Uncertainty-based Forgetting Mitigation for Generalized Few-Shot Object Detection

Karim Guirguis [1,2] George Eskandar [3] Mingyang Wang [1] Matthias Kayser [1]
Eduardo Monari [1] Bin Yang [3] Jürgen Beyerer [2,4]
Robert Bosch GmbH[1] Karlsruhe Institute of Technology[2] University of Stuttgart[3] Fraunhofer IOSB[4]

Abstract

Generalized Few-Shot Object Detection (G-FSOD) seeks to jointly detect base classes with abundant data and novel classes with limited data. Due to data scarcity, predictive uncertainties are more pronounced in G-FSOD than in conventional object detection. Unaccounting for these uncertainties leads to degraded overall detection performance and forgetting the base classes. However, previous G-FSOD works have not exploited these uncertainties. Upon examining the basic two-stage G-FSOD framework, which includes a Region Proposal Network (RPN) and a subsequent R-CNN, we observe that a straightforward integration of uncertainty estimation leads to detrimental performance. To this end, we first increase the model capacity by increasing the depth of the RPN and cascading multiple R-CNNs in an end-to-end manner. Next, we interleave the stages with uncertainty estimation and attention blocks. The aim is to progressively refine the proposals by exploiting the estimated uncertainties while attending to the discriminative features through the attention mechanism. Extensive experiments on the well-established G-FSOD benchmarks, MS-COCO and PASCAL-VOC, show that our proposed method sets a new G-FSOD standard.

1. Introduction

Acquiring diverse and extensive labeled datasets to train data-hungry Object Detection (OD) models [1, 2, 7, 8, 18, 22–24, 26] can be time-consuming, labor-intensive, and costly in numerous applications, such as autonomous driving and industrial production. Few-Shot Object Detection (FSOD) [4, 13, 25, 28, 32] strives to emulate the cognitive capabilities of humans by rapidly acquiring meaningful representations, provided by a limited number of training examples. Specifically, FSOD utilizes prior knowledge by pre-training on a base dataset containing abundant training samples. This acquired knowledge is then leveraged in the subsequent novel training phase, allowing the model to learn novel classes using limited data rapidly. Although meta-learning and transfer learning paradigms have achieved notable success in FSOD, most methods prioritize the detection performance of novel classes, often neglecting the performance of base classes. This may lead to catastrophic forgetting, where the model loses knowledge of the base classes, compromising the safety of the operational system. Another essential yet entirely neglected aspect of FSOD, is the reliability of the model predictions in the presence of model and data uncertainties.

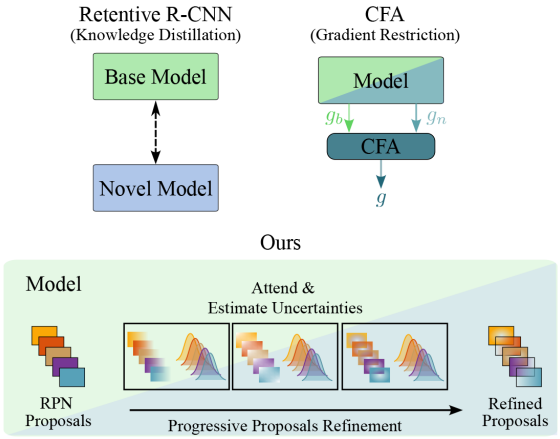

Figure 1. An abstract comparison of the current G-FSOD works tackling the base forgetting, Retentive R-CNN [5] and CFA [9], compared to the proposed approach.

While the primary objective of Generalized Few-Shot Object Detection (G-FSOD) [5, 9, 10, 20, 28] is to detect both base and novel classes jointly, recent works focus explicitly on mitigating the previously mentioned problem of forgetting base classes. These approaches can be categorized into base data-dependent [5, 9] and base data-free [10] approaches. Retentive R-CNN [5] is a base data-dependent approach that adopts a student-teacher framework to retain base knowledge while learning new classes at the expense of added computational and memory requirements. CFA [9]

introduces a plug-and-play gradient update rule to restrain the update gradients during novel training. By constraining the gradients, CFA aims to find optima with better overall detection performance and less forgetting. On the other hand, NIFF [10] is a base data-free approach that trains a lightweight base feature generator using statistics from the base model, eliminating the need for base data during the novel training phase. These approaches provide different strategies for addressing the challenges of forgetting and retaining base knowledge in G-FSOD whether or not base data is available. However, none of these methods address the inherent data and model uncertainties affecting the reliability of the network predictions.

Predictive uncertainties [14] can be decomposed into aleatoric uncertainties and epistemic uncertainties. The former represents the inherent variability in the data itself, such as sensor noise. Aleatoric uncertainty is commonly addressed by explicitly incorporating it into the neural network as learnable parameters associated with the predicted outputs. For instance, in the context of OD, these additional parameters can represent the aleatoric uncertainty related to class probabilities or bounding box coordinates [11, 15]. Epistemic uncertainty, on the other hand, captures the uncertainty arising from the lack of knowledge or limited training data. In particular for OD, epistemic uncertainty is typically addressed by incorporating dropouts [19] during the training phase of the model [11, 15], where a portion of the neurons is randomly dropped during training, effectively creating an ensemble of models. By examining the variance among the predictions generated by these diverse models, we can approximate the level of epistemic uncertainty in the model. However, predictive uncertainties have been mainly exploited in standard OD [6, 11, 15, 30] and have not yet been addressed in FSOD or G-FSOD scenarios. Given that the majority of G-FSOD approaches are based on the two-stage Faster R-CNN architecture [24], we argue that the introduction of novel classes results in higher epistemic uncertainties, significantly degrading the quality of object proposals generated by the Region Proposal Network (RPN) and consequently the subsequent R-CNN stage.

Contribution: In this paper, we introduce Uncertainty-based Progressive Proposal Refinement (UPPR), a method that leverages uncertainty estimation to enhance object proposals, improving overall detection performance and reducing forgetting. UPPR specifically focuses on modeling predictive uncertainties within a two-stage G-FSOD framework, allowing for the refinement of object proposals. This approach aims to enhance detection performance while mitigating the issue of forgetting by explicitly incorporating uncertainty modeling. An illustration of the proposed approach in comparison to other G-FSOD works is shown in Figure 1.

