Incremental Few-Shot Object Detection

Juan-Manuel Pérez-Rúa[1] j.perez-rua
Xiatian Zhu[1] xiatian.zhu
Timothy Hospedales[1,3] t.hospedales
Tao Xiang[1,2] tao.xiang
*@samsung.com
[1]Samsung AI Centre, Cambridge
[2]University of Surrey [3]University of Edinburgh
United Kingdom

Abstract

Most existing object detection methods rely on the availability of abundant labelled training samples per class and offline model training in a batch mode. These requirements substantially limit their scalability to open-ended accommodation of novel classes with limited labelled training data. We present a study aiming to go beyond these limitations by considering the Incremental Few-Shot Detection (iFSD) problem setting, where new classes must be registered incrementally (without revisiting base classes) and with few examples. To this end we propose OpeN-ended Centre nEt (ONCE), a detector designed for incrementally learning to detect novel class objects with few examples. This is achieved by an elegant adaptation of the CentreNet detector to the few-shot learning scenario, and meta-learning a class-specific code generator model for registering novel classes. ONCE fully respects the incremental learning paradigm, with novel class registration requiring only a single forward pass of few-shot training samples, and no access to base classes – thus making it suitable for deployment on embedded devices. Extensive experiments conducted on both the standard object detection and fashion landmark detection tasks show the feasibility of iFSD for the first time, opening an interesting and very important line of research.

1. Introduction

Despite the success of deep convolutional neural networks (CNNs) [21, 24, 45, 49] in object detection [43, 42, 30, 28], most existing models can be only trained offline via a lengthy process of many iterations in a *batch* setting. Under this setting, all the target classes are known, each class has a large number of annotated training samples, and all training images are used for training. This annotation cost and training complexity severely restricts the potential for these methods to grow and accommodate new classes online. Such a missing capability is required in robotics applications [32, 1], when the detector is running on embedded devices, or simply to scale up to addressing the long tail of object categories to recognise [31]. In contrast, humans learn new concepts such as object classes incrementally without forgetting previously learned knowledge [14], and often requiring only a few visual examples per class [36, 3]. Motivated by the vision of closing this gap between state-of-the-art object detection and human-level intelligence, a couple of very recent studies [53, 22] proposed methods for few-shot object detector learning.

Nonetheless, both methods [53, 22] are fundamentally *unscalable* to real-world deployments in open-ended or robotic learning settings, due to lacking the capability of *incremental* learning of novel concepts from a data stream over time. Specifically, they have to perform a costly training/updating of the detection model using the data of both old (base) and new (novel) classes together, whenever a novel class should be added. Consequently, while they successfully reduce annotation requirements, these models essentially reduce to the conventional batch learning paradigm. This leads to a prohibitively expensive quadratic computation cost in number of categories in an incremental scenario, and also raises issues in data privacy over time [9, 40]. Meanwhile, storage and compute requirements prohibit on-device deployment in robotic scenarios where a robot might want to incrementally register objects encountered in the world for future detection [32, 1].

To overcome the aforementioned limitation, we study a very practical learning setting – *Incremental Few-Shot Detection* (iFSD). The iFSD setting is defined by: **(1)** The detection model can be pre-trained in advance on a set of base classes each with abundant training samples available – it makes sense to use existing annotated datasets to bootstrap a model [32]. **(2)** Once trained, an iFSD model should be capable of deployment to real-world applications where novel classes can be registered at any time using only a few

annotated examples. The model should provide good performance for all classes observed so far (i.e., learning without forgetting). **(3)** The learning of novel classes from an unbounded stream of examples should be feasible in terms of memory footprint, storage, and compute cost. Ideally the model should support deployment on resource-limited devices such as robots and smart phones.

Conventional object detection methods are unsuited to the proposed setting due to the intrinsic need for batch learning on large datasets as discussed earlier. An obvious idea is to fine-tune the trained model with novel class training data. However without revisiting old data (batch setting) this causes a dramatic degradation in performance for existing categories due to the *catastrophic forgetting* challenge [14]. The state-of-the-art few-shot object detection methods [53, 22] suffer from the same problem too, if denied access to the base (old) class training data and adapted sequentially to novel classes (see the evaluations in Table 2).

In this work, as the first step towards the proposed incremental few-shot object detection problem in the context of deep neural networks, we introduce *OpeN-ended Centre nEt* (ONCE). The model is built upon the recently proposed CentreNet [56], which was originally designed for conventional batch learning of object detection. We take a feature-based knowledge transfer strategy, decomposing CentreNet into class-generic and class-specific components for enabling incremental few-shot learning. More specifically, ONCE uses the abundant base class training data to first train a class-generic feature extractor. This is followed by meta-learning a class-specific code generator with simulated few-shot learning tasks. Once trained, given a handful of images of a novel object class, the meta-trained class code generator elegantly enables the ONCE detector to incrementally learn the novel class in an efficient feed-forward manner during the meta-testing stage (novel class registration). This is achieved without requiring access to base class data or iterative updating. Compared with [22, 53], ONCE better fits the iFSD setting in that its performance is *insensitive* to the arrival order and choice of novel classes. This is due to not using softmax-based classification but per-class thresholding in decision making. Importantly, since each class-specific code is generated independent of other classes, ONCE is intrinsically able to maintain the detection performance of the base classes and any novel classes registered so far.

