Few-Shot Object Detection via Variational Feature Aggregation

Jiaming Han[1,2*], **Yuqiang Ren**[3], **Jian Ding**[1,2], **Ke Yan**[3†], **Gui-Song Xia**[1,2†]

[1]NERCMS, School of Computer Science, Wuhan University
[2]State Key Lab. LIESMARS, Wuhan University
[3]YouTu Lab, Tencent
{hanjiaming, jian.ding, guisong.xia}@whu.edu.cn, {condiren, kerwinyan}@tencent.com

Abstract

As few-shot object detectors are often trained with abundant base samples and fine-tuned on few-shot novel examples, the learned models are usually biased to base classes and sensitive to the variance of novel examples. To address this issue, we propose a meta-learning framework with two novel feature aggregation schemes. More precisely, we first present a Class-Agnostic Aggregation (CAA) method, where the query and support features can be aggregated regardless of their categories. The interactions between different classes encourage class-agnostic representations and reduce confusion between base and novel classes. Based on the CAA, we then propose a Variational Feature Aggregation (VFA) method, which encodes support examples into class-level support features for robust feature aggregation. We use a variational autoencoder to estimate class distributions and sample variational features from distributions that are more robust to the variance of support examples. Besides, we decouple classification and regression tasks so that VFA is performed on the classification branch without affecting object localization. Extensive experiments on PASCAL VOC and COCO demonstrate that our method significantly outperforms a strong baseline (up to 16%) and previous state-of-the-art methods (4% in average).

Introduction

This paper studies the problem of few-shot object detection (FSOD), a recently-emerged challenging task in computer vision (Yan et al. 2019; Kang et al. 2019). Different from generic object detection (Girshick et al. 2014; Redmon et al. 2016; Ren et al. 2017), FSOD assumes that we have abundant samples of some base classes but only a few examples of novel classes. Thus, a dynamic topic is how to improve the recognition capability of FSOD on novel classes by transferring the knowledge of base classes to novel ones.

In general, FSOD follows a two-stage training paradigm. In stage-I, the detector is trained with abundant base samples to learn generic representations required for the object detection task, such as object localization and classification. In stage-II, the detector is fine-tuned with only K shots (K=1, 2, 3, ...) novel examples. Despite the great success of this paradigm, the learned models are usually biased to base classes due to the imbalance between base and novel classes. As a result, the model will confuse novel objects with similar base classes. See Fig. 5 (top) for an instance, the novel class, *cow*, has high similarities with several base classes such as *dog, horse* and *sheep*. Besides, the model is sensitive to the variance of novel examples. Since we only have K shots examples per class, the performance highly depends on the quality of the support sets. As shown in Fig. 1, appearance variations are common in FSOD. Previous methods (Yan et al. 2019) consider each support example as a single point in the feature space and average all features as class prototypes. However, it is difficult to estimate the real class centers with a few examples.

In this paper, we propose a meta-learning framework to address this issue. Firstly, we build a strong meta-learning baseline based on Meta R-CNN (Yan et al. 2019), which even outperforms a representative two-stage fine-tuning ap-

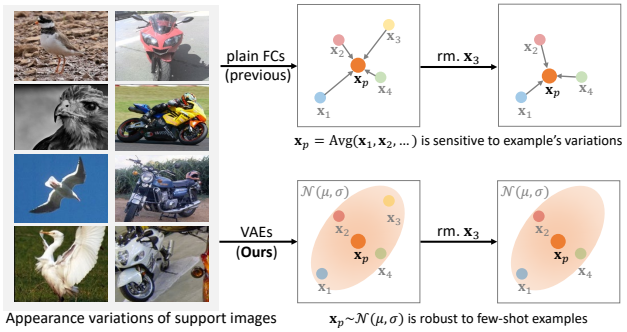

Figure 1: Comparisons of different support feature encoding methods. Previous methods use plain fully-connected (FC) layers to encode support features and obtain class prototypes by averaging these features: $\mathbf{x}_p = \text{Avg}(\mathbf{x}_1, \mathbf{x}_2, \ldots)$. In contrast, our method uses variational autoencoders (VAEs) pre-trained on abundant base examples to estimate the distributions of novel classes. Since intra-class variance is shared across classes and can be modeled with common distributions (Lin et al. 2018), we use a shared VAE to transfer the distributions of base classes to novel classes. Finally, we can sample class prototypes $\mathbf{x}_p$ from the distributions $\mathcal{N}(\mu, \sigma)$ that are robust to the variance of few-shot examples. rm.: remove.

*Work done during internship at Tencent YouTu Lab.
†Corresponding author.

proach TFA (Wang et al. 2020). By revisiting the feature aggregation module in meta-learning frameworks, we propose Class-Agnostic Aggregation (CAA) and Variational Feature Aggregation (VFA) to reduce class bias and improve the robustness to example's variances, respectively.

Feature aggregation is a crucial design in FSOD, which defines how query and support examples interact. Previous works such as Meta R-CNN adopt a class-specific aggregation scheme (Fig. 2 (a)), *i.e.*, query features are aggregated with support features of the same class, ignoring cross-class interactions. In contrast, we propose CAA (Fig. 2 (b)) which allows feature aggregation between different classes. Since CAA encourages the model to learn class-agnostic representations, the bias towards base classes is reduced. Besides, the interactions between different classes simultaneously model class relations so that novel classes will not be confused with base classes.

Based on CAA, we propose VFA which encodes support examples into class-level support features. Our motivation is that intra-class variance (*e.g.* appearance variations) is shared across classes and can be modeled with common distributions (Lin et al. 2018). So we can use base classes' distributions to estimate novel classes' distributions. We achieve this by modeling each class as a common distribution with variational autoencoders (VAEs). We firstly train the VAE on abundant base examples and then fine-tune it on few-shot novel examples. By transferring the learned intra-class variance to novel classes, our method can estimate novel classes' distributions with only a few examples (Fig. 1). Finally, we sample support features from distributions and aggregate them with query features to produce more robust predictions.

