# iFS-RCNN: An Incremental Few-shot Instance Segmenter

Khoi Nguyen
VinAI Research, Ha Noi, Vietnam
ducminhkhoi@gmail.com

Sinisa Todorovic
Oregon State University, Oregon, USA
sinisa@oregonstate.edu

## Abstract

*This paper addresses incremental few-shot instance segmentation, where a few examples of new object classes arrive when access to training examples of old classes is not available anymore, and the goal is to perform well on both old and new classes. We make two contributions by extending the common Mask-RCNN framework in its second stage – namely, we specify a new object class classifier based on the probit function and a new uncertainty-guided bounding-box predictor. The former leverages Bayesian learning to address a paucity of training examples of new classes. The latter learns not only to predict object bounding boxes but also to estimate the uncertainty of the prediction as a guidance for bounding box refinement. We also specify two new loss functions in terms of the estimated object-class distribution and bounding-box uncertainty. Our contributions produce significant performance gains on the COCO dataset over the state of the art – specifically, the gain of +6 on the new classes and +16 on the old classes in the AP instance segmentation metric. Furthermore, we are the first to evaluate the incremental few-shot setting on the more challenging LVIS dataset.*

## 1. Introduction

This paper addresses the two related problems of incremental few-shot object detection (iFSOD) and instance segmentation (iFSIS). Initially, we are given a large training set of base object classes, which can be used for pre-training an instance segmenter. After this pre-training, access to training examples of the base classes becomes unavailable. With an arrival of a few training examples of new classes, the goal is to achieve successful object detection and instance segmentation on both new and base classes. Our key challenges include: how to address a paucity of data for new classes, and how to train on the new classes such that the base classes are not "forgotten".

Tab. 1 compares iFSOD and iFSIS with other related problems. iFSOD and iFSIS are important problems arising in many applications, where access to old training data becomes unavailable, due to, e.g., privacy and security issues or new legal regulations of data access. Also, they are critical in applications where limited time budgets prohibit retraining on both base and new classes.

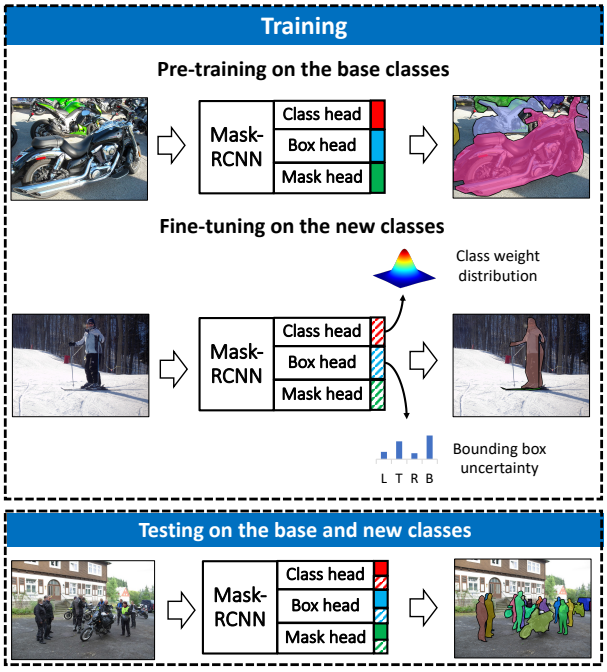


Figure 1. Our iFS-RCNN is first pre-trained on abundant examples of base classes, and then fine-tuned on a few examples of new classes. iFS-RCNN modifies the classification head of Mask-RCNN by estimating the class-weight distribution via Bayesian learning. iFS-RCNN modifies the bounding-box head of Mask-RCNN by computing uncertainty of predicting the left (L), right (R), top (T), and bottom (B) sides of the bounding boxes. For testing, the last layers learned on the new classes (diagonal stripes) are concatenated with the corresponding ones learned on the base classes (solid color).

There is scant work on iFSOD and iFSIS. Following recent FSIS approaches, we use Mask-RCNN [13], and modify its prediction heads, as shown in Fig. 1. Mask-RCNN is first pre-trained with abundant examples of base classes, and then fine-tuned on new classes by "freezing" all mod-

| Settings | Pretrained on | Fine-tuned on | Tested on |
|---|---|---|---|
| FSOD - FSIS | base | new | new |
| gFSOD - gFSIS | base | base + new | base + new |
| CL | base | new | base + new |
| iFSOD - iFSIS | base | new | base + new |

Table 1. A comparison of related problems. The blue and red indicate abundant and a few examples, respectively, of base and new classes. FSOD (FSIS): few-shot object detection (instance segmentation), gFSOD (gFSIS): generalized FSOD (FSIS), CL: continual learning, iFSOD (iFSIS): incremental FSOD (FSIS). iFSOD (iFSIS) are more challenging than: FSOD (FSIS), since we test on both classes; gFSOD (gFSIS), since our training cannot access the base classes; CL, since they work with more examples.

ules except for the classification head, bounding-box head, and segmentation-mask head. Finally, for testing on both base and new classes, weights learned on the new classes are concatenated with weights learned on the base classes to make the corresponding last layers in the classification, bounding-box, and segmentation-mask heads.

As shown in Fig. 1 and depicted in more detail in Fig. 2, we make two contributions aimed at addressing overfitting of Mask-RCNN in few-shot fine-tuning and improving its generalization to query images with large appearance-shape-scale variations of both base and new classes.

