Any-Shot Object Detection

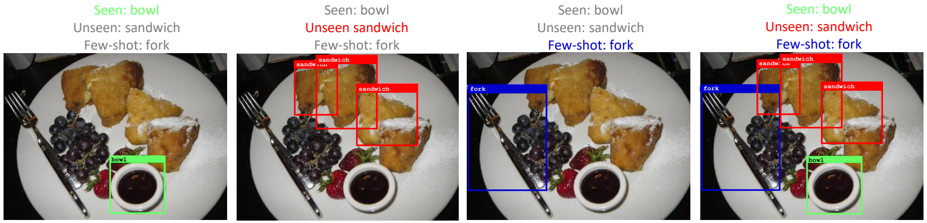
Shafin Rahman[1,2,3][0000−0001−7169−0318], Salman Khan[4,2][0000−0002−9502−1749],
Nick Barnes[2][0000−0002−9343−9535], and Fahad Shahbaz
Khan[4][0000−0002−4263−3143]

[1] North South University, Dhaka, Bangladesh
[2] Australian National University, Canberra, ACT 2601, Australia
[3] Data61, CSIRO, Canberra, ACT 2601, Australia
[4] MBZ University of Artificial Intelligence, Abu Dhabi, UAE
shafin.rahman@northsouth.edu,nick.barnes@anu.edu.au,
{salman.khan,fahad.khan}@mbzuai.ac.ae

Abstract. Previous work on novel object detection considers zero or few-shot settings where none or few examples of each category are available for training. In real world scenarios, it is less practical to expect that '*all*' the novel classes are either unseen or have few-examples. Here, we propose a more realistic setting termed '*Any-shot detection*', where totally unseen and few-shot categories can simultaneously co-occur during inference. Any-shot detection offers unique challenges compared to conventional novel object detection such as, a high imbalance between unseen, few-shot and seen object classes, susceptibility to forget base-training while learning novel classes and distinguishing novel classes from the background. To address these challenges, we propose a unified any-shot detection model, that can concurrently learn to detect both zero-shot and few-shot object classes. Our core idea is to use class semantics as prototypes for object detection, a formulation that naturally minimizes knowledge forgetting and mitigates the class-imbalance in the label space. Besides, we propose a rebalanced loss function that emphasizes difficult few-shot cases but avoids overfitting on the novel classes to allow detection of totally unseen classes. Without bells and whistles, our framework can also be used solely for Zero-shot object detection and Few-shot object detection tasks. We report extensive experiments on Pascal VOC and MS-COCO datasets where our approach is shown to provide significant improvements.

1 Introduction

Traditional object detectors are designed to detect the categories on which they were originally trained. In several applications, such as self-driving cars, it is important to extend the base object detector with novel categories that were never seen before. The current 'novel' object detection models proposed in the literature target either of the two distinct settings, *Zero-shot detection* (ZSD) and *Few-shot detection* (FSD). In the former setting, it is assumed that totally unseen objects appear during inference and a model must learn to adapt for novel



(a) Traditional Object Detection (b) Zero-shot Object Detection (c) Few-shot object Detection (d) Any-shot Object Detection

Fig. 1. (a) A traditional object detection method only detects seen objects. In the same vein, zero and few-shot object detection methods can detect (b) unseen or (c) few-shot objects. (d) Our proposed Any-shot detection method can *simultaneously* detect seen, unseen and few-shot objects.

categories using only their class description (semantics). In the latter setting, a small and fixed-number of novel class samples are available for model adaptation. However, in a practical scenario, restricting the novel classes to be always unseen (with zero visual examples) or always with few-shot examples can limit the generality of the model.