We also propose to decouple classification and regression tasks so that our feature aggregation module can focus on learning translation-invariant features without affecting object localization. We conduct extensive experiments on two FSOD datasets, PASCAL VOC (Everingham et al. 2010) and COCO (Lin et al. 2014) to demonstrate the effectiveness of our method. We summarize our contributions as follows:

- We build a strong meta-learning baseline Meta R-CNN++ and propose a simple yet effective Class-Agnostic Aggregation (CAA) method.
- We propose Variational Feature Aggregation (VFA), which transforms instance-wise features into class-level features for robust feature aggregation. To our best knowledge, we are the first to introduce variational feature learning into FSOD.
- Our method significantly improves the baseline Meta R-CNN++ and achieves a new state-of-the-art for FSOD. For example, we outperform the strong baseline by 9%$\sim$16% and previous best results by 3%$\sim$7% on the Novel Set 1 of PASCAL VOC.

Related Work

Generic Object Detection. Object detection has witnessed significant progress in the past decade, which can be roughly divided into two groups: one-stage and two-stage detectors. One-stage detectors predict bounding boxes and class labels by presetting dense anchor boxes (Redmon et al. 2016; Liu et al. 2016; Lin et al. 2017), points (Law and Deng 2018; Zhou, Wang, and Krähenbühl 2019), or directly output sparse predictions (Carion et al. 2020; Chen et al. 2021). Two-stage detectors (Girshick et al. 2014; Girshick 2015; Ren et al. 2017) first generate a set of object proposals with Region Proposal Network (RPN) and then perform proposal-wise classification and regression. However, most generic detectors are trained with abundant samples and not designed for data-scarce scenarios.

Few-Shot Object Detection. Early attempts (Kang et al. 2019; Yan et al. 2019; Wang, Ramanan, and Hebert 2019) in FSOD adopt **meta-learning** architectures. FSRW (Kang et al. 2019) and Meta R-CNN (Yan et al. 2019) aggregate image/RoI-level query features with support features generated by a meta learner. Following works explore different designs of meta-learning architectures, *e.g.*, feature aggregation scheme (Xiao and Marlet 2020; Fan et al. 2020; Hu et al. 2021; Zhang et al. 2021; Han et al. 2021) and feature space augmentation (Li et al. 2021a; Li and Li 2021). Different from meta-learning, Wang et al. propose a simple two-stage **fine-tuning** approach, TFA (Wang et al. 2020). TFA shows that only fine-tuning the last layers can significantly improve the FSOD performance. Due to the simple structure of TFA, a line of works (Sun et al. 2021; Zhu et al. 2021; Qiao et al. 2021; Cao et al. 2021) following TFA are proposed. **In this work**, we build a strong meta-learning baseline that even surpasses the fine-tuning baseline TFA. Then we revisit the feature aggregation scheme and propose two novel feature aggregation methods, CAA and VFA, achieving a new state-of-the-art in FSOD.

Variational Feature Learning. Given an input image/feature, we can transform it into a distribution with VAEs. By sampling features from the distribution, we can model intra-class variance that defines the class's character. The variational feature learning paradigm has been used in various tasks, *e.g.*, zero/few-shot learning (Zhang et al. 2019; Xu et al. 2021; Kim et al. 2019), metric learning (Lin et al. 2018) and disentanglement learning (Ding et al. 2020). In this work, we use VAEs trained on abundant base examples to estimate novel classes' distributions with only a few examples. Besides, we also propose a consistency loss to make the model produce class-specific distributions. To our best knowledge, we are the first to introduce variational feature learning into FSOD.

Background and Meta R-CNN++

Preliminaries

Problem Definition. We follow the FSOD settings in previous works (Yan et al. 2019; Wang et al. 2020). Assume we have a dataset $D = \{(x, y), x \in X, y \in Y\}$ with a set of classes C, where x is the input image and $y = \{c_i, \mathbf{b}_i\}_{i=1}^{N}$ is the corresponding class label c and bounding box $\mathbf{b}$ annotations. We then split the dataset into base classes C_b and novel classes C_n where $C_b \cup C_n = C$ and $C_b \cap C_n = \varnothing$. Generally, we have abundant samples of C_b and K shots samples of C_n (K=1, 2, 3, ...). The goal is to detect objects

setting	TFA	Meta R-CNN*	Meta R-CNN++		
param freeze	✓	✗	✓	✓	✓
cosine cls.	✓	✗	✗	✓	✓
last layer init.	copy	rand	rand	rand	copy
bAP (stage-I)	**80.8**	72.8	77.6	77.6	77.6
bAP (stage-II)	79.6	47.4	64.9	68.2	**76.8**
nAP	39.8	20.7	**42.0**	40.5	41.6

Table 1: Difference analysis between Meta R-CNN and TFA. The results are evaluated under the 1 shot setting of PASCAL VOC Novel Set 1. stage-I and stage-II: base training and fine-tuning stages. *: Our re-implemented results.

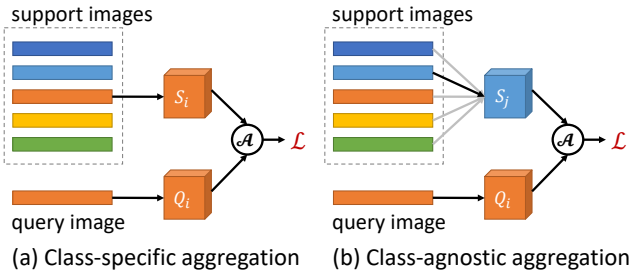

Figure 2: Illustration of two feature aggregation methods. S_i/Q_i: support and query features of class i. $\mathcal{A}$: feature aggregation. $\mathcal{L}$: loss functions.

of C_n with only K shots annotated instances. Existing few-shot detectors usually adopt a two-stage training paradigm: base training and few-shot fine-tuning, where the representations learned from C_b are transferred to detect novel objects in the fine-tuning stage.

