Meta R-CNN : Towards General Solver for Instance-level Low-shot Learning (Supplementary Material)

Xiaopeng Yan[1*]; Ziliang Chen[1*], Anni Xu[1], Xiaoxi Wang[1], Xiaodan Liang[1,2], Liang Lin[1,2,†]
[1] Sun Yat-sen University [2]DarkMatter AI Research
{yanxp3,wangxx35}@mail2.sysu.edu.cn, c.ziliang@yahoo.com, 466783266@qq.com, xdliang328@gmail.com, linliang@ieee.org

1. Accelerated task adaptation

Meta-learning facilitates Faster R-CNN to detect novel-class low-shot objects. Through the lens of stochastic optimization, it gives the credits to the task adaptation acceleration. More specifically, we observe the performance comparison between Faster R-CNN (trained by two-phase strategy, *i.e.*, FRCN+ft-full) and Meta R-CNN over iterations. As shown in Fig 1, Meta R-CNN presents as an envelope that upper bounds Faster R-CNN. It indicates meta-learning encouraging faster performance improvement to novel-class object detection.

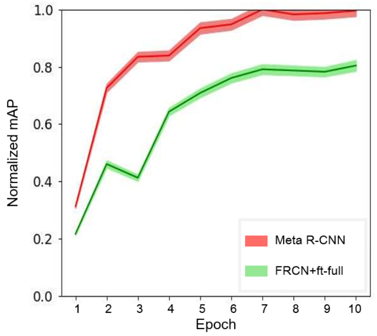
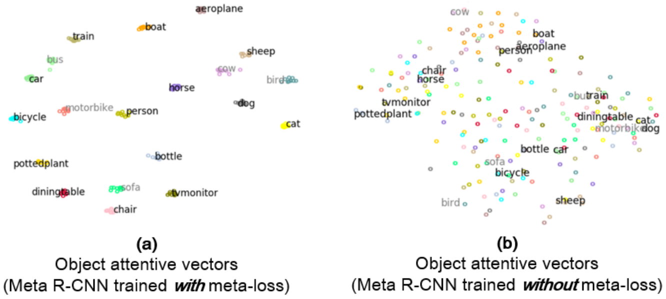


Figure 1. Normalized mAP *w.r.t.* novel-class object detection over iterations. The mean and variance values of Normalized mAP are computed by class-specific Normalized AP, which is normarlized by the converged value of AP against number of training iterations.

2. Attentive vector analysis

As we mentioned in the paper, Meta R-CNN takes class attentive vectors to remodel Faster R-CNN, while class attentive vectors are inferred by averaging the object attentive vectors in each class. It implies that learning good representation of object attentive vectors would lead to the success of Meta R-CNN. To this end, we visualize the object attentive vectors used for testing by t-SNE [2], and compare the same visualization when Meta R-CNN is trained without meta-loss $\left(L_{\mathrm{meta}}(\phi)\right)$. All are illustrated in Fig 2. First, we find that object attentive vectors tend to cluster together when they belong to the same class and repulse those from the other classes (See Fig 2 (a)). These object attentive vectors produce more deterministic class attentive vector (less inter-class variance when choosing different objects to induce class attentive vectors). To this Meta R-CNN is endowed with more stable performance, since class attentive vectors would not significant change when objects change. Distinct from this, when Meta R-CNN is trained without meta-loss $\big($Fig 2 (b)$\big)$, object attentive vectors become more diverse and the inter-class variance is very large. These object attentive vectors bring about two negative effects to Meta R-CNN: **1).** Due to the large inter-class variance, the trained model suffers unstable performances: if we change the objects, the according class attentive vectors will significantly change. **2).** The inferred class attentive vectors are probably close, resulting ambiguous object detection produced by the corresponding class-specific predictor heads.

In Fig 2 (a), it is also observed that the classes with similar semantics would be closer to those with different semantics. For instance, 'Car', 'Bus', 'Train' are close together, as they all belong to vehicle. The observation unveils that Meta R-CNN may achieve novel-class object detection by the aid of the base-class objects that share similar semantic information.