In a real-world scenario, both unseen and few-shot classes can be simultaneously of interest. Moreover, we may encounter a few examples of a novel class that was previously supposed to be unseen. In such a case, an adaptive model must leverage from new information to improve its performance in an online fashion. To address these requirements, we introduce a new '*Any-shot Detection*' (ASD) protocol where a novel class can have zero or a few training examples. Since, the existing object detection models can either work for zero-shot or few-shot settings, we develop a unified framework to address the ASD problem (see Fig. 1). Remarkably, since the ASD task sits at the continuum between ZSD and FSD, our model can be directly applied to both these problems as well.

The ASD task poses new challenges for novel object detection. First, a high data imbalance between unseen, few-shot and seen classes can lead to a biased detection model. Additionally, the fine-tuning performed on few-shot examples can lead to forgetting previously acquired knowledge, thereby deteriorating model performance on seen and unseen classes. To overcome these challenges, we propose to learn a mapping from the visual space to the semantic space where class descriptors serve as fixed prototypes. The semantic prototypes in our approach encode inherent class relationships (thus enabling knowledge transfer from the seen to the unseen), helps us disentangle totally unseen concepts from the background and can be automatically updated to align well with the visual information. Besides, we introduce a novel rebalancing loss function for the fine-tuning stage that functions on few-shot examples. This loss serves two objectives, i.e., to focus on the errors made for few-shot classes and at the same time avoid overfitting them so that it remains generalizable to totally unseen categories.

Our main contributions are:

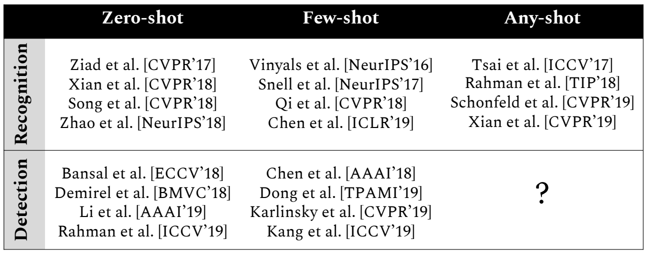
- A unified framework that can accommodate ZSD, FSD, ASD and their generalized settings.
- Learning with semantic class-prototypes that are well aligned with visual information and help minimize forgetting old concepts.
- An end-to-end solution with a novel loss function that rebalances errors to penalize difficult cases yet remains generalizable to unseen objects.
- Extensive experiments with new ASD setup, as well as comparisons with traditional FSD and ZSD frameworks demonstrating significant improvements.

2 Related Work

N-shot recognition: There exist three types of methods for n-shot recognition. The *first* body of work targets only zero-shot recognition (ZSR) [1–3]. They perform training with seen data and test on unseen (or unseen+seen) data. To relate seen and unseen classes, they use semantic embeddings e.g., attributes [4] or word vectors [5, 6]. The ZSR task has been investigated under popular themes such as transduction [7, 8], domain adaptation [9, 8], adversarial learning [10] and class-attribute association [11, 12]. The *second* body of work targets only few-shot recognition (FSR) task [13]. This task leverages few labeled examples to classify novel classes. Most popular methods to solve FSR are based on meta-learning where approaches perform metric learning to measure the similarity between input and novel classes [14–16], adapt the meta-learner by calculating gradient updates for novel classes [16] or predict the classifier weights for novel classes [17]. The *third* body of work addresses both zero and few-shot learning together [18–21]. These approaches are the extended version of ZSR or FSR methods that consider word vectors to accommodate both problems within a single framework. Our current work belongs to the third category, but instead of a recognition task, we focus on the detection problem, that is more challenging.

Zero-shot detection: Different from traditional object detection (where only seen objects are detected), ZSD aims to detect both seen and/or unseen objects. Pioneering works on ZSD attempt to extend established object detection methods to enable ZSD. For example, [22], [23, 24] and [25, 26] employ pre-computed object proposals [27], YOLOv2 [28] and Faster-RCNN [29] based methods for ZSD, respectively. Recent methods for ZSD employ specialized polarity loss [30], explore transductive settings [31] and use raw textual description instead of only class-names [32]. All the above methods focus on only ZSD and Generalized ZSD tasks but cannot accommodate FSD scenario when new instances of unseen images become available. In this paper, we propose a method that can perform ZSD, FSD, and ASD tasks seamlessly, including their generalized cases.