Meta-Learning Based FSOD. We take Meta R-CNN (Yan et al. 2019) for an example. As shown in Fig. 3, the main framework is a siamese network with a query feature encoder $\mathcal{F}_Q$, a support feature encoder $\mathcal{F}_S$, a feature aggregator $\mathcal{A}$ and a detection head $\mathcal{F}_D$. Typically, $\mathcal{F}_Q$ and $\mathcal{F}_S$ share most parameters and $\mathcal{A}$ refers to the channel-wise product operation. Meta R-CNN follows the episodic training paradigm (Vinyals et al. 2016). Each episode is composed of a set of support images and binary masks of annotated objects, $\{x_i, M_i\}_{i=1}^N$, where N is the number of training classes. Specifically, we first feed the support set $\{x_i, M_i\}_{i=1}^N$ to $\mathcal{F}_S$ to generate class-specific support features $\{S_i\}_{i\in C}$, and the query image to $\mathcal{F}_Q$ to generate a set of RoI features $\{Q^m\}$ (m is the index of RoIs). Then we aggregate each Q^m and S_i with the feature aggregator $\mathcal{A}$. Finally, the aggregated features $\widetilde{Q}_i^m$ are fed to the detection head $\mathcal{F}_D$ to produce final predictions.

Meta R-CNN++: Stronger Meta-Learning Baseline

Meta-learning has proved a promising approach, but the fine-tuning based approach receives more and more attention recently due to its superior performance. Here we aim to bridge the gap between the two approaches. We choose Meta R-CNN and TFA as baselines and explore how to build a strong FSOD baseline with meta-learning.

Although both methods follow a two-stage training paradigm, TFA optimizes the model with advanced techniques in the fine-tuning stage: **(a)** TFA freezes most network parameters, and only trains the last classification and regression layers so that the model will not overfit to few-shot examples. **(b)** Instead of randomly initializing the classification layer, TFA copies pre-trained weights of base classes and only initializes the weights of novel classes. **(c)** TFA adopts cosine classifier (Gidaris and Komodakis 2018) rather than a linear classifier.

Considering the success of TFA, we build Meta R-CNN++, which follows the architecture of Meta R-CNN but aligns most hyper-parameters with TFA. Here we explore different design choices to mitigate the gap between the two approaches, shown in Tab. 1. **(a) Parameter freeze.** By adopting the same parameter freezing strategy, Meta R-CNN++ significantly outperforms Meta R-CNN and even achieves higher novel AP than TFA. **(b) Cosine classifier.** Different from TFA, Meta R-CNN++ with the cosine classifier does not surpass the linear classifier in nAP (41.6 *vs.* 42.0), but its performance on base classes is better than the linear classifier (68.2 *vs.* 64.9). **(c) Alleviate base forgetting.** We follow TFA and copy the pre-trained classifier weights of base classes. We find Meta R-CNN++ can also maintain the performance on base classes (76.8 *vs.* 77.6).

The above experiments indicate that meta-learning remains a promising approach for FSOD as long as we carefully handle the fine-tuning stage. Therefore, we choose Meta R-CNN++ as our baseline in the following sections.

The Proposed Approach

Class-Agnostic Aggregation

Feature aggregation is an important module in meta-learning based FSOD (Kang et al. 2019; Yan et al. 2019). Many works adopt a class-specific aggregation (CSA) scheme. Let us assume that a query image has an object of class $C_Q = \{i\}$ and the corresponding RoI features $\{Q_i^m\}$. In the training phase, as shown in Fig. 2 (a), CSA aggregates each RoI feature Q_i^m with the support features S_i of the same class: $\widetilde{Q}_{ii}^m = \mathcal{A}(Q_i^m, S_i)$. In the testing phase, CSA aggregates the RoI feature with support features of all classes: $\widetilde{Q}_{ij}^m = \mathcal{A}(Q_i^m, S_j), j \in C$, and each support feature S_j is to predict objects of its corresponding class. Notably, if the query image contains multiple classes, CSA aggregates the query features with each support feature in C_Q: $\widetilde{Q}_{ij}^m = \mathcal{A}(Q_i^m, S_j), j \in C_Q$. But CSA still follows the class-specific way, as support features not belonging to C_Q will never be aggregated with the query feature.

As discussed before, the learned models are usually biased to base classes due to the imbalance between base and novel classes. Therefore, we revisit CSA and propose a simple yet effective Class-Agnostic Aggregation (CAA). See Fig. 2 (b) for an instance, CAA allows feature aggregation between different classes, which encourages the model to learn class-agnostic representations and thereby reduces the class bias. Besides, the interactions between different classes can simultaneously model class relations so that

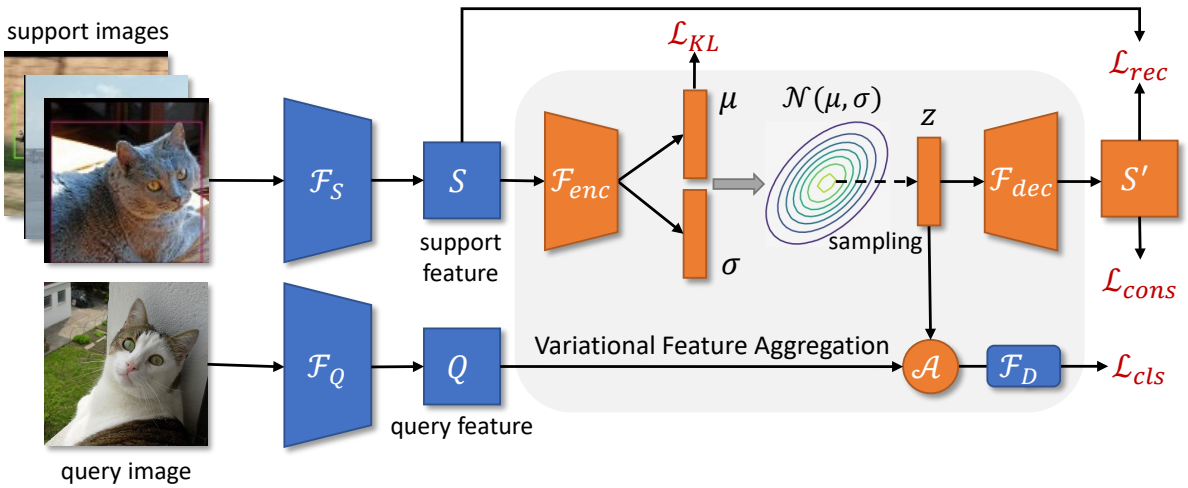

Figure 3: Overview of our framework. $\mathcal{F}_Q$ and $\mathcal{F}_S$ denote query and support feature extractors, respectively. $\mathcal{F}_{enc}$ and $\mathcal{F}_{dec}$ are the variational feature encoder and decoder. $\mathcal{F}_D$: the detection head. $\mathcal{A}$: feature aggregation. Note that we do not visualize RPN and the regression branch for simplicity.