Inspired by deep Bayesian learning [2], *our first contribution* is about learning a distribution of the classification head's weights on the new classes, and using the estimated distribution for regularization of the fine-tuning. Instead of using the standard Monte Carlo sampling of weights for this Bayesian learning, our key technical novelty is in casting the weight-distribution learning as a Bayesian logistic regression problem, and specifying an efficient approximation to this intractable problem using the probit function. From our ablation studies, our probit-based approximation gives a significantly better performance than the Monte Carlo sampling, even under a reasonable training-time budget.

*Our second contribution* is about estimating the uncertainty of bounding-box localization on the new classes, and using the estimated uncertainty for two purposes – to refine the bounding-box prediction and to appropriately weight the loss of bounding-box prediction. As shown in Fig. 2, we use the estimated uncertainty along with the ROI-aligned-pooled feature map as input to a new bounding-box refinement module. The refined bounding box is subsequently input to the segmentation head. Also, we define a new loss between the ground truth and predicted bounding box, such that the *loss becomes smaller for highly uncertain predictions*, i.e., our fine-tuning stronger penalizes errors on training examples with highly certain bounding-box predictions.

It is worth noting that we neither use Bayesian learning nor explicitly estimate uncertainty for fine-tuning of the segmentation head of Mask-RCNN. This is because fine-tuning of the segmentation head on a few examples of the new classes does not face the common challenges of few-shot learning. Recall that the segmentation head predicts pixel labels independently with a $1 \times 1$ convolution. Consequently, every (pixel, label) pair is an independent training example, giving rise to a sufficiently large training set for fine-tuning of the segmentation mask.

Our extension of Mask-RCNN for incremental few-shot setting gives the name to our approach – iFS-RCNN. iFS-RCNN is evaluated on the COCO dataset [24] for few-shot object detection and instance segmentation with the iFSOD, iFSIS, FSOD, and FSIS tasks. iFS-RCNN significantly outperforms a recent approach [11]. In comparison with the standard Mask-RCNN trained in the gFSOD and gFSIS settings, iFS-RCNN shows a considerable improvement on the new classes while retaining the same performance on the base classes. On iFSOD and iFSIS, we also achieve the higher COCO AP rates by +6 and +16 for the new and base classes, respectively, relative to the state of the art.

In addition, we are the first to report the results of iFSOD, FSIS, and iFSIS on the more challenging LVIS dataset [12] having significantly more classes and long-tailed class distributions.

In the following, Sec. 2 reviews prior work; Sec. 3 specifies iFS-RCNN; and Sec. 4 presents our implementation details and experimental results.

## 2. Related Work

This section reviews closely related work.

**FSOD** approaches [6, 8, 9, 16, 17, 19, 32, 33, 35, 36, 38] typically adapt Faster-RCNN [29], YOLO [28], or DETR [5] for standard object detection to few-shot setting. These approaches can be divided into two subgroups based on: episodic training [8, 16, 35, 36, 38] and fine-tuning [6, 9, 32, 33]. The former uses episodic training to mimic the setting of few-shot learning, by limiting access to a few annotated support images for each base class. The latter fine-tunes the weights of some layers while freezing the rest to preserve knowledge learned on the base classes.

**FSIS** approaches typically use Mask-RCNN [13] as their backbone network. As in FSOD, these methods either train episodically [10, 25, 26, 37] or pretrain on the base classes then fine-tune the last layers of each head [32] on the new classes. Our iFS-RCNN follows the second training strategy and applies it to all three heads – the classification, bounding-box, and segmentation heads – of Mask-RCNN.

**iFSOD** approaches [20, 21, 27] typically use Faster-RCNN as their backbone network. In [20], knowledge distillation ensures the prediction of the base classes, after fine-tuning on the new classes, to match its pretrained prediction. In [27], weights of the box head are generated on the fly, based on a class-specific code extracted from examples of the target classes. In this way, each class has a distinct box