We make three **contributions** in this work: **(1)** We investigate the heavily understudied Incremental Few-Shot Detection problem, which is crucial to many real-world applications. To the best of our knowledge, this is the first attempt to reduce the reliance of a deeply-learned object detector on batch training with large base class datasets during few-shot enrolment of novel classes, unlike recent few-shot detection alternatives [22, 53]. **(2)** We formulate a novel *OpeN-ended Centre nEt* (ONCE) by adapting the recent CentreNet detector to the incremental few-shot scenario. **(3)** We perform extensive experiments on both standard object detection (COCO [29], PASCAL VOC [12]) and fashion landmark detection (DeepFashion2 [15]) tasks. The results show ONCE's significant performance advantage over existing alternatives.

2. Related Work

Object detection Existing deep object detection models fall generally into two categories: (1) Two-stage detectors [19, 18, 43, 20, 8], (2) One-stage detectors [30, 41, 42, 28, 56, 57, 25]. While usually being superior in detection performance, the two-stage methods are less efficient than the one-stage ones due to the need for object region inference (and subsequent classification from the set of object proposals). Typically, both approaches assume a large set of training images per class and need to train the detector in an offline batch mode. This restricts their usability and scalability when novel classes must be added on the fly during model deployment. Despite being non-incremental, they can serve as the detection backbone of a few-shot detector. Our ONCE method is based on the one-stage CentreNet [56] which is chosen because of its efficiency and competitive detection accuracy, as well as the fact that it can be easily decomposed into class-generic and specific parts for adaptation to the Incremental Few-Shot Detection problem.

Few-shot learning For image recognition, efficiently accommodating novel classes on the fly is widely studied under the name of few-shot learning (FSL) [51, 38, 34, 13, 46, 48, 5]. Assuming abundant labelled examples of a set of base classes, FSL methods aim to meta-learn a data-efficient learning strategy that subsequently allows novel classes to be learned from very limited per-class examples. A large body of FSL work has investigated how to learn from such scarce data without overfitting [34, 44, 6, 26, 7, 17, 47, 55, 50, 27, 39, 11, 16]. Nonetheless, these FSL works usually focus on classification of whole images, or *well cropped* object images. This is much simpler than object detection as object instances do not need to be separated from diverse background clutter, or localised in space and scale. In this work we extend few-shot classification to the more challenging object detection task.

Few-shot object detection and beyond A few recent works have attempted to exploit few-shot learning techniques for object detection [53, 22, 23]. However, these differ significantly from ours in that they consider a *non-incremental* learning setting, which restricts dramatically their scalability and applicability in scenarios where access to either large-scale base class data is prohibitive. This is the case for example, due to limited computational resources and/or data privacy issues. We therefore consider the more

practical *incremental* few-shot learning setting, eliminating the infeasible requirement of repeatedly training the model on the large-scale base class training data[1]. While this more challenging scenario inevitably lead to inferior performance compared to *non-incremental* learning, as shown in our experiments, it is more representative of natural human learning capabilities and thus provides a good research target with great application potential once solved.

There are other techniques that aim to minimise the amount of data labelling for object detection such as weakly supervised learning [4, 10] and zero-shot learning [58, 37, 2]. They assume different forms of training data and prior knowledge, and conceptually are complementary to our iFSD problem setting. They can thus be combined when there are multiple input sources available (e.g., unlabelled data or semantic class descriptor).

3. Methodology

Problem Definition We consider the problem of *Incremental Few-Shot Detection (iFSD)*: obtaining a learner able to *incrementally* recognise *novel* classes using only a few labelled examples per class. We consider two *disjoint* object class sets: the *base classes* used to bootstrap the system, which are assumed to come with abundant labelled data; and the *novel classes* which are sparsely annotated, and to be enrolled incrementally. That is, in a computationally efficient way, and without revisiting the base class data.

3.1. Object Detection Architecture

To successfully learn object detection from sparsely annotated novel classes, we wish to build upon an effective architecture, and exploit knowledge transfer from base classes already learned by this architecture. However, the selection of the base object detection architecture *cannot* be arbitrary given that we need to adapt the detection model to novel classes on-the-fly. For example, while Faster R-CNN [43] is a common selection and provides strong performance given large scale annotation, it is less flexible to accommodate novel classes due to a two-stage design and the use of softmax-based classification.

In this work, we build upon the recently developed object detection model CentreNet [56] based on several considerations: **(1)** It is a highly-efficient one-stage object detection pipeline with better speed-accuracy trade-off than alternatives such as SSD [30], RetinaNet [28], and YOLO [41, 42]. **(2)** Importantly, it follows a *class-specific* modelling paradigm, which allows easy and efficient introduction of novel classes in a plug-in manner. Indeed, the core feature of CentreNet, the per-class heatmap-based centroid prediction is naturally appropriate for incremental learning as required in our setting.

[1]Some works have studied incremental learning of object detectors [35]. However, they do not tackle the few shot regime.