Our two key findings in this work are as follows. First, we show that careful architectural considerations can significantly impact the G-FSOD performance. Providing more model capacity and better feature representations help to improve detection performance and enhances generalization to novel classes. Secondly, we demonstrate that modeling predictive uncertainties in G-FSOD does not only contribute to improved detection performance, but also effectively mitigates the issue of forgetting when it comes to base classes.

2. Related Works

2.1. Object Detection

Two primary types of object detectors exist: two-stage and one-stage. Two-stage detectors [1, 7, 8, 24] involve a proposal generation stage. In the case of Faster R-CNN, this stage includes a Region Proposal Network (RPN), which utilizes a three-layer CNN to classify and refine the proposed regions. The proposals are then passed through an R-CNN output the predicted boxes. Cascade R-CNN [1] improves upon Faster R-CNN by employing a multi-stage architecture to learn more intricate object representations. Additionally, Cascade R-CNN introduces a weighted loss function that assigns greater importance to incorrectly classified proposals. On the other hand, one-stage detectors [18, 22, 23, 26] directly classify and locate the objects. Rather than relying on anchors, CenterNet [2], is a one-stage detector representing the object using three keypoints: two for the corners and one for the center. CenterNet detects keypoints in an image and groups those belonging to the same object, assigning them a preliminary bounding box based on their positions. Subsequently, CenterNet generates heatmaps, indicating the probability of a keypoint existing at various locations within the image.

2.2. Generalized Few-Shot Object Detection

G-FSOD focuses on detecting both base and novel classes. TFA [28] was the initial work in the realm of G-FSOD based on Faster R-CNN, intending to mitigate forgetting by optimizing training samples that maintain a balance between base and novel classes. On the other hand, ONCE [20] tackles the issue of incremental class learning in G-FSOD by employing a meta-learning approach with the YOLOv2 [22] detection model. Decouple Faster R-CNN (DeFRCN) [21], a pioneering transfer-learning-based strategy, highlights the conflicting objectives of the RPN and class-aware ROI head. To address this, they eliminate the RPN gradients and downscale the ROI head gradients that flow to the backbone, which has been shown to mitigate forgetting of base classes. Retentive R-CNN [5] utilizes the base-trained model in a distillation-like manner to alleviate forgetting.

To account for the forgetting without significant compu-

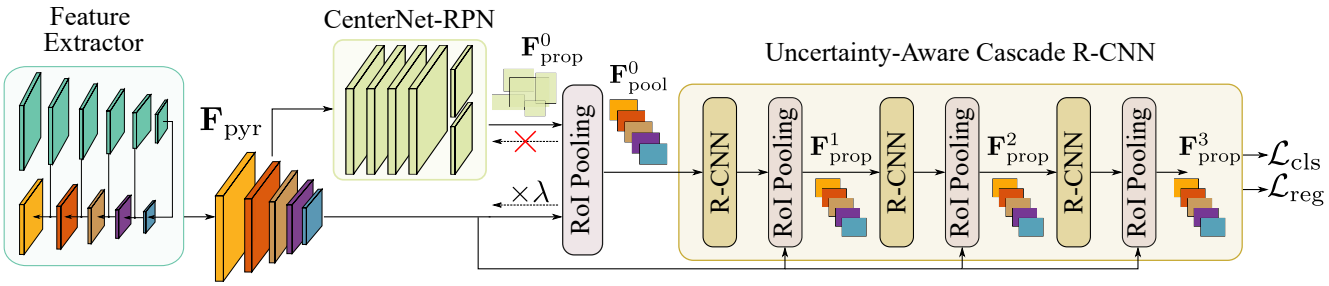

Figure 2. An illustration of the overall proposed UPPR method applied using the proposed DeCRCN as the base framework.

tational and memory overhead, CFA [9] introduces a novel gradient update mechanism based on the angle between gradients for base and novel samples, which helps mitigate forgetting. While the previous methods depend on stored base samples, NIFF [10] instead learns a base sample generator based on class-wise statistics. This enables generating base samples in a student-teacher fashion, allowing the model to retain the knowledge of the base classes without explicitly storing their samples. However, none of the abovementioned methods tackle the predictive uncertainties in a G-FSOD task.

2.3. Uncertainty-based Object Detection

Considering the importance of reliability and robustness in object detection models for various applications, it becomes necessary to consider both model and data uncertainties. Typically, aleatoric and epistemic uncertainties are jointly estimated by using direct modeling in combination with MC-dropout. Although, most existing works utilize the output layers to model aleatoric uncertainties directly, they differ in how they model epistemic uncertainties. Feng et al. [6] specifically employ dropout inferences only in the RoI-head of a Faster R-CNN model. Bayesian-YOLO [15] extends a YOLOv2 network and perform the dropout inference in both the backbone network and the detection head. Wirges et al. [30] adopt a similar architecture to [6] but introduce dropout layers in either the CNN backbone or the head networks. Finally, BayesOD [11] modifies a RetinaNet by incorporating MC-dropout in its detection head. Nonetheless, none of the above mentioned works address the problem in a FSOD or G-FSOD setting.