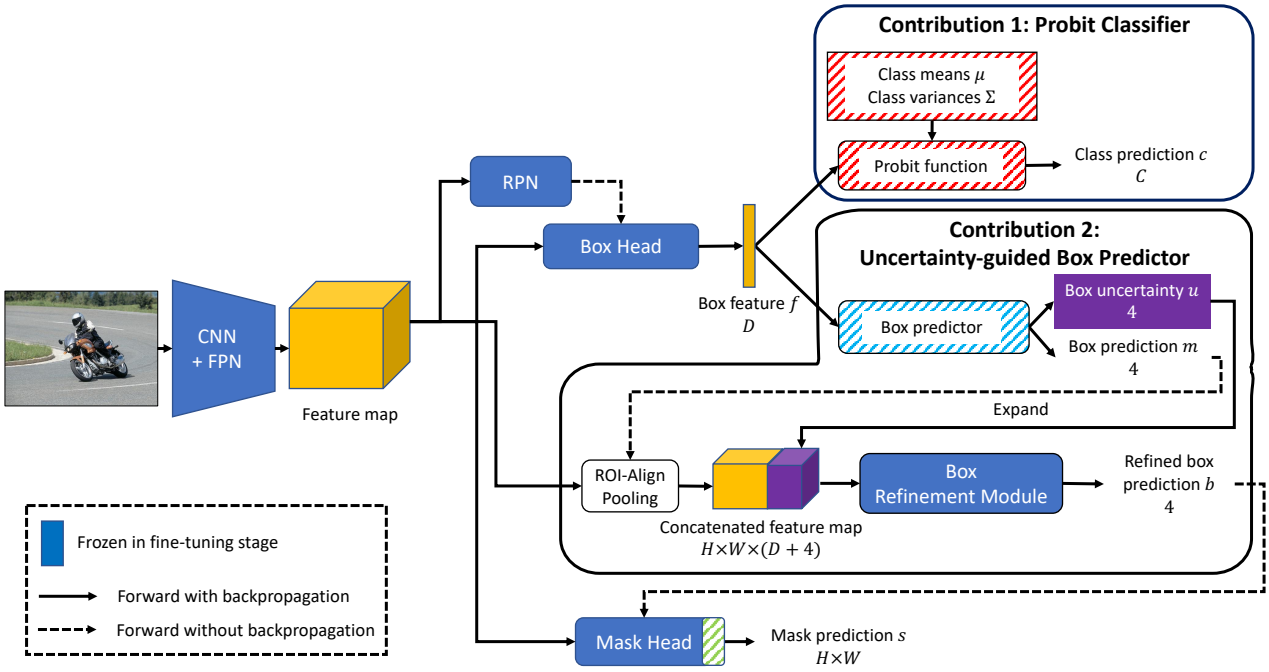


Figure 2. iFS-RCNN extends Mask-RCNN with two contributions: probit classifier and uncertainty-guided box predictor. The former uses Bayesian learning to estimate a probability distribution of the classifier head's weights (red diagonal stripes). Our contribution 1 is the efficient, analytical formulation of this Bayesian learning using the probit function. The latter explicitly estimates uncertainty of predicting bounding boxes (blue diagonal stripes), and uses the estimated uncertainty (violet) as input features along with the ROI-align-pooled features (dark yellow) for refining the boxes. The colored stripes depict the last layers of the classification, box, and segmentation heads learned in the few-shot setting during fine-tuning of the new classes.

head for object detection. Our iFS-RCNN also uses for each class a distinct set of the classification, bounding-box, and segmentation heads of Mask-RCNN, where the difference between the respective heads is in the last layer fine-tuned for each new class separately.

**iFSIS**: A recent approach to iFSIS [11] replaces the standard fully-connected classifier in Mask-RCNN with the cosine-similarity classifier. Unlike our iFS-RCNN, they do not use Bayesian learning and do not estimate bounding-box uncertainty. They convert activations of the classification head to a softmax distribution, and train the head with cross-entropy loss. In this way, the activation scores of all classes compete with each other to determine the class of the bounding box. As training examples of the new classes are scarce, their activation scores are likely to be smaller than those of the base classes in the softmax function. Consequently, their classifier is likely to be biased toward favoring the base classes. They address this bias by using the cosine-similarity classifier, where both the box feature and class weights are normalized to have unit length before the dot product for reducing a statistical difference between the base and new classes. In contrast, we directly use sigmoid activation of the classification head for predicting the class of the bounding box, and train our fully-connected classifier with the focal loss [23]. Our iFS-RCNN uses the sigmoid activation to predict each class independently, and thus alleviate the aforementioned bias.

## 3. Specification of iFS-RCNN

### 3.1. Problem Statement

We address N-way K-shot iFSOD and iFSIS, where abundant training examples of $N_b$ base classes are provided for pre-training. After that, access to training examples of the base classes is not available. When a few training examples $K$ of additional $N_n$ new classes are arbitrarily provided, our goal is to detect and segment all object instances belonging to all $N = N_b + N_n$ classes in a query image.

In the following, we specify our two contributions – the probit classifier and uncertainty-guided box predictor.

### 3.2. The Probit Classifier

As mentioned in Sec. 2, our iFS-RCNN uses sigmoid activation instead of softmax activation for predicting the class of a bounding box. With sigmoid activation, the scores for all classes are independently predicted, and thus our classi-

fication head effectively addresses the statistical difference between the base and new classes. However, in the FSOD and FSIS settings, our experiments (see Tab. 2) demonstrate that the classification head with the sigmoid activation usually gives a lower performance than that with the softmax activation, when weights of the classifier head (also referred to as class weights) are learned as point estimation.

To address this problem, we resort to Bayesian learning of a distribution of the class weights, and adopt the common variational framework. Formally, the class weights $w$ are characterized by the normal distribution, $w \sim \mathcal{N}(\mu, \Sigma)$, with mean $\mu \in \mathbb{R}^D$ and diagonal covariance matrix $\Sigma \in \mathbb{R}_+^{D\times D}$. Our goal is to learn $\mu$ and $\Sigma$ by minimizing the following variational objective:

$$L_c = l_d(p(c|f,\mu,\Sigma), c^*) + \mathrm{KL}(\mathcal{N}(\mu,\Sigma)||\mathcal{N}(\mathbf{0},\mathbf{1})), \quad (1)$$

where $f \in \mathbb{R}^D$ is a feature extracted from the bounding box; $c$ and $c^*$ are the predicted and ground-truth classes; $l_d$ is the incurred sigmoid focal loss [23] for predicting $c$; KL stands for the Kullback–Leibler divergence; and $p(c|f,\mu,\Sigma)$ is the posterior predictive distribution defined as

$$p(c|f,\mu,\Sigma) = \int \sigma(f^\top w)\mathcal{N}(w|\mu,\Sigma)dw, \quad (2)$$

where $\sigma(\cdot)$ denotes the sigmoid activation.