novel classes will not confuse with base classes. Formally, for each RoI feature Q_i^m of class $i \in C$ and a set of support features $\{S_j\}_{j \in C}$, we **randomly** select a support feature S_{j^*} of class j^* to aggregate with the query feature,

$$\widetilde{Q}_{ij^*}^m = \mathcal{A}(Q_i^m, S_{j^*}), j^* \in C. \quad (1)$$

Then we feed the aggregated feature $\widetilde{Q}_{ij^*}^m$ to the detection head $\mathcal{F}_D$ to output classification scores $\mathbf{p} = \mathcal{F}_D(\widetilde{Q}_{ij^*}^m)$, which is supervised with the label of class i. Note that CAA is used for training; the testing phase still follows CSA.

Variational Feature Aggregation

Prior works usually encode support examples into single feature vectors that are difficult to represent the whole class distribution. Especially when the data is scarce and example's variations are large, we cannot make an accurate estimation of class centers. Inspired by recent progress in variational feature learning (Lin et al. 2018; Zhang et al. 2019; Xu et al. 2021), we transform support features into class distributions with VAEs. Since the estimated distribution is not biased to specific examples, features sampled from the distribution are robust to the variance of support examples. Then we can sample class-level features for robust feature aggregation. The framework of VFA is shown in Fig. 3.

Variational Feature Learning. Formally, we aim to transform the support feature S into a class distribution $\mathcal{N}$, and sample the variational feature z from $\mathcal{N}$ for feature aggregation. We optimize the model in a similar way to VAEs, but our goal is to sample the latent variable z instead of the reconstructed feature $S^{'}$. Following the definition of VAEs, we assume z is generated from a prior distribution $p(z)$ and S is generated from a conditional distribution $p(S|z)$. As the process is hidden and z is unknown, we model the posterior distribution with variational inference. More specifically, we approximate the true posterior distribution $p(z|S)$ with another distribution $q(z|S)$ by minimizing the Kullback-Leibler (KL) divergence:

$$D_{\mathrm{KL}}(q(z|S)||p(z|S)) = \int q(z|S) \log \frac{q(z|S)}{p(z|S)}, \quad (2)$$

which is equivalent to maximizing the evidence lower bound (ELBO):

$$ELBO = \mathbb{E}_{q(z|S)}[\log p(S|z))] - D_{\mathrm{KL}}(q(z|S)||p(z)). \quad (3)$$

Here we assume the prior distribution of z is a centered isotropic multivariate Gaussian, $p(z) = \mathcal{N}(0, I)$, and set the posterior distribution $q(z|S)$ to be a multivariate Gaussian with diagonal covariance: $q(z|S) = \mathcal{N}(\mu, \sigma)$. The parameters μ and σ can be implemented by a feature encoder $\mathcal{F}_{enc}$: $\mu, \sigma = \mathcal{F}_{enc}(S)$. Then we obtain the variational feature z with the reparameterization trick (Kingma and Welling 2013): $z = \mu + \sigma \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. The first term of Eq. 3 can be simplified to a reconstruction loss $\mathcal{L}_{rec}$ which is usually defined as the L2 distance between the input S and the reconstructed target $S^{'}$,

$$\mathcal{L}_{rec} = \|S - S^{'}\| = \|S - \mathcal{F}_{dec}(z)\|, \quad (4)$$

where $\mathcal{F}_{dec}$ denotes a feature decoder. As for the second term of Eq. 3, we directly minimize the KL divergence of $q(z|S)$ and $p(z)$,

$$\mathcal{L}_{\mathrm{KL}} = D_{\mathrm{KL}}(q(z|S)||p(z)), \quad (5)$$

which forces the variation feature z to follow a normal distribution.

By optimizing the two objectives, $\mathcal{L}_{rec}$ and $\mathcal{L}_{\mathrm{KL}}$, we transform the support feature S into a distribution $\mathcal{N}$. Then we can sample the variational feature z from $\mathcal{N}$. Since z still lacks class-specific information, we apply a **consistency loss** $\mathcal{L}_{cons}$ to the reconstructed feature $S^{'}$, which is defined as the cross-entropy between $S^{'}$ and its class label c,

$$\mathcal{L}_{cons} = \mathcal{L}_{\mathrm{CE}}(\mathcal{F}_{cls}^{S^{'}}(S^{'}), c), \quad (6)$$