3. Methodology

Our goal is to design a two-stage G-FSOD framework that enhances the object proposals to improve the overall detection performance without forgetting the base classes. The main contributions of our proposed approach can be summarized as follows:

- As a straightforward application of predictive uncertainties in the adopted base framework, Decoupled Faster R-CNN (DeFRCN) [21], initially leads to a deterioration in the detection quality, thus, we implement two architectural modifications: (1) We opt for a deeper RPN with five layers, departing from the conventional two-layer design. Drawing inspiration from the generalization capabilities of CenterNet [2], we employ a key-point based detection approach. Meaning that the detector directly predicts object center points in the image, eliminating the need for predefined anchor boxes. The center point computation is based on the provided bounding boxes, requiring no extra data or labels. This modified RPN is named CenterNet-RPN. (2) We deepen the R-CNN by cascading three R-CNNs, each with gradually increasing IoU thresholds, allowing the network to learn how to refine the proposals gradually. This design choice ensures that the network has sufficient learning capacity to refine the proposals and improve detection performance effectively. We call this modified architecture as Decoupled Cascaded R-CNN (DeCRCN).
- We estimate the aleatoric and epistemic uncertainties in each R-CNN stage. Thereby, each stage is considered as an ensemble model that refines the proposals based on IoU thresholds and the estimated uncertainties. During training, we impose increasing IoU thresholds so that the latter stages are more confident than earlier ones.
- To selectively focus on discriminative features, we interleave the R-CNN stages with CBAM [31] attention blocks *only* during the novel training phase, where they can learn on a balanced set of base and novel classes. This prevents the attention blocks from overfitting the base features.

In this section, we describe these contributions, but, first, we begin with defining the G-FSOD

3.1. Problem Formulation

G-FSOD splits the training dataset $\mathcal{D}_{train}$ into two subsets: a base dataset $\mathcal{D}_b$ containing a large number of instances of base classes $\mathcal{C}_b$, and a novel dataset $\mathcal{D}_n$ containing a limited number of instances of novel classes $\mathcal{C}_n$. Note that there is no overlap between both classes, $\mathcal{C}_b \cap \mathcal{C}_n = \emptyset$. Each input image $x \in \mathcal{X}$ is paired with an annotation $y \in \mathcal{Y}$ that includes the class label c_i and the corresponding bounding box coordinates b_i for each instance i.

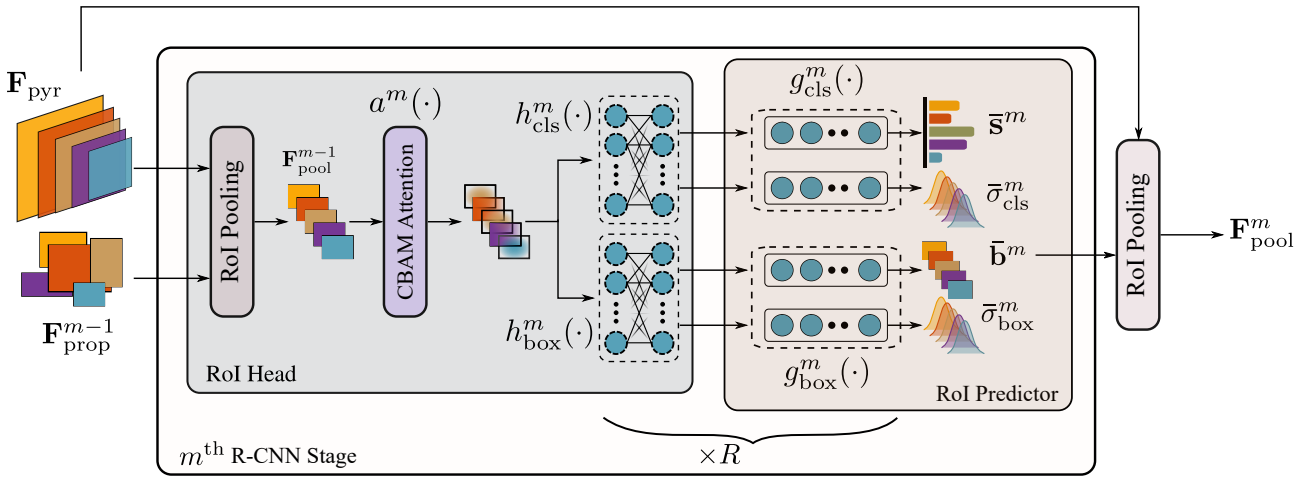

Figure 3. An illustration of the single stage R-CNN at test-time of the cascaded R-CNNs. The dotted neurons represent the dropouts. The epistemic uncertainties are computed by R forward runs and the predictions are averaged.

More specifically, $\mathcal{D}_b = \{(x, y) \mid y = (c_i, b_i), c_i \in \mathcal{C}_b\}$, and $\mathcal{D}_n = \{(x, y) \mid y = (c_i, b_i), c_i \in \mathcal{C}_n\}$.

The G-FSOD training process involves two stages. In the first stage, the model is trained on the base dataset $\mathcal{D}_b$ to establish transferable prior knowledge. In the second stage, the model leverages the acquired knowledge to rapidly learn novel classes from $\mathcal{D}_n$ along with a handful examples of base samples from $\mathcal{D}_b$. In contrast to FSOD, the primary goal of the G-FSOD is maximizing the overall Average Precision (AP), which is a weighted average of the AP of the base classes (bAP) and the AP of the novel classes (nAP). Formally, $\text{AP} = |\mathcal{C}_b| \cdot \text{bAP} + |\mathcal{C}_n| \cdot \text{nAP}/(|\mathcal{C}_b| + |\mathcal{C}_n|)$.

G-FSOD frameworks are mostly based on a two-stage Faster R-CNN model. One of the main bottlenecks encountered during standard object detection is the poor quality of object proposals [27]. The proposals quality further deteriorates in G-FSOD due to the introduction of new classes. There are three main reasons for this: (1) the training data for these new classes is limited and does not represent the true class distribution, (2) the novel classes might be considered as background by the network due to a low IoU with the ground truth boxes, and (3) the scale distribution of the novel objects differs from that in the base training data. Moreover, the limited novel samples result in higher epistemic uncertainty because the true data distribution is not fully captured, causing the model to overfit or underfit the data. None of the previous G-FSOD works have explicitly tackled the aforementioned limitations.

3.2. Model Architecture

An overview of the proposed model architecture is shown in Figure 2 and is described in the following:

3.2.1 Multiscale Keypoint-based RPN

Although the RPN is designed to be a simple three-layer architecture class-agnostic module [24], it usually generates subpar proposals for the subsequent R-CNN detector. The issue arises from the reliance on anchors of fixed sizes, which can result in numerous background and low-quality foreground proposals. Additionally, the misalignment of the anchors and the convolutional features adds to the difficulty of the bounding box classification task. On the other hand, keypoint-based approaches promise to alleviate the aforementioned limitations by representing each object keypoints, thus providing more accurate spatial information. We replace the anchor-based RPN with a keypoint-based CenterNet, and denote it by *CenterNet-RPN*. Additionally, to explicitly address the variability in object scale, we enhance the feature extractor by integrating a Feature Pyramid Network (FPN) [17]. This facilitates the refinement of object proposals at different scales.