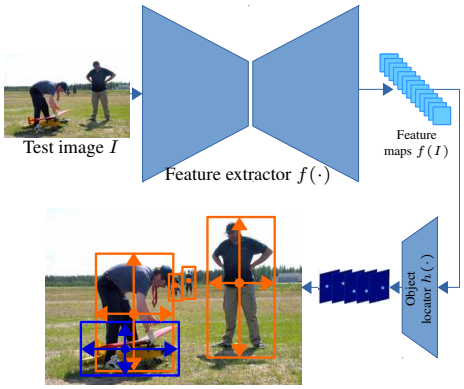

Figure 1: Overview of Centre-Net. A backbone network, which can be implemented with resolution-reducing blocks followed by upsampling operations, generates feature maps (top). These feature maps are further processed by a small CNN (the object locator) and transformed into a set of per-class heatmaps encoding the object box centre points and its size (bottom).

3.1.1 A Review of the CentreNet Model

The key idea of CentreNet is to reformulate object detection as a *point+attribute* regression problem. It is inspired by keypoint detection methods [33, 52], taking a similar spirit to [25, 57] without the need for point grouping and post-processing. The architecture of CentreNet is depicted in Fig. 1. Specifically, as the name suggests, CentreNet takes the centre point and the spatial size (*i.e.* the width & height) of an object bounding box as the regression target. Both are represented with 2D heatmaps, generated according to the ground-truth annotation. In training, the model is optimised to predict such heatmaps, supervised by an L_1 regression loss. For model details, we refer the readers to the original paper due to space limit.

Remarks It is worth mentioning that this keypoint estimation based formulation for object detection not only eliminates the need for region proposal generation, but also enables object location and size prediction in a common format by predicting corresponding pixel-wise aligned heatmaps. For few-shot object detection in particular, a key merit of CentreNet is that each individual class maintains its own prediction heatmap and makes independent detection by activation thresholding. We show next how to exploit this property of CentreNet in order to support incremental enrolment of novel classes in an order-free and combination-insensitive manner, without interference between old & new classes. This is in contrast to the softmax classification used in existing models [53, 22] where interactions between classes make this vision hard to achieve.

3.2. Incremental Few-shot Object Detection

As CentreNet is a batch learning model, it is *unsuited* for iFSD. We address this problem by incorporating a meta-learning strategy [51, 44] to CentreNet architecture, which

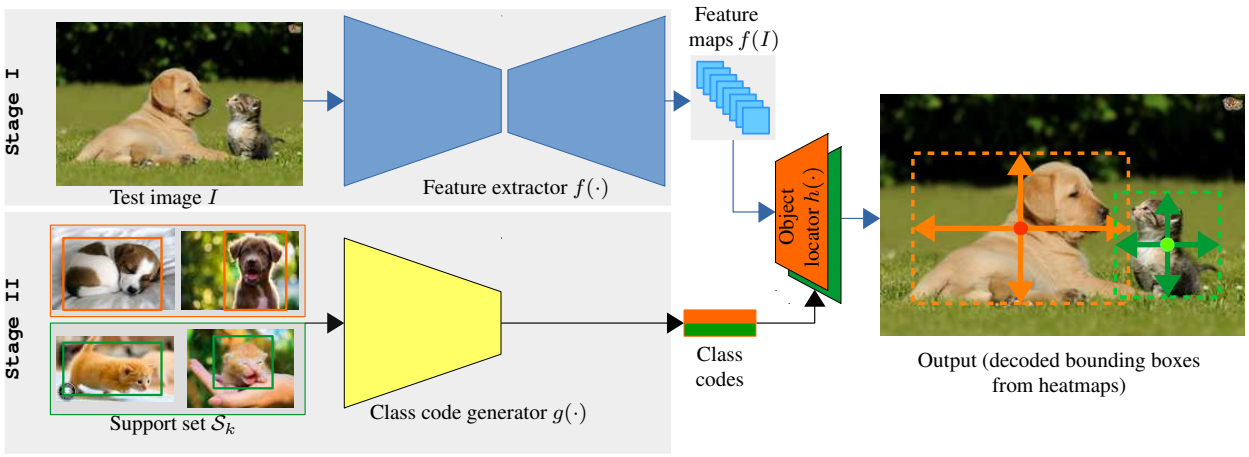

Figure 2: The architecture of our *OpeN-ended Centre nEt* (ONCE) model. Specifically, the feature extractor (an encoder-decoder model in our implementation, coloured blue in the top-left) generates the class-generic feature maps $f(I)$ of a test image. These maps are further convolved with the class-specific codes (coloured orange for the dog class, and green for the cat class) predicted by the class code generator (bottom-left, coloured yellow) from a few labelled support samples per class, to generate the object detection result in heatmap format (not shown for simplicity). The model training of ONCE involves two stages: **(1)** Stage I: a regular CentreNet-like supervised learning is performed on the abundant training data of base classes. **(2)** Stage II: episodic meta-training is performed with the weights of the feature extractor being frozen, allowing the class code generator to learn how to generate a class-specific code from a small per-class support set such that the model can generalise well to unseen novel classes (right). The base classes are used as *fake* novel classes in meta-training. It is noted that ONCE can also be applied for other detection problems, *e.g.*, fashion landmark localisation.

results in the proposed ***OpeN-ended Centre nEt*** (ONCE).