(a)
Object attentive vectors
(Meta R-CNN trained ***with*** meta-loss)

(b)
Object attentive vectors
(Meta R-CNN trained ***without*** meta-loss)

Figure 2. The t-SNE visualization of object attentive vectors with respect to Meta R-CNN trained w/wo meta-loss. For each class, 10 objects are taken to produce the object attentive vectors for visualization. Color indicates class (Best viewed in color).

[*] indicate equal contribution (Xiaopeng Yan and Ziliang Chen). † indicates corresponding author: Liang Lin.

Shot	Baselines	Novel-class Split-1						Novel-class Split-2						Novel-class Split-3					
		bird	bus	cow	mbike	sofa	mean	aero	bottle	cow	horse	sofa	mean	boat	cat	mbike	sheep	sofa	mean
1	YOLO-Low-shot[1]	**13.5**	10.6	**31.5**	13.8	**4.3**	**14.8**	**11.8**	**9.1**	15.6	**23.7**	**18.2**	**15.7**	**10.8**	**44.0**	**17.8**	**18.1**	**5.3**	**19.2**
	FRCN+joint	9.7	0.0	1.5	0.5	1.8	2.7	1.6	0.3	3.2	3.6	**0.8**	1.9	0.2	21.9	0.0	1.1	**3.0**	5.2
	FRCN+ft	13.4	14.8	4.9	**25.6**	0.7	11.9	0.5	0.2	15.9	12.2	0.6	5.9	**10.4**	7.3	13.1	3.5	0.6	5.0
	FRCN+ft-full	**14.3**	**16.6**	**16.4**	18.7	**2.9**	13.8	0.5	0.4	**22.7**	**15.0**	0.7	7.9	0.8	26.4	12.3	9.3	0.1	9.8
	Meta R-CNN (ours)	6.1	**32.8**	15.0	**35.4**	0.2	**19.9**	**23.9**	**0.8**	**23.6**	3.1	0.7	**10.4**	0.6	**31.1**	**28.9**	**11.0**	0.1	**14.3**
2	YOLO-Low-shot[1]	**21.2**	12.0	16.8	**17.9**	**9.6**	15.5	**28.6**	**0.9**	27.6	0.0	**19.5**	**15.3**	**5.3**	**46.4**	**18.4**	**26.1**	**12.4**	**21.7**
	FRCN+joint	12.4	0.1	2.2	0.3	0.5	3.1	2.3	0.2	3.9	5.4	1.0	2.6	1.3	25.0	0.2	9.7	**1.5**	7.5
	FRCN+ft	5.4	19.0	39.8	16.6	1.2	16.4	3.6	**1.3**	13.1	23.3	**1.4**	8.5	**5.3**	16.9	10.2	14.3	1.1	9.6
	FRCN+ft-full	8.1	**25.9**	**49.3**	13.0	**1.5**	**19.6**	3.5	0.1	**36.1**	**35.7**	1.1	**15.3**	2.2	**25.6**	13.9	13.9	0.9	11.3
	Meta R-CNN (ours)	**17.2**	**34.4**	**43.8**	**31.8**	0.4	**25.5**	**12.4**	0.1	**44.4**	**50.1**	0.1	**19.4**	**10.6**	24.0	**36.2**	**19.2**	0.8	**18.2**