3.2.2 Cascade R-CNN

To refine the RPN proposals, we replace the conventional R-CNN with a Cascade R-CNN [1] and set increasing IoU thresholds. Each stage improves the quality of object proposals from the previous stage, thereby increasing the number of true positives passed to the next stage. Inside each R-CNN stage, we opt to decouple the classification and localization features by introducing dual classification and bounding box regressor heads.

3.2.3 Multi-Stage Instance-level Attention

While feeding the instance-level features to cascaded R-CNN stages helps refine the proposals, we note that not all instance-level features are of equal importance. In order to assign more importance to features that correlate with correct classification, we interleave the R-CNN stages with attention modules. We employ a convolutional block attention module (CBAM) [31] to selectively focus on the most relevant features for the G-FSOD task. Specifically, the channel and spatial attention components of CBAM capture both channel-wise and spatial-wise relations between the instance-level features, which enables the model to comprehend better semantically-rich information for both the novel and base classes. Another advantage of using CBAM for our task is its lightweight design [31], which is particularly important as we are incorporating it after each R-CNN stage in the network. To prevent the CBAM from favoring the base classes over the novel classes, multi-stage attention blocks are *only* added during the novel training phase to ensure a balanced representation of both base and novel features.

3.3. Uncertainty-based Proposals Refinement

As previously mentioned, inherent data and model uncertainties exist and should be taken into account to alleviate forgetting and enhance the detection of novel classes. As we show later in the experiments (see Section 4), adding uncertainty on DeFRCN in a straightforward manner results in a deterioration of base and novel AP, thereby motivating the presented architectural design choices. We propose to estimate aleatoric and epistemic uncertainties in each stage of the Cascade R-CNN.

3.3.1 Stage-wise Epistemic Uncertainty-based Refinement

During training, we model the epistemic uncertainty by adding dropout layers in each R-CNN stage. The process starts by taking the pyramid feature maps $\mathbf{F}_{\text{pyr}}$ generated by the backbone network and the object proposals generated by the previous stage (with CenterNet-RPN being the initial stage). Then, the proposal features are extracted using RoI-pooling, passed through the CBAM attention block to focus, and fed into the classification and bounding box regressor heads to obtain the class scores and bounding box offsets. This constitutes a single forward run in an R-CNN stage. During testing, we activate the dropout layers, perform R forward runs per stage, aggregate the predictions, and pass them to the next stage along with $\mathbf{F}_{\text{pyr}}$. Figure 3 illustrates the operation of one R-CNN stage at test-time. Formally, for M stages, the classification features for the m^{th} stage are denoted by:

$$\mathbf{F}^{m}_{\text{cls}} = h^{m}_{\text{cls}}(a^{m}(\mathbf{F}^{m-1}_{\text{pool}})), \tag{1}$$

where for stage m, $a^{m}(\cdot)$ is the CBAM attention module. h^{m}_{cls} is the classification head in the RoI-head. $\mathbf{F}^{m-1}_{\text{pool}}$ is the pooled instance-level features from the previous stage. Similarly, the bounding-box features are computed as:

$$\mathbf{F}^{m}_{\text{box}} = h^{m}_{\text{box}}(a^{m}(\mathbf{F}^{m-1}_{\text{pool}})), \tag{2}$$

where for stage m, h^{m}_{box} is the bounding-box head in the RoI-head. The $\mathbf{F}^{m}_{\text{cls}}$ and $\mathbf{F}^{m}_{\text{box}}$ undergo the RoI-predictor to compute the classification and regression offsets along with their corresponding uncertainties. The RoI-predictor consists of a classifier head $\mathbf{g}^{m}_{\text{cls}}(\cdot)$ and a box head $\mathbf{g}^{m}_{\text{box}}(\cdot)$. During inference, we perform R forward runs with dropouts and aggregate the classification logits $\mathbf{s}^{m}$ along with the class aleatoric variances $\mathbf{\Sigma}^{m}_{\text{cls}}$ and similarly the box offsets $\mathbf{b}^{m}$ along with the box aleatoric variances $\mathbf{\Sigma}^{m}_{\text{box}}$. The final classification logits and associated aleatoric variances is computed via an average over the R forward runs as follows:

$$\bar{\mathbf{s}}^{m}_{\text{cls}},\ \bar{\mathbf{\Sigma}}^{m}_{\text{cls}} = \frac{1}{R}\sum_{r=1}^{R} g^{m}_{\text{cls}}(\mathbf{F}^{m,r}_{\text{cls}}), \tag{3}$$

where $\mathbf{F}^{m,r}_{\text{cls}}$ represent different RoI-head classification features each forward run r due to the stochastic dropouts in the classification head. For the box offsets and predicted variances,

$$\bar{\mathbf{b}}^{m}_{\text{box}},\ \bar{\mathbf{\Sigma}}^{m}_{\text{box}} = \frac{1}{R}\sum_{r=1}^{R} g^{m}_{\text{box}}(\mathbf{F}^{m,r}_{\text{box}}), \tag{4}$$

where $\mathbf{F}^{m,r}_{\text{box}}$ are the RoI-head box features for the forward run r.