Once $\mu$ and $\Sigma$ are learned, class prediction amounts to the MAP problem: $c = \arg\max_{c'} p(c'|f,\mu,\Sigma)$. However, the integral in (2) is intractable. Prior work typically approximates the MAP problem with Monte Carlo sampling: $p(c|f,\mu,\Sigma) \approx \frac{1}{T}\sum_{w_t\sim\mathcal{N}(\mu,\Sigma)} \sigma(f^\top w_t)$ where $T$ is the number of Monte-Carlo samples. The Monte Carlo approximation, however, has a poor trade off between efficiency and accuracy, where using a sufficiently large $T$ would make our training and testing prohibitively slow.

Instead of the stochastic Monte Carlo sampling, we specify a more efficient deterministic approximation of the posterior predictive distribution. We first observe that our fine-tuning of the classification head's last layer using Bayesian learning is equivalent to learning a *Bayesian logistic regression* (BLR). Conveniently, the well-known probit function $\Phi(x)$ [30], [1, p. 219] provides a deterministic approximation to BLR. The probit function approximates the sigmoid function as $\sigma(x) \approx \Phi(\lambda x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\lambda x}{\sqrt{2}}\right)\right]$, where $\mathrm{erf}(\cdot)$ is the error function, and $\lambda^2 = \pi/8$ ensures that the two functions have the same slope at the origin. An important property of the probit function is that its convolution with a Gaussian function can be expressed analytically. Let $a = f^\top w \in \mathbb{R}$ be a random variable whose expectation and variance can be expressed as $\mathbb{E}[a] = \mu_a = f^\top\mu \in \mathbb{R}$ and $\mathbb{V}[a] = \Sigma_a = f^\top\Sigma f \in \mathbb{R}$. Then the posterior predictive distribution given by (2) can be efficiently approximated as

$$p(c|f,\mu,\Sigma) = \int \sigma(a)\mathcal{N}(a|\mu_a,\Sigma_a)da, \quad (3)$$

$$\approx \int \Phi(\lambda a)\mathcal{N}(a|\mu_a,\Sigma_a)da = \Phi\left(\frac{\lambda\mu_a}{(1+\lambda^2\Sigma_a)^{\frac{1}{2}}}\right), \quad (4)$$

$$\approx \sigma\left(\frac{f^\top\mu}{(1+\frac{\pi}{8}f^\top\Sigma f)^{\frac{1}{2}}}\right). \quad (5)$$

As our classifier head uses the probit function for the MAP class prediction, we call it the probit classifier. It is suitable for iFSOD and iFSIS for two reasons. It leverages Bayesian learning to address the paucity of training data. Also, it predicts a score for each class independently to address the incremental-learning setting.

### 3.3. Uncertainty-Guided Bounding Box Predictor

Object appearances, shapes, and scales in test images may significantly differ from a few training examples available. Also, target objects in query images may be subject to partial occlusion. All this gives rise to uncertainty in bounding box prediction. We seek to explicitly model this uncertainty when predicting four offset values $\{m_k\}_{k=1..4} \in \mathbb{R}^4$ that initially identify the location of bounding boxes. Specifically, as shown in Fig. 2, our box predictor additionally estimates four uncertainty values $\{u_k\}_{k=1..4} \in \mathbb{R}_+^4$ of the bounding box prediction, one for each of the $\{m_k\}$ predictions. The estimated uncertainty $\{u_k\}$ is then used as input along with the ROI-align-pooled features – extracted from the initially predicted box $m$ – to the box refinement module for the final offset bounding-box prediction $\{b_k\}_{k=1..4} \in \mathbb{R}^4$.

To learn how to predict uncertainty $\{u_k\}$, initial bounding-box $\{m_k\}$, and refined bounding-box $\{b_k\}$ on a few training examples, we specify the following box loss:

$$L_b = L_{\mathrm{u}} + L_{\text{refine}}, \quad (6)$$

where $L_{\mathrm{u}}$ is our new uncertainty-weighted box loss and $L_{\text{refine}}$ is loss incurred by the box refinement module.

We define $L_{\mathrm{u}}$ as

$$L_{\mathrm{u}} = \sum_{k=1}^{4} \frac{1}{2}\left(\frac{(m_k - b_k^*)^2}{u_k^2} + u_k^2\right), \quad (7)$$

where $b^*$ is the ground-truth box. The first term in (7) is aimed at minimizing a weighted difference between the ground-truth and predicted boxes. The weighting is inversely proportional to the predicted uncertainty $u_k$ such that the lower loss is incurred for box predictions with high uncertainty. The second term in (7) is aimed at minimizing uncertainty values such that the network incurs a penalty for predicting high uncertainty when trying to reduce the first term in (7).

When the box refinement module makes the final prediction $b$, it incurs the following loss:

$$L_{\text{refine}} = \sum_{k=1}^{4} \text{smooth L1}(b_k, b_k^*). \quad (8)$$

It is worth noting that our loss formulation fundamentally differs from other recent approaches aimed at estimating uncertainty in object detection. For example, recent approaches [15, 18] make the assumption that the bounding-box location and its uncertainty are governed by a Gaussian distribution. In contrast, we do not explicitly specify any probability distribution of box locations. iFS-RCNN appears related to Cascade-RCNN [3, 4] which also refines the initial box. However, these approaches do not explicitly predict uncertainty and thus cannot use uncertainty as an input feature for the box refinement as we do. By contrast, our uncertainty estimation in training serves to transfer "knowledge" from the base classes to the new testing classes. In experiments, these approaches show worse performance than our uncertainty-guided box refinement module.