3	YOLO-Low-shot [1]	26.1	19.1	40.7	20.4	**27.1**	26.7	29.4	**4.6**	34.9	6.8	**37.9**	22.7	**11.2**	**39.8**	**20.9**	23.7	**33.0**	**25.7**
	FRCN+joint	13.7	0.4	6.4	0.8	0.2	4.3	16.7	0.2	7.4	15.7	0.5	8.1	0.2	37.2	0.6	17.2	0.1	11.1
	FRCN+ft	**31.1**	24.9	**51.7**	23.5	13.6	29.0	**29.8**	0.1	40.3	43.8	2.9	23.4	3.7	32.8	18.2	30.7	5.0	18.1
	FRCN+ft-full	29.1	**34.1**	**55.9**	**28.6**	**16.1**	**32.8**	**31.9**	**0.3**	**45.2**	**50.4**	3.4	**26.2**	10.6	27.2	16.5	**31.7**	9.5	19.1
	Meta R-CNN (ours)	**30.1**	**44.6**	50.8	**38.8**	10.7	**35.0**	25.2	0.1	**50.7**	**53.2**	**18.8**	**29.6**	**16.3**	**39.7**	**32.6**	**38.8**	**10.3**	**27.5**
5	YOLO-Low-shot[1]	31.5	21.1	39.8	**40.0**	**37.0**	33.9	**33.1**	**9.4**	38.4	25.4	**44.0**	30.1	**14.2**	**57.3**	**50.8**	38.9	**41.6**	**40.6**
	FRCN+joint	17.4	7.9	9.6	14.0	9.1	11.8	3.2	**4.5**	16.1	24.8	0.6	9.9	1.6	39.7	3.2	16.4	3.4	12.9
	FRCN+ft	31.3	36.5	54.1	26.5	36.2	36.9	17.5	2.3	39.6	**55.0**	31.2	29.1	5.1	41.7	33.1	36.2	37.9	30.8
	FRCN+ft-full	**36.1**	**44.6**	**56.0**	33.5	**37.2**	**41.5**	23.1	3.9	**44.7**	54.0	32.2	**31.6**	11.0	**51.8**	36.0	**41.3**	34.6	35.0
	Meta R-CNN (ours)	**35.8**	**47.9**	**54.9**	**55.8**	34.0	**45.7**	**28.5**	0.3	**50.4**	**56.7**	**38.0**	**34.8**	**16.6**	45.8	**53.9**	**41.5**	**48.1**	**41.2**
10	YOLO-Low-shot [1]	30.0	**62.7**	43.2	**60.6**	39.6	**47.2**	43.2	**13.9**	41.5	58.1	39.2	**39.2**	**20.1**	51.8	55.6	**42.4**	36.6	41.3
	FRCN+joint	14.6	20.3	19.2	24.3	2.2	16.1	17.6	**9.1**	13.8	21.6	0.8	12.6	2.3	43.0	17.4	12.6	1.0	15.3
	FRCN+ft	31.3	36.5	**54.1**	26.5	36.2	36.9	**46.5**	4.5	34.0	57.9	1.1	28.8	15.5	**65.2**	53.6	40.9	41.9	43.4
	FRCN+ft-full	**40.1**	47.8	45.5	47.5	**47.0**	45.6	44.3	3.0	**42.9**	**59.4**	**46.2**	39.1	**19.4**	64.3	**57.3**	40.9	**43.4**	**45.1**
	Meta R-CNN (ours)	**52.5**	**55.9**	**52.7**	**54.6**	**41.6**	**51.5**	**52.8**	3.0	**52.1**	**70.0**	**49.2**	**45.4**	13.9	**72.6**	**58.3**	**47.8**	**47.6**	**48.1**

Table 1. AP and mAP on VOC2007 test set for novel classes and base classes of the first base/novel split. We evaluate the performance for different shots novel-class examples with FRCN under ResNet-101. RED/BLUE indicate the SOTA/the second best. (Best viewd in color)

COCO **Novel-class Split-1**		Box						Mask					
shot	method	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
5	MRCN+ft-full	1.3	3.0	1.1	0.3	1.1	2.4	1.3	2.7	1.1	**0.3**	0.6	2.2
	Meta R-CNN (224x224)	2.4$^{+1.1}$	5.8$^{+2.8}$	**1.5**$^{+0.4}$	0.8$^{+0.5}$	2.5$^{+1.4}$	3.7$^{+1.3}$	2.2$^{+0.9}$	4.9$^{+2.2}$	**1.7**$^{+0.6}$	0.2$^{-0.1}$	1.7$^{+1.1}$	3.6$^{+1.4}$
	Meta R-CNN (600x600)	**3.5**$^{+2.2}$	**9.9**$^{+6.9}$	1.2$^{+0.1}$	**1.2**$^{+0.9}$	**3.9**$^{+2.8}$	**5.8**$^{+3.4}$	**2.8**$^{+1.5}$	**6.9**$^{+4.2}$	**1.7**$^{+0.6}$	**0.3**$^{+0.0}$	**2.3**$^{+1.7}$	**4.7**$^{+2.5}$
10	MRCN+ft-full	2.5	5.7	1.9	**2.0**	2.7	3.9	1.9	4.7	1.3	0.2	1.4	3.2
	Meta R-CNN (224x224)	4.3$^{+1.8}$	9.4$^{+3.7}$	**3.3**$^{+1.4}$	1.3$^{-0.7}$	0.4$^{-2.3}$	6.4$^{+2.5}$	3.7$^{+1.8}$	8.4$^{+3.7}$	2.9$^{+1.6}$	0.3$^{+0.1}$	0.2$^{-1.2}$	5.6$^{+2.4}$
	Meta R-CNN (600x600)	**5.6**$^{+3.1}$	**14.2**$^{+8.5}$	3.0$^{+1.1}$	**2.0**$^{+0.0}$	**6.6**$^{+3.9}$	**8.8**$^{+4.9}$	**4.4**$^{+2.5}$	**10.6**$^{+5.9}$	**3.3**$^{+2.0}$	**0.5**$^{+0.3}$	**3.6**$^{+2.2}$	**7.2**$^{+4.0}$
20	MRCN+ft-full	4.5	9.8	**3.4**	**2.0**	4.6	6.2	3.7	8.5	2.9	0.3	2.5	5.8
	Meta R-CNN (224x224)	**6.2**$^{+1.7}$	**16.6**$^{+6.8}$	2.5$^{-0.9}$	1.7$^{-0.3}$	**6.7**$^{+2.1}$	**9.6**$^{+3.4}$	**6.4**$^{+2.7}$	**14.8**$^{+6.3}$	**4.4**$^{+1.5}$	**0.7**$^{+0.4}$	**4.9**$^{+2.4}$	**9.3**$^{+3.5}$

Table 2. Low-shot detection and instance segmentation performance on COCO minival set for novel classes under Mask R-CNN with ResNet-50. The evaluation based on 5/10/20-shot-object in novel classes.

COCO **Novel-class Split-2**		Box						Mask					
shot	method	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
5	MRCN+ft-full	2.3	4.4	**2.3**	0.6	2.3	3.2	2.1	3.9	**2.0**	0.3	1.8	3.1
	Meta R-CNN (224x224)	**3.3**$^{+1.0}$	**9.4**$^{+5.0}$	1.1$^{-1.2}$	**1.7**$^{+1.1}$	**3.9**$^{+1.6}$	4.4$^{+1.2}$	**2.3**$^{+0.2}$	**5.1** $^{+1.2}$	1.8$^{-0.2}$	**0.4**$^{+0.1}$	**2.2**$^{+0.4}$	**3.8**$^{+0.7}$
	Meta R-CNN (600x600)	3.1$^{+0.8}$	8.9$^{+4.5}$	1.1$^{-1.2}$	1.1$^{+0.6}$	3.0$^{+0.7}$	**5.1**$^{+1.9}$	2.2$^{+0.1}$	4.7$^{+0.8}$	1.9$^{-0.1}$	**0.4**$^{+0.1}$	1.7$^{-0.1}$	3.2$^{+0.1}$
10	MRCN+ft-full	2.6	6.0	**1.8**	1.2	2.7	3.6	2.8	5.7	**2.3**	**0.5**	2.6	4.1
	Meta R-CNN (224x224)	**3.9**$^{+1.3}$	**11.2**$^{+5.2}$	1.4$^{-0.4}$	**1.9**$^{+0.7}$	**4.0**$^{+1.3}$	5.9 $^{+2.3}$	**2.9**$^{+0.1}$	6.3$^{+0.6}$	2.1$^{-0.2}$	**0.5**$^{+0.0}$	**2.8**$^{+0.2}$	**5.0**$^{+0.9}$
	Meta R-CNN (600x600)	**3.9**$^{+1.3}$	11.0$^{+5.0}$	1.7$^{-0.1}$	1.7$^{+0.5}$	3.9$^{+1.2}$	**6.2**$^{+2.6}$	2.8$^{+0.0}$	**6.4**$^{+0.7}$	2.1$^{-0.2}$	**0.5**$^{+0.0}$	2.7$^{+0.1}$	4.5$^{+0.4}$
20	MRCN+ft-full	3.4	8.1	2.3	2.2	3.7	4.9	3.3	7.4	2.3	0.8	3.2	5.5
	Meta R-CNN (ours)	**4.7**$^{+1.3}$	**10.2**$^{+2.1}$	**3.8**$^{+1.5}$	**2.8** $^{+0.6}$	**5.4**$^{+1.7}$	**7.2**$^{+2.3}$	**4.5**$^{+1.2}$	**9.4**$^{+2.0}$	**3.8**$^{+1.5}$	**1.1**$^{+0.3}$	**4.5**$^{+1,3}$	**7.8**$^{+2.3}$