3.3.2 Stage-wise Aleatoric Uncertainty-based Refinement

The aleatoric uncertainties are considered for both the classification and bounding boxes regression. First, the classification logits are modelled as a multivariate Gaussian distribution parametrized by the mean $\mathbf{s}_{\text{cls}}$ of the predicted classification logits and the diagonal covariance matrix $\mathbf{\Sigma}_{\text{cls}}$ computed by the predicted class variances σ^{2}_{cls}. Next, we draw N_{cls} classification logits $\mathbf{s}^{[n]}_{\text{cls}}$ from the created Gaussian distribution. The resulting matrix containing all samples is denoted by $\mathbf{S}_{\text{cls}}$ and is expressed as:

$$\mathbf{S}_{\text{cls}} = \{\mathbf{s}^{[n]}_{\text{cls}}\}^{N_{\text{cls}}}_{n=1} \in \mathbb{R}^{N_{\text{cls}}\times|\mathcal{C}|},\ \mathbf{s}^{[n]}_{\text{cls}} \sim \mathcal{N}(\mathbf{s}_{\text{cls}}, \mathbf{\Sigma}_{\text{cls}}). \tag{5}$$

The classification loss is then the softmax cross-entropy between the stochastic classification logits $\mathbf{S}_{\text{cls}}$ and the associated ground-truth labels.

Methods / Shots	w/E	5 shot			10 shot			30 shot		
		AP	bAP	nAP	AP	bAP	nAP	AP	bAP	nAP
FRCN-ft-full[28]	✗	18.0	22.0	6.0	18.1	21.0	9.2	18.6	20.6	12.5
TFA w/ fc[28]	✗	27.5	33.9	8.4	27.9	33.9	10.0	29.7	35.1	13.4
TFA w/ cos[28]	✗	28.1	34.7	8.3	28.7	35.0	10.0	30.3	35.8	13.7
MPSR[32]	✗	-	-	-	15.3	17.1	9.7	17.1	18.1	14.1
DeFRCN [21]	✗	28.7	33.1	15.3	30.6	34.6	18.6	31.6	34.7	**22.5**
ONCE [20]	✗	13.7	17.9	1.0	13.7	17.9	1.2	-	-	-
Meta R-CNN [34]	✗	3.6	3.5	3.8	5.4	5.2	6.1	7.8	7.1	9.9
FSRW[13]	✗	-	-	-	-	-	5.6	-	-	9.1
FsDetView [33]	✗	5.9	5.7	6.6	6.7	6.4	7.6	10.0	9.3	12.0
CFA w/ fc [9]	✗	**30.1**	**37.1**	9.0	30.8	**37.6**	10.5	31.9	**37.7**	14.7
CFA w/ cos [9]	✗	29.7	36.3	9.8	30.3	36.6	11.3	31.7	37.0	15.6
CFA-DeFRCN [9]	✗	**30.1**	35.0	**15.6**	**31.4**	35.5	**19.1**	**32.0**	35.0	**23.0**
DeCRCN-UPPR	✗	**33.7**	**38.9**	**17.9**	**35.0**	**40.2**	**19.2**	**36.0**	**40.1**	**24.0**
Retentive R-CNN[5]	✓	31.5	39.2	8.3	32.1	39.2	10.5	32.9	39.3	13.8
CFA w/ fc [9]	✓	31.8	**39.5**	8.8	32.2	**39.5**	10.4	33.2	**39.5**	14.3
CFA w/ cos [9]	✓	32.0	**39.5**	9.6	32.4	39.4	11.3	33.4	**39.5**	15.1
CFA-DeFRCN [9]	✓	**33.0**	38.9	**15.6**	**34.0**	39.0	**18.9**	**34.9**	39.0	**22.6**
DeCRCN-UPPR	✓	**35.9**	**41.9**	**17.8**	**36.2**	**41.9**	**19.1**	**37.8**	**42.0**	**23.8**

Table 1. G-FSOD results on MS-COCO for $5, 10, 30$-shot settings. w/E denotes the ensemble-based evaluation protocol. The best and second-best results are color coded.

Second, the bounding box regression are similarly modelled as a Gaussian distribution with the mean being the predicted box offsets $\mathbf{b}_{\text{box}}$ and the diagonal covariance matrix based on the predicted box variances $(\sigma_x^2, \sigma_y^2, \sigma_w^2, \sigma_h^2)$. As a result, the bounding box regression loss is computed using the negative log-likelihood from [15].

3.4. Overall DeCRCN-UPPR Pipeline

The entire pipeline, namely DeCRCN-UPPR, can be summarized as follows:

1. The initial proposals from CenterNet-RPN are sent to the first R-CNN stage along with the pyramid feature maps from the backbone.
2. The RoI-head attends to the pooled features, extracting classification and bounding box features that undergo the RoI-predictor, resulting in classification logits and variances and bounding box offsets and variances.
3. To capture epistemic uncertainties, stochasticity is introduced through dropout layers during training. During inference, R forward passes are conducted, and the network predictions are aggregated and averaged to obtain the final predictions.
4. The predicted box offsets are then applied to the input proposals, resulting in refined boxes that serve as input for the next R-CNN stage.

This progressive refinement generates more reliable boxes by leveraging the averaged epistemic predictions, which are more robust than single-run predictions.

4. Experiments

Our proposed approach is evaluated on widely recognized G-FSOD benchmarks, namely MS-COCO [16] and PASCAL-VOC [3] datasets. To ensure a fair comparison with previous works, we use the same data splits as employed in earlier works [9, 21, 28].

4.1. Experimental Setup

Datasets. In our experiments, we utilize the MS-COCO dataset, which consists of 80 classes. Among these, 60 classes are considered base categories and do not overlap with the classes in the PASCAL-VOC dataset. The remaining 20 classes in MS-COCO are unique and treated as novel classes. During testing, we use a validation set of 5000 images while the rest of the dataset is used for training. The results are reported for different shot settings, including 5-shot, 10-shot, and 30-shot. For the PASCAL-VOC dataset, it is divided into three distinct splits, with each split containing 20 classes. Among these, 15 classes are designated base classes, and the remaining 5 classes are considered novel classes. The training data for base and novel classes is drawn from the VOC 2007 and VOC 2012 train/val sets. The VOC 2007 test set is used for evaluation. We report the results for various shot settings, including 1-shot, 2-shot, 3-shot, 5-shot, and 10-shot.