Model formulation ONCE starts by decomposing CentreNet into two components: (i) *feature extractor*, which is shared by all the base and novel classes, and (ii) *object locator*, which contains class-specific parameters for each individual class to be detected. Specifically, feature extractor takes as input an image, and outputs a 3D feature map. Then, the object locator analyses the feature map with a class-specific code used as convolutional kernel and yields the object detection result for that class in form of a heatmap.

In a standard extractor/locator decomposition of CentreNet, the object locator still needs to be trained in batch mode, and with large scale training data. In ONCE, we further paramaterise the object locator by a meta-learned generator network, where the generator network synthesises the parameters of the locator network (*i.e.*, the class-specific convolutional kernel weights) given a few-shot support set. In this way, we transform the conventional batch-mode detector learning problem (second CNN in Fig. 1, indicated with the tag "object locator") into a feed-forward pass of the parameter generator meta-network (class code generator in Fig. 2). To achieve this, we perform meta-learning to train the class code generator to solve few-shot detector learning tasks via weight synthesis given a support set (resulting in the green and orange class-specific object locators in Fig. 2).

Meta-Training: Learning a Few-Shot Detector To fully exploit the base classes with rich training data, we train ONCE in two sequential stages. In the first stage, we train the class-agnostic feature extractor on the base-class training data. This feature extractor is then fixed in subsequent steps as per other few-shot strategies [48]. In the second stage, we learn few-shot object-detection by jointly training the object locator, which is conditioned on a class-specific code; along with a meta-network that generates it given a support set. This is performed by episodic training that simulates few-shot episodes that will be encountered during deployment. In the following sections we describe the training process in more detail.

Meta-Testing: Enrolling New Classes At test time, given a support set of novel classes each with a few labelled bounding boxes, we directly deploy the trained feature extractor, object locator, and code generator learned during meta-training above. The meta-network generates object-specific weights from the support-set (few-shot) and the object locator uses these to detect objects in the test images. This means that novel class objects are detected in test images in a feed forward manner, *without* model training and/or adaptation.

3.2.1 Stage I: Feature Extractor Learning

We aim to learn a class-agnostic feature extractor f in ONCE. This can be simply realised by standard supervised learning with bound-box level supervision on the base classes, similarly to the original CentreNet [56]. Concretely, we perform keypoint detection and train the detection model (including both feature extractor $f(\cdot)$ and object locator $h(\cdot)$) by heatmap regression loss. In this stage we train a complete feature extractor and object locator pipeline, although the objective of this stage is solely to learn a robust feature extractor $f(\cdot)$. The locator learned

in this stage is a regular CentreNet locator, which will be discarded in stage II, but will be used at the test time for the base classes.

Given a training image $I \in \mathcal{R}^{h\times w\times 3}$ of height h and width w, we extract a class-agnostic feature map $\mathbf{m} = f(I)$, $\mathbf{m} \in \mathcal{R}^{\frac{h}{r}\times\frac{w}{r}\times c}$. The object locator then detects objects of each class k by processing the feature map with a learned class convolutional kernel $\mathbf{c}_k \in \mathcal{R}^{1\times 1\times c}$, where r is the output stride and c the number of feature channels. We then obtain the heatmap prediction $Y_k \in \mathcal{R}^{\frac{h}{r}\times\frac{w}{r}}$ for class k as:

$$Y_k = h(\mathbf{m}, \mathbf{c}_k) = \mathbf{m} \odot \mathbf{c}_k, \quad k \in \{1, 2, \cdots, K_b\}, \tag{1}$$

where $\odot$ denotes the convolutional operation, and K_b denotes the number of base classes.

For locating the object instances of class k, we start by identifying local peaks $\mathcal{P}_k = \{(x_i, y_i)\}_{i=1}^{n}$, which are the points whose activation value is higher or equal to its 8-connected spatial neighbours in Y_k. The bounding box prediction is inferred as

$$\begin{aligned}(x_i + \delta x_i - w_i/2, \;\; y_i + \delta y_i - h_i/2, \\ x_i + \delta x_i + w_i/2, \;\; y_i + \delta y_i + h_i/2)\end{aligned} \tag{2}$$

where $(\delta x_i, \delta y_i) = O_{x_i,y_i}$ is the offset prediction, $O \in \mathcal{R}^{\frac{h}{r}\times\frac{w}{r}\times 2}$, and $(h_i, w_i) = S_{x_i,y_i}$ is the size prediction, $S \in \mathcal{R}^{\frac{h}{r}\times\frac{w}{r}\times 2}$, generated by the offset and size codes in the same fashion as the class codes. Given the ground-truth bounding boxes and this prediction, we use the L_1 regression loss for model optimisation on the parameters of feature extractor f and parameters $\mathbf{c}$ of locator h. In practice, the feature extractor model is implemented with the ResNet-based backbone of [52].

3.2.2 Stage II: Class Code Generator Learning

The class code parameters $\mathbf{c}$ learned for detection above are fixed parameters for base-classes only. To deal with the iFSD setting, besides these base-class codes we need an *inductive* class code generator $g(\cdot)$ that can efficiently synthesise class codes for novel classes on the fly during deployment, given only a few labelled samples. To train the class code generator $g(\cdot)$, we exploit an episodic meta-learning strategy [51]. This uses the base class data to sample a large number of few-shot tasks, thus simulating the test-time requirement of few-shot learning of new tasks. While episodic meta-learning is widely used in few-shot recognition, we customise a strategy for *detection* here.