### 3.4. Our Training and Testing Strategies

**Training on the base classes**

1. Obtain a variant of Mask-RCNN named Mask+Sigmoid by replacing the standard softmax classifier with the sigmoid classifier. Train Mask+Sigmoid with the following loss functions: sigmoid focal loss, $L_{\text{refine}}$ in (8), and mask-BCE loss.
2. Obtain a variant of Mask+Sigmoid named Mask+Sigmoid+Uncertainty by replacing the box predictor with the uncertainty-guided box predictor (contribution 2 in Fig. 2). While freezing other modules, train the uncertainty-guided box predictor of Mask+Sigmoid+Uncertainty with loss $L_b$ in (6).
3. Store the class weights $\mu_b$ of the base classes for the sigmoid classifier of Mask+Sigmoid+Uncertainty.

**Fine-tuning on the new classes**

1. Obtain a variant of Mask+Sigmoid+Uncertainty named iFS-RCNN by replacing the sigmoid classifier with the probit classifier (contribution 1 in Fig. 2).
2. While freezing other modules, train the probit classifier with loss $L_c$ in (1) to obtain the class weights $\mu_n, \Sigma_n$ of the new classes. Also, train the last layer of the box predictor with $L_b$ in (6); and the last layer of the segmentation head with the mask-BCE loss.

**Testing on the base and new classes**

1. Set $\mu = [\mu_b; \mu_n]$ and $\Sigma = [\mathbf{0}; \Sigma_n]$ for the probit classifier, where $[\cdot;\cdot]$ is a concatenation. Also, concatenate the weights estimated for the base and new classes to obtain the box and segmentation-mask heads.
2. Run iFS-RCNN on query images.

## 4. Experimental Results

**Datasets & Metrics**: We evaluate iFS-RCNN on the modified version of the COCO 2014 dataset [24] introduced by [16] for FSIS and FSOD. Also, we are the first to evaluate iFSOD, FSIS, and iFSIS on a new split of the LVIS dataset [12] introduced by [32] for FSOD. We report the common COCO-style evaluation metrics of both object detection and instance segmentation – namely, the average precision (AP) at multiple intersection-over-union (IoU) thresholds ranging from 0.5 to 0.95.

For COCO, the 20 categories shared with PASCAL VOC [7] are used as new classes while the remaining 60 classes are used as the base classes. We vary the number of examples for the new classes, i.e. $K = \{1, 2, 3, 5, 10, 30\}$, and report average results with $95\%$ confidence interval over 10 runs with different sets of few-shot examples for each $K$. This paper reports results for $K = \{1, 5, 10\}$ for brevity. Other results are in the supplementary material.

LVIS has 1230 classes where some have a large number of examples and some other, called rare classes, have only a few examples (less than 10 examples per class). Hence, the number of images for each class in LVIS has a long-tail distribution. We take the frequent (appearing in more than 100 images) classes and the common classes (10-100 images) in LVIS as the base classes, while the 454 rare classes (appearing less than 10 images) as the new classes. Due to the small number of training examples for the rare classes in LVIS, we cannot have multiple runs as in COCO, thus we follow [32] on their split with $K \leq 10$.

### 4.1. Implementation Details

Our backbone CNN is ResNet-50 [14] with the FPN of [22], as in recent work on FSIS. All variants of iFS-RCNN are implemented using the detectron2 toolbox [34] with the codebase of [32]. All variants of our approach, specified in Sec. 4.2, are trained using SGD and a batch size of 16 on 8 NVIDIA GPUs V100s, with two images per GPU. The learning rate is set to 0.02 and 0.01 for the pre-training and fine-tuning stages respectively. The number of iterations for the pre-training stage is 110000 with two weight decay steps with the rate of 10 at 80000 and 100000 iterations. The number of iterations for fine-tuning stage depends on the number of examples ranging from 500 iterations (with $K = 1$) to 6000 iterations (with $K = 30$). The hyper-parameters of the focal loss are $\gamma = 0.25, \alpha = 2$.

The threshold for filtering predictions before the NMS is 0.05. To select proposals for training the uncertainty-guided box predictor, we choose the predicted boxes with IoU larger than 0.7 with their closest ground-truth boxes. The threshold for deciding foreground and background in the segmentation mask head is 0.5. We use the softplus function $f(x) = \ln(1 + \exp(x))$ to ensure the class variances $\Sigma$ and box uncertainty $u$ are non-negative.