Evaluation Metrics. Consistent with previous G-FSOD frameworks [5, 9, 21, 28], we use the same performance metrics: the overall average precision (AP), base class average precision (bAP), and novel class average precision (nAP). Additionally, we report the average recall (AR) for

Methods / Shots	w/E	All Set 1					All Set 2					All Set 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FRCN-ft-full[28]	✗	55.4	57.1	56.8	60.1	60.9	50.1	53.7	53.6	55.9	55.5	58.5	59.1	58.7	61.8	60.8
TFA w/ fc[28]	✗	69.3	66.9	70.3	73.4	73.2	64.7	66.3	67.7	68.3	68.7	67.8	68.9	70.8	72.3	72.2
TFA w/ cos[28]	✗	69.7	68.2	70.5	73.4	72.8	65.5	65.0	67.7	68.0	68.6	67.9	68.6	71.0	72.5	72.4
MPSR[32]	✗	56.8	60.4	62.8	66.1	69.0	53.1	57.6	62.8	64.2	66.3	55.2	59.8	62.7	66.9	67.7
DeFRCN[21]	✗	73.1	73.2	73.7	75.1	**74.4**	68.6	69.8	71.0	72.5	71.5	72.5	73.5	72.7	74.1	73.9
Meta R-CNN[34]	✗	17.5	30.5	36.2	49.3	55.6	19.4	33.2	34.8	44.4	53.9	20.3	31.0	41.2	48.0	55.1
FSRW[13]	✗	53.5	50.2	55.3	56.0	59.5	55.1	54.2	55.2	57.5	58.9	54.2	53.5	54.7	58.6	57.6
FsDetView[33]	✗	36.4	40.3	40.1	50.0	55.3	36.3	43.7	41.6	45.8	54.1	37.0	39.5	40.7	50.7	54.8
CFA w/ fc [9]	✗	69.5	68.2	69.8	73.5	74.3	66.0	66.9	69.2	70.1	71.1	67.7	69.0	70.9	72.6	73.5
CFA w/ cos [9]	✗	69.1	69.8	71.9	73.6	73.9	64.8	66.5	68.3	69.5	70.5	67.7	69.7	71.9	73.0	73.5
CFA-DeFRCN [9]	✗	**73.8**	**74.6**	**74.5**	**76.0**	**74.4**	**69.3**	**71.4**	**72.0**	**73.3**	**72.0**	**72.9**	**73.9**	**73.0**	**74.1**	**74.6**
DeCRCN-UPPR	✗	**74.3**	**75.1**	**75.4**	**76.3**	**75.1**	**71.4**	**72.6**	**73.2**	**74.9**	**73.2**	**73.2**	**74.3**	**74.2**	**75.3**	**75.8**
Retentive R-CNN[5]	✓	71.3	72.3	72.1	74.0	74.6	66.8	68.4	70.2	70.7	71.5	69.0	70.9	72.3	73.9	74.1
CFA w/ fc [9]	✓	70.3	69.5	71.0	74.4	74.9	67.0	68.0	70.2	70.8	71.5	69.1	70.1	71.6	73.3	74.7
CFA w/ cos [9]	✓	71.4	71.8	73.3	74.9	75.0	66.8	68.4	70.4	71.1	71.9	69.7	71.2	72.6	74.0	74.7
CFA-DeFRCN [9]	✓	**75.0**	**76.0**	**76.8**	**77.3**	**77.3**	**70.4**	**72.7**	**73.7**	**74.7**	**74.2**	**74.7**	**75.5**	**75.0**	**76.2**	**76.6**
DeCRCN-UPPR	✓	**76.1**	**77.0**	**77.9**	**78.2**	**78.4**	**71.3**	**73.5**	**74.4**	**75.1**	**75.2**	**75.1**	**76.9**	**76.2**	**77.3**	**77.5**

Table 2. The overall G-FSOD (AP50) results on PASCAL-VOC for $1, 2, 3, 5, 10$-shot settings for all three splits. The best and second-best results are color coded.

both base class (bAR) and novel class (nAR). Finally, for a fair comparison with Retentive R-CNN [5], we present ensemble-inference results (w/E), where we utilize the parameters of the base model during the inference process.

Implementation Details. We employ a Cascade R-CNN [1] as the base detector, with a ResNet-101 [12] backbone that has been pre-trained on ImageNet. The model consists of four stages: the RPN based on the CenterNet architecture [2] and three subsequent R-CNN stages, each with a gradually increasing Intersection over Union (IoU) threshold $[0.5, 0.6, 0.7]$. To optimize the network end-to-end, we employ Stochastic Gradient Descent (SGD) with a mini-batch size of 16. The SGD algorithm incorporates a momentum of 0.9 and a weight decay of $5e^{-5}$. During base training, the total number of iterations is 110000, the learning rate is set to 0.02, with two learning step decays at 85000 and 100000 iterations. Moreover, the gradients backpropagating from the RPN are killed, and the Gradient Descent Layer (GDL) [21] scale of R-CNN is $\lambda = 0.75$. During novel finetuning, the total number of iterations is 4000, the learning rate is set to 0.01 with a decay step at 2000 iterations. The R-CNN GDL scale is $\lambda = 0.04$ and the gradients of the R-CNN layers are down-scaled by a factor of 0.1. For epistemic uncertainty, we perform 40 forward runs and set all dropout layers with a probability of 0.5. For aleatoric uncertainty, we set the number of classification samples to 10. We denote the overall proposed architecture as Decoupled Cascade R-CNN (DeCRCN).

4.2. Comparison Results

We compare our method (UPPR) with the proposed DeCRCN architecture against state-of-the-art G-FSOD [5, 9] and FSOD models on MS-COCO and PASCAL-VOC benchmarks. We opt to apply our approach on top of the recent state-of-the-art transfer learning-based approach DeFRCN [21]. We denote our model by *DeCRCN-UPPR*.

4.2.1 MS-COCO Results

In Table 1, we show the results on MS-COCO. UPPR outperforms all previous state-of-the-art results by a significant margin on the AP and bAP in all settings while achieving slightly better nAP. Moreover, we evaluate our model using the ensemble evaluation protocol in Retentive R-CNN [5] and outperform the other approaches. The multiple run results over different seeds are presented in the supplementary material.