Specifically, we define an iFSD task T as uniform distribution over possible class label sets L, each with one or a few unique classes. To form an *episode* to compute gradients and train the class code generator $g(\cdot)$, we start by sampling a class label set L from T (*e.g.*, $L = \{person, bottle, \cdots\}$). With L, we sample a *support* (meta-training) set $\mathcal{S}$ and a *query* (meta-validation) set $\mathcal{Q}$. Both $\mathcal{S}$ and $\mathcal{Q}$ are labelled samples of the classes in L.

In the forward pass, the support set $\mathcal{S}$ is used for generating a class code for each sampled class k as:

$$\tilde{\boldsymbol{c}}_k = g(\mathcal{S}_k), \tag{3}$$

where $\mathcal{S}_k$ are the support samples of class k. With these codes $\{\tilde{\boldsymbol{c}}_k\}$, our method then performs object detection for query images I by using the feature extractor (Eq. (4)) and object locator (Eq.(5)):

$$\boldsymbol{m} = f(I), \;\; \text{with} \;\; I \in \mathcal{Q}, \tag{4}$$

$$\tilde{Y} = h(\boldsymbol{m}, \tilde{\boldsymbol{c}}_k). \tag{5}$$

ONCE is then trained to minimise the mean prediction error on $\mathcal{Q}$ by updating solely the parameters of the code generator (cf. Eq. (3)). Same as CentreNet, L_1 loss is used as the objective function in this stage, defined as $|\tilde{Y} - Z|$ where Z is the ground-truth heatmap.

3.2.3 Meta Testing: Enrolling New Classes

Given the feature extractor (f, trained in Stage I), the code generator (g, trained in stage II), and the object locator (h, Eq. (1)), At test time, ONCE can efficiently enrol any new class with a few labelled samples in a feed forward manner without model adaptation and update.

The meta-testing for a novel class is summarised as:

1. Obtaining its class code with a few-shot labelled set using Eq. (3);
2. Computing the test image features by using Eq. (4);
3. Locating object instances of the novel class by Eq. (1);
4. Obtaining all the object candidates using Eq. (2);
5. Finding the heatmap local maxima to output the final detection result of that class.

This process applies for the base classes except that *Step 1* is no longer needed since their class codes are already obtained from the training stage I (cf. Eq. (1)). In doing so, we can easily introduce novel classes independently which facilitates the model iFSD deployment.

3.2.4 Architecture

For the feature extractor function f, we start from a strong and simple baseline architecture [52] that uses ResNet [21] as backbone. This architecture consists of an encoder-decoder pair that first extracts a low resolution 3D map and then expands it by means of learnable upsampling convolutions, outputting high resolution feature maps $f(I)$ for an input image I. We leverage the same backbone for the class code generator (without the upsampling operations). Before

Method	Novel Classes		Base Classes		All Classes	
	AP	AR	AP	AR	AP	AR
Fine-Tuning	1.4	8.2	20.7	23.4	15.8	24.4
Feature-Reweight [22]	5.6	10.1	-	-	-	-
Meta R-CNN* [53]	**8.7**	**12.6**	-	-	-	-
ONCE	5.1	9.5	22.9	29.9	18.4	24.8

Table 1: Non-incremental few-shot object detection performance on **COCO *val2017*** set. Training setting: 10-shot per novel class and all the base class training data. '*': Using different (unknown) support sets of novel classes. '-': No reported results.

meta-training (stage II), the class code generator weights are initialised by cloning the weights of the encoder part of the feature extractor. The final convolution outputs are globally pooled to form the class codes $\mathbf{c}_k$, giving a code size of 256[2]. To handle support sets with variable size, we adopt the invariant set representation of [54] by average pooling of the class code generator outputs for every image $I_i^{k,s}$ in $\mathcal{S}_k$. The code and trained model will be released.

4. Experiments

4.1. Non-Incremental Few-Shot Detection

We start the experimental section with an important contextual experiment. We evaluated the performance of ONCE in the *non-incremental setting* as studied in [22, 53]. In particular, we use COCO [29], a popular object detection benchmark, covering 80 object classes from which 20 are left-out to be used as novel classes. These meta-testing classes happen to be the 20 categories covered by the PASCAL VOC dataset [12]. The remaining 60 classes in COCO serve as base classes. For model training, we used 10 shots per novel class along with all the base class training data. The results on COCO in Table 1 show that, while not directly comparable due to using different detection backbones and/or data split, ONCE approaches the performance of the two state-of-the-art models [22, 53]. We continue our experimental analysis hereafter with the incremental setting, which happens to be much more challenging and not trivially tackled with previous methods [22, 53].

4.2. Incremental Few-Shot Object Detection

Experimental setup For evaluating iFSD, we followed the evaluation setup of [22, 53] but with the key differences that the base class training data is *not* accessible during meta-testing, and incremental update for novel classes is required. In particular, the widely used object detection benchmarks COCO [29] and PASCAL VOC [12] are used. As mentioned before, COCO covers 80 object classes including all 20 classes in PASCAL VOC. We treated the 20 VOC/COCO shared object classes as novel classes, and the remaining 60 classes in COCO serve as base classes. This leads to two dataset splits: same-dataset on COCO and cross-dataset with 60 COCO classes as base.