| | Object Detection | | | | | | Instance Segmentation | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number of shots | 1 | 2 | 3 | 5 | 10 | 30 | 1 | 2 | 3 | 5 | 10 | 30 |
| Mask-RCNN | 3.58 | 5.07 | 5.79 | 7.81 | 8.59 | 12.68 | 3.71 | 5.24 | 5.29 | 7.66 | 8.46 | 11.09 |
| Mask+Cosine | 3.39 | 4.87 | 5.19 | 6.87 | 7.96 | 12.52 | 3.40 | 5.00 | 4.75 | 6.68 | 7.72 | 11.03 |
| Mask+Sigmoid | 3.60 | 4.49 | 5.63 | 7.06 | 7.68 | 11.40 | 3.92 | 4.63 | 5.63 | 7.15 | 7.67 | 10.94 |
| Mask+Probit | 5.18 | 5.90 | 7.82 | 9.45 | 10.43 | 13.48 | 5.15 | 6.03 | 7.67 | 9.34 | 9.52 | 12.07 |
| Mask+MC | 4.52 | 5.45 | 7.02 | 8.85 | 9.57 | 13.05 | 4.54 | 5.19 | 6.91 | 8.26 | 8.98 | 11.55 |
| Mask+Sig+Uncert | 4.48 | 5.32 | 6.76 | 8.49 | 9.16 | 12.85 | 4.84 | 5.88 | 7.00 | 8.62 | 9.22 | 11.98 |
| Mask+Sig+Gauss | 3.74 | 4.77 | 5.89 | 7.33 | 7.86 | 11.65 | 3.94 | 4.72 | 5.87 | 7.24 | 7.76 | 11.02 |
| Mask+Sig+Refine | 3.87 | 4.57 | 5.78 | 7.48 | 8.23 | 11.95 | 3.99 | 4.77 | 5.68 | 7.40 | 7.87 | 11.01 |
| iFS-RCNN | 6.34 | 6.93 | 8.93 | 10.53 | 11.27 | 14.66 | 5.54 | 6.33 | 7.80 | 9.41 | 10.23 | 13.08 |

Table 2. Our ablation study on FSOD and FSIS with different $K = \{1, 2, 3, 5, 10, 30\}$ on COCO. The best results are in red, second-best results are in blue. Mask-RCNN and Mask+Cosine use the softmax activation with cross-entropy loss for training. The remaining ablations use the sigmoid activation with the focal loss for training.

| | Object Detection | | | | | | | | | Instance Segmentation | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tested on | New classes | | | Base classes | | | All classes | | | New classes | | | Base classes | | | All classes | | |
| Number of shots | 1 | 5 | 10 | 1 | 5 | 10 | 1 | 5 | 10 | 1 | 5 | 10 | 1 | 5 | 10 | 1 | 5 | 10 |
| TFA [32] | 2.90 | 7.00 | 9.10 | 31.90 | 32.30 | 32.40 | 3.60 | 11.50 | 14.20 | - | - | - | - | - | - | - | - | - |
| FSDetView [35] | 3.35 | 8.53 | 12.50 | 25.75 | 25.05 | 24.82 | 20.15 | 20.92 | 21.74 | - | - | - | - | - | - | - | - | - |
| GIFSOD [21] | - | - | 8.50 | - | - | 28.10 | - | - | 23.20 | - | - | - | - | - | - | - | - | - |
| ONCE [27] | 0.70 | 1.00 | 1.20 | 17.90 | 17.90 | 17.90 | 13.60 | 13.70 | 13.70 | - | - | - | - | - | - | - | - | - |
| LEAST [20] | 4.40 | 9.40 | 12.50 | 24.60 | 25.20 | 23.10 | 7.50 | 13.70 | 16.20 | - | - | - | - | - | - | - | - | - |
| iMTFA [11] | 3.23 | 6.07 | 6.97 | 27.81 | 24.13 | 23.36 | 21.67 | 19.62 | 19.26 | 2.81 | 5.19 | 5.88 | 25.90 | 22.56 | 21.87 | 20.13 | 18.22 | 17.87 |
| Mask+Sigmoid | 2.85 | 6.34 | 8.04 | 38.55 | 38.53 | 38.53 | 29.62 | 30.49 | 30.91 | 3.06 | 6.52 | 8.00 | 35.70 | 35.69 | 35.69 | 27.54 | 28.76 | 29.37 |
| iFS-RCNN | **4.54** | **9.91** | **12.55** | **40.08** | **40.06** | **40.05** | **31.19** | **32.52** | **33.02** | **3.95** | **8.80** | **10.06** | **36.35** | **36.33** | **36.32** | **28.45** | **29.89** | **30.41** |

Table 3. iFSOD and iFSIS results on COCO with different $K = \{1, 5, 10\}$. '-' indicates no results are reported. Best results are in **bold**.

Our segmentation head is the same as in Mask-RCNN [13]. It is trained with the binary cross-entropy (BCE) loss. Our box refinement module has the same architecture as the box head. The final layers of the uncertainty-guided box predictor and segmentation head are class-specific, and obtained by concatenating the class weights learned for the base and new classes.

## 4.2. Ablation Study

The following ablations are evaluated on the first run of COCO for studying how each component of iFS-RCNN affects the final performance.

- *Mask-RCNN*: the original Mask-RCNN [13] with the softmax classifier.
- *Mask+Cosine*: replace the dot product with the cosine-similarity classifier in Mask-RCNN as in [11].
- *Mask+Sigmoid*: replace the softmax classifier with the sigmoid classifier in Mask-RCNN (our strong baseline).
- *Mask+Probit*: replace the sigmoid classifier with the probit classifier in Mask+Sigmoid, our contribution 1 in Fig. 2.
- *Mask+MC*: replace the probit approximation with the Monte Carlo (MC) sampling in Mask+Probit, where the number of samplings is $T = 10$.
- *Mask+Sig+Uncert*: replace the box predictor in Mask+Sigmoid with the uncertainty-guided box predictor, our contribution 2 in Fig. 2.
- *Mask+Sig+Gauss*: a variant of Mask+Sigmoid which additionally predicts box uncertainty (similar to [15]) with Gaussian distribution assumption.
- *Mask+Sig+Refine*: similar to Cascade RCNN [3] as it does not explicitly predict uncertainty when refining the initial box prediction
- Our *iFS-RCNN* in Fig. 2.

