DeFRCN: Decoupled Faster R-CNN for Few-Shot Object Detection

Limeng Qiao   Yuxuan Zhao   Zhiyuan Li   Xi Qiu*   Jianan Wu   Chi Zhang

Megvii Technology

{qiaolimeng, zhaoyuxuan, lizhiyuan, qiuxi, wjn, zhangchi}@megvii.com

Abstract

Few-shot object detection, which aims at detecting novel objects rapidly from extremely few annotated examples of previously unseen classes, has attracted significant research interest in the community. Most existing approaches employ the Faster R-CNN as basic detection framework, yet, due to the lack of tailored considerations for data-scarce scenario, their performance is often not satisfactory. In this paper, we look closely into the conventional Faster R-CNN and analyze its contradictions from two orthogonal perspectives, namely multi-stage (RPN vs. RCNN) and multi-task (classification vs. localization). To resolve these issues, we propose a simple yet effective architecture, named Decoupled Faster R-CNN (DeFRCN). To be concrete, we extend Faster R-CNN by introducing Gradient Decoupled Layer for multi-stage decoupling and Prototypical Calibration Block for multi-task decoupling. The former is a novel deep layer with redefining the feature-forward operation and gradient-backward operation for decoupling its subsequent layer and preceding layer, and the latter is an offline prototype-based classification model with taking the proposals from detector as input and boosting the original classification scores with additional pairwise scores for calibration. Extensive experiments on multiple benchmarks show our framework is remarkably superior to other existing approaches and establishes a new state-of-the-art in few-shot literature [1].

1. Introduction

Recently, deep neural networks have achieved state-of-the-art on a variety of visual tasks, *e.g.* image classification [9, 17, 18] and object detection [4, 8, 14, 15, 23, 33, 34, 36]. However, these leaps of performance arrive only when a large amount of annotated data is available. Since it is often labor-intensive to obtain adequate labelled data, the number of available samples severely limits the applications of current vision systems. Besides, compared to the ability of human to quickly extract novel concepts from extremely few examples, these deep models are still far from satisfactory.
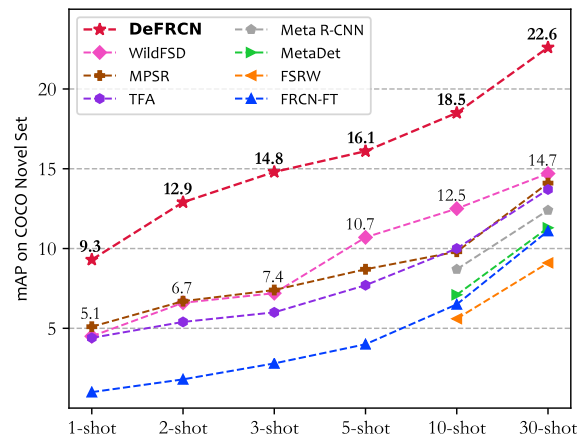

Figure 1: FSOD performance (mAP) on COCO [24] novel set at different shot numbers. The proposed DeFRCN is remarkably superior to other state-of-the-art approaches.

It is thus of attracting major research interest on few-shot learning [6, 21, 22, 29, 38, 40, 44], which employs the idea of learning novel concepts rapidly and generalizing well in data-scarce scenario. As one of the research branches, few-shot object detection (FSOD) is a much more challenging task than both few-shot classification and object detection [5, 19, 46, 51, 53]. At present, most FSOD approaches prefer to follow the meta-learning paradigm to acquire more task-level knowledge and generalize better to novel classes. However, these methods usually suffer from a complicated training process and data organization, which results in limited application scenarios. In contrast, the finetune-based methods that exist as another research branch of FSOD, are very simple and efficient [46]. By adopting a two-stage finetuning scheme, this series is comparable to meta methods. Yet, due to most parameters are pre-trained on base domain and then frozen on novel set, they may fall down the severe shift in data distribution and underutilization of novel data.

Regardless of the meta-based or finetune-based method, Faster R-CNN [36] has been widely used as the basic detector and achieved good performance. However, its original architecture is designed for conventional detection and lacks of tailored consideration for few-shot scenario, which limits the upper bound of existing approaches. Concretely,
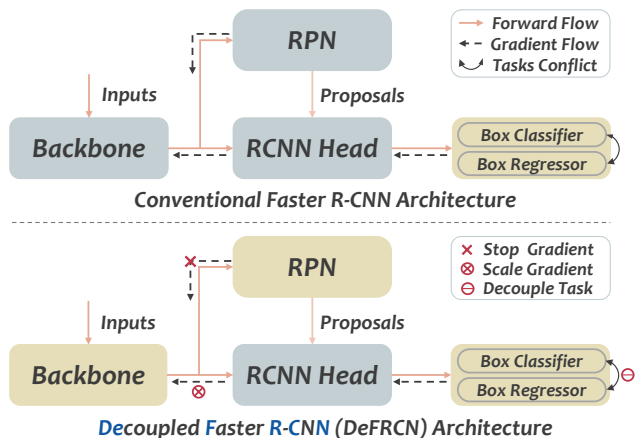

Xi Qiu (qiuxi@megvii.com) is the corresponding author.
[1] https://github.com/er-muyue/DeFRCN

Conventional Faster R-CNN Architecture

Decoupled Faster R-CNN (DeFRCN) Architecture

Figure 2: Comparison of Faster R-CNN and our motivation. We performs **stop-gradient** between RPN and backbone, meanwhile, **scale-gradient** between RCNN and backbone, as well as decouple conflict tasks between classifier and regressor. The yellow blocks are trainable during fine-tuning.

on the one hand, as a classic two-stage stacking architecture, (*i.e.*, backbone, RPN and RCNN, see Fig.2), Faster R-CNN may encounter an intractable conflict when it performs joint optimization end-to-end between **class-agnostic** RPN and **class-relevant** RCNN through the shared backbone. On the other hand, as a multi-task learning paradigm (*i.e.*, classification and localization), RCNN needs translation-invariant features for box classifier whereas translation-covariant features for box regressor. These mismatched goals potentially generate so many low-quality scores and then further lead to the reduced classification power. Moreover, since there are only a few samples available during learning, these above contradictions will be further exacerbated.