4.2.2 PASCAL-VOC Results

We report the overall performance on PASCAL-VOC (AP50) in Table 2 and the novel performance (nAP50) in Table 3. We show that adopting UPPR achieves state-of-the-art results with and without the ensemble evaluation protocol on all shot settings. The multiple run results using various seeds are shown in the supplementary material.

The findings from our experiments provide empirical evidence to support the integration of predictive uncertainties into the detection process, empowering the progressive refinement of object proposals. This approach leads to enhanced prediction confidence, improving overall detection performance. Furthermore, using predictive uncertainties demonstrates its potential to mitigate the issue of forgetting, thereby preserving knowledge effectively during the object detection task.

4.3. Ablation Study

In Table 4, we perform an ablation study to analyze our contribution. In config A, we start with our baseline De-

Methods / Shots	w/E	Novel Set 1					Novel Set 2					Novel Set 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FRCN-ft-full[28]	✗	15.2	20.3	29.0	25.5	28.7	13.4	20.6	28.6	32.4	38.8	19.6	20.8	28.7	42.2	42.1
TFA w/ fc[28]	✗	36.8	29.1	43.6	55.7	57.0	18.2	29.0	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2
TFA w/ cos[28]	✗	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[32]	✗	42.8	43.6	48.4	55.3	61.2	29.8	28.1	41.6	43.2	47.0	35.9	40.0	43.7	48.9	51.3
DeFRCN[21]	✗	57.0	58.6	64.3	67.8	67.0	35.8	42.7	51.0	54.4	52.9	52.5	56.6	55.8	**60.7**	62.5
Meta R-CNN[34]	✗	16.8	20.1	20.3	38.2	43.7	7.7	12.0	14.9	21.9	31.1	9.2	13.9	26.2	29.2	36.2
FSRW[13]	✗	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	39.2	19.2	21.7	25.7	40.6	41.3
MetaDet[29]	✗	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
FsDetView*[33]	✗	25.4	20.4	37.4	36.1	42.3	22.9	21.7	22.6	25.6	29.2	32.4	19.0	29.8	33.2	39.8
CFA w/ fc [9]	✗	40.0	35.5	40.9	54.1	56.9	22.2	27.1	35.2	38.5	40.9	29.7	35.1	39.5	47.2	51.3
CFA w/ cos [9]	✗	41.2	43.6	49.5	56.5	57.3	21.3	27.4	35.3	39.1	42.1	31.7	39.1	44.6	49.9	52.6
CFA-DeFRCN [9]	✗	**58.2**	**63.3**	**65.8**	**68.9**	**67.1**	**37.1**	**45.5**	**51.3**	**55.2**	**53.8**	**54.7**	**57.8**	**56.9**	60.0	**63.3**
DeCRCN-UPPR	✗	**60.2**	**64.7**	**66.4**	**70.1**	**68.4**	**38.7**	**46.4**	**52.8**	**56.2**	**54.6**	**55.5**	**58.7**	**57.9**	**61.2**	**64.7**
Retentive R-CNN[5]	✓	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
CFA w/ fc [9]	✓	39.0	34.9	41.4	54.8	57.0	21.8	26.1	35.3	37.1	40.1	29.9	34.3	40.1	47.0	52.6
CFA w/ cos [9]	✓	42.4	43.9	50.3	56.6	57.3	21.0	27.5	35.3	38.6	41.4	32.3	38.0	44.5	49.8	52.7
CFA-DeFRCN [9]	✓	**59.0**	**63.5**	**66.4**	**68.4**	**68.3**	**37.0**	**45.8**	**50.0**	**54.2**	**52.5**	**54.8**	**58.5**	**56.5**	**61.3**	**63.5**
DeCRCN-UPPR	✓	**61.0**	**64.5**	**67.8**	**69.7**	**69.0**	**38.5**	**46.9**	**51.4**	**55.9**	**53.6**	**55.3**	**59.4**	**57.5**	**62.8**	**64.1**

Table 3. PASCAL-VOC G-FSOD (nAP50) results for $1, 2, 3, 5, 10$-shot settings for all three splits are reported. Similar to [5, 9], w/E denotes the ensemble-based inference paradigm [5]. The best and second-best results are color coded.

Model Configuration	Base Training		Base		Novel		Overall	
	AP	AR	AP	AR	nAP	nAR	bAP	bAR
A DeFRCN	38.5	33.2	36.5	32.4	16.8	20.2	31.6	29.4
B DeFRCN (NT: UE)	38.5	33.2	34.9	31.3	17.5	20.8	30.5	28.6
C DeCRCN	41.4	35.7	38.2	34.1	17.5	21.6	33.0	31.0
D DeCRCN (NT: UE)	41.4	35.7	38	34.2	18.2	22.5	33.1	31.3
E DeCRCN (BT + NT: UE)	42.0	36.1	40.2	36.6	19.0	23.6	34.7	32.6
F DeCRCN (BT: UE, NT: ATT)	42.0	36.1	40.2	36.6	19.3	24.2	34.8	32.8
G DeCRCN (BT + NT: UE + ATT)	41.7	36.2	37.3	34.6	18.7	23.6	32.6	31.8
H DeCRCN (BT: UE, NT: UE + ATT) (UPPR)	42.0	36.1	40.5	36.7	19.2	24.0	35.0	32.8

Table 4. An ablation study performed on MS-COCO for the 10-shot setting to highlight the impact of different design choices. BT and NT denote base training and novel training, respectively. UE denotes uncertainty estimation (aleatoric and epistemic). ATT is the stage-wise instance-level attention.

FRCN. In config B, we add aleatoric and epistemic uncertainty estimation to the RoI head during novel training only and observe a decline in the bAP with a slight upgrade in the nAP. Replacing the RoI head with a Cascade R-CNN and the RPN with CenterNet in configuration C yields improvements in the base metrics. We denote this model by DeCRCN. We observe that adding uncertainty estimation in a stagewise manner D maintains the bAP of the previous configuration while increasing the nAP. This highlights the importance of applying uncertainty estimation in a stagewise manner instead of applying it only once in a single stage. While uncertainty estimation applied naively causes some forgetting, it can enhance the proposal refinement progressively in the Cascade R-CNN. In E, we improve all metrics by learning uncertainty during base training as well. Finally, in F, H and G, we show the impact of applying attention blocks during novel training. With or without UE, they can boost bAP and nAP when applied during the novel-training stage. We note, however, that when included in the base training phase, the performance drops on the base classes, indicating the importance of our design choice to train attention blocks on a balanced set of both classes.