Method / Shots	Backbone	Novel Set 1					Novel Set 2					Novel Set 3					Avg.
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10	
FSRW (Kang et al. 2019)	YOLOv2	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9	28.4
MetaDet (Wang et al. 2019)	VGG16	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	31.0
Meta R-CNN (Yan et al. 2019)	ResNet-101	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	31.1
TFA w/ cos (Wang et al. 2020)	ResNet-101	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	39.9
MPSR (Wu et al. 2020)	ResNet-101	41.7	-	51.4	55.2	61.8	24.4	-	39.2	39.9	47.8	35.6	-	42.3	48.0	49.7	-
Retentive (Fan et al. 2021)	ResNet-101	42.4	45.8	45.9	53.7	56.1	21.7	27.8	35.2	37.0	40.3	30.2	37.6	43.0	49.7	50.1	41.1
Halluc (Zhang and Wang 2021)	ResNet-101	47.0	44.9	46.5	54.7	54.7	26.3	31.8	37.4	37.4	41.2	40.4	42.1	43.3	51.4	49.6	43.2
CGDP+FSCN (Li et al. 2021b)	ResNet-101	40.7	45.1	46.5	57.4	62.4	27.3	31.4	40.8	42.7	46.3	31.2	36.4	43.7	50.1	55.6	43.8
CME (Li et al. 2021a)	ResNet-101	41.5	47.5	50.4	58.2	60.9	27.2	30.2	41.4	42.5	46.8	34.3	39.6	45.1	48.3	51.5	44.4
SRR-FSD (Zhu et al. 2021)	ResNet-101	47.8	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4	44.8
FSOD-UP (Wu et al. 2021)	ResNet-101	43.8	47.8	50.3	55.4	61.7	31.2	30.5	41.2	42.2	48.3	35.5	39.7	43.9	50.6	53.5	45.0
FSCE (Sun et al. 2021)	ResNet-101	44.2	43.8	51.4	61.9	63.4	27.3	29.5	43.5	44.2	50.2	37.2	41.9	47.5	54.6	58.5	46.6
QA-FewDet (Han et al. 2021)	ResNet-101	42.4	51.9	55.7	62.6	63.4	25.9	37.8	46.6	48.9	51.1	35.2	42.9	47.8	54.8	53.5	48.0
FADI (Cao et al. 2021)	ResNet-101	50.3	54.8	54.2	59.3	63.2	30.6	35.0	40.3	42.8	48.0	45.7	49.7	49.1	55.0	**59.6**	49.2
Zhang et al. (Zhang et al. 2021)	ResNet-101	48.6	51.1	52.0	53.7	54.3	**41.6**	45.4	45.8	46.3	48.0	46.1	51.7	52.6	54.1	55.0	49.8
Meta FR-CNN (Han et al. 2022)	ResNet-101	43.0	54.5	60.6	66.1	65.4	27.7	35.5	46.1	47.8	51.4	40.6	46.4	53.4	**59.9**	58.6	50.5
DeFRCN (Qiao et al. 2021)	ResNet-101	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	51.9
VFA (Ours)	ResNet-101	**57.7**	**64.6**	**64.7**	**67.2**	**67.4**	41.4	**46.2**	**51.1**	51.8	**51.6**	**48.9**	**54.8**	**56.6**	59.0	58.9	**56.1**

Table 2: Results on PASCAL VOC. The results are sorted by the averaged score (Avg.). See our appendix for the generalized FSOD results.

where $\mathcal{F}_{cls}^{S^{'}}$ denotes a linear classifier. The introduction of $\mathcal{L}_{cons}$ transforms the learned distributions into class-specific distributions. The support feature S_i is forced to approximate a parameterized distribution $\mathcal{N}(\mu_i, \sigma_i)$ of class i, so that the sampled z can preserve class-specific information.

Variational Feature Aggregation. Since the support features are transformed into class distributions, we can sample features from the distribution and aggregate them with query features. Compared with the original support feature S and reconstructed feature $S^{'}$, the latent variable z contains more generic features of the class (Zhang et al. 2019; Lin et al. 2018), which is robust to the variance of support examples.

Specifically, VFA follows the class-agnostic approach in CAA but aggregates the query feature Q with a variational feature z. Given a query feature Q_i of class i and support feature S_j of class j, we firstly approximate the class distribution $\mathcal{N}(\mu_j, \sigma_j)$ and sample a variational feature $z_j = \mu_j + \sigma_j$ from $\mathcal{N}(\mu_j, \sigma_j)$. Then we aggregate them together with the following equation:

$$\widetilde{Q}_{ij} = \mathcal{A}(Q_i, z_j) = Q_i \odot \mathrm{sig}(z_j), \tag{7}$$

where $\odot$ means channel-wise multiplication and sig is short for the *sigmoid* operation. In the training phase, we randomly select a support feature S_j (*i.e.*, one support class j) for aggregation. In the testing phase (especially $K > 1$), we average K support features of class j into one $\bar{S}_j$, and approximate the distribution $\mathcal{N}(\mu_j, \sigma_j)$ with the averaged feature, $\mu_j, \sigma_j = \mathcal{F}_{enc}(\bar{S}_j)$. Instead of adopting complex distribution estimation methods, we find the averaging approach works well in our method.

Network and Objective. VFA only introduces a light encoder $\mathcal{F}_{enc}$ and decoder $\mathcal{F}_{dec}$. $\mathcal{F}_{enc}$ contains a linear layer and two parallel linear layers to produce μ and σ, respectively. $\mathcal{F}_{dec}$ consists of two linear layers to generate the reconstructed feature $S^{'}$. We keep all layers the same dimension (2048 by default). VFA is trained in an end-to-end manner with the following multi-task loss:

$$\mathcal{L} = \mathcal{L}_{rpn} + \mathcal{L}_{reg} + \mathcal{L}_{cls} + \mathcal{L}_{cons} + \mathcal{L}_{rec} + \alpha\mathcal{L}_{KL}, \tag{8}$$

where $\mathcal{L}_{rpn}$ is the total loss of RPN, $\mathcal{L}_{reg}$ is the regression loss, and α is a weight coefficient (α=2.5$\times 10^{-4}$ by default).

Classification-Regression Decoupling

Generally, the detection head $\mathcal{F}_D$ contains a shared feature extractor $\mathcal{F}_{share}$ and two separate network $\mathcal{F}_{cls}$ and $\mathcal{F}_{reg}$ for classification and regression, respectively. In previous works, the aggregated feature is fed to $\mathcal{F}_D$ to produce both classification scores and bounding boxes. However, the classification task requires translation-invariant features, while regression needs translation-covariant features (Qiao et al. 2021). Since support features are always translation-invariant to represent class centers, the aggregated feature harms the regression task. Therefore, we decouple the two tasks in the detection head. Let Q and $\widetilde{Q}$ denote the original and aggregated query features. Previous methods take $\widetilde{Q}$ for both tasks, where the classification score $\mathbf{p}$ and predicted bounding boxes $\mathbf{b}$ are defined as:

$$\mathbf{p} = \mathcal{F}_{cls}(\mathcal{F}_{share}(\widetilde{Q})), \mathbf{b} = \mathcal{F}_{reg}(\mathcal{F}_{share}(\widetilde{Q})). \tag{9}$$

To decouple these tasks, we adopt separate feature extractors and use the original query feature Q for regression,

$$\mathbf{p} = \mathcal{F}_{cls}(\mathcal{F}_{share}^{cls}(\widetilde{Q})), \mathbf{b} = \mathcal{F}_{reg}(\mathcal{F}_{share}^{reg}(Q)), \tag{10}$$

where $\mathcal{F}_{share}^{cls}$ and $\mathcal{F}_{share}^{reg}$ are the feature extractor for classification and regression, respectively.