Motivated by the above observations, we extend Faster R-CNN for few-shot scenario from two orthogonal perspectives: (1) multi-stage view. As shown in Fig.2, the Faster R-CNN contains three components, *i.e.*, backbone, RPN and RCNN, which interact with each other through feature-forward and gradient-backward. Due to the contradiction mentioned above between RPN and RCNN, we present to alleviate the entire model from being dominated by one of them with tailoring the degree of decoupling between three modules through gradient. (2) multi-task view. The task conflict between classification and regression affects the quality of features, which in turn damages the performance of box head outputs, *i.e.*, category scores and box coordinates. We employ an efficient score calibration module only on the classification branch to achieve the purpose of decoupling the above two tasks.

This paper proposes a simple yet effective approach, named Decoupled Faster R-CNN (DeFRCN), to perform both multi-stage decoupling and multi-task decoupling for few-shot object detection. The overall architecture is very straightforward as demonstrated in Fig.3. Compared to the

standard Faster R-CNN [36], DeFRCN additionally contains two Gradient Decoupled Layer (GDL) and an offline Prototypical Calibration Block (PCB). The former ones are inserted between the shared backbone and RPN, meanwhile, between the backbone and RCNN to adjust the degree of decoupling among three modules, and the latter is parallel to the box classifier for further score calibration. Specifically, during the forward-backward propagation, GDL performs a learnable affine transformation on the forward feature maps and simply multiplies the backward gradient by a constant, which decouples the subsequent module and preceding module efficiently. Moreover, PCB is initially equipped with a well pre-trained classification model (*e.g.* ImageNet Pretrain) and a set of novel support prototypes. Then it takes the region proposals from few-shot detector as input and boosts the original softmax scores with additional prototype-based pairwise scores. As an interesting by-product, we find that just adopting PCB only in the inference phase can greatly improve the performance of few-shot detectors, with no extra training effort, which makes the PCB data-efficient and plug-and-play. The main contributions of our approach are three-folds:

- We look closely into the conventional Faster R-CNN and propose a simple yet effective architecture for few-shot detection, named Decoupled Faster R-CNN, which can be learned end-to-end via straightforward fine-tuning.

- To deal with the data-scarce scenario, we further present two novel modules, *i.e.* GDL and PCB, to perform decoupling among multiple components of Faster R-CNN and boost classification performance respectively.

- DeFRCN is remarkably superior to SOTAs on various benchmarks, revealing the effectiveness of our approach.

2. Related Work

2.1. General Object Detection

General object detection based on deep neural networks are currently divided into two main branches, *i.e.*, two-stage proposal-based paradigm [4, 8, 14, 15, 16, 23, 36] and one-stage proposal-free one [3, 25, 33, 34, 35], which both have witnessed fantastic progress on numerous large-scale benchmarks. The R-CNN series falls into the former line of work, which firstly generates a set of potential objects with region proposal network (RPN) [36] and then performs category classification and box localization for end-to-end detection. In contrast, one-stage detectors endeavour to directly produce final predictions from the feature map without RPN module, usually have the advantages of inference speed but the detection performance is often not as good as two-stage approaches. However, all these frameworks uniformly assume that a large amount of annotated data from seen domain can be accessed, which may be stuck in troubles in data-scarce scenarios or novel unseen domains.
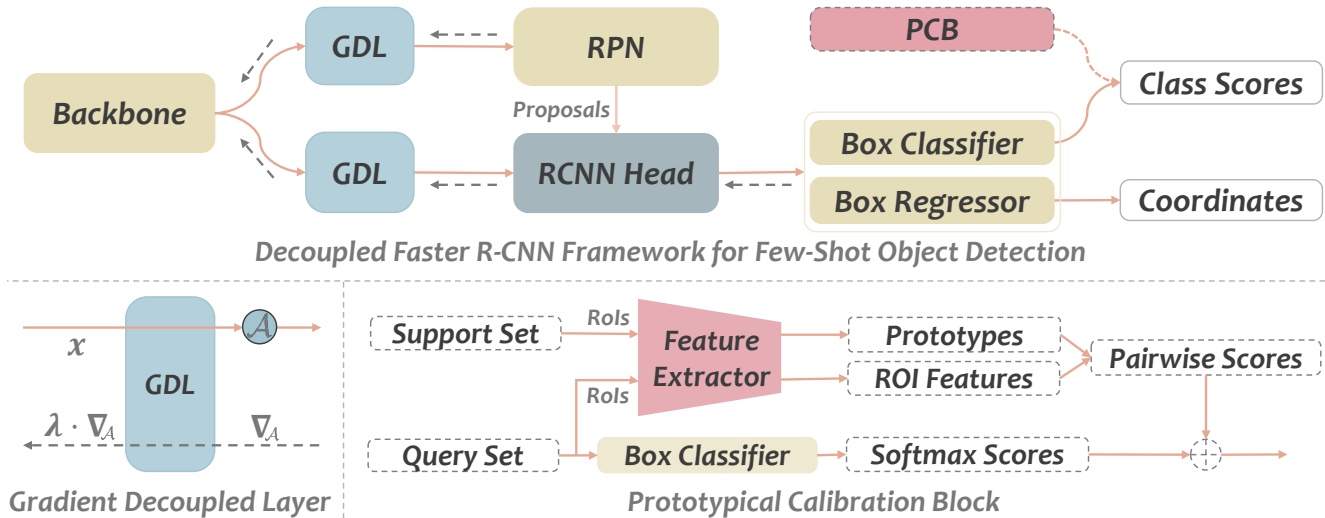
Figure 3: **The architecture of Decoupled Faster R-CNN (DeFRCN) for few-shot object detection**. Compared to the standard Faster R-CNN, there are two Gradient Decoupled Layers (sky-blue) and an offline Prototypical Calibration Block (red) are inserted into the framework to perform decoupling for multi-stage and multi-task, respectively. The $\mathcal{A}$ is the affine transformation layer in GDL and $\oplus$ is score fusion operation in PCB. Moreover, yellow and dark-blue indicate that the block is trainable and frozen during fine-tuning. The orange solid and black dotted lines represent forward flow and gradient flow.

2.2. Few-Shot Learning

Few-shot learning, which aims at learning to learn general knowledge slowly from abundant base data and extracting novel concepts rapidly from extremely few examples of new-coming classes, has been recently featured into the meta-learning based [43] and fine-tuning based [27] paradigms. As a recognition case of few-shot learning, few-shot classification has been widely investigated until now. In the literature, a large amount of studies that follow the idea of meta-learning to alleviate severe over-fitting can be divided into two streams, namely, optimization approaches [1, 13, 22, 26, 30, 32] and metric approaches [21, 29, 38, 40, 44]. The former intents to learn efficient parameter updating rules [32] or good parameters initialization strategies [13], and the latter focuses on obtaining a generalizable embedding metric space to perform pairwise similarity of inputs. In addition to meta-based approaches, some simple fine-tuning based methods [6, 41] are attaching more and more attention in the few-shot community. These methods show that just fine-tuning a linear classifier on top of a pre-trained model surprisingly achieves competitive performance with the meta-based approaches. Compared to classification, the solutions for other tasks, such as object detection and segmentation, are still underdeveloped.

2.3. Few-Shot Object Detection

Since previous detectors usually require a large amount of annotated data, few-shot detection has attracted more and more interest recently [2, 10, 12, 28, 31, 45, 47, 52, 54]. Similar to classification task [38, 39], most of the current

few-shot detectors focus on the meta-learning paradigm. FSRW [19] is a light-weight meta-model based on YOLOv2 [34] to re-weight the importance of features with channel-wise attention, and then adapt these features to promote novel detection. Yet, instead of employing attention on the whole feature map, Meta R-CNN [53] focuses on the attention of each RoI feature. Furthermore, FSDView [51] puts forward a novel feature aggregation scheme, which leverages on base classes feature information to improve the performance on novel classes. From the perspective of attention on RPN, FSOD [11] utilizes support information to filter out most background boxes and those in non-matching categories. Although meta-based approaches have been extensively studied recently, there are still some other meta-free methods. RepMet [20] incorporates a modified prototypical network as classification head into a standard object detector. And TFA [46] proposes a simple approach based on transfer learning, that only fine-tunes the last layer of existing detectors on rare classes, which are comparable to the previous meta-based methods. Instead, our approach, which also follows the idea of fine-tuning, jointly trains the almost entire detector with novel gradient decoupled layer and prototypical calibration block, outperforming all above meta-based and finetune-based approaches.

3. Methods

In this section, we first introduce the setup of few-shot object detection in Section 3.1. Then we revisit conventional Faster R-CNN in Section 3.2 and elaborate our Decoupled Faster R-CNN (DeFRCN) in Section 3.3.

3.1. Problem Setting

As in various previous work [11, 19, 46, 51], we follow the standard problem settings of few-shot object detection in our paper. Specifically, the whole learning procedure is organized into the form of two-stage fine-tuning paradigm, which gradually collects transferable knowledge across a large base set $\mathcal{D}_{base}$ with abundant annotated instances and performs adaptation quickly on novel support set $\mathcal{D}_{novel}$ with only a few samples per category. Note that the base classes $\mathcal{C}_{base}$ in $\mathcal{D}_{base}$ and the novel classes $\mathcal{C}_{novel}$ in $\mathcal{D}_{novel}$ are non-overlapping, namely, $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \varnothing$. Given a sample $(x, y) \in \mathcal{D}_{base} \cup \mathcal{D}_{novel}$, where $x = \{o_i, i = 1, ..., N\}$ is the input image with N objects and $y = \{(c_i, b_i), i = 1, ..., N\}$ denotes the category $c_i \in \mathcal{C}_{base} \cup \mathcal{C}_{novel}$ and the structured box annotations b_i. Under this setting, the ultimate goal of our algorithm is to optimize a robust detector $\mathcal{F}$ based on the $\mathcal{D}_{base}$ and $\mathcal{D}_{novel}$, then classify and localize unlabelled objects of a novel query set $\mathcal{D}_{query}$ with classes $\mathcal{C}_{query}$, where $\mathcal{C}_{query} \subseteq \mathcal{C}_{base} \cup \mathcal{C}_{novel}$. The overall procedure, which follows the standard transfer learning, can be summarized as follow,

$$\mathcal{F}_{init} \overset{\mathcal{D}_{base}}{\rightsquigarrow} \mathcal{F}_{base} \overset{\mathcal{D}_{novel}}{\rightsquigarrow} \mathcal{F}_{novel} \qquad (1)$$

where $\mathcal{F}_{init}$, $\mathcal{F}_{base}$ and $\mathcal{F}_{novel}$ denote the learned detectors in initialization, base training and novel fine-tuning stages respectively. The symbol $\rightsquigarrow$ indicates model training.

3.2. Revisiting Faster R-CNN

As a two-stage stacking architecture, Faster R-CNN [36] consists of three function-detached modules for end-to-end training, *i.e.*, a shared convolutional backbone for extracting *generalized* features, an efficient Region Proposal Network (RPN) for generating *class-agnostic* proposals and a task-specific RCNN head [14] for performing *class-relevant* classification and localization. The whole learning procedure is illustrated in Fig.2 (a). Concretely, the input image is first fed into the backbone to generate a high-level feature map, and then parallelly provided to the next two modules, *i.e.*, RPN and RCNN. Second, with classifying and regressing a group of scale varying anchors of the feature map simultaneously, RPN generates a sparse set of high-quality region proposals. Finally, on top of the shared feature map and proposals, RCNN pools each region-of-interest into a fixed size feature map with RoI pooling [16], and then performs box classifier and regressor for computing the object category probabilities and fine-tuning the box boundaries respectively. All these modules are jointly optimized end-to-end by minimizing an unify objective function, which follows the multi-task learning paradigm as:

$$\mathcal{L}_{total} = \underbrace{(\mathcal{L}_{rpn}^{cls} + \mathcal{L}_{rpn}^{reg})}_{rpn\ task} + \eta \cdot \underbrace{(\mathcal{L}_{rcnn}^{cls} + \mathcal{L}_{rcnn}^{reg})}_{rcnn\ task} \qquad (2)$$

where η is a balanced hyper-parameter for different tasks.

Problem of multi-task learning. It can be seen that the above-mentioned three modules of Faster R-CNN constitute an unified multi-task learning (MTL) framework, yet there is a certain inconsistency among the optimization goals of these sub-networks. Specifically, with utilizing the feature maps extracted from hard-parameter shared [42] backbone, RPN aims at generating class-agnostic region proposals to tell the network *where to look*, while RCNN targets to perform region-based detection category by category to identify *what to look*. Furthermore, the classification head needs translation invariant features whereas the localization head needs translation covariant features on the contrary. In spite of multi-task learning generally helps to improve the end-to-end performance of object detection as shown in Faster R-CNN [36], the joint optimization with the Eq.2 may lead to possible suboptimal solution on individual tasks in order to balance the mismatched goals of them [7, 49].

Problem of shared backbone. According to the arguments in [36], the ultimate goal of shared backbone is to extract general features that are as suitable as possible for all down-stream tasks. In fact, from the perspective of gradient flow in Fig.2 (a), RPN and RCNN mutually exchange information of optimization through the shared backbone. However, due to the potential contradictions between RPN and RCNN, we notice that the current architecture may lead to the reduced few-shot detection power of the entire framework. Moreover, following the setting of Eq.1, the shared backbone of few-shot novel detector $\mathcal{F}_{novel}$ is usually fine-tuned from a base domain detector $\mathcal{F}_{base}$. During this two-stage cross-domain procedure, RPN may suffer from the foreground-background confusion, which means a proposal that belongs to background in the base training phase is likely to be foreground in the novel fine-tuning phase. Through the gradient from RPN, the shared convolutional layers propagate the tendency of over-fitting on base classes to backbone and RCNN. Although this is one of the convergence schemes to behave well on base domain, it potentially damages the ability to transfer to the novel set quickly and efficiently, especially in the data-scarce scenario.

3.3. Decoupled Faster R-CNN

Motivated by the above arguments, we propose a simple yet effective approach, named Decoupled Faster R-CNN (DeFRCN), to tap into more potential of Faster R-CNN styled detectors in few-shot literature. Based on the idea of decoupling three functional modules (*i.e.*, backbone, RPN and RCNN) and two kinds of tasks (*i.e.*, classification and localization), the overall architecture of our method is very straightforward as demonstrated in Fig.3, which has two Gradient Decoupled Layers (GDL) to adjust the degree of decoupling among three modules and an offline Prototypical Calibration Block (PCB) to improve the classification power of RCNN during the inference phase.

3.3.1 Gradient Decoupled Layer

In this section, we look into a different aspect of network design - how to customize the relationship between the upstream and downstream modules of the model. From the perspective of feature-forward and gradient-backward, we introduce a novel architectural unit, denoted as the Gradient Decoupled Layer (GDL). During the forward propagation, GDL employs an affine transformation layer $\mathcal{A}$, which is parameterized by learnable channel-wise weights ω and bias b, to simply enhance feature representations and perform forward-decoupling. During the backward propagation, GDL takes the gradient from the subsequent layer, multiplies it by a constant $\lambda \in [0, 1]$ and passes it to the preceding layer, as illustrated in Fig.3. Concretely, along with the back-propagation process passes through the GDL, the partial derivatives of the loss $\mathcal{L}_d$ that is downstream of the GDL with respect to the layer parameters θ_u that are upstream of the GDL get multiplied by λ, $i.e.$, $\frac{\partial \mathcal{L}_d}{\partial \theta_u}$ (denoted as ∇ in Eq.4) is simply replaced with $\lambda \frac{\partial \mathcal{L}_d}{\partial \theta_u}$. Mathematically, we can formally treat GDL as a pseudo-function $\mathbb{G}_{(\mathcal{A},\lambda)}$ defined by two equations describing its forward- and backward-propagation behaviour as follows:

$$\mathbb{G}_{(\mathcal{A},\lambda)}(x) = \mathcal{A}(x) \qquad (3)$$

$$\frac{d\mathbb{G}_{(\mathcal{A},\lambda)}}{dx} = \lambda \nabla_{\mathcal{A}} \qquad (4)$$

where $\mathcal{A}$ is an affine transformation layer, $\lambda \in [0, 1]$ is a decoupling coefficient and $\nabla_{\mathcal{A}}$ is the Jacobian matrix from the affine layer. In general, implementing such layer with existing deep learning frameworks are extremely simple, as defining procedures for forwardprop (affine transformation) and backprop (multiplying by a constant) is trivial. We provide the pseudo-code of GDL in Algorithm 1.

Perform Decoupling with GDL. Given a standard Faster R-CNN [36], two GDLs are respectively inserted between the shared backbone and RPN ($i.e.$, $\mathbb{G}_{rpn}$), as well as the shared backbone and RCNN ($i.e.$, $\mathbb{G}_{rcnn}$), which brings the part of DeFRCN architecture depicted in Fig.3. Specifically, during the forward propagation, the feature from shared backbone is transformed into different feature spaces through $\mathcal{A}_{rpn}$ and $\mathcal{A}_{rcnn}$. Moreover, during the backward propagation, we adjust the decoupling degree of three modules ($i.e.$, backbone, RPN and RCNN) by applying different λ_{rpn} and λ_{rcnn} on gradients. More formally, we consider the following loss function with two separate GDLs as:

$$\mathcal{L} = \mathcal{L}_{rpn}\left(F_{rpn}(\mathbb{G}_{rpn}(F_b(x;\theta_b)); \theta_{rpn}), y_{rpn}\right) + \eta \cdot$$
$$\mathcal{L}_{rcnn}\left(F_{rcnn}(\mathbb{G}_{rcnn}(F_b(x;\theta_b)); \theta_{rcnn}), y_{rcnn}\right) \quad (5)$$

Here, $\mathbb{G}_{\cdot}$ is the Gradient Decoupled Layer we proposed in this section, θ_b, θ_{rpn} and θ_{rcnn} are learnable parameters for the backbone, RPN and RCNN respectively. Moreover, η

is a hyper-parameter to control the trade-off between $\mathcal{L}_{rpn}$ and $\mathcal{L}_{rcnn}$ (usually is set to 1).

Optimization with GDL. Consistent with the optimization goal of Faster R-CNN, we seek the optimal parameters θ_b, θ_{rpn} and θ_{rcnn}, denoted as Θ, for the function Eq.5 as:

$$\Theta = \arg\min_{\Theta} \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \mathcal{L}, \ \ \Theta = \{\theta_b, \theta_{rpn}, \theta_{rcnn}\} \quad (6)$$

where $\mathcal{N}$ is the number of training samples, and $\mathcal{L}$ is from the Eq.5. Concretely, a gradient descent step can be described as:

$$\theta_b \ \ \leftarrow \ \ \theta_b \ - \ \gamma\left(\lambda_1 \frac{\partial \mathcal{L}_{rpn}}{\partial \theta_b} + \lambda_2 \frac{\partial \mathcal{L}_{rcnn}}{\partial \theta_b}\right) \quad (7)$$

$$\theta_{rpn} \ \ \leftarrow \ \ \theta_{rpn} \ - \ \gamma \frac{\partial \mathcal{L}_{rpn}}{\partial \theta_{rpn}} \quad (8)$$

$$\theta_{rcnn} \ \ \leftarrow \ \ \theta_{rcnn} \ - \ \gamma \frac{\partial \mathcal{L}_{rcnn}}{\partial \theta_{rcnn}} \quad (9)$$

where γ is the learning rate, λ_1 and λ_2 are decoupling coefficients for RPN and RCNN respectively. It can be seen from Eq.8 and Eq.9 that adding GDL does not affect the optimization of RPN and RCNN. However, the parameter update of sharing backbone is deeply affected by GDL in Eq.7. We mainly analyze three important situations: (1) $\lambda_1 = 0$ (or $\lambda_2 = 0$), it is equivalent to *stopping gradient* from RPN (or RCNN), and the update of θ_b will only be dominated by RCNN (or RPN); (2) $\lambda_1 \in (0, 1]$ (or $\lambda_2 \in (0, 1]$), it is equivalent to *scaling gradient* from RPN (or RCNN), which means that the RPN (or RCNN) has individual contributions to the update of shared backbone; (3) $\lambda_1 = \lambda_2 = \tilde{\lambda}$, which is equivalent to multiplying the learning rate γ of backbone by a small coefficient, $i.e.$, $\tilde{\lambda}$, ensures that the update speed of θ_b is slower than θ_{rpn} and θ_{rcnn}. Note that $\lambda < 0$ is meaningless for detection and more discussion about λ is mentioned in the supplementary material.

Algorithm 1 Gradient Decoupled Layer, PyTorch-like

```
# A: learnable channel-wise affine layer
# _lambda: gradient decoupling coefficient

class GradientDecoupledLayer(Function):

    # feature forward
    def forward(ctx, x, A, _lambda):
        ctx._lambda = _lambda
        x = A(x)
        return x.view_as(x)

    # gradient backward
    def backward(ctx, grad_output):
        grad_output = grad_output * ctx._lambda
        return grad_output, None, None

def decouple_layer(x, A, _lambda):
    return GradientDecoupleLayer(x, A, _lambda)
```

3.3.2 Prototypical Calibration Block

In this section, we introduce a novel metric-based score refinement module, termed as Prototypical Calibration Block (PCB), to effectively decouple the classification and localization tasks during the inference time. In general, most of detectors parallelly deploy a classifier and a regressor on top of the shared network. However, classification needs translation invariant features whereas localization needs translation covariant features. Thus the localization branch may force the backbone to gradually learn translation covariant property, which potentially downgrades the performance of classifier. Due to model complexity, the extreme lack of annotated samples will further exacerbate this contradiction.

We notice that the under-explored few-shot classification branch generates a large amount of low-quality scores, which motivates us to eliminate high-scored false positives and remedy low-scored missing samples by introducing a Prototypical Calibration Block (PCB) for score refinement. The overall pipeline is illustrated in Fig.3 (c). Concretely, our PCB consists of a strong classifier from ImageNet pre-trained model, a RoIAlign layer and a prototype bank. Given a M-way K-shot task with support set $\mathcal{S}$, the PCB first extracts original image feature map and then employs RoIAlign with ground-truth boxes to produce MK instance representations. Based on these features, we shrink the support set $\mathcal{S}$ to a prototype bank $\mathcal{P} = \{p_c\}_{c=1}^{M}$ with Eq.10:

$$p_c = \frac{1}{|\mathcal{S}_c|} \sum_{(x_i, \, y_i) \in \mathcal{S}_c} x_i \qquad (10)$$

where $\mathcal{S}_c$ is a subset which contains samples with the same label c in $\mathcal{S}$. Given an object proposal $\hat{y}_i = (c_i, s_i, b_i)$ produced by fine-tuned few-shot detector, where b_i is the box boundaries, c_i is the predicted category and s_i is the corresponding score, PCB first performs RoIAlign on predicted box b_i to generate object feature x_i, and then calculate the cosine similarity s_i^{cos} between x_i and p_{c_i} as:

$$s_i^{cos} = \frac{x_i \cdot p_{c_i}}{\|x_i\| \|p_{c_i}\|} \qquad (11)$$

In the end, we perform weighted aggregation between the s_i^{cos} from PCB and s_i from few-shot detector for final classification score $s_i^{\ddagger}$ as follow:

$$s_i^{\ddagger} = \alpha \cdot s_i + (1 - \alpha) \cdot s_i^{cos} \qquad (12)$$

where α is the trade-off hyper-parameter.

Moreover, we do not share any parameters between the few-shot detector and PCB module, so that the PCB can not only preserve the quality of classification-aimed translation invariance feature, but also better decouple the classification task and regression task within the RCNN. Furthermore, since the PCB module is offline without any further training, it can be plug-and-play and easily equipped to any other architectures to build stronger few-shot detectors.

4. Experiments

In this section, we first introduce the experimental settings in Sec.4.1 and then compare our approach with previous SOTAs on multiple benchmarks in Sec.4.2. Finally, we provide comprehensive ablation studies in Sec.4.3.

4.1. Experimental Setting

Existing benchmarks. We follow the previous work [19, 46, 51] and utilize the same data splits with [46] to evaluate our approach for a fair comparison. As for PASCAL VOC, we have three random split groups and each of them covers 20 categories, which are randomly divided into 15 base classes and 5 novel classes. Each novel category has $K = 1, 2, 3, 5, 10$ objects sampled from the combination of VOC07 and VOC12 train/val set for few-shot training. And VOC07 test set is used for evaluation. As for COCO, the 60 categories disjoint with VOC are denoted as base classes while the remaining 20 classes are used as novel classes with $K = 1, 2, 3, 5, 10, 30$ shots. We utilize 5k images from the validation set for evaluation and the rest for training.

Evaluation setting. We take two popular evaluation protocols into consideration to access the effectiveness of our approach, including few-shot object detection (**FSOD**) and generalized few-shot object detection (**G-FSOD**). The former protocol is widely adopted by most previous methods [5, 19, 51, 53] and only focuses on the performance of novel classes. Yet, the latter presents to not only observe the performance on novel classes, but also base and overall performance of the few-shot detector, which is more comprehensive and monitors the occurrence of catastrophic forgetting [46]. For evaluation metrics, we report AP_{50} for VOC and the COCO-style mAP for COCO. Moreover, all results are averaged over multiple repeated runs.

Implementation details. Our approach employs Faster R-CNN [36] (termed as FRCN) as the basic detection framework and ResNet-101 [17] pre-trained on ImageNet [37] as the backbone. We adopt SGD to optimize our network end-to-end with a mini-batch size of 16, momentum of 0.9 and weight decay of $5e^{-5}$. The learning rate is set to 0.02 during base training and 0.01 during few-shot fine-tuning. Moreover, the λ in GDL of RPN is set to 0 for stopping gradient and the λ in GDL of RCNN is set to 0.75 during base training and 0.01 during novel fine-tuning for scaling gradient. The α in PCB is uniformly set to 0.5 in all settings.

4.2. Comparison Results

PASCAL VOC. We present our evaluation results of VOC on three different data splits in Table 1. It can be seen that, no matter under the *FSOD* or *G-FSOD* setting, our De-FRCN is significantly superior to the recent state-of-the-art approaches by a large margin (up to 21.4%), which demonstrates the effectiveness of our approach. Based on the re-

Method / Shots	w/G	Novel Set 1					Novel Set 2					Novel Set 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
YOLO-ft [19]	✗	6.6	10.7	12.5	24.8	38.6	12.5	4.2	11.6	16.1	33.9	13.0	15.9	15.0	32.2	38.4
FRCN-ft [53]	✗	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
LSTD [5]	✗	8.2	1.0	12.4	29.1	38.5	11.4	3.8	5.0	15.7	31.0	12.6	8.5	15.0	27.3	36.3
FSRW [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 [47]	✗	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
Meta R-CNN [53]	✗	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 [46]	✗	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 [48]	✗	**41.7**	-	**51.4**	55.2	**61.8**	**24.4**	-	**39.2**	**39.9**	**47.8**	**35.6**	-	42.3	48.0	49.7
DeFRCN (Ours)	✗	**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**
FRCN-ft [53]	✓	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 [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 [46]	✓	**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 [51]	✓	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 (Ours)	✓	**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**

Table 1: Experimental results on VOC dataset. We evaluate DeFRCN performance (AP_{50}) on three different splits. The term *w/G* indicates whether we use the *G-FSOD* setting [46]. **RED**/**BLUE** indicate SOTA/the second best. Note that our results are averaged over multiple runs and the base/overall performance are presented in supplementary materials, the same below.

Method / Shots	w/G	Shot Number					
		1	2	3	5	10	30
FRCN-ft [53]	✗	1.0*	1.8*	2.8*	4.0*	6.5	11.1
FSRW [19]	✗	-	-	-	-	5.6	9.1
MetaDet [47]	✗	-	-	-	-	7.1	11.3
Meta R-CNN [53]	✗	-	-	-	-	8.7	12.4
TFA [46]	✗	4.4*	5.4*	6.0*	7.7*	10.0	13.7
MPSR [48]	✗	**5.1***	**6.7***	**7.4***	8.7*	9.8	14.1
FSDetView [51]	✗	4.5	6.6	7.2	**10.7**	**12.5**	**14.7**
DeFRCN (Ours)	✗	**9.3**	**12.9**	**14.8**	**16.1**	**18.5**	**22.6**
FRCN-ft [53]	✓	1.7	3.1	3.7	4.6	5.5	7.4
TFA [46]	✓	1.9	3.9	5.1	7.0	9.1	12.1
FSDetView [51]	✓	**3.2**	**4.9**	**6.7**	**8.1**	**10.7**	**15.9**
DeFRCN (Ours)	✓	**4.8**	**8.5**	**10.7**	**13.6**	**16.8**	**21.2**

Table 2: Experimental results on COCO dataset. We evaluate DeFRCN performance (mAP) over multiple runs. The superscript * indicates that the results are reproduced by us.

Method	FRCN-ft	FSRW	MetaDet	MetaRCNN	MPSR	Ours
mAP	31.2	32.3	33.9	37.4	42.3	**55.9**

Table 3: The 10-shot cross-domain *FSOD* performance on COCO base set $\rightarrow$ VOC novel set. All detection results for comparison refer from [19, 48, 53].

COCO. The Table 2 shows all evaluation results on COCO dataset with the standard COCO-style averaged AP (mAP). Obviously, our approach consistently outperforms recent SOTAs in all setups, including *FSOD* and *G-FSOD* for K=1,2,3,5,10,30. For *FSOD*, we achieve around 6.0% and 7.9% improvement over the best method in 10-shot and 30-shot respectively, which demonstrates the strong robustness and generalization ability of our method in the few-shot scenario. Furthermore, compared to the fine-tuning based methods, the number of learnable parameters of DeFRCN is almost the same as FRCN-ft and much more than TFA. The results in Table 2 reveal that our method not only guarantees the sufficient learning of these parameters, but also does not fall into the severe over-fitting. All base/overall results of *G-FSOD* are presented in supplementary materials.

COCO to VOC. We conduct the cross-domain *FSOD* experiments on the standard VOC 2007 test set with following the same setting from [19, 48], which uses the base dataset with 60 classes as in the previous COCO within-domain setting and the novel dataset with 10-shot objects for each of the 20 classes from VOC. As shown in the Table 3, our approach achieves the best performance with 55.9%, which has 13.6% improvement than MPSR [48]. This huge upswing demonstrates that our proposed DeFRCN has better generalization ability in cross-domain situations.

sults of Table 1, we further notice that two interesting phenomena exist in few-shot detection: (1) For *FSOD* setting, the increment of novel shots does not necessarily lead to an advance in final performance. Take Novel Set 1 as an example, the AP_{50} of 5-shot is 64.1% but 10-shot is 60.8% (-3.3%). There is a similar case in TFA. We conjecture that the quality of sample is vital in data-scarce scenario and adding low-quality samples may be harmful to the detector. (2) For the comparison between *FSOD* and *G-FSOD*, we find that as the number of shots increases, the final performance of *G-FSOD* grows faster than that of *FSOD* (40.2% $\rightarrow$ 66.5% *vs.* 53.6% $\rightarrow$ 60.8%), which is due to the addition of more negative samples under the *G-FSOD* setting.

| FRCN	GDL-B	GDL-N	PCB	Base	Novel	
					10	30
✓					7.9	12.2
✓			✓	38.4	10.4	14.8
✓		✓			15.2	19.0
✓		✓	✓		16.6	20.5
✓	✓				8.2	13.1
✓	✓		✓	39.0	10.8	15.1
✓	✓	✓			16.9	21.0
✓	✓	✓	✓		18.5	22.6

Table 4: Effectiveness of different modules in DeFRCN. All results are conducted on COCO dataset. The GDL-B and GDL-N indicates that we use GDL in base training phase and novel fine-tuning phase respectively.

Backbone	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
R50-C4-1x [50]	35.7	56.1	38.0	19.2	40.9	48.7
+ GDL	36.5	**57.6**	39.2	19.8	41.7	50.3
R101-C4-3x [50]	41.1	61.4	44.0	22.2	45.5	55.9
+ GDL	41.9	**62.3**	45.1	22.3	46.6	57.8

Table 5: Conventional object detection results on COCO.

4.3. Ablation Study

Effectiveness of different modules. We conduct relative ablations in 10/30-shot scenarios on the COCO dataset to carefully analyze how much each module contributes to the ultimate performance of DeFRCN. All results are shown in Table 4 in great details. Specifically, the first row shows the results of plain FRCN, which only achieves 7.9%/12.2% for 10/30-shot respectively, indicating that the original model without any few-shot techniques is severely over-fitting due to the lack of training data. Next, we take four progressive steps to complete the exploration of our DeFRCN: (1) add GDL in base training phase (GDL-B). Through the results of rows 1-4 and 5-8, we find that the GDL-B improves by 0.6% on base classes and also a certain improvement (0.3% ∼ 2.1%) on novel classes. This indicates that a better base model is beneficial to the performance of few-shot detector. (2) add GDL in novel fine-tuning phase (GDL-N). The results of first row and third row show that GDL-N makes an amazing boost with 7.3%/6.8% for 10/30-shot, which are mainly from two aspects: i) more learnable parameters guarantee sufficient ability to transfer to novel domain, and ii) GDL greatly reduces the risk of over-fitting.(3) add PCB in the inference phase. As a plug-and-play module, PCB is orthogonal to GDL, so no matter which setting PCB is added, our model further gains 1.4% ∼ 2.6% points on mAP. (4) Finally, we integrate the above three modules into original FRCN, and the last line shows the final performance of DeFRCN. Compared to the plain results in the first row, we obtain a marvelous promotion of **10.6%/10.4%** for 10/30-shot, which proves the effectiveness of our approach.
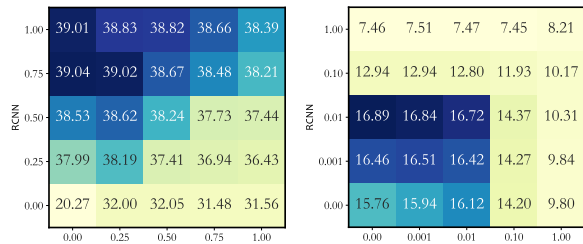
(a) The base training stage (b) The novel fine-tune stage

Figure 4: The effectiveness of different degree of coupling. The horizontal and vertical axis represent the λ in GDL of RPN and RCNN respectively. Note the results in (b) do not use PCB to ensure the impact of GDL is considered only.

Effectiveness of the degree of decoupling. We carefully explore the influence of decoupling with setting different λ_{rpn} and λ_{rcnn} in GDL during both base training and novel fine-tuning, and all results are illustrated in Fig.4. No matter in the base training or the novel fine-tuning stage, the model tends to achieve higher performance when λ_{rpn} is set to a smaller value (close to 0), while λ_{rcnn} needs an appropriate value to ensure that the backbone can be optimized better. This observation prompts us to perform *stop-gradient* for RPN and *scale-gradient* for RCNN in DeFRCN. In addition, we further get a very interesting conclusion from four corners in Fig.4(a): in term of FRCN backbone optimization, RPN plays a negative role in this procedure (39.01 *vs.* 38.39), while RCNN has a positive effect (31.56 *vs.* 38.39).

Can GDL boost conventional detection? The above analysis shows that GDL brings a significant improvement on FSOD. Since the problem it solves (that is, the contradiction in Faster R-CNN) also potentially exists in conventional detection, we conjecture that our GDL is as well as effective in data-sufficient scenarios. Thus we further conduct experiments on COCO 2017 dataset with standard setup [36] and the results are shown in Table 5. It can be seen that the proposed GDL outperforms baselines on all evaluation metrics. Specifically, adding GDL to original FRCN gains 1.5% and 0.9% AP50 for Res-50 and Res-101 respectively.

5. Conclusion

In this paper, we look closely into the visual task of few-shot object detection and propose a simple yet effective fine-tuning based framework, named Decoupled Faster R-CNN, which remarkably alleviates the potential contradictions of conventional Faster R-CNN in data-scarce scenario with introducing novel GDL and PCB. Despite its simplicity, our method still achieves new state-of-the-art on various benchmarks, which demonstrates its effectiveness and versatility.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.

[2] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision*, pages 384–400, 2018.

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.

[5] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. Lstd: A low-shot transfer detector for object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[6] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

[7] Bowen Cheng, Yunchao Wei, Honghui Shi, Rogerio Feris, Jinjun Xiong, and Thomas Huang. Revisiting rcnn: On awakening the classification power of faster rcnn. In *Proceedings of the European Conference on Computer Vision*, pages 453–468, 2018.

[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.

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 764–773, 2017.

[10] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-example object detection with model communication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1641–1654, 2018.

[11] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2020.

[12] Zhibo Fan, Jin-Gang Yu, Zhihao Liang, Jiarong Ou, Changxin Gao, Gui-Song Xia, and Yuanqing Li. Fgn: Fully guided network for few-shot instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9172–9181, 2020.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

[14] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[15] 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.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[17] 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.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[19] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8420–8429, 2019.

[20] 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.

[21] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning Workshop*, volume 2. Lille, 2015.

[22] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[24] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[26] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[27] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

[28] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13846–13855, 2020.

[29] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3603–3612, 2019.

[30] 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.

[31] Shafin Rahman, Salman Khan, Nick Barnes, and Fahad Shahbaz Khan. Any-shot object detection. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[32] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

[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.

[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.

[35] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[38] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[39] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

[40] 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.

[41] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *Proceedings of the European Conference on Computer Vision*, 2020.

[42] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[43] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

[44] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[45] Tao Wang, Xiaopeng Zhang, Li Yuan, and Jiashi Feng. Few-shot adaptive faster r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7173–7182, 2019.

[46] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *International Conference on Machine Learning*, 2020.

[47] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9925–9934, 2019.

[48] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *Proceedings of the European Conference on Computer Vision*, pages 456–472. Springer, 2020.

[49] Yue Wu, Yinpeng Chen, Lu Yuan, Zicheng Liu, Lijuan Wang, Hongzhi Li, and Yun Fu. Rethinking classification and localization for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10186–10195, 2020.

[50] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[51] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *Proceedings of the European Conference on Computer Vision*, 2020.

[52] Zixuan Xiao, Jiahao Qi, Wei Xue, and Ping Zhong. Few-shot object detection with self-adaptive attention network for remote sensing images. *IEEE J-STARS*, 14:4854–4865, 2021.

[53] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9577–9586, 2019.

[54] Ze Yang, Yali Wang, Xianyu Chen, Jianzhuang Liu, and Yu Qiao. Context-transformer: tackling object confusion for few-shot detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12653–12660, 2020.