Few-shot detection: FSD methods attempt to detect novel classes for which only a few instances (1-10) are available during the inference stage [33, 34]. Among the early attempts of FSD, [35] proposed a regularizer that works on standard object detection models to detect novel classes. Later, [36] proposed a distant metric learning-based approach that learned representative vectors to facilitate FSD. The drawback of the above FSD methods is that they cannot



	Zero-shot	Few-shot	Any-shot
Recognition	Ziad et al. [CVPR'17] Xian et al. [CVPR'18] Song et al. [CVPR'18] Zhao et al. [NeurIPS'18]	Vinyals et al. [NeurIPS'16] Snell et al. [NeurIPS'17] Qi et al. [CVPR'18] Chen et al. [ICLR'19]	Tsai et al. [ICCV'17] Rahman et al. [TIP'18] Schonfeld et al. [CVPR'19] Xian et al. [CVPR'19]
Detection	Bansal et al. [ECCV'18] Demirel et al. [BMVC'18] Li et al. [AAAI'19] Rahman et al. [ICCV'19]	Chen et al. [AAAI'18] Dong et al. [TPAMI'19] Karlinsky et al. [CVPR'19] Kang et al. [ICCV'19]	?

Fig. 2. The any-shot object detection (ASD) problem has not been addressed in the literature before. Importantly, an ASD system can automatically perform both ZSD and FSD, which no prior approach can simultaneously offer.

handle seen/base classes during test time. Recently, [37] proposed to train a base network with seen data and then fine-tune it by meta-network learning that predicts scores for both seen and novel classes by re-weighing the base network features. This approach can perform generalized FSD but cannot accommodate ZSD or ASD scenario. In this paper, we address the mentioned gap in the literature (see Fig. 2).

3 Novel Object Detection

Novel object detection refers to enhancing the ability of a traditional object detector model to detect a new set of classes that were not present during training. We propose a unified Any-shot Detection (ASD) setting[5] where novel classes include both few-shot and unseen (zero-shot) classes. This is in contrast to the existing works on novel object detection that treat zero and few-shot detection in an isolated manner. In the absence of the unseen class and few-shot classes, our problem becomes identical to a conventional FSD and ZSD task, respectively. In this way, our proposed ASD settings unifies ZSD and FSD in a *single* framework.

3.1 Problem Formulation

Assume a total of C object classes are present in a given test set that need to be detected. Out of these, $S(>0)$ seen classes have many, $Q(\geq 0)$ few-shot classes have few and $U(\geq 0)$ unseen classes has no examples available in the training dataset and $C = S{+}Q{+}U$. Here, $T = Q{+}U$ represents the total number of novel classes that become available during the inference stage. For each class, a semantic description is available as a d-dimensional embedding vector. The semantic embeddings of all classes are denoted by $\boldsymbol{W} = [\boldsymbol{W}_s, \boldsymbol{W}_f, \boldsymbol{W}_u] \in \mathbb{R}^{d\times C}$, where $\boldsymbol{W}_s \in \mathbb{R}^{d\times S}$, $\boldsymbol{W}_f \in \mathbb{R}^{d\times Q}$ and $\boldsymbol{W}_u \in \mathbb{R}^{d\times U}$ are semantic vectors for seen, few-shot and unseen classes, respectively.

[5] Our any-shot detection setting is different from [19], which considers zero and few-shot problems *separately* for a simpler *classification* task.