Tab. 2 shows our evaluation of the above ablations on FSOD and FSIS (see Tab. 1). From Tab. 2, Mask-RCNN with the softmax activation outperforms other point-estimation-based ablations with the sigmoid activation. However, Mask-RCNN gives worse performance than the ablations which use the sigmoid activation with Bayesian learning for the Probit or MC. This justifies our choice of Bayesian-based classifiers with the sigmoid activation. Also, our probit classifier outperforms the MC classifier, suggesting that MC requires longer stochastic sampling $T \gg 10$, which would be prohibitively slow in practice. Importantly, our modeling of uncertainty in Mask+Sig+Uncert gives a substantial performance gain over Mask+Sigmoid, Mask+Sig+Gauss, and Mask+Sig+Refine. This justifies our contribution 2. Finally, our full approach, iFS-RCNN yields

| Settings | FSOD | FSIS | Object Detection | | | | Instance Segmentation | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Tested on | New | New | New | Base-c | Base-f | All | New | Base-c | Base-f | All |
| TFA [32] (gFSOD) | 18.35 | - | 16.90 | 24.30 | 27.90 | 24.40 | - | - | - | - |
| Mask-RCNN [13] (gFSOD & gFSIS) | 16.50 | 18.31 | 12.11 | 24.54 | 28.59 | 24.04 | 12.75 | 25.35 | 27.75 | 24.36 |
| Mask+Sigmoid (iFSOD & iFSIS) | 16.93 | 19.18 | 15.02 | 23.33 | 27.23 | 23.55 | 17.39 | 25.26 | 27.05 | 24.75 |
| iFS-RCNN (iFSOD & iFSIS) | **20.76** | **21.06** | **18.38** | **26.11** | **30.12** | **26.46** | **18.26** | **26.29** | **28.46** | **25.90** |

Table 4. Object detection and instance segmentation results with AP metric for FSOD, FSIS, iFSOD, and iFSIS tasks on LVIS. The best results are in **bold**. TFA and Mask-RCNN are trained with more supervision than Mask+Sigmoid and iFS-RCNN as described in Tab. 1. Base-c and Base-f indicate the common classes ($\geq$ 100 images) and frequent classes (10-100 images) among the base classes.

the best performance, about +2.5 performance gain over the strong baseline Mask+Sigmoid and +2 over Mask-RCNN. In the following experiments, we choose Mask+Sigmoid and iFS-RCNN to compare with prior work.

### 4.3. Comparison with Prior Work on COCO

Tab. 3 compares our results on COCO in iFSOD and iFSIS with the strong baseline Mask+Sigmoid, approaches designed for FSOD and FSIS (TFA [32], FSDetView [35]), and approaches designed for iFSOD and iFSIS (GIFSOD [21], ONCE [27], LEAST [20], MTFA, and iMTFA [11]).

TFA (a fine-tuning-based approach) and FSDetView (an episodic-training-based approach) are adapted to the iFSOD setting as follows. TFA is first trained on the base classes, resulting in model 1. Then, TFA is fine-tuned on the new classes, resulting in model 2. Finally, we run the models 1 and 2 on the same test images and select the top 100 predictions, as in the COCO evaluation protocol. For FSDetView, after pre-training the model on training examples of the base classes, we use the pretrained model to estimate the prototype for each base class, and then run the pretrained model on the new classes to extract their prototypes. FSDetView uses the prototypes of both new and base classes for object detection in test images. As TFA and FSDetView were not originally designed for iFSOD, their aforementioned adaptation from FSOD to iFSOD has two limitations: (1) Storage and running of two distinct models/prototypes for predicting the base and new classes; (2) The two distinct models may independently each yield a high score on a base and new class in the test image, which is then hard to resolve.

Among the approaches aimed at iFSOD, ONCE is based on YOLO [28], and LEAST and GIFSOD are based on Faster-RCNN [29]. iMTFA is a variant of Mask+Cosine, where the weight $\mu_n$ is set to the box feature $f$ extracted from an example of the new class.

For the comparison in Tab. 3, results are averaged over 10 runs with different sets of few-shot examples. From this table, for the iFSOD setting, iFS-RCNN outperforms state of the art (SOTA) approaches slightly on new classes (+0.05) while significantly on the base classes (+12) for $K = 10$. For the iFSIS setting, iFS-RCNN outperforms the SOTA iMTFA with significant margins. Specifically, for $K = 10$, our performance gains are about +6 for the new classes and +16 for the base classes on segmentation. Our performance gains are very large, especially for the challenging COCO dataset.

### 4.4. Results on LVIS

Tab. 4 reports our results on LVIS with $K \leq 10$ on iFSIS and FSIS. Although Mask+Sigmoid uses less supervision than Mask-RCNN and TFA, they give comparable results. iFS-RCNN significantly outperforms Mask-RCNN with the gains of +3 on FSIS and +6 on iFSIS on the new classes. These results demonstrate the effectiveness of our iFS-RCNN on the more challenging dataset LVIS.