For the ***same-dataset*** evaluation on COCO, we used the *train* images of base classes for model meta-training. In each episode, we randomly sampled 32 tasks, each of which consisted of a 3-class detection problem, and for which each class had 5 annotated bounding boxes per-class. Larger learning tasks may be beneficial for performance but requiring more GPU memory in training, thus not possible with the resources at our disposal. For meta-testing, we randomly sampled a support set from the training split of all 20 novel classes. To enrol these 20 novel classes to the model, we consider two settings: *incremental batch*, or *continuous incremental learning*. In the incremental batch setting, all 20 novel classes are added at once with a single model update. In the continual incremental learning setting, the 20 novel classes are added one by one with 20 models updates. We evaluated few-shot detection learning with $k \in \{1, 5, 10\}$) bounding boxes annotated for each novel class. In practice, we used the same support sets of novel classes as [22] for enabling a direct comparison between incremental learning and non-incremental learning (the data split used in [53] is not publicly available). We then evaluated the model performance on the *validation* set of the novel classes.

For the ***cross-dataset*** evaluation from COCO to VOC, we used the same setup and train/test data partitions as above, except that the model was evaluated on the PASCAL VOC 2007 *test* set. That is, the meta-training support/query sets were drawn from COCO, while meta-testing was performed using VOC training data for few-shot detector learning, and VOC testing data for evaluation.

Competitors We compared our ONCE method with several alternatives: (1) The standard *Fine-Tuning* method, (2) a popular meta-learning method *MAML* (the first-order variant) [13, 34], and (3) a state-of-the-art (non-incremental) few-shot object detection method *Feature-Reweight* [22]. In particular, since [22] was originally designed for the non-incremental setting, we adapted it to the iFSD setting based on its publicly released code[3]. We note that *Meta R-CNN* [53] shares the same formulation as [22], with the difference of reweighting the regional proposals rather than the whole images. However, without released code we cannot reproduce the Meta R-CNN. All methods are implemented on CentreNet/ResNet50 for both the backbone of the detector network and the code generator meta-networks.

Object detection on COCO We first evaluated *the incremental batch setting*. The results of all the methods are compared in Table 2. We have several observations: **(1)** The standard Fine-Tuning method is not only *ineffective* for

[2]In practice, a novel class code is 3×256 when accounting for class-specific width and height heatmaps. This is omitted during description of our method for simplicity.

[3]The main modification is that only novel classes are visited in meta-test time.

Figure 3: Novel class object detection on the COCO *val2017* set. Top: our method. Bottom: Fine-Tuning. The 10-shot iFSD setting.

Shot	Method	Novel Classes		Base Classes		All Classes	
		AP	AR	AP	AR	AP	AR
1	Fine-Tuning	0.0	0.0	1.1	1.8	0.8	1.4
	MAML [13]	0.1	0.5	N/A	N/A	N/A	N/A
	Feature-Reweight† [22]	0.1	0.3	2.5	4.3	1.9	3.3
	ONCE	**0.7**	**6.3**	**17.9**	**19.5**	**13.6**	**16.2**
5	Fine-Tuning	0.2	3.5	2.6	7.4	2.0	6.4
	MAML [13]	0.4	3.9	N/A	N/A	N/A	N/A
	Feature-Reweight† [22]	0.8	5.1	3.3	8.2	2.6	7.4
	ONCE	**1.0**	**7.4**	**17.9**	**19.5**	**13.7**	**16.4**
10	Fine-Tuning	0.6	4.2	2.8	8.0	2.3	7.0
	MAML [13]	0.8	4.9	N/A	N/A	N/A	N/A
	Feature-Reweight† [22]	**1.5**	**8.3**	3.7	8.9	3.1	8.7
	ONCE	1.2	7.6	**17.9**	**19.5**	**13.7**	**16.5**

Table 2: Incremental few-shot object detection performance on **COCO *val2017*** set. *Setting*: Incremental learning in batch of all 20 novel classes. '†': the code of [22] is adapted to use the same detection backbone (CentreNet) and setting for fair comparison.

learning from few-shot samples per novel class, but also suffers from catastrophic forgetting (massive base class performance drop), making it unsuited for iFSD. **(2)** As a representative meta-learning method[4], MAML improves slightly the few-shot detection capability over Fine-Tuning. However, without access to the support sets of base (old) classes in iFSD, it is incapable of performing object detection for base classes. After all, MAML is not designed for incremental learning. **(3)** Feature-Reweight, similarly to Fine-Tuning, suffers from catastrophic forgetting when used in the incremental setting. It is inferior to our method for most metrics, with a slight edge on novel class detection for the 10-shot experiment. This occurs, at the cost of intensive optimisation in testing time, which is not ideal in many practical scenarios. **(4)** ONCE achieves the best performance on most experiments for both novel and base classes simultaneously. The improvements over baselines are more significant with fewer shots. In particular, by class-specific detector learning ONCE keeps the performance on base classes unchanged, naturally solving the learning without forgetting challenge.[5] **(5)** While the absolute performance on novel classes is still low, this is a new and extremely challenging problem, for which ONCE provides a promising first solution without requiring test-time optimisation. Some qualitative results are shown in Fig. 3.

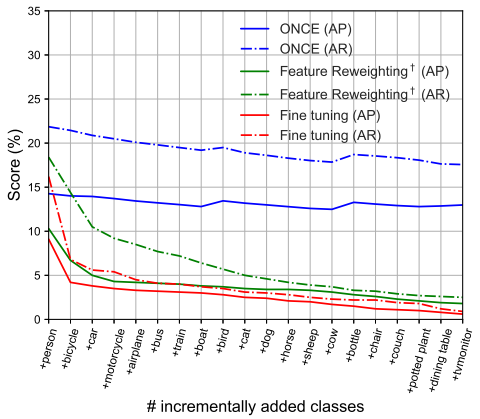

Figure 4: Incremental few-shot object detection performance on **COCO *val2017*** set. We plot accuracy for all-classes vs. the number of incrementally-added novel classes. Training setting: 10-shot per novel class. '†': the code of [22] is adapted to use the same detection backbone (CentreNet) and setting for fair comparison.

We also evaluated *the continuous incremental learning setting*: novel classes are added one at a time. We reported the all-classes accuracy. In this test, we excluded MAML since it is not designed for iFSD with no capability of detecting base class objects. The results in Fig. 4 show that the performance of ONCE changes little, while the performance of the competitors drops quickly due to increased forgetting as more classes are added. These results validate our ONCE's ability to add new classes on-the-fly.

[4]Note that there are many more recent FSL methods [6, 26, 7, 17, 47, 55, 50, 27, 39, 11, 16] that produce better performance on image classification tasks than MAML. However, they are designed for classification only and cannot be easily adapted for the object detection task here.

[5]The base class AP we get is lower than in the normal supervised setting. This is due to early stopping during base class training. Without it, we do reach the results reported in [56]. However, early stopping for training base class detector is important for iFSD as otherwise the features will be overfit to known classes and generalise less to novel classes.

Shot	Method	AP	AP_S	AP_M	AP_L	AR	AR_S	AR_M	AR_L
5	Fine-Tuning	0.1	0.1	0.8	0.3	1.9	0.7	2.9	7.6
	MAML [34]	0.6	0.2	1.1	1.3	2.2	1.2	4.0	10.6
	ONCE	**2.4**	**1.2**	**2.4**	**3.4**	**12.2**	**5.9**	**16.4**	**33.6**
10	Fine-Tuning	0.3	0.1	0.8	1.0	2.8	0.9	3.3	10.2
	MAML [34]	1.0	0.4	1.7	2.1	3.2	7.9	5.1	12.2
	ONCE	**2.6**	**5.7**	**2.2**	**4.9**	**11.6**	**8.3**	**19.4**	**32.6**

Table 3: Incremental few-shot object detection *transfer* performance on **VOC2007 test** set. Training data: COCO. Setting: *Incremental batch*. We only reported the *novel class* performance as there are no base class images in VOC.

Object detection transfer from COCO to VOC. We evaluated iFSD in a cross-dataset setting from COCO to VOC. We considered *the incremental batch setting*, and reported *the novel class performance* since there are no base class images in VOC. The results in Table 3 show that: **(1)** The same relative results are obtained as in Table 2, confirming that the performance advantages of our model transfers to a test domain different from the training one. **(2)** Higher performance is obtained on VOC by all methods compared to COCO. This makes sense as COCO images are more challenging and unconstrained than those in VOC.

4.3. Few-Shot Fashion Landmark Detection

Experimental setup. Besides object detection, we further evaluated our method for fashion landmark detection on the DeepFashion2 benchmark [15]. This dataset has 801K clothing items from 13 categories. A single clothing category contains 8∼19 landmark classes, giving a total of 294 classes. This forms a two-level hierarchical semantic structure, which is not presented in the COCO/VOC dataset. Each image, captured either in commercial shopping stores or in-the-wild consumer scenarios, presents one or multiple clothing items.

On top of the *original* train/test data split, we developed an iFSD setting. Specifically, we split the 294 landmark classes into three sets: 153 for training (8 clothing categories), 95 for validation (3 clothing categories), and 46 for testing (2 clothing categories). This split is category-disjoint across train/val/test sets for testing model generalisation over clothing categories. The most sparse clothing categories are assigned to the test set.

In each episode of iFSD training, we randomly sampled 1 task each with k-shot annotated landmarks per class and a total of 5 landmark classes, with $k \in \{1, 5, 10\}$. We used the val set for model selection, *i.e.* selecting the final model based on the validation accuracy. To avoid overfitting to the training clothing categories, we randomly sampled 5 out of all the available (8∼19) landmark classes in each episode. This is made possible by the class-specific modelling nature of ONCE, while [22, 53] cannot do this. In meta-testing, we randomly sampled a support set from the *original* training set of novel landmark classes (part of the iFSD test set), including $k \in \{1, 5, 10\}$ bounding boxes annotated for each novel landmark class, and used it for model learning. We tested the model performance on the *original* testing set of novel landmark classes (part of the iFSD test set). We repeated the test process 100 times and report the average.

Figure 5: Fashion landmark detection by ONCE. Column **1**: a support sample with five randomly selected landmark ground-truth. Column **2**: a test image with the predicted landmarks. Columns **3-7**: predicted heatmaps.

Shot	Method	AP	AP_{50}	AR	AR_{50}
1	Fine-Tuning	2.8	9.0	6.7	16.5
	ONCE	**4.6**	**26.5**	**11.8**	**49.9**
5	Fine-Tuning	17.0	35.9	25.1	42.1
	ONCE	**29.5**	**77.5**	**42.1**	**87.4**
10	Fine-Tuning	17.1	37.1	24.6	43.0
	ONCE	**32.2**	**79.5**	**44.3**	**88.3**

Table 4: Incremental few-shot landmark detection performance on the novel classes of **DeepFashion2**. Setting: *Incremental batch*.

Competitors. In this controlled test, we compared ONCE with the *Fine-Tuning* baseline. Other methods for few-shot object detection (*i.e.*, [22, 53]) are not trivially adaptable for this task due to its hierarchical semantic structure. Indeed, the inter-class independence obtained by adopting CentreNet as detection backbone, and our proposed few-shot detection framework allows for such general applicability of ONCE.

Evaluation results. We evaluated *the incremental batch setting* and reported *the novel class performance*. The results in Table 4 show that ONCE consistently and significantly outperforms *Fine-Tuning*. This suggests that our model is better at transferring the landmark appearance information from base classes to novel classes, even when only as little as *one shot* training example is available for learning. due to not having to perform iterative optimisation during meta-test. Note that the absolute accuracy achieved on this task is much higher than the object detection tasks (Table 4 vs. Tables 2&3). This is due to the fact that all classes are clothing landmarks so there is much more transferable knowledge from base to novel classes. An example of one-shot landmark detection by ONCE is shown in Fig. 5. It can be seen that the model can detect landmarks accurately after seeing them only once.

5. Conclusion

We have investigated the challenging yet practical incremental few-shot object detection problem. Our proposed ONCE provides a promising initial solution to this problem. Critically, ONCE is capable of incrementally registering novel classes with few examples in a feed-forward manner, without revisiting the base class training data. It yields superior performance over a number of alternatives on both object and landmark detection tasks in the incremental few-shot setting. Our work is evidence of the need for further efforts towards solving more effectively the iFSD problem.

References

[1] Saif Alabachi, Gita Sukthankar, and Rahul Sukthankar. Customizing object detectors for indoor robots. In *ICRA*, 2019. 1

[2] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, 2018. 3

[3] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 92(2):115–147, 1987. 1

[4] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 3

[5] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. 2

[6] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *CVPR*, 2019. 2, 7

[7] Wen-Hsuan Chu, Yu-Jhe Li, Jing-Cheng Chang, and Yu-Chiang Frank Wang. Spot and learn: A maximum-entropy patch sampler for few-shot image classification. In *CVPR*, 2019. 2, 7

[8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*, 2016. 2

[9] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*, 2019. 1

[10] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. In *CVPR*, 2017. 3

[11] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *ICCV*, 2019. 2, 7

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 2, 6

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2, 6, 7

[14] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 1, 2

[15] Yuying Ge, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *CVPR*, 2019. 2, 8

[16] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Perez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *ICCV*, 2019. 2, 7

[17] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *CVPR*, 2019. 2, 7

[18] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 2

[19] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 37(9):1904–1916, 2015. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5

[22] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *ICCV*, 2019. 1, 2, 3, 6, 7, 8

[23] 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 *CVPR*, 2019. 2

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[25] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018. 2, 3

[26] Aoxue Li, Tiange Luo, Zhiwu Lu, Tao Xiang, and Liwei Wang. Large-scale few-shot learning: Knowledge transfer with class hierarchy. In *CVPR*, 2019. 2, 7

[27] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *ICCV*, 2019. 2, 7

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1, 2, 3

[29] 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 *ECCV*. Springer, 2014. 2, 6

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 1, 2, 3

[31] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[32] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *CORL*, 2017. 1

[33] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 3

[34] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv*, 2018. 2, 6, 8

[35] Andreas Opelt, Axel Pinz, and Andrew Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *CVPR*, 2006. 3

[36] Steven Pinker. *How the mind works*. Penguin UK, 2003. 1

[37] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In *ACCV*, 2018. 3

[38] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2016. 2

[39] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *ICCV*, 2019. 2, 7

[40] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 1

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2, 3

[42] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018. 1, 2, 3

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 2, 3

[44] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. In *ICML*, 2017. 2, 3

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[46] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 2

[47] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, 2019. 2, 7

[48] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2, 4

[49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1

[50] Pavel Tokmakov, Yu-Xiong Wang, and Martial Hebert. Learning compositional representations for few-shot recognition. In *ICCV*, 2019. 2, 7

[51] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016. 2, 3, 5

[52] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 3, 5

[53] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *ICCV*, 2019. 1, 2, 3, 6, 8

[54] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *NeurIPS*, 2017. 6

[55] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. In *CVPR*, 2019. 2, 7

[56] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv*, 2019. 2, 3, 4, 7

[57] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019. 2, 3

[58] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Zero shot detection. *IEEE TCSVT*, 2019. 3