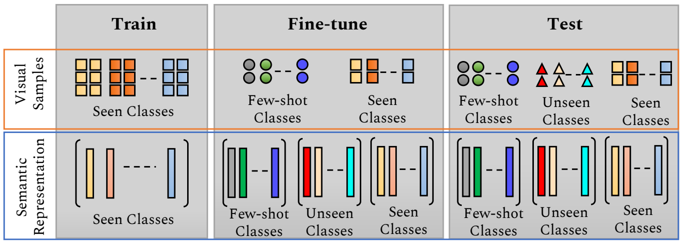


Fig. 3. An overview of Any-shot Detection setting.

The base training set ($\mathcal{D}_{tr}$) contains N_{tr} images with instances from S seen classes. Each training image $\boldsymbol{I}$ is provided with a set of bounding boxes, where each box $\mathbf{b}_{tr}$ is provided with a seen label $\mathbf{y}_{tr} \in \{0,1\}^S$. Similarly, when $Q{>}0$, we have a fine-tuning dataset ($\mathcal{D}_{ft}$) with N_{ft} images containing instances from both seen and few-shot classes for which bounding box $\mathbf{b}_{ft}$ and class label $\mathbf{y}_{ft} \in \{0,1\}^C$ annotations are present. During inference, we have a testing dataset $\mathcal{D}_{ts}$ with N_{ts} images where each image can contain any number of seen and novel class objects (see Fig. 3).

Our task is to perform any-shot detection, defined as:

Definition 1. Any-shot detection: *When $Q{>}0$ and $U{>}0$, predict object labels and associated bounding boxes for T novel classes, that include both zero and few-shot classes.*

In comparison, the traditional zero and few-shot problems can be defined as: (a) **Few-shot detection:** When $Q{>}0$ but $U{=}0$, predict object labels and associated bounding boxes for all Q classes. (b) **Zero-shot detection:** When $Q{=}0$ but $U{>}0$, predict object labels and associated boxes for all U classes. Note, if $Q{=}U{=}0$ then the problem becomes equivalent to a traditional detection task.

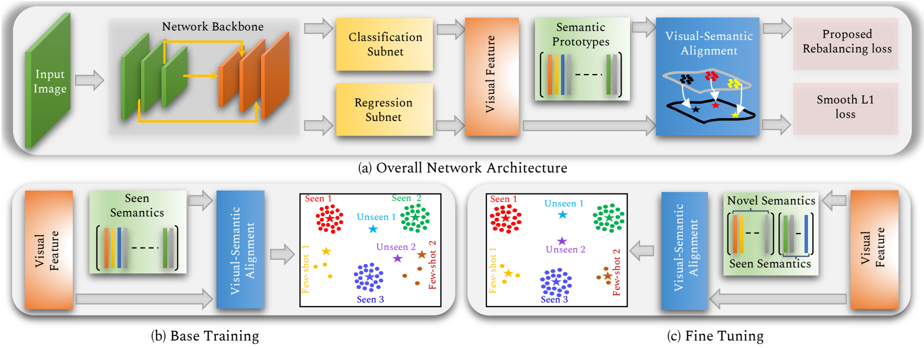
We also study the generalized ASD problem defines as:

Definition 2. Generalized any-shot detection: *When $\{S,Q,U\}{\subset}\mathbb{Z}^+$, predict object labels and box locations for all C classes, that include seen, zero and few-shot classes.*

In the same vein, generalized zero and few-shot detection problems aim to detect seen classes in addition to novel ones. Next, we describe our approach for ASD and GASD.

3.2 Method

Our goal is to design a model that can *simultaneously* perform zero, few, and many-shot detection. This model is trained with seen classes, and is quickly adapted during inference for zero-shot and few-shot classes. This problem set-up has the following major challenges: *(a) Adaptability:* A trained model must be flexible enough to incorporate new classes (with no or few examples) on the go, *(b) Learning without Forgetting:* While the model is adapted for new classes, it must not forget the previous knowledge acquired on seen classes, and *(c) Class*



(a) Overall Network Architecture

(b) Base Training

(c) Fine Tuning

Fig. 4. (a) Network architecture. The visual-semantic alignment is performed using (b) seen semantics during base training and with (c) seen and novel semantics during fine-tuning. The visual features from the classification and regression units are separately used for visual-semantic alignment and subsequent loss calculation.