### 4.5. Qualitative Evaluation

Fig. 3 illustrates some of our results. The top two rows show success cases while the bottom row shows failure cases. For the failure cases from left to right: the train is misclassified as bus due to very similar appearance, the bird detection has a large bounding box due to the occlusion, the human leg is segmented as a part of the motorbike due to similar appearance, and the small boats far behind are not detected. Fig. 4 shows our box refinement results. As can be seen, high uncertainty about the box prediction gives a considerable box refinement.

## 5. Conclusion and Discussion

We have specified iFS-RCNN to address N-way K-shot incremental few-shot object detection (iFSOD) and instance segmentation (iFSIS). iFS-RCNN leverages Mask-RCNN, but modifies the standard softmax classifier and bounding-box prediction head. The softmax classifier has been replaced with the sigmoid classifier for alleviating a statistical imbalance of the base and new classes. The sigmoid classifier has been further improved via Bayesian learning to robustly estimate a distribution of the classifier head's weights on a few examples of the new classes. For this Bayesian learning, we have proposed an analytical approximation using the probit function. Also, the standard box predictor of Mask-RCNN has been extended to explicitly predict uncertainty of box prediction, and use the estimated uncertainty as input to a bounding-box refinement module. iFS-RCNN significantly outperforms iMTFA – the state of

Figure 3. Representative results of iFS-RCNN on COCO-new with $K = 30$. The top two rows show success cases while the bottom row shows failure cases. In the last row, from left to right: the train is detected as bus, bonding-box is too large for the bird, the human leg is segmented as a part of the motorbike, the small boats far behind are not detected.


Figure 4. Bounding-box refinement with our uncertainty-guided box predictor on COCO-new with $K = 1$. For each pair, left: initial box, right: refined box. The box labels show left (L), top (T), right (R), and bottom (B) side uncertainties; red text indicates high uncertainty ($\geq$ 10) that lead to large refinements. Yellow arrows indicate the refined direction of each box side.

the art in iFSIS – with very large performance gains on the COCO dataset, +6 on the new classes and +16 on the base classes with 10 training examples. Moreover, we are the first to report the results on the LVIS dataset for the iFSOD, FSIS, and iFSIS problems, where we outperform strong baselines.

As for potential limitations, our contribution 1, in general, could be applied to any object detector, including RPN-based ones like Faster-RCNN [29] and Mask-RCNN [13]), point-based ones like FCOS [31] and Center-Net [39]), and transformer-based ones like DETR [5]. However, in our experiments, we have succeeded in applying it only to the RPN-based detector. This might be due to the pre-selected balanced training examples of background and foreground classes in the second stage of Faster-RCNN, which is absent in the other two detector frameworks. Also, while our estimation of uncertainty of box prediction and our new uncertainty-weighted box loss have enabled significant performance gains, our contribution 2 lacks a theoretical underpinning as to why our formulation outperforms a Gaussian-based uncertainty estimation.

As any system for object detection and segmentation, ours could be weaponized and misused for malicious violations of privacy, as well as used for efficiently discovering and fighting against such misuses due to its few-shot learning capabilities.

**Acknowledgement.** This work was supported in part by DARPA MCS Award N66001-19-2-4035.

## References


[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

[3] 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.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[6] 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*, 2018.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[8] 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.

[9] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4527–4536, 2021.

[10] 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.

[11] Dan Andrei Ganea, Bas Boom, and Ronald Poppe. Incremental few-shot instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1185–1194, June 2021.

[12] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[13] 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.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2888–2897, 2019.

[16] 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/CVF International Conference on Computer Vision*, pages 8420–8429, 2019.

[17] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2019.

[18] Youngwan Lee, Joong-won Hwang, Hyung-Il Kim, Kimin Yun, and Joungyoul Park. Localization uncertainty estimation for anchor-free object detection. *arXiv preprint arXiv:2006.15607*, 2020.

[19] Aoxue Li and Zhenguo Li. Transformation invariant few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3094–3102, 2021.

[20] Pengyang Li, Yanan Li, and Donghui Wang. Class-incremental few-shot object detection. *arXiv preprint arXiv:2105.07637*, 2021.

[21] Yiting Li, Haiyue Zhu, Jun Ma, Chek Sing Teo, Cheng Xiang, Prahlad Vadakkepat, and Tong Heng Lee. Towards generalized and incremental few-shot object detection. *arXiv preprint arXiv:2109.11336*, 2021.

[22] 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.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *European conference on computer vision (ECCV)*, June 2016.

[25] Claudio Michaelis, Ivan Ustyuzhaninov, Matthias Bethge, and Alexander S. Ecker. One-shot instance segmentation. *arXiv preprint arXiv:1811.11507*, 2018.

[26] Khoi Nguyen and Sinisa Todorovic. Fapis: A few-shot anchor-free part-based instance segmenter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11099–11108, 2021.

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

[28] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[29] 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.

[30] David J Spiegelhalter and Steffen L Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20(5):579–605, 1990.

[31] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9627–9636, 2019.

[32] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020.

[33] 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. Springer, 2020.

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

[35] 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. Springer, 2020.

[36] 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 *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2019.

[37] 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/CVF International Conference on Computer Vision*, pages 9577–9586, 2019.

[38] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, and Shijian Lu. Meta-detr: Few-shot object detection via unified image-level meta-learning. *arXiv preprint arXiv:2103.11731*, 2021.

[39] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.