Experiments and Analysis

Experimental Setting

Datasets. We evaluate our method on PASCAL VOC (Everingham et al. 2010) and COCO (Lin et al. 2014), following

Method / Shots	10	30
Fine-tuning		
MPSR (Wu et al. 2020)	9.8	14.1
TFA w/ cos (Wang et al. 2020)	10.0	13.7
Retentive (Fan et al. 2021)	10.5	13.8
FSOD-UP (Wu et al. 2021)	11.0	15.6
SRR-FSD (Zhu et al. 2021)	11.3	14.7
CGDP+FSCN (Li et al. 2021b)	11.3	15.1
FSCE (Sun et al. 2021)	11.9	16.4
FADI (Cao et al. 2021)	12.2	16.1
DeFRCN (Qiao et al. 2021)	18.5	22.6
Meta-learning		
FSRW (Kang et al. 2019)	5.6	9.1
MetaDet (Wang, Ramanan, and Hebert 2019)	7.1	11.3
Meta R-CNN (Yan et al. 2019)	8.7	12.4
QA-FewDet (Han et al. 2021)	11.6	16.5
FSDetView (Xiao and Marlet 2020)	12.5	14.7
Meta FR-CNN (Han et al. 2022)	12.7	16.6
DCNet (Hu et al. 2021)	12.8	18.6
CME (Li et al. 2021a)	15.1	16.9
VFA (Ours)	**16.2**	**18.9**

Table 3: Results on COCO. The backbone is the same as Tab. 2. The results are sorted by 10-shot nAP. See our appendix for the generalized FSOD results.

previous works (Kang et al. 2019; Wang et al. 2020). We use the data splits and annotations provided by TFA (Wang et al. 2020) for a fair comparison. For PASCAL VOC, we split 20 classes into three groups, where each group contains 15 base classes and 5 novel classes. For each novel set, we have K={1, 2, 3, 5, 10} shots settings. For COCO, we set 60 categories disjoint with PASCAL VOC as base classes and the remaining 20 as novel classes. We have K={10, 30} shots settings.

Evaluation Metrics. For PASCAL VOC, we report the Average Precision at IoU=0.5 of base classes (bAP) and novel classes (nAP). For COCO, we report the mean AP at IoU=0.5:0.95 of novel classes (nAP).

Implementation Details. We implement our method with Mmdetection (Chen et al. 2019). The backbone is ResNet-101 (He et al. 2016) pre-trained on ImageNet (Russakovsky et al. 2015). We adopt SGD as the optimizer with batch size 32, learning rate 0.02, momentum 0.9 and weight decay 1e-4. The learning rate is changed to 0.001 in the few-shot fine-tuning stage. We fine-tune the model with {400, 800, 1200, 1600, 2000} iterations for K={1, 2, 3, 5, 10} shots in PASCAL VOC, and {10000, 20000} iterations for K={10, 30} shots in COCO. We keep other hyper-parameters the same as Meta R-CNN (Yan et al. 2019) if not specified.

Main Results

PASCAL VOC. As shown in Tab. 2, VFA significantly outperforms existing methods. VFA achieves the best (13/16) or second-best (3/16) results on all settings. In Novel Set 1, VFA outperforms previous best results by 3.2%∼7.1%. Our 2-shot result even surpasses previous best 10-shot results (64.6% *vs.* 63.4%), which indicates that our method is more robust to the variance of few-shot examples. Besides, we notice that our gains are stable and consistent. This phenomenon demonstrates that VFA is not biased to specified class sets and can be generalized to more common scenarios. Furthermore, VFA obtains a 56.1% average score and surpasses the second-best result by 4.2%, which further demonstrates its effectiveness.

COCO. As shown in Tab. 3, VFA achieves the best nAP among meta-learning based methods and second-best results among all methods. We notice that a fine-tuning based method, DeFRCN (Qiao et al. 2021), outperforms our method in nAP. To concentrate on the feature aggregation module in meta-learning, we do not utilize advanced techniques, *e.g.*, the gradient decoupled layer (Qiao et al. 2021) in DeFRCN. We believe the performance of VFA can be further boosted with more advanced techniques.

Method	CRD	CAA	VFA	Shots 1	3	5
Meta R-CNN++				42.0	56.5	58.3
Ours	✓			46.0	61.7	62.3
	✓	✓		51.3	62.8	66.4
	✓	✓	✓	**57.7**	**64.7**	**67.2**

Table 4: Effect of different modules.

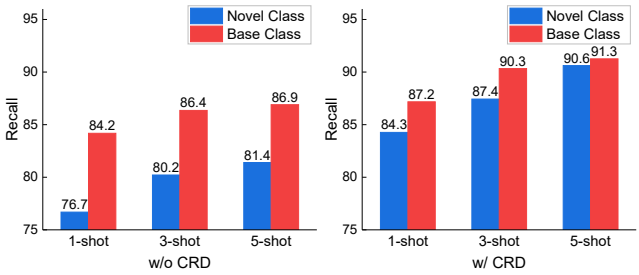

Figure 4: Comparisons of recall without/with CRD.

Ablation Studies

We conduct a series of ablation experiments on Novel Set 1 of PASCAL VOC.

Effect of different modules. As shown in Tab. 4, we evaluate the effect of different modules by gradually applying the proposed modules to Meta R-CNN++. Although Meta R-CNN++ is competitive enough, we show **CRD** improves the performance on nAP, where the absolute gains exceed 4%. Besides, we find CRD significantly improves the recall on all classes (Fig. 4) and narrows the gap between base and novel classes because it uses separate networks to learn translation-invariant and -covariant features. Then, we apply **CAA** to the model and obtain further improvements. The confusions between different classes are reduced. Finally, we build **VFA** and achieve a new state-of-the-art. The 1-shot performance is even comparable with 5-shot Meta R-CNN++ in nAP, indicating that VFA is robust to the variance of support examples especially when the data is scarce.

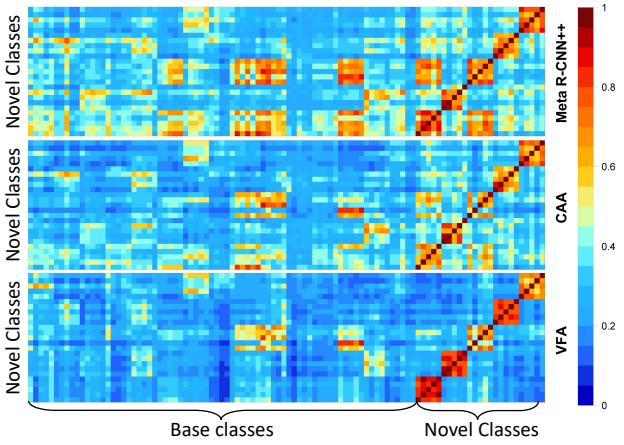

Figure 5: Similarity matrix visualization. We calculate cosine similarities of support features in the 5-shot setting of PASCAL VOC Novel Set 1. *sofa, motorbike, cow, bus* and *bird* are novel classes. Warmer color denotes higher similarity. Zoom in for details.

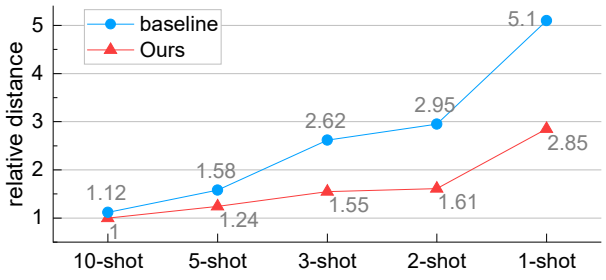

Figure 6: The distance from the estimated prototype of K-shot examples to the real class center. For each novel class, we take the mean feature of all training examples as its real class center. Our 10-shot result is the reference distance, while other results are relative distances. We only report the averaged distance of all novel classes for simplicity.

Visual analysis of different feature aggregation. Fig. 5 gives a visual analysis of different feature aggregation methods. Due to the imbalance between base and novel classes, some novel classes are confused with base classes in Meta R-CNN++ (with CSA), *e.g.*, a novel classe, *cow* have higher similarity (>0.8) with *horse* and *sheep*. In contrast, CAA reduces class bias and confusion by learning class-agnostic representations. The inter-class similarities are also reduced so that a novel example will not be classified to base classes. Finally, we use VFA to transforms support examples into class distributions. By learning intra-class variances from abundant base examples, we can estimate novel classes' distributions even with a few examples. In Fig. 5 (bottom), we can see VFA significantly improves intra-class similarities.

Robust and accurate class prototypes. In the testing phase, detectors take the mean feature of K-shot examples as the class prototype. As shown in Fig. 6, our estimated class prototypes are more robust and accurate than the baseline. The distances to real class centers do not increase much as the shot decreases, because our method can fully leverage base classes' distributions to estimate novel classes' distributions. The prototypes sampled from distributions are robust to the variance of support examples. While the baseline is sensitive to the number of support examples.

Features		S	$S^{'}$	μ	σ	$\widetilde{z}$	z
bAP		**78.8**	78.1	78.6	78.3	78.0	78.6
nAP	1	55.2	54.4	56.6	55.4	53.0	**57.7**
	3	63.7	63.6	63.7	**64.9**	63.2	64.7
	5	66.6	66.9	66.7	66.9	66.3	**67.2**
	avg.	61.8	61.6	62.3	62.4	60.8	**63.2**

Table 5: Comparisons of different support features. S and $S^{'}$ are the original and reconstructed features. μ, σ, $\widetilde{z} = \mu + \epsilon \cdot \sigma$ and $z = \mu + \sigma$ are latent variables. avg.: The average score.

Setting / Shots		1	3	5
w/o VFA		51.3	62.8	66.4
w/ VFA	w/o $\mathcal{L}_{cons}$	53.6	64.3	66.7
	$\mathcal{L}_{cons}$ on S	52.9	64.1	**67.3**
	$\mathcal{L}_{cons}$ on $S^{'}$	**57.7**	**64.7**	67.2

Table 6: Effect of $\mathcal{L}_{cons}$. w/o: without. $\mathcal{L}_{cons}$ on S/$S^{'}$: apply $\mathcal{L}_{cons}$ to S or $S^{'}$. The results are averages of multiple runs.

Which feature to aggregate? In Tab. 5, we explore different features for aggregation. All types of features achieve comparable performance on base classes but vary on novel classes. The performance of original feature S and reconstructed feature $S^{'}$ lag behind the latent encoding μ, σ and z. We hypothesize that the latent encoding contains more class-generic features. Besides, $\widetilde{z} = \mu + \epsilon \cdot \sigma$ performs worst among these features due to its indeterminate inference process. Instead, a simplified version $z = \mu + \sigma$ achieves satisfactory results, which is the default setting of VFA.

Effect of $\mathcal{L}_{cons}$. We use a shared VAE to encode support features but still need to preserve class-specific information. Therefore, we add a consistency loss $\mathcal{L}_{cons}$ to produce class-wise distributions. Tab. 6 shows that $\mathcal{L}_{cons}$ is important for VFA. $\mathcal{L}_{cons}$ applied to $S^{'}$ forces the model to produce class-conditional distributions so that the latent variable z can retrain meaningful information to represent class centers.

Design of VFA. The variational feature encoder $\mathcal{F}_{enc}$ and decoder $\mathcal{F}_{dec}$ are not sensitive to the number and dimension of hidden layers. Please see our appendix for details.

Conclusion

This paper revisits feature aggregation schemes in meta-learning based FSOD and proposes Class-Agnostic Aggregation (CAA) and Variational Feature Aggregation (VFA). CAA can reduce class bias and confusion between base and novel classes; VFA transforms instance-wise support features into class distributions for robust feature aggregation. Extensive experiments on PASCAL VOC and COCO demonstrate our effectiveness.

Acknowledgements

This work was partially supported by National Nature Science Foundation of China under the grants 41820104006, 61922065, and U22B2011.

References

Cao, Y.; Wang, J.; Jin, Y.; Wu, T.; Chen, K.; Liu, Z.; and Lin, D. 2021. Few-Shot Object Detection via Association and DIscrimination. *NeurIPS*, 34.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*, 213–229. Springer.

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; Zhang, Z.; Cheng, D.; Zhu, C.; Cheng, T.; Zhao, Q.; Li, B.; Lu, X.; Zhu, R.; Wu, Y.; Dai, J.; Wang, J.; Shi, J.; Ouyang, W.; Loy, C. C.; and Lin, D. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*.

Chen, T.; Saxena, S.; Li, L.; Fleet, D. J.; and Hinton, G. 2021. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*.

Ding, Z.; Xu, Y.; Xu, W.; Parmar, G.; Yang, Y.; Welling, M.; and Tu, Z. 2020. Guided variational autoencoder for disentanglement learning. In *CVPR*, 7920–7929.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV*, 88(2): 303–338.

Fan, Q.; Zhuo, W.; Tang, C.-K.; and Tai, Y.-W. 2020. Few-shot object detection with attention-RPN and multi-relation detector. In *CVPR*, 4013–4022.

Fan, Z.; Ma, Y.; Li, Z.; and Sun, J. 2021. Generalized few-shot object detection without forgetting. In *CVPR*, 4527–4536.

Gidaris, S.; and Komodakis, N. 2018. Dynamic few-shot visual learning without forgetting. In *CVPR*, 4367–4375.

Girshick, R. 2015. Fast R-CNN. In *ICCV*, 1440–1448.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 580–587.

Han, G.; He, Y.; Huang, S.; Ma, J.; and Chang, S.-F. 2021. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *ICCV*, 3263–3272.

Han, G.; Huang, S.; Ma, J.; He, Y.; and Chang, S.-F. 2022. Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. In *AAAI*, volume 36, 780–789.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hu, H.; Bai, S.; Li, A.; Cui, J.; and Wang, L. 2021. Dense relation distillation with context-aware aggregation for few-shot object detection. In *CVPR*, 10185–10194.

Kang, B.; Liu, Z.; Wang, X.; Yu, F.; Feng, J.; and Darrell, T. 2019. Few-shot object detection via feature reweighting. In *ICCV*, 8420–8429.

Kim, J.; Oh, T.-H.; Lee, S.; Pan, F.; and Kweon, I. S. 2019. Variational prototyping-encoder: One-shot learning with prototypical images. In *CVPR*, 9462–9470.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Law, H.; and Deng, J. 2018. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 734–750.

Li, A.; and Li, Z. 2021. Transformation invariant few-shot object detection. In *CVPR*, 3094–3102.

Li, B.; Yang, B.; Liu, C.; Liu, F.; Ji, R.; and Ye, Q. 2021a. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *CVPR*, 7363–7372.

Li, Y.; Zhu, H.; Cheng, Y.; Wang, W.; Teo, C. S.; Xiang, C.; Vadakkepat, P.; and Lee, T. H. 2021b. Few-shot object detection via classification refinement and distractor retreatment. In *CVPR*, 15395–15403.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *ICCV*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Lin, X.; Duan, Y.; Dong, Q.; Lu, J.; and Zhou, J. 2018. Deep variational metric learning. In *ECCV*, 689–704.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. SSD: Single shot multibox detector. In *ECCV*, 21–37.

Qiao, L.; Zhao, Y.; Li, Z.; Qiu, X.; Wu, J.; and Zhang, C. 2021. DeFRCN: Decoupled Faster R-CNN for Few-Shot Object Detection. In *ICCV*, 8681–8690.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *CVPR*, 779–788.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, 1137–1149.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *IJCV*, 115(3): 211–252.

Sun, B.; Li, B.; Cai, S.; Yuan, Y.; and Zhang, C. 2021. Fsce: Few-shot object detection via contrastive proposal encoding. In *CVPR*, 7352–7362.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *NeurIPS*, 3630–3638.

Wang, X.; Huang, T. E.; Darrell, T.; Gonzalez, J. E.; and Yu, F. 2020. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*.

Wang, Y.-X.; Ramanan, D.; and Hebert, M. 2019. Meta-learning to detect rare objects. In *ICCV*, 9925–9934.

Wu, A.; Han, Y.; Zhu, L.; and Yang, Y. 2021. Universal-prototype enhancing for few-shot object detection. In *ICCV*, 9567–9576.

Wu, J.; Liu, S.; Huang, D.; and Wang, Y. 2020. Multi-scale positive sample refinement for few-shot object detection. In *ECCV*, 456–472. Springer.

Xiao, Y.; and Marlet, R. 2020. Few-shot object detection and viewpoint estimation for objects in the wild. In *ECCV*, 192–210. Springer.

Xu, J.; Le, H.; Huang, M.; Athar, S.; and Samaras, D. 2021. Variational Feature Disentangling for Fine-Grained Few-Shot Classification. In *ICCV*, 8812–8821.

Yan, X.; Chen, Z.; Xu, A.; Wang, X.; Liang, X.; and Lin, L. 2019. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *ICCV*, 9577–9586.

Zhang, J.; Zhao, C.; Ni, B.; Xu, M.; and Yang, X. 2019. Variational few-shot learning. In *ICCV*, 1685–1694.

Zhang, L.; Zhou, S.; Guan, J.; and Zhang, J. 2021. Accurate few-shot object detection with support-query mutual guidance and hybrid loss. In *CVPR*, 14424–14432.

Zhang, W.; and Wang, Y.-X. 2021. Hallucination improves few-shot object detection. In *CVPR*, 13008–13017.

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as Points. *arXiv preprint arXiv:1904.07850*.

Zhu, C.; Chen, F.; Ahmed, U.; Shen, Z.; and Savvides, M. 2021. Semantic relation reasoning for shot-stable few-shot object detection. In *CVPR*, 8782–8791.