Imbalance: The classes representations are highly imbalanced: some with many instances, some with none and others with only a few. Therefore, the learning regime must be robust against the inherent imbalance in the ASD setting.

At a high-level, our proposed approach has two main components that address the above mentioned problems. First, we consider the semantic class prototypes to serve as anchors in the prediction space, thereby providing the flexibility to extend to any number of novel classes without the need to retrain network parameters. We show that such a representation also helps in avoiding catastrophic forgetting that is likely to occur otherwise. Furthermore, we propose a new loss formulation to address the class imbalance problem, that specifically focuses on difficult cases and minimizes model's bias against rare classes. We elaborate the novel aspects of our approach below.

Learning without Forgetting. Traditional object detection models are static approaches that cannot dynamically adapt to novel classes. The flexibility to introduce novel object classes, after the base model training, requires special consideration, e.g., by posing it as an incremental learning problem [38–40]. In such cases, since classifier weights for novel categories are learned from scratch, the knowledge distillation concept [41] is applied to avoid forgetting the old learning. Such a strategy is not useful in our case because unlike previous approaches, we do not have access to many examples of the new task and a subset of novel classes has no training examples available.

To allow adding novel classes without forgetting old concepts, our approach seeks to disentangle the feature learning and classification stages. Precisely, we develop a training mechanism in which adding new classes does not require re-training base-network's parameters. This is achieved by defining the output decision space in the form of semantic class prototypes. These semantic class representatives are obtained in an unsupervised manner using a large text corpus,

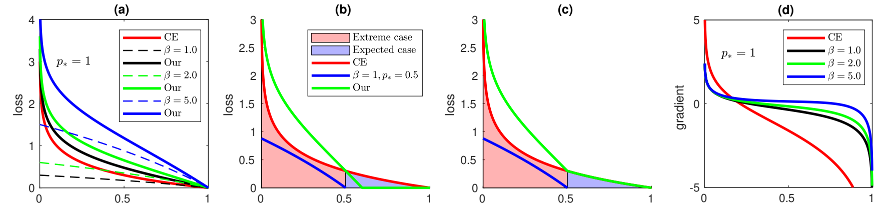


Fig. 5. Loss visualization. The colored dotted lines represent β-controlled penalty function, $h(.)$ and the solid lines represent loss functions. **(a)** The red line represents cross entropy (CE) which is compared to our loss with $\beta = 1, 2$ and 5 shown as black, green and blue lines, respectively. **(b)** Our loss (green line) with fixed p_*=0.5. Here, our loss can be less than CE (red line) for the expected case. **(c)** Our loss curve (green line) with dynamic p_*. Here, our loss calculates the same value as CE (red line) for the expected case. The red shaded region represents extreme case since $p{<}p_*$ and blue shaded region represents expected or moderate case $p{\geq}p_*$. (d) Derivatives of CE and our loss for different β. See the supplementary material for the gradient analysis.

such as the Wikipedia, and encode class-specific attributes as well as the inter-class relationships [5, 6].

During base-model training, the model learns to map visual features to semantic space. At the same time, the semantic space is well-aligned with the visual concepts using a learnable projection. Note that only seen class semantics are used during the training stage. Once the base-model is trained, novel classes (both zero and few-shot) can be accommodated at inference time taking advantage of the semantic class descriptions of the new classes. For novel classes whose new few-examples are available during inference, we fine-tune the model to adapt semantic space, but keeping the base-model's architecture unchanged. Essentially, this means that adding new classes does not demand any changes to the architecture of the base-network. Still, the model is capable of generating predictions for novel classes since it has learned to relate visual information with semantic space during training (see Fig. 4).

