# Meta R-CNN : Towards General Solver for Instance-level Low-shot Learning

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

## Abstract

*Resembling the rapid learning capability of human, low-shot learning empowers vision systems to understand new concepts by training with few samples. Leading approaches derived from meta-learning on images with a single visual object. Obfuscated by a complex background and multiple objects in one image, they are hard to promote the research of low-shot object detection/segmentation. In this work, we present a flexible and general methodology to achieve these tasks. Our work extends Faster /Mask R-CNN by proposing meta-learning over RoI (Region-of-Interest) features instead of a full image feature. This simple spirit disentangles multi-object information merged with the background, without bells and whistles, enabling Faster /Mask R-CNN turn into a meta-learner to achieve the tasks. Specifically, we introduce a Predictor-head Remodeling Network (PRN) that shares its main backbone with Faster /Mask R-CNN. PRN receives images containing low-shot objects with their bounding boxes or masks to infer their class attentive vectors. The vectors take channel-wise soft-attention on RoI features, remodeling those R-CNN predictor heads to detect or segment the objects consistent with the classes these vectors represent. In our experiments, Meta R-CNN yields the new state of the art in low-shot object detection and improves low-shot object segmentation by Mask R-CNN. Code:* `https://yanxp.github.io/metarcnn.html`.

## 1. Introduction

Deep learning frameworks dominate the vision community to date, due to their human-level achievements in supervised training regimes with a large amount of data. But distinguished with human that excel in rapidly understanding visual characteristics with few demonstrations, deep neural networks significantly suffer performance drop when training data are scarce in a class. The exposed bottleneck triggers many researches that rethink the generalization of deep learning [46, 11], among which *low(few)-shot learning* [26] is a popular and very promising direction. Provided with very *few labeled data* (1∼10 shots) in *novel classes*, low-shot learners are trained to recognize the data-starve-class objects by the aid of *base classes* with *sufficient labeled data* (See Fig 1.a). Its industrial potential increasingly drives the emergence of solution, falling under the umbrellas of Bayesian approaches [10, 26], similarity learning [25, 36] and meta-learning [40, 42, 41, 37].

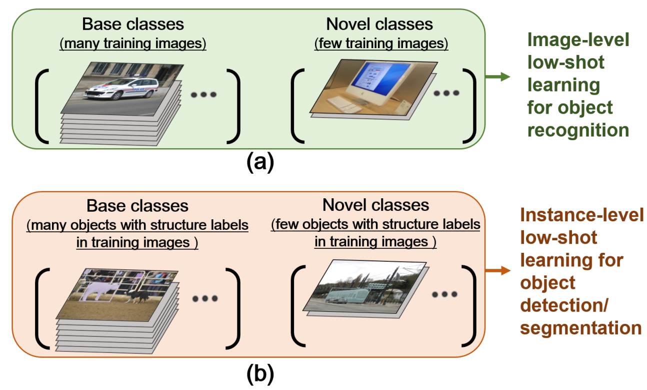

Figure 1. The illustration of labeled training images in low-shot setups for visual object recognition and class-aware object structure (bounding-boxs or masks) prediction. Compared with recognition, novel-class few objects in low-shot object detection/ segmentation blend with other objects in diverse backgrounds, yet requiring a low-shot learner to predict their classes and structure labels.

However, recognizing a single object in an image is solely a tip of the iceberg in real-world visual understanding. In terms of *instance-level learning tasks*, *e.g.*, object detection [35, 33]/ segmentation [2], prior works in low-shot learning contexts remain rarely explored (See Fig 1.b). Since learning the instance-level tasks requires bounding-box or masks (structure labels) consuming more labors than image-level annotations, it would be practically impactful if the novel classes, object bounding boxes and segmentation masks can be synchronously predicted by a low-shot learner. Unfortu-

*indicate equal contribution (Xiaopeng Yan and Ziliang Chen). † indicates corresponding author: Liang Lin. This work was supported in part by the National Key Research and Development Program of China under Grant No. 2018YFC0830103, in part by National High Level Talents Special Support Plan (Ten Thousand Talents Program), and in part by National Natural Science Foundation of China (NSFC) under Grant No. 61622214, 61836012, and 61876224, and in part by the Natural Science Foundation of Guangdong Province under Grant No. 2017A030312006.

nately, these tasks in object-starve conditions become much tougher, as a learner needs to locate or segment the novel-class number-rare objects beside of classifying them. Moreover, due to multiple objects in one image, novel-class objects might blend with the objects in other classes, further obfuscating the information to predict their structure labels. Given this, researchers might expect a complicated solution, as what were done to solve low-shot recognition [10, 26].

Beyond their expectation, we present an intuitive and general methodology to achieve low-shot object detection and segmentation : we propose a novel *meta-learning paradigm based on the RoI (Region-of-Interest) features produced by Faster/Mask R-CNN* [35, 17]. Faster /Mask R-CNN should be trained with considerable labeled objects and unsuited in low-shot object detection. Existing meta-learning techniques are powerful in low-shot recognition, whereas their successes are mostly based on recognizing a single object. Given an image with multi-object information merged in background, they almost fail as the meta-optimization could not disentangle this complex information. But interestingly, we found that the blended undiscovered objects could be "pre-processed" via the RoI features produced by the first-stage inference in Faster /Mask R-CNNs. Each RoI feature refers to a single object or background, so Faster /Mask R-CNN may disentangle the complex information that most meta-learners suffer from.

Our observation motivates the marriage between Faster /Mask R-CNN and meta-learning. Concretely, we extend Faster /Mask R-CNN by introducing a Predictor-head Remodeling Network (PRN). PRN is fully-convoluted and shares the main backbone's parameters with Faster /Mask R-CNN. Distinct from the R-CNN counterpart, PRN receives low-shot objects drawn from base and novel classes with their bboxes or masks, inferring class-attentive vectors, corresponding to the classes that low-shot input objects belong to. Each vector takes channel-wise attention to all RoI features, inducing the detection or segmentation prediction for the classes. To this end, a Faster /Mask R-CNN predictor head has been remodeled to detect or segment the objects that refer to the PRN's inputs, including the category, position, and structure information of low-shot objects. Our framework exactly boils down to a typical meta-learning paradigm, encouraging the name *Meta R-CNN*.

Meta R-CNN is **general** (available in diverse backbones in Faster/Mask R-CNN), **simple** (a lightweight PRN) yet **effective** (a huge performance gain in low-shot object detection/ segmentation) and remains **fast inference** (class-attentive vectors could be pre-processed before testing). We conduct the experiments across 3 benchmarks, 3 backbones for low-shot object detection/ segmentation. Meta R-CNN has achieved the new state of the art in low-shot novel-class object detection/ segmentation, and more importantly, kept competitive performance to detect base-class objects. It verifies Meta R-CNN significantly improve the generalization capability of Faster/ Mask R-CNN.

## 2. Related Work

**Low-shot object recognition** aims to recognize novel visual objects given very few corresponding labeled training examples. Recent studies in vision are mainly classed into three streams based on Bayesian approaches, metric learning and meta-learning, respectively. Bayesian approaches [10, 26] presume a mutual organization rule behind the objects, and design probabilistic model to discover the information among latent variables. Similarity learning [25, 36, 38] tend to consider the same-category examples's features should be more similar than those between different classes. Distinct from them, meta-learning [40, 37, 12, 32, 3, 16, 43, 11] designs to learn a *meta-learner* to *parametrize* the optimization algorithm or *predict* the parameters of a classifier, so-called "learning-to-learn". Recent theories [1, 23] show that meta-learner achieves a generalization guarantee, attracting tremendous studies to solve low-shot problems by meta-learning techniques. However, most existing methods focus on single-object recognition.

**Object detection** based on neural network is mainly resolved by two solver branches: one-stage / two-stage detectors. One-stage detectors attempt to predict bounding boxes and detection confidences of object categories directly, including YOLO [33], SSD [28] and the variants. R-CNN [14] series [18, 13, 35, 8] fall into the second stream. The methods apply covnets to classify and regress the location by the region proposals generated by different algorithms [39, 35]. More recently, low-shot object detection has been extended from recognition [4, 22, 21]. [21] follows full-image meta-learning principle to address this problem. Instead, we discuss the similarity and difference between low-shot object recognition and detection in Sec 3, to reasonably motivate our RoI meta-learning approach.

**Object segmentation** is expected to pixel-wise segment the objects of interest in an image. Leading methods are categorized into image-based and proposal-based. Proposal-based methods [30, 31, 7, 6] predict object masks based on the generated region proposals while image-based methods [47, 48, 44, 2] produce a pixel-level segmentation map over the image to identify object instance. The relevant researches in few-shot setup remain absent.

## 3. Tasks and Motivation

Before introducing Meta R-CNN, we consider low-shot object detection /segmentation tasks it aims to achieve. The tasks could be derived from low-shot object recognition in terms of *meta-learning* methods that motivate our method.

### 3.1. Preliminary: low-shot visual object recognition by meta-learning

In low-shot object recognition, a learner $h(;\boldsymbol{\theta})$ receives training data from *base classes* $C_{\mathrm{base}}$ and *novel classes*

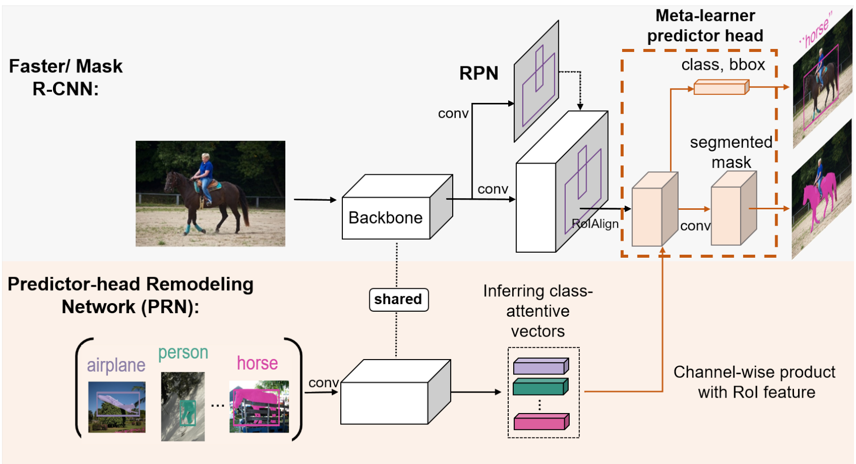


Figure 2. Our Meta R-CNN consists of 1) Faster/ Mask R-CNN; 2) Predictor-head Remodeling Network (PRN). Faster/ Mask R-CNN (module) receives an image to produce RoI features, by taking RoIAlign on the image region proposals extracted by RPN. In parallel, our PRN receives $K$-shot-$m$-class resized images with their structure labels (bounding boxes/segmentaion masks) to infer $m$ class-attentive vectors. Given a class attentive vector representing class $c$, it takes a channel-wise soft-attention on each RoI feature, encouraging the Faster/ Mask R-CNN predictor heads to detect or segment class-$c$ objects based on the RoI features in the image. As the class $c$ is dynamically determined by the inputs of PRN, Meta R-CNN is a meta-learner.

$C_{\text{novel}}$. So the data can be divided into two groups: $D_{\text{base}} = \{(\mathbf{x}_i^{\text{base}}, y_i^{\text{base}})\}_{i=1}^{n_1} \sim P_{\text{base}}$ contains sufficient samples in each base class; $D_{\text{novel}} = \{(\mathbf{x}_i^{\text{novel}}, y_i^{\text{novel}})\}_{i=1}^{n_2} \sim P_{\text{novel}}$ contains very few samples in each novel class. $h(;\boldsymbol{\theta})$ aims to classify test samples drawn from $P_{\text{novel}}$. Notably, training $h(;\boldsymbol{\theta})$ with small dataset $D_{\text{novel}}$ to identify $C_{\text{novel}}$ suffers model overfitting, whereas training $h(;\boldsymbol{\theta})$ with $D_{\text{base}} \cup D_{\text{novel}}$ still fails, due to the extreme data quantity imbalance between $D_{\text{base}}$ and $D_{\text{novel}}$ ($n_2 << n_1$).

Recent wisdoms tend to address this problem by recasting it into a meta-learning paradigm [40, 41] to encourage a fast model adaptation (generalization) to novel tasks, *e.g.*, classifying objects in $C_{\text{novel}}$. In each iteration, the meta-learning paradigm draws a subset of classes $C_{\text{meta}} \sim C_{\text{base}} \cup C_{\text{novel}}$ and thus, uses the images belonging to $C_{\text{meta}}$ to construct two batches: a training mini-batch $D_{\text{train}}$ and a small-size *meta(reference)-set* $D_{\text{meta}}$ (low-shot samples in each class). Given this, a *meta-learner* $h(\mathbf{x}_i, D_{\text{meta}}; \boldsymbol{\theta})$ simultaneously receives an image $\mathbf{x}_i \sim D_{\text{train}}$ and the entire $D_{\text{meta}}$ and then, is trained to classify $D_{\text{train}}$ into $C_{\text{meta}}$[1]. By replacing $D_{\text{meta}}$ with $D_{\text{novel}}$, recent theories [1, 23] present generalization bounds to the meta-learner, enabling $h(, D_{\text{novel}}; \boldsymbol{\theta})$ to correctly recognize the objects $\sim P_{\text{novel}}$.

### 3.2. Low-shot object detection / segmentation

From visual recognition to detection /segmentation, low-shot learning on objects becomes more complex: An image $\mathbf{x}_i$ might contain $n_i$ objects $\{\mathbf{z}_{i,j}\}_{j=1}^{n_i}$ in diverse classes, positions and shapes. Therefore the low-shot learners need to identify novel-class objects $\mathbf{z}_{i,j}^{\text{novel}}$ from other objects and background, and then, predict their classes $y_{i,j}^{\text{novel}}$ and structure labels $s_{i,j}^{\text{novel}}$ (bounding-boxes or masks). Most existing detection/ segmentation baselines address the problems by modeling $h(\mathbf{x}_i; \boldsymbol{\theta})$, performing poorly in a low-shot scenario. However, meta-predictor $h(\mathbf{x}_i, D_{\text{meta}}; \boldsymbol{\theta})$ is also unsuitable, since $\mathbf{x}_i$ contains multi-object complex information merged in diverse backgrounds.

**Motivation.** The real goal of meta-learning for low-shot object detection/ segmentation is to model $h(\mathbf{z}_{i,j}, D_{\text{meta}}; \boldsymbol{\theta})$ rather than $h(\mathbf{x}_i, D_{\text{meta}}; \boldsymbol{\theta})$. Since visual objects $\{\mathbf{z}_{i,j}\}_{j=1}^{n_i}$ are blended with each other and merge with the background in $\mathbf{x}_i$, meta-learning with $\{\mathbf{z}_{i,j}\}_{j=1}^{n_i}$ is prevented. Howbeit in two-stage detection models, *e.g.*, Faster/ Mask R-CNNs, multi-object and their background information can be disentangled into RoI (Region-of-Interest) features $\{\hat{\mathbf{z}}_{i,j}\}_{j=1}^{\hat{n}_i}$, which are produced by taking RoIAlign on the image region proposals extracted by the region proposal network (RPN). These RoI features are fed into the second-stage predictor head to achieve RoI-based object classification, position location and silhouette segmentation for $\{\mathbf{z}_{i,j}\}_{j=1}^{n_i}$. Given this, it is preferable to remodel the R-CNN predictor head into $h(\hat{\mathbf{z}}_{i,j}, D_{\text{meta}}; \boldsymbol{\theta})$ to classify, locate and segment the object $\mathbf{z}_{i,j}$ behind each region of interest (RoI) feature $\hat{\mathbf{z}}_{i,j}$.

## 4. Meta R-CNN

Aiming at meta-learning over regions of interest (RoIs), Meta R-CNN is conceptually simple: its pipeline consists of 1). Faster/ Mask R-CNN; 2). *Predictor-head Remodeling Network* (**PRN**). Faster/ Mask R-CNN produces object proposals $\{\hat{\mathbf{z}}_{i,j}\}_{j=1}^{n_i}$ by their region proposal networks (**RPN**). Then each $\hat{\mathbf{z}}_{i,j}$ combines with *class-attentive vectors* inferred by our **PRN**, which plays the role of $h(, D_{\text{meta}}; \boldsymbol{\theta})$ to detect or segment the novel-class objects. The Meta R-CNN framework is illustrated in Fig 2 and we elaborate it by starting from Faster/ Mask R-CNN.

### 4.1. Review the R-CNN family

Faster R-CNN system is known as a two-stage pipeline. The first stage is a region proposal network (**RPN**), receiv-

[1] In a normal setup, meta-learning includes two phases, *meta-train* and *meta-test*. The first phase only use a subset of $C_{\text{base}}$ to train a meta-learner.

ing an image $\mathbf{x}_i$ to produce the candidate object bounding-boxes (so-called object region proposals) in this image. The second stage, *i.e.*, Fast R-CNN [13], shares the RPN backbone to extract RoI (Region-of-Interest) features $\{\hat{\mathbf{z}}_{i,j}\}_{j=1}^{\hat{n}_i}$ from $\hat{n}_i$ object region proposals after RoIAlign [2], enabling its predictor head $h(\hat{\mathbf{z}}_{i,j}, \boldsymbol{\theta})$ to classify and locate the object $\mathbf{z}_{i,j}$ behind the RoI feature $\hat{\mathbf{z}}_{i,j}$ of $\mathbf{x}_i$. Mask R-CNN activates the segmentation ability in Faster R-CNN by adding a parallel mask branch in the predictor head $h(\cdot, \boldsymbol{\theta})$. Due to our identical technique applied in Faster/ Mask R-CNN, we unify their predictor heads by $h(\cdot, \boldsymbol{\theta})$.

As previously discussed, predictor head $h(\cdot, \boldsymbol{\theta})$ in Faster/ Mask R-CNN is inappropriate to make low-shot object detection/ segmentation. To this we propose PRN that remodels $h(\cdot, \boldsymbol{\theta})$ into a meta-predictor head $h(\cdot, D_{\mathrm{meta}}; \boldsymbol{\theta})$.

### 4.2. Predictor-head Remodeling Network (PRN)

A straightforward approach to design $h(\cdot, D_{\mathrm{meta}}; \boldsymbol{\theta})$ is to learn $\boldsymbol{\theta}$ to predict the optimal parameter *w.r.t.* an arbitrary meta-set $D_{\mathrm{meta}}$ like [42]. Such explicit "learning-to-learn" manner is sensitive to the architectures and $h(\cdot, \boldsymbol{\theta})$ in Faster/ Mask R-CNN is abandoned. Instead, our work is inspired by the concise spirit of SNAIL [29], thus, incorporating class-specific soft-attention vectors to achieve channel-wise feature selection on each RoI feature in $\{\mathbf{z}_{i,j}\}_{j=1}^{\hat{n}_i}$ [5]. This soft-attention mechanism is implemented by the class-attentive vectors $\mathbf{v}^{\mathrm{meta}}$ inferred from the objects in a meta-set $D_{\mathrm{meta}}$ via PRN. In particular, suppose that PRN denotes as $\mathbf{v}^{\mathrm{meta}} = f(D_{\mathrm{meta}}; \boldsymbol{\phi})$, given each RoI feature $\hat{\mathbf{z}}_{i,j}$ that belongs to image $\mathbf{x}_i$, it holds

$$\begin{aligned} h(\hat{\mathbf{z}}_{i,j}, D_{\mathrm{meta}}; \boldsymbol{\theta}') &= h(\hat{\mathbf{z}}_{i,j} \otimes \mathbf{v}^{\mathrm{meta}}, \boldsymbol{\theta}) \\ &= h(\hat{\mathbf{z}}_{i,j} \otimes f(D_{\mathrm{meta}}; \boldsymbol{\phi}), \boldsymbol{\theta}) \end{aligned} \tag{1}$$

where $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ denote the parameters of Faster/Mask R-CNN and our PRN (most of them are shared, $\boldsymbol{\theta}' = \{\boldsymbol{\theta}, \boldsymbol{\phi}\}$); $\otimes$ indicates the channel-wise multiplication operator. Eq 1 implies that PRN remodels $h(\cdot, \boldsymbol{\theta})$ into $h(\cdot, D_{\mathrm{meta}}; \boldsymbol{\theta})$ in principles. It is intuitive, flexibly-applied and allows end-to-end joint training with its Faster/ Mask R-CNN counterpart.

Suppose $\mathbf{x}_i$ as the image Meta R-CNN aiming to detect. After RoIAlign in its R-CNN module, it turns to a set of RoI features $\{\hat{\mathbf{z}}\}_{i,j}^{\hat{n}_i}$. Here we explain how PRN acts on them.

**Infer class-attentive vectors.** As can be observed, PRN $f(D_{\mathrm{meta}}; \boldsymbol{\phi})$ receives all objects in meta-set $D_{\mathrm{meta}}$ as input. In the context of object detection/ segmentation, $D_{\mathrm{meta}}$ denotes a series of objects distributed across images, whose classes belong to $C_{\mathrm{meta}}$ and there exist $K$ objects per class ($K$-shot setup). Each object in $D_{\mathrm{meta}}$ presents a 4-channel input, *i.e.*, an RGB image $\mathbf{x}$ with the same-spatial-size foreground structure label $s$ that are combined to represent this object ($s$ is a binary mask derived from the object bounding-box or segmentation mask). Hence given $m$ as the size of $C_{\mathrm{meta}}$, PRN receives $mK$ 4-channel object inputs in each inference process. To ease the computation burden, we standardize the spatial size of object inputs into 224×224. During inference, after passing the first convolution layer of our PRN, each object feature would be fed into the second layer of its R-CNN counterpart, undergoing the shared backbone before RoIAlign. Instead of accepting RoIAlign, the feature passes a channel-wise soft-attention layer to produce its *object attentive vector* $\mathbf{v}$. To this end, PRN encodes $mK$ objects in $D_{\mathrm{meta}}$ into $mK$ object attentive vectors and then, applies average pooling to obtain the class-attentive vectors $\mathbf{v}_c^{\mathrm{meta}}$, *i.e.*, $\mathbf{v}_c^{\mathrm{meta}} = \frac{1}{K}\sum_{j=1}^{K} \mathbf{v}_k^{(c)}$, ($\forall c \in C_{\mathrm{meta}}$, $\mathbf{v}_k^{(c)}$ represents an object attentive vector inferred from a class-$c$ object and there are $K$ objects per class).

**Remodel R-CNN predictor heads.** After obtaining the class-attentive vectors $\mathbf{v}_c^{\mathrm{meta}}$ ($\forall c \in C_{\mathrm{meta}}$), PRN applies them to make channel-wise soft-attention on each RoI feature $\mathbf{z}_{i,j}$. Suppose that $\hat{\mathbf{Z}}_i = [\hat{\mathbf{z}}_{i,1}; \cdots; \hat{\mathbf{z}}_{i,128}] \in \mathbb{R}^{2048\times128}$ denotes the RoI feature matrix generated from $\mathbf{x}_i$ (128 denotes the number of RoI). PRN replaces $\hat{\mathbf{Z}}_i$ by $\hat{\mathbf{Z}}_i \otimes \mathbf{v}_c^{\mathrm{meta}} = [\hat{\mathbf{z}}_{i,1} \otimes \mathbf{v}_c^{\mathrm{meta}}; \cdots; \hat{\mathbf{z}}_{i,128} \otimes \mathbf{v}_c^{\mathrm{meta}}]$ to feed the primitive predictor heads in Faster /Mask R-CNNs. The refinement leads to detecting or segmenting all class-$c$ objects in the image $\mathbf{x}_i$. In this spirit, each RoI feature $\hat{\mathbf{z}}_{i,j}$ produces $m$ binary detection outcomes that refers to the classes in $C_{\mathrm{meta}}$. To this Meta R-CNN categorizes $\hat{\mathbf{z}}_{i,j}$ into the class $c^*$ with the highest confidence score and use the branch $\hat{\mathbf{z}}_{i,j} \otimes \mathbf{v}_{c^*}^{\mathrm{meta}}$ to locate or segment the object. But if the highest confidence score is lower than the objectness threshold, this RoI would be treated as background and discarded.

## 5. Implementation

Meta R-CNN is trained under a meta-learning paradigm. Our implementation based on Faster/ Mask R-CNN, whose hyper-parameters follow their original report.

**Mini-batch construction.** Simulating the meta-learning paradigm we have discussed, a training mini-batch in Meta R-CNN is comprised of $m$ classes $C_{meta} \sim C_{\mathrm{base}} \cup C_{\mathrm{novel}}$, a $K$-shot $m$-class meta-set $D_{\mathrm{meta}}$ and $m$-class training set $D_{\mathrm{train}}$ (classes in $D_{\mathrm{meta}}$, $D_{\mathrm{train}}$ consistent with $C_{meta}$). In our implementation, $D_{\mathrm{train}}$ represent the objects in the input $\mathbf{x}$ of Faster/ Mask R-CNNs. To keep the class consistency, we choose $C_{meta}$ as the object classes image $\mathbf{x}$ refers to, and only uses the attentive vectors inferred from the objects belonging to the classes in $C_{\mathrm{meta}}$. Therefore, if the R-CNN module receives an image input $\mathbf{x}$ that contains objects in $m$ classes, a mini-batch consists of $\mathbf{x}$ ($D_{\mathrm{train}}$) and $mK$ resized images with their structure label masks.

**Channel-wise soft-attention layer.** This layer receives the features induced from the main backbone of the R-CNN counterpart. It performs a spatial pooling to align the object features maintaining the identical size of RoI features. Then

[2]RoIAlign operation is first introduced by Mask R-CNN yet can be used by Faster R-CNN. Faster R-CNNs in our work are based on RoIAlign.

the features undergo an element-wise sigmoid to produce attentive vectors (the size is 2048×1 in our experiment).

**Meta-loss.** Given an RoI feature $\hat{\mathbf{z}}_{i,j}$, to avoid the prediction ambiguity after soft-attention, attentive vectors from different-class objects should lead to diverse feature selection effects on $\hat{\mathbf{z}}_{i,j}$. To this we propose a simple *meta-loss* $L(\phi)_{\mathrm{meta}}$ to diversify the inferred object attentive vectors in meta-learning. It is implemented by a cross-entropy loss encouraging the object attentive vectors to fall in the class each object belongs to. This auxiliary loss powerfully boosts Meta R-CNN performance (see Table 6 Ablation 2).

**RoI meta-learning.** Following the typical optimization routines in [40, 37, 41], meta-learning Meta R-CNN is divided into two phases. In the first phase (so-called *meta-train*), we solely consider base-class objects to construct $D_{\mathrm{meta}}$ and $D_{\mathrm{train}}$ in per iter. In the case that an image simultaneously includes base-class and novel-class objects, we ignore the novel-class objects in meta-train. In the second phase (so-called *meta-test*), objects in base and novel classes are both considered. The objective is formulated as

$$\min_{\boldsymbol{\theta},\,\boldsymbol{\phi}} \underbrace{L(\boldsymbol{\theta},\boldsymbol{\phi})_{\mathrm{cls}} + L(\boldsymbol{\theta},\boldsymbol{\phi})_{\mathrm{reg}} + \lambda L(\boldsymbol{\theta},\boldsymbol{\phi})_{\mathrm{mask}}}_{\text{Losses derived from Faster/Mask R-CNN}} + L(\boldsymbol{\phi})_{\mathrm{meta}} \tag{2}$$

where $\lambda = \{0, 1\}$ indicates the activator of mask branch. The illustration of meta-learning for Meta R-CNN is below:

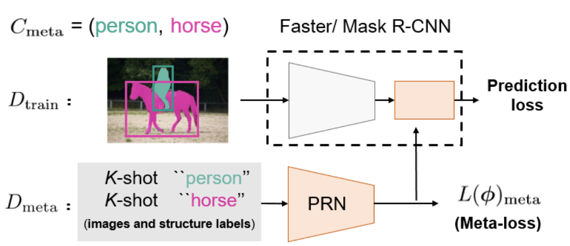


Figure 3. The illustrative instance of RoI meta-learning process in Meta R-CNN. Suppose the image Faster/ Mask R-CNN receiving contains objects in "person", "horse". Then $C_{\mathrm{meta}}$ = ( "person", "horse" ) and $D_{\mathrm{meta}}$ includes $K$-shot "person" and "horse" images with their structure labels, respectively. As the training image iteratively changes, $C_{\mathrm{meta}}$ and $D_{\mathrm{meta}}$ would adaptively change.

**Inference.** Meta R-CNN entails two inference processes based on Faster/Mask R-CNN module and PRN. In training, the object attentive vectors inferred from $D_{\mathrm{meta}}$ would replace the class-attentive vectors to take soft-attention effects on $\hat{\mathbf{Z}}_i$ and produce the object detection/ segmentation losses in Eq 2. In testing, we choose $C_{\mathrm{meta}} = C_{\mathrm{base}} \cup C_{\mathrm{novel}}$. It is because that unknown objects in a test image may cover all possible categories. PRN receives $K$-shot visual objects in all classes to produce class-attentive vectors to achieve low-shot object detection/ segmentation. Note that, no matter of object or class attentive vectors, they can be pre-processed before testing, and parallelly take soft-attention on RoI feature matrices. It promises the fast inference of Faster/ Mask R-CNN will not be decelerated: In our experiment (using a single GTX TITAN XP), if shot is 3, the inference speed of Faster R-CNN is 83.0 ms/im; Meta R-CNN is 84.2ms/im; if shot is 10, the speed of Meta R-CNN is 85.4ms/im.

## 6. Experiments

In this section, we propose thorough experiments to evaluate Meta R-CNN on low-shot object detection, the related ablation, and low-shot object segmentation.

### 6.1. Low-shot object detection

In low-shot object detection, we employ a Faster R-CNN [35] with ResNet-101 [17] backbone as the R-CNN module in our Meta R-CNN framework.

**Benchmarks and setups.** Our low-shot object detection experiment follows the setup [21]. Concretely, we evaluate all baselines on the generic object detection tracks of PASCAL VOC 2007 [9], 2012 [9], and MS-COCO [27] benchmarks. We adopt the PASCAL Challenge protocol that a correct prediction should have more than 0.5 IoU with the ground truth and set the evaluation metric to the mean Average Precision (mAP). Among these benchmarks, VOC 2007 and 2012 consists of images covering 20 object categories for training, validation and testing sets. To create a low-shot learning setup, we consider three different novel/base-class split settings, *i.e.*, ("bird", "bus", "cow", "mbike", "sofa"/ rest); ("aero", "bottle","cow","horse","sofa" / rest) and ("boat", "cat", "mbike","sheep", "sofa"/ rest). During the first phase of meta-learning, only base-class objects are considered. In the second phase, there are $K$-shot annotated bounding boxes for objects in each novel class and $3K$ annotated bounding boxes for objects in each base class for training, where $K$ is set 1, 2, 3, 5 and 10. We also evaluate our method on COCO benchmark with 80 object categories including the 20 categories in PASCAL VOC. In this experiment, we set the 20 classes included in PASCAL VOC as the novel classes, then the rest 60 classes in COCO as base classes. The union of 80k train images and a 35k subset of validation images (trainval35k) are used for training, and our evaluation is based on the remaining 5k val images (minival). Finally, we consider the cross-benchmark transfer setup of low-shot object detection from *COCO* to *PASCAL* [21], which leverages 60 base classes of COCO to learn knowledge representations and the evaluation is based on 20 novel classes of PASCAL VOC.

**Baselines.** In methodology, Meta R-CNN can be treated as the meta-learning extension of Faster R-CNN (FRCN) [35] in the background of low-shot object detection. To this a question about detector generalization is probably raised:

***Does Meta R-CNN help to improve the generalization capability of Faster R-CNN?***

To answer this question, we compare our Meta R-CNN with its base FRCN. This detector is derived into three base-

Table 1. Low-shot detection mAP on VOC2007 test set in novel classes. We evaluate the baselines under three different splits of novel classes. RED and BLUE indicate state-of-the-art (SOTA) and the second best. (Best viewd in color)

| | Novel-class Setup 1 | | | | | Novel-class Setup 2 | | | | | Novel-class Setup 3 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method/Shot | 1 | 2 | 3 | 5 | 10 | 1 | 2 | 3 | 5 | 10 | 1 | 2 | 3 | 5 | 10 |
| YOLO-Low-shot [21] | **14.8** | **15.5** | 26.7 | 33.9 | **47.2** | **15.7** | **15.3** | 22.7 | 30.1 | **39.2** | **19.2** | **21.7** | **25.7** | **40.6** | 41.3 |
| FRCN+joint | 2.7 | 3.1 | 4.3 | 11.8 | 29.0 | 1.9 | 2.6 | 8.1 | 9.9 | 12.6 | 5.2 | 7.5 | 6.4 | 6.4 | 6.4 |
| FRCN+ft | 11.9 | 16.4 | 29.0 | 36.9 | 36.9 | 5.9 | 8.5 | 23.4 | 29.1 | 28.8 | 5.0 | 9.6 | 18.1 | 30.8 | 43.4 |
| FRCN+ft-full | 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** |
| Meta R-CNN (ours) | **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** |

Table 2. The ablation study of backbones (mAP on VOC2007 testset in novel classes and base classes of the first base/novel split based on FRCN).

| Shot | Baselines | Base | Novel |
|---|---|---|---|
| 3 | ResNet-34+ft-full | **57.9** | 19.6 |
| | ResNet-34+Ours | 57.6 | **25.3** |
| | ResNet-101+ft-full | 63.6 | 32.8 |
| | ResNet-101+Ours | **64.8** | **35.0** |
| 10 | ResNet-34+ft-full | 61.1 | 40.2 |
| | ResNet-34+Ours | **61.3** | **44.5** |
| | ResNet-101+ft-full | 61.3 | 45.6 |
| | ResNet-101+Ours | **67.9** | **51.5** |

Table 3. AP and mAP on VOC2007 test set for novel classes and base classes of the first base/novel split. We evaluate the performance for 3/10-shot novel-class examples with FRCN under ResNet-101. RED/BLUE indicate the SOTA/the second best. (Best viewd in color)

| | | Novel classes | | | | | | Base classes | | | | | | | | | | | | | | | | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Shot | Baselines | bird | bus | cow | mbike | sofa | mean | aero | bike | boat | bottle | car | cat | chair | table | dog | horse | person | plant | sheep | train | tv | mean | |
| 3 | YOLO-Low-shot [21] | 26.1 | 19.1 | 40.7 | 20.4 | **27.1** | 26.7 | **73.6** | **73.1** | 56.7 | 41.6 | **76.1** | 78.7 | 42.6 | **66.8** | 72.0 | 77.7 | 68.5 | **42.0** | 57.1 | **74.7** | 70.7 | **64.8** | 55.2 |
| | FRCN+joint | 13.7 | 0.4 | 6.4 | 0.8 | 0.2 | 4.3 | **75.9** | **80.0** | **65.9** | **61.3** | **85.5** | **86.1** | **54.1** | **68.4** | **83.3** | **79.1** | **78.8** | **43.7** | **72.8** | **80.8** | **74.7** | **72.7** | 55.6 |
| | FRCN+ft | **31.1** | 24.9 | **51.7** | 23.5 | 13.6 | 29.0 | 65.4 | 56.4 | 46.5 | 41.5 | 73.3 | 84.0 | 40.2 | 55.9 | 72.1 | 75.6 | 74.8 | 32.7 | 60.4 | 71.2 | **71.2** | 61.4 | 53.3 |
| | FRCN+ft-full | 29.1 | **34.1** | **55.9** | **28.6** | **16.1** | **32.8** | 67.4 | 62.0 | 54.3 | 48.5 | 74.0 | **85.8** | 42.2 | 58.1 | 72.0 | **77.8** | 75.8 | 32.3 | **61.0** | 73.7 | 68.6 | 63.6 | **55.9** |
| | Meta R-CNN (ours) | **30.1** | **44.6** | 50.8 | **38.8** | 10.7 | **35.0** | 67.6 | 70.5 | **59.8** | **50.0** | 75.7 | 81.4 | **44.9** | 57.7 | **76.3** | 74.9 | **76.9** | 34.7 | 58.7 | **74.7** | 67.8 | **64.8** | **57.3** |
| 10 | YOLO-Low-shot [21] | 30.0 | **62.7** | 43.2 | **60.6** | 39.6 | **47.2** | 65.3 | 73.5 | 54.7 | 39.5 | 75.7 | 81.1 | 35.3 | 62.5 | 72.8 | 78.8 | 68.6 | **41.5** | 59.2 | **76.2** | 69.2 | 63.6 | **59.5** |
| | FRCN+joint | 14.6 | 20.3 | 19.2 | 24.3 | 2.2 | 16.1 | **78.1** | **80.0** | **65.9** | **64.1** | **86.0** | **87.1** | **56.9** | **69.7** | **84.1** | **80.0** | **78.4** | **44.8** | **74.6** | **82.7** | **74.1** | **73.8** | 59.4 |
| | FRCN+ft | 31.3 | 36.5 | **54.1** | 26.5 | 36.2 | 36.9 | **68.4** | **75.2** | 59.2 | **54.8** | 74.1 | 80.8 | 42.8 | 56.0 | 68.9 | 77.8 | 75.5 | 34.7 | **66.1** | 71.2 | 66.2 | 64.8 | 57.8 |
| | FRCN+ft-full | **40.1** | 47.8 | 45.5 | 47.5 | **47.0** | 45.6 | 65.7 | 69.2 | 52.6 | 46.5 | 74.6 | 73.6 | 40.7 | 55.0 | 69.3 | 73.5 | 73.2 | 33.8 | 56.5 | 69.8 | 65.1 | 61.3 | 57.4 |
| | Meta R-CNN (ours) | **52.5** | **55.9** | **52.7** | **54.6** | **41.6** | **51.5** | 68.1 | 73.9 | **59.8** | 54.2 | **80.1** | **82.9** | **48.8** | **62.8** | **80.1** | **81.4** | **77.2** | 37.2 | 65.7 | 75.8 | **70.6** | **67.9** | **63.8** |

lines according to the different training strategies they use. Specifically, **FRCN+joint** is to jointly train the FRCN detector with base-class and novel-class objects. The identical number of iteration is used for training this baseline and our Meta R-CNN. **FRCN+ft** takes a similar two-phase training strategy in Meta R-CNN: it only uses base-class objects (with bounding boxes) to train FRCN in the first phase, then use the combination of base-class and novel-class objects to fine-tune the network. For a fair comparison, the objects in images used to train FRCN+ft is identical to Meta R-CNN, and FRCN+ft also takes the same number of iteration (in both training phases) of Meta R-CNN. Finally, **FRCN+ft-full** employ the same training strategy of FRCN+ft in the first phase, yet train the detector to fully converge in the second phase. Beyond these baselines, Meta R-CNN is also compared with the state-of-the-art low-shot object detector [21] modified from YOLOv2 [34] (**YOLO-Low-shot**). Note that, YOLO-Low-shot also employs meta-learning, whereas distinct from Meta R-CNN based on RoI features, it is based on a full image. Their comparison reveals whether the motivation of Meta R-CNN is reasonable.

**PASCAL VOC.** The experimental evaluation are shown in Table 1. The $K$-shot object detection is performed based on $K = (1, 2, 3, 5, 10)$ across three novel/base class splits. As can be observed, Meta R-CNN consistently outperforms the three FRCN baselines by a large margin across splits. It uncovers the generalization weakness of FRCN: without adequate number of bounding-box annotations, FRCN performs poorly to detect novel-class objects, and this weakness could not be overcome by changing the training strategies. In a comparison, by simply deploying a lightweight PRN, FRCN turns into Meta R-CNN and significantly improve the performance on novel-class object detection. It implies that our approach endows FRCN with the generalization ability in low-shot learning.

Besides, Meta R-CNN outperforms YOLO-Low-shot in the majority of the cases (except for 1/2-shot in the third split). Since the YOLO-Low-shot results are borrowed from their report, the 1/2-shot objects are probably different from what we use. Extremely-low-shot setups are sensitive to the change of the low-shot object selection and thus, hard to reveal the superiority of low-shot learning algorithms. In the more robust 5/10-shot setups, Meta R-CNN significantly exceeds YOLO-Low-shot ($+11.8\%$ in the 5-shot of the first split; $+6.8$ in the 10-shot of the third split.)

Let's consider detailed evaluation in Table 3 based on the first base/novel-class split. Note that, FRCN+joint achieved SOTA in base classes, however, at the price of the performance disaster in novel classes (72.7 in base classes yet 4.3 in novel classes given $K$=3). This sharp contrast caused by the extreme object quantity imbalance in the low-shot setup, further reveal the fragility of FRCN in the generalization problem. On the other hand, we find that Meta R-CNN outperforms YOLO-Low-shot both in base classes and novel

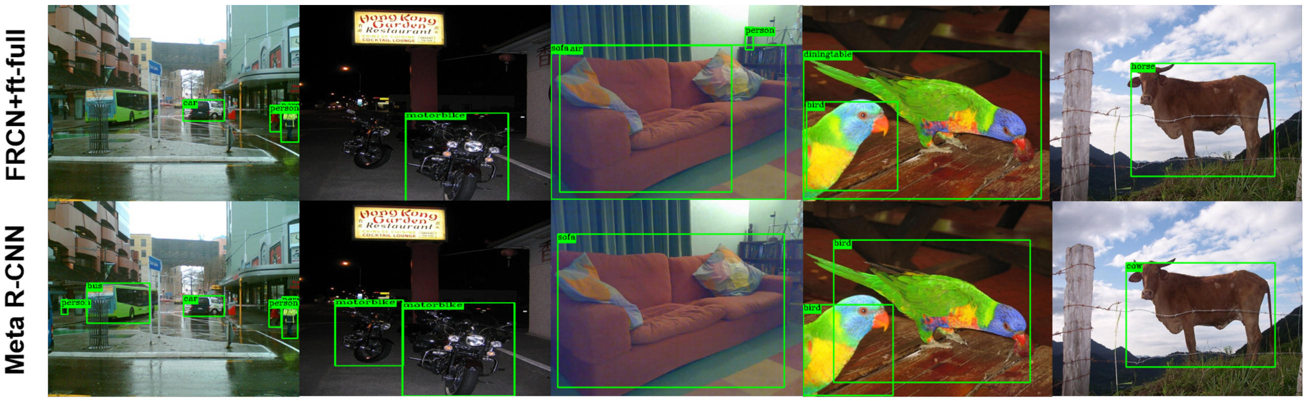


Figure 4. The visualization of novel-class objects detected by FRCN+ft-full and Meta R-CNN. Compared with Meta R-CNN, FRCN+ft-full is inferior: bboxes in the first two columns are missed; in the middle column is duplicate and the classes are wrong in the last two columns.

Table 4. Low-shot detection performance on COCO minival set for novel classes. We evaluate the performance for different shot examples of novel classes under FRCN pipeline with ResNet-50. RED/BLUE indicate the SOTA/the second best. (Best viewd in color)

| Shot | Baselines | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ | $AR_1$ | $AR_{10}$ | $AR_{100}$ | $AR_S$ | $AR_M$ | $AR_L$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | YOLO-Low-shot [21] | 5.6 | 12.3 | 4.6 | 0.9 | 3.5 | 10.5 | 10.1 | 14.3 | 14.4 | 1.5 | 8.4 | **28.2** |
| | FRCN+ft | 1.3 | 4.2 | 0.4 | 0.4 | 0.9 | 2.1 | 5.5 | 8.0 | 8.0 | 2.4 | 6.4 | 13.0 |
| | FRCN+ft-full | **6.5** | **13.4** | **5.9** | **1.8** | **5.3** | **11.3** | **12.6** | **17.7** | **17.8** | **6.5** | **14.4** | **28.6** |
| | Meta R-CNN (ours) | **$8.7^{+2.2}$** | **$19.1^{+5.7}$** | **$6.6^{+0.7}$** | **$2.3^{+0.5}$** | **$7.7^{+2.4}$** | **$14.0^{+2.7}$** | **$12.6^{+0}$** | **$17.8^{+0.1}$** | **$17.9^{+0.1}$** | **$7.8^{+1.3}$** | **$15.6^{+1.2}$** | $27.2^{-1.4}$ |
| 30 | YOLO-Low-shot [21] | 9.1 | 19.0 | 7.6 | 0.8 | 4.9 | 16.8 | 13.2 | 17.7 | 17.8 | 1.5 | 10.4 | **33.5** |
| | FRCN+ft | 1.5 | 4.8 | 0.5 | 0.3 | 1.8 | 2.0 | 7.0 | 10.1 | 10.1 | 5.8 | 8.3 | 13.5 |
| | FRCN+ft-full | **11.1** | **21.6** | **10.3** | **2.9** | **8.8** | **18.9** | **15.0** | **21.1** | **21.3** | **10.1** | **17.9** | **33.2** |
| | Meta R-CNN (ours) | **$12.4^{+1.3}$** | **$25.3^{+4.3}$** | **$10.8^{+0.5}$** | **$2.8^{-0.1}$** | **$11.6^{+2.8}$** | **$19.0^{+1.0}$** | **$15.0^{+0}$** | **$21.4^{+0.3}$** | **$21.7^{+0.4}$** | **$8.6^{-1.5}$** | **$20.0^{+2.1}$** | $32.1^{-1.4}$ |

Table 5. The ablation of image-level and RoI-level meta-learning

| shot | Ablation | Base | Novel |
|---|---|---|---|
| 3 | full-image meta-learning | 43.4 | 8.1 |
| | RoI meta-learning | **64.8** | **35.0** |
| 10 | full-image meta-learning | 61.2 | 32.0 |
| | RoI meta-learning | **67.9** | **51.5** |

Table 6. Ablation studies of (1) ***meta-learning*** and (2) ***meta-loss*** (mAP on VOC2007 test set for novel classes and base classes of the first base/novel split under FRCN pipeline with ResNet-101) .

| shot | Ablation (1) | Base | Novel | Ablation (2) | Base | Novel |
|---|---|---|---|---|---|---|
| 3 | meta-learning (w/o) | 38.5 | 9.0 | meta-loss (w/o) | 24.2 | 57.7 |
| | meta-learning (w) | **64.8** | **35.0** | meta-loss (w) | **35.0** | **64.8** |
| 10 | meta-learning (w/o) | 56.9 | 40.5 | meta-loss (w/o) | 46.6 | 64.3 |
| | meta-learning (w) | **67.9** | **51.5** | meta-loss (w) | **51.5** | **67.9** |

classes, which means that Meta R-CNN is the SOTA low-shot detector. Finally, Meta R-CNN outperforms all other baselines in mAP. This observation is significant: **Meta R-CNN would not sacrifice the overall performance to make low-shot learning.** In Fig 4, we visualize some comparison between FRCN+ft+full and Meta R-CNN on detecting novel-class objects.

**MS COCO.** We evaluate 10-shot /30-shot setups on MS COCO [27] benchmark and report the standard COCO metrics. The results on novel classes are presented in Table 4. It shows that Meta R-CNN significantly outperforms other baselines and YOLO-Low-shot. Note that, the performance gain is obtained by our method compared to YOLO-Low-shot (12.4% vs. 11.1%). The improvement is lower than those on PASCAL VOC, since MS COCO is more challenging with more complex scenarios such as occlusion, ambiguities and small objects.

**MS COCO to PASCAL.** In this cross-dataset low-shot object detection setup, all the baselines are trained with 10-shot objects in novel classes on MS COCO while they are evaluated on PASCAL VOC2007 test set. Distinct from the previous experiments that focus on evaluating cross-category model generalization, this setup further to reveal the cross-domain generalization ability. FRCN+ft and FRCN+ft-full get the detection performances of 19.2% and 31.2% respectively. The low-shot object detector YOLO-Low-shot obtains 32.3%. Instead, Meta R-CNN achieves 37.4%, reaping a significant performance gain (approximately 5% mAP) against the second best.

## 6.2. Ablation

Here we conduct comprehensive ablation studies to uncover Meta R-CNN. These ablations are based on 3/10-shot object detection performances on PASCAL VOC in the first base/novel split setup.

**Backbone.** We ablate the backbone (i.e. ResNet-34 [17] and ResNet-101 [17]) of Meta R-CNN to observe the object detection performances in base and novel classes (Table 2). It's observed that our framework significantly outperforms the FRCN-ft-full on base and novel classes across

Table 7. 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 (More comprehensive results see our supplementary material).

| | | 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 (ours) | $\mathbf{3.5}^{+2.2}$ | $\mathbf{9.9}^{+6.9}$ | $\mathbf{1.2}^{+0.1}$ | $\mathbf{1.2}^{+0.9}$ | $\mathbf{3.9}^{+2.8}$ | $\mathbf{5.8}^{+3.4}$ | $\mathbf{2.8}^{+1.5}$ | $\mathbf{6.9}^{+4.2}$ | $\mathbf{1.7}^{+0.6}$ | $\mathbf{0.3}^{+0.0}$ | $\mathbf{2.3}^{+1.7}$ | $\mathbf{4.7}^{+2.5}$ |
| 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 (ours) | $\mathbf{5.6}^{+3.1}$ | $\mathbf{14.2}^{+8.5}$ | $3.0^{+1.1}$ | $\mathbf{2.0}^{+0.0}$ | $\mathbf{6.6}^{+3.9}$ | $\mathbf{8.8}^{+4.9}$ | $\mathbf{4.4}^{+2.5}$ | $\mathbf{10.6}^{+5.9}$ | $\mathbf{3.3}^{+2.0}$ | $\mathbf{0.5}^{+0.3}$ | $\mathbf{3.6}^{+2.2}$ | $\mathbf{7.2}^{+4.0}$ |
| 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 (ours) | $\mathbf{6.2}^{+1.7}$ | $\mathbf{16.6}^{+6.8}$ | $2.5^{-0.9}$ | $1.7^{-0.3}$ | $\mathbf{6.7}^{+2.1}$ | $\mathbf{9.6}^{+3.4}$ | $\mathbf{6.4}^{+2.7}$ | $\mathbf{14.8}^{+6.3}$ | $\mathbf{4.4}^{+1.5}$ | $\mathbf{0.7}^{+0.4}$ | $\mathbf{4.9}^{+2.4}$ | $\mathbf{9.3}^{+3.5}$ |

different backbones (large margins of 35.0% vs. 32.8% with ResNet-34 and 51.5% vs. 45.6% with ResNet-101 on novel classes). These verify the potential of Meta R-CNN that can be flexibly-deployed across different backbones and consistently outperforms the baseline methods.

**RoI meta-learning.** Since Meta R-CNN is formally devised as a meta-learner, it would be important to observe whether it is truly improved by RoI meta-learning. To verify our claim, we ablate Meta R-CNN from two aspects: 1). using meta-learning or not (Ablation 1 in Table 6); 2). meta-learning on full-image or RoI features (Table 5). As illustrated in Table 6 (Ablation 1), meta-learning significantly boosts Meta R-CNN performance by clear large margins both in novel classes (35.0% vs.9.0% in 3-shot; 51.5% vs.40.5% in 10-shot) and in base classes (38.5% vs.64.8% in 3-shot; 67.9% vs.56.9% in 10-shot). As $K$ decreases, the improvement will be more significant. In Table 5, we have observed that full-image meta-learning suffers heavy performance drop compared with RoI meta-learning and moreover, it even performs worse than the Faster R-CNN trained without meta-strategy. It shows that RoI meta-learning indeed encourages the generalization of the R-CNN family.

**Meta-loss $L_{\mathrm{meta}}(\phi)$.** Meta R-CNN takes the control of Faster R-CNN by way of class attentive vectors. Their reasonable diversity would lead to the performance improvement when detecting the objects in different classes. To verify our claim, we ablate the meta-loss $L_{\mathrm{meta}}(\phi)$ used to increase the diversity of class-attentive vectors. The ablation is shown in Table 6 Ablation 2. Obviously, the Meta R-CNN performances in base and novel classes are significantly improved by adding the meta-loss.

### 6.3. Low-shot object segmentation

As we demonstrated in our methodology, Meta R-CNN is a versatile meta-learning framework to achieve low-shot object structure prediction, especially, not just limited in the object detection task. To verify our claim, we deploy PRN to change a Mask R-CNN [17] (MRCN) into its Meta R-CNN version. This Meta R-CNN using ResNet-50 [19] as its backbone, would be evaluated on the instance-level object segmentation track on MS COCO benchmark. We report the standard COCO metrics based on object detection and segmentation. Noted that, AP in object segmentation is evaluated by using *mask* IoU. We use the trainval35k images for training and val5k for testing where the 20 classes in PASCAL VOC [9] as novel classes and the remaining 60 categories in COCO [27] as base classes. Base classes have abundant labeled samples with instance segmentation while novel classes only have $K$-shot annotated bounding boxes and instance segmentation masks. $K$ is set to 5,10 and 20 in our object segmentation experiments.

**Results.** Due to the relatively competitive performances of FRCN+ft+full shown in low-shot object detection, we adopt the same-style training strategy for MRCN, leading to **MRCN+ft+full** on object detection and instance-level object segmentation results in Table. 7. It could be observed that our proposed Meta R-CNN is consistently superior to MRCN+ft+full across 5,10,20-shot settings with significant margins in low-shot object segmentation tasks. For instance, Meta R-CNN achieves a 1.7% performance improvement (6.2% vs.4.5%) on object detection and 2.7% performance improvement (6.4% vs.3.7%) on instance segmentation. These evidences further demonstrate the superiority and universality of our Meta R-CNN presenting. Comprehensive results are found in our supplementary material.

## 7. Discussion and Future Work

Low-shot object detection/ segmentation are very valuable as their successes would lead to an extensive variety of visual tasks generalizing to newly-emerged concepts without heavily consuming labor annotation. Our work takes an insightful step towards the successes by proposing a flexible and simple yet effective framework, *e.g.*, Meta R-CNN. Standing on the shoulders of Faster/ Mask R-CNN, Meta R-CNN overcomes the shared weakness of existing meta-learning algorithms that almost disable to recognize the semantic information entangled with multiple objects. Simultaneously, it endows traditional Faster/ Mask R-CNN with the generalization capability in front of low-shot objects in novel classes. It is lightweight, plug-and-play, and performs impressively in low-shot object detection/ segmentation. It is worth noting that, as Meta R-CNN solely remodels the predictor branches into a meta-learner, it potentially can be extended to a broad range of models [15, 20, 24, 45] in the entire R-CNN family. To this Meta R-CNN might enable visual structure prediction in the more challenging low-shot conditions, *e.g.*, low-shot relationship detection and others.

## References


[1] Ron Amit and Ron Meir. Meta-learning by adjusting priors based on extended pac-bayes theory. *arXiv preprint arXiv:1711.01244*, 2017. 2, 3

[2] Anurag Arnab and Philip HS Torr. Bottom-up instance segmentation using deep higher-order crfs. *arXiv preprint arXiv:1609.02583*, 2016. 1, 2

[3] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Advances in Neural Information Processing Systems*, pages 523–531, 2016. 2

[4] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. Lstd: A low-shot transfer detector for object detection. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[5] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. 2016. 4

[6] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2018. 2

[7] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, pages 534–549. Springer, 2016. 2

[8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 2

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5, 8

[10] Li Fei-Fei, Rob Fergus, and Pietro Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *IEEE International Conference on Computer Vision*, 2008. 1, 2

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 1, 2

[12] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 2

[13] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2, 3

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[15] Georgia Gkioxari, Ross Girshick, Piotr Dollr, and Kaiming He. Detecting and recognizing human-object interactions. 2017. 8

[16] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 2

[17] Kaiming He, Georgia Gkioxari, Piotr Dollr, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 5, 7, 8

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[20] Ronghang Hu, Piotr Dollr, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. 2017. 8

[21] 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, 5, 6, 7

[22] Leonid Karlinsky, Joseph Shtok, Sivan Harary, Eli Schwartz, Amit Aides, Rogerio Feris, Raja Giryes, and Alex M Bronstein. Repmet: Representative-based metric learning for classification and few-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2019. 2

[23] Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. *arXiv preprint arXiv:1902.10644*, 2019. 2, 3

[24] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollr. Panoptic feature pyramid networks. 2019. 8

[25] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015. 1, 2

[26] B. M. Lake, R Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 1, 2

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5, 7, 8

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[29] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. 2017. 4

[30] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015. 2

[31] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016. 2

[32] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7229–7238, 2018. 2

[33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 2

[34] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 6

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. 2015. 1, 2, 5

[36] Pranav Shyam, Shubham Gupta, and Ambedkar Dukkipati. Attentive recurrent comparators. 2017. 1, 2

[37] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. 2017. 1, 2, 5

[38] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2

[39] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 2

[40] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. 2016. 1, 2, 3, 5

[41] Yu Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. 2018. 1, 3, 5

[42] Yu Xiong Wang and Martial Hebert. *Learning to Learn: Model Regression Networks for Easy Small Sample Learning*. 2016. 1, 4

[43] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pages 7029–7039, 2017. 2

[44] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Bridging category-level and instance-level semantic image segmentation. *arXiv preprint arXiv:1605.06885*, 2016. 2

[45] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *European Conference on Computer Vision*, 2018. 8

[46] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. 2017. 1

[47] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 669–677, 2016. 2

[48] Ziyu Zhang, Alexander G Schwing, Sanja Fidler, and Raquel Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2614–2622, 2015. 2