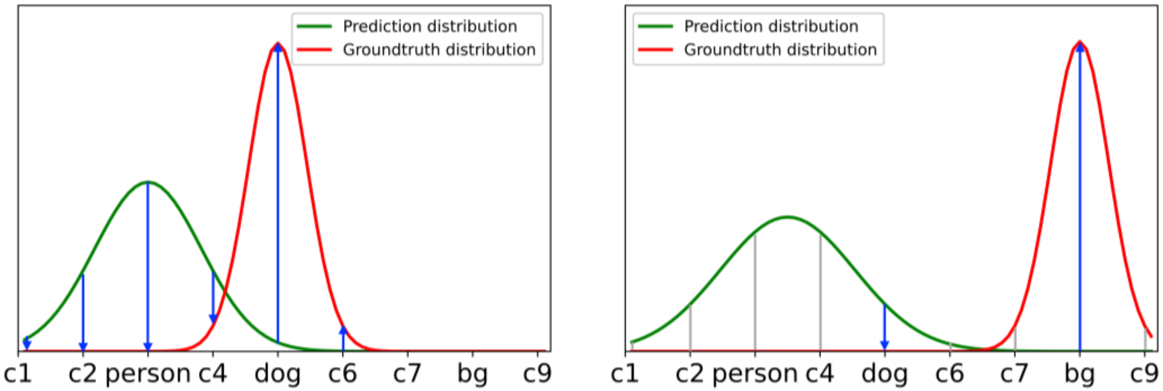
The decoupling classifier is more efficient because of no additional parameters or computation cost.

Discussion

Tencent YouTu Lab

Why does the DC work well?

Gradient optimization:



(a) Positive head (b) Negative head

Positive head:

$$\frac{\partial \mathcal{L}^{\mathrm{fg}}_{\mathrm{CLS}}}{\partial \theta_{cls}} = (\vec{\hat{p}} - \vec{y}^{\mathrm{fg}})\frac{\partial \vec{x}}{\theta_{cls}}$$

Negative head:

$$\frac{\partial \mathcal{L}^{\mathrm{bg}}_{\mathrm{CLS}}}{\partial \theta_{cls}} = \boxed{\vec{m}}(\vec{\hat{p}} - \vec{y}^{\mathrm{bg}})\frac{\partial \vec{x}}{\theta_{cls}}$$

Positive head: the gradient is updated in each dimension of the class space.

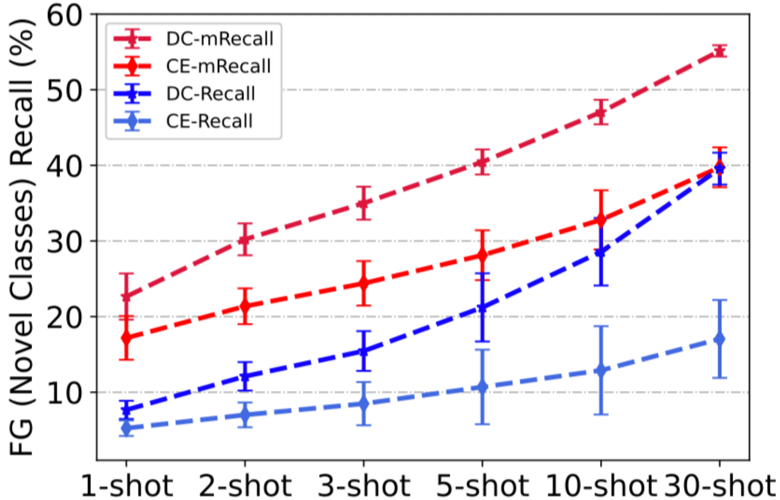
Negative head: the gradient is limited in some special dimension because of the introduced $\vec{m}$ and thus the biased classification may be alleviated.

Discussion

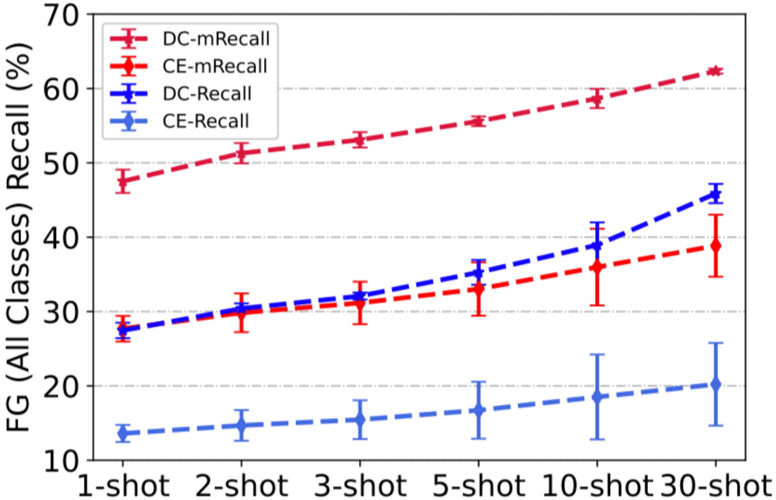
Tencent YouTu Lab

Why does the DC work well?

Generalization ability (Reacll and mRecall):



(a) FSIS



(b) gFSIS

The mRecall and Recall of our decoupling classifier is significantly higher that of the standard classifier on each shot with two few-shot settings. This means that the decoupling classifier is helpful to mitigate the bias classification thus boosting instance-level few-shot performance.

Discussion

Qualitative Evaluation

The baseline (Mask-DeFRCN) could fail to detect some objects, because it may tend to incorrectly recognize positive objects as background. However, the bias classification is well mitigated using our method, and thus better detection results are obtained.

Conclusions

Tencent YouTu Lab

- We rethink instance-level few-shot methods from the perspective of label completeness and discover that existing few-shot methods severely suffer from bias classification.
- We propose a simple but effective decoupling classifier for mitigating the bias classification in instance-level few-shot settings.
- We achieve state-of-the-art results on two instance-level few-shot tasks without any additional parameters and computation cost.

Thanks !

DCFS Project Page