5. Conclusion

We propose DeCRCN-UPPR, a novel G-FSOD framework that alleviates forgetting on the base images while improving the detection of the novel classes that only have a few labeled instances during training. Our main contribution is the design of a detector that learns to refine proposals in a stagewise manner by leveraging predictive uncertainties to detect objects despite the few training instances. Furthermore, we append each R-CNN stage with an attention block during novel training allowing the next stage to selectively focus on the discriminative features that enable a better classification. The interleaving of ensemble stages and attention blocks significantly boosts the detection of base and novel classes. We hope our proposed approach sheds light on the potential of predictive uncertainties in enhancing the performance of few-shot models and promoting their use in robotics and industrial applications.

References

[1] 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. 1, 2, 4, 7

[2] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. CenterNet: Keypoint triplets for object detection. In IEEE International Conference on Computer Vision, pages 6568–6577, 2019. 1, 2, 3, 7

[3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. International Journal of Computer Vision, 88(2):303–338, 2010. 6

[4] Qi Fan, Wei Zhuo, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In IEEE Conference on Computer Vision and Pattern Recognition, pages 4012–4021, 2020. 1

[5] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In IEEE Conference on Computer Vision and Pattern Recognition, pages 4527–4536, 2021. 1, 2, 6, 7, 8

[6] Di Feng, Lars Rosenbaum, and Klaus Dietmayer. Towards Safe Autonomous Driving: Capture Uncertainty in the Deep Neural Network For Lidar 3D Vehicle Detection. In International Conference on Intelligent Transportation Systems (ITSC), pages 3266–3273, 2018. 2, 3

[7] Ross Girshick. Fast R-CNN. In IEEE International Conference on Computer Vision, pages 1440–1448, 2015. 1, 2

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587, 2014. 1, 2

[9] Karim Guirguis, Ahmed Hendawy, George Eskandar, Mohamed Abdelsamad, Matthias Kayser, and Jürgen Beyerer. CFA: Constraint-based Finetuning Approach for Generalized Few-Shot Object Detection. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 4038–4048, 2022. 1, 3, 6, 7, 8

[10] Karim Guirguis, Johannes Meier, George Eskandar, Matthias Kayser, Bin Yang, and Jürgen Beyerer. NIFF: Alleviating Forgetting in Generalized Few-Shot Object Detection via Neural Instance Feature Forging. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 1, 2, 3

[11] Ali Harakeh, Michael Smart, and Steven L. Waslander. BayesOD: A Bayesian Approach for Uncertainty Estimation in Deep Object Detectors. In IEEE International Conference on Robotics and Automation (ICRA), pages 87–93, 2020. 2, 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016. 7

[13] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In IEEE International Conference on Computer Vision, pages 8419–8428, 2018. 1, 6, 7, 8

[14] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In Advances in Neural Information Processing Systems (NIPS), 2017. 2

[15] Florian Kraus and Klaus Dietmayer. Uncertainty Estimation in One-Stage Object Detection. In IEEE Intelligent Transportation Systems Conference (ITSC), pages 53–60, 2019. 2, 3, 6

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In European Conference on Computer Vision, pages 740–755. Springer, 2014. 6

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In IEEE Conference on Computer Vision and Pattern Recognition, pages 2117–2125, 2017. 4

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In European Conference on Computer Vision, pages 21–37, 2016. 1, 2

[19] Nitish Srivastava and Geoffrey Hinton and Alex Krizhevsky and Ilya Sutskever and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958, 2014. 2

[20] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M. Hospedales, and Tao Xiang. Incremental few-shot object detection. In IEEE Conference on Computer Vision and Pattern Recognition, pages 13843–13852, 2020. 1, 2, 6

[21] Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. DeFRCN: Decoupled faster R-CNN for few-shot object detection. In IEEE International Conference on Computer Vision, 2021. 2, 3, 6, 7, 8

[22] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In IEEE Conference on Computer Vision and Pattern Recognition, pages 7263–7271, 2017. 1, 2

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In IEEE Conference on Computer Vision and Pattern Recognition, pages 779–788, 2016. 2

[24] 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. 1, 2, 4

[25] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. FSCE: few-shot object detection via contrastive proposal encoding. In IEEE Conference on Computer Vision and Pattern Recognition, pages 7352–7362, 2021. 1

[26] Mingxing Tan, Ruoming Pang, and Quoc Le. EfficientDet: Scalable and efficient object detection. In IEEE Conference on Computer Vision and Pattern Recognition, pages 10778–10787, 2020. 1, 2

[27] Thang Vu, Hyunjun Jang, Trung X Pham, and Chang D Yoo. Cascade RPN: Delving into High-Quality Region Proposal Network with Adaptive Convolution. In Conference on Neural Information Processing Systems (NeurIPS), 2019. 4

[28] 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, pages 9919–9928, 2020. 1, 2, 6, 7, 8

[29] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In IEEE International Conference on Computer Vision, pages 9924–9933, 2019. 8

[30] Sascha Wirges, Marcel Reith-Braun, Martin Lauer, and Christoph Stiller. Capturing Object Detection Uncertainty in Multi-Layer Grid Maps. In IEEE Intelligent Vehicles Symposium (IV), pages 1520–1526, 2019. 2, 3

[31] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional Block Attention Module. In European Conference on Computer Vision (ECCV), pages 3–19, 2018. 3, 5

[32] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In European Conference on Computer Vision, pages 456–472, 2020. 1, 6, 7, 8

[33] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In European Conference on Computer Vision, pages 192–210, 2020. 6, 7, 8

[34] 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 IEEE International Conference on Computer Vision, pages 9577–9586, 2019. 6, 7, 8