Table 3. Low-shot detection and instance segmentation performance on COCO minival set for novel classes under Mask R-CNN with ResNet-50. The evaluation based on 5/10/20-shot-object in novel classes.

3. Construction ablation of PRN

We additionally test four designs to model a predictor head in different manners: **concate** (Concatenate the class attentive vector and RoI feature for the class-specific prediction), **plus** (elementwise-plus of class attentive feature and RoI feature for the class-specific prediction), **unshare** (The parameters of PRN and R-CNN counterpart are not shared), **limited meta set** (Only use the image-related classes to generate D_{meta}). Results are shown in Table.4. **concate** shows superior in "Base" object detection while **ours** (channel-wise attention) performs better in "Novel" object detection.

Table 4. The ablation of different variations on PRN

shot	Variations	Base	Novel	shot	Variations	Base	Novel
3	concate	**67.0**	33.6	10	concate	**68.4**	50.5
	plus	64.1	32.9		plus	67.9	48.7
	unshare	59.8	21.2		unshare	67.3	40.5
	limited meta set	55.8	33.4		limited meta set	61.4	49.9
	ours	64.8	**35.0**		ours	67.9	**51.5**

4. Low-shot object detection

In Table 1, we conduct the PASCAL VOC experimental results based on low-shot object detection in details. These experiments are based on three different novel / base-class split settings: **Novel-class Split-1** *("bird", "bus", "cow", "mbike", "sofa"/ rest)*; **Novel-class Split-2** *("aero", "bottle","cow","horse","sofa" / rest)* and **Novel-class Split-3** *("boat", "cat", "mbike","sheep", "sofa"/ rest).*

5. Low-shot object segmentation

In Table 2 3, we conduct the COCO experiments based on low-shot object segmentation in two different novel/base-class split settings. In novel-class split-1, the novel class selection follows the classes in PASCAL VOC. In novel-class split-2, we randomly choose ('person','car', 'motorcycle', 'airplane', 'bus', 'train', 'cow','elephant','zebra','tennis racket','bed', 'refrigerator','pizza', 'toilet','microwave','truck','umbrella', 'handbag', 'parking meter', 'teddy bear') as novel classes. In the 5-/10-shot experiment in Split-1, we develop two variants from our Meta R-CNN, *i.e.*, (224x224) and (600x600). They indicate different resolution of the input in meta (reference)-set D_{meta}. Since object segmentation concerns more detailed semantic than object detection, increasing the resolution of reference image can significantly improve the segmentation performance on those objects in the data-starve categories. For a fair comparison with other baselines, the images used for training ($\mathcal{D}_{\text{train}}$) and evaluation ($\mathcal{D}_{\text{test}}$) are consistent in 224x224 across all baselines.

References

[1] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. *arXiv preprint arXiv:1812.01866*, 2018. 2

[2] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 1