Formally, a training image $\boldsymbol{X}$ is fed to a deep network $f(\cdot)$, that produces a visual feature vector $f(\boldsymbol{X}) \in \mathbb{R}^n$ corresponding to a particular box location. In a parallel stream, seen word vectors $\boldsymbol{W}_s$, are passed through a light-weight subnet denoted by $g(\cdot)$, producing the transformed word vectors $g(\boldsymbol{W}_s)$. Then, we connect the visual and semantic streams via a trainable layer, $\boldsymbol{U} \in \mathbb{R}^{n \times d}$. Finally, a set of seen scores $\boldsymbol{p}_s$ is obtained by applying a sigmoid activation function (σ). The overall computation is given by,

$$\boldsymbol{p}_s = \sigma\big(f(\boldsymbol{X})^T \boldsymbol{U} g(\boldsymbol{W}_s)\big). \tag{1}$$

The mapping layer $\boldsymbol{U}$ can be understood as the bridge between semantic and visual domains. Given the visual and semantic mappings, $f(\boldsymbol{X})$ and $g(\boldsymbol{W}_s)$ respectively, $\boldsymbol{U}$ seeks to maximize the alignment between the visual feature and its corresponding semantic class such that the prediction $\boldsymbol{p}_s$ is maximized. In a way, $\boldsymbol{p}_s$ is the alignment compatibility scores where a higher score means more compatibility between feature and semantics. The function $f(\cdot)$ can be implemented with a convolutional network backbone (e.g. ResNet [42]) and $g(\cdot)$ can be implemented as a fixed or trainable layer. In the fixed case, fixed word vectors can be directly used for alignment i.e., $g(\boldsymbol{W}_s) = \boldsymbol{W}_s$. In the trainable case, $\boldsymbol{W}_s$ can be updated using a trainable metric $\boldsymbol{M} \in \mathbb{R}^{d \times v}$ and a word vocabulary $\boldsymbol{D} \in \mathbb{R}^{v \times d}$, resulting in $g(\boldsymbol{W}_s) = \delta(\boldsymbol{W}_s \boldsymbol{M} \boldsymbol{D})$ where, v is the size of the word vocabulary and δ is a tanh activation. In our experiments, we find a trainable $g(\boldsymbol{W}_s)$ achieves better performance.

We propose a two step training procedure. The first step involves training the base model, where Eq. 1 is trained with only images and semantics of seen classes. In the second step of fine-tuning, when novel class information becomes available, we replace $\boldsymbol{W}_s$ by $\boldsymbol{W}$ and train it with few-shot examples. Eq. 1 then becomes $\boldsymbol{p} = \sigma\big(f(\boldsymbol{X})^T \boldsymbol{U} g(\boldsymbol{W})\big)$, where, $\boldsymbol{p}$ contains scores of both seen and novel classes. In this manner, model is quickly adapted for novel classes.

Notably, although our network can predict scores for all classes, no new tunable weights are added. In both steps, our network tries to align the feature with its corresponding semantics. From the network's perspective it does not matter how many classes are present. It only cares how compatible a feature is with the corresponding class semantics. This is why our network does not forget the seen classes as these semantic prototypes serve as an anchor to retain previous knowledge. Adding new classes is therefore not a new task for the network. During fine-tuning, the model still performs the same old task of aligning feature and semantics.

Learning with Imbalance Data. After base training on seen classes, novel classes become available during inference stage i.e., few-shot images and the word vectors of both zero and few-shot classes. Here, the few-shot image may contain seen instances as well. In this way, the fine-tuning stage contains an imbalanced data distribution and the model must minimize bias towards the already seen classes. To this end, we propose a rebalancing loss for the fine-tuning stage.

Suppose, $p \in \boldsymbol{p}$ is the alignment score for a visual feature and the corresponding class semantics. As the fine-tuning is done on rare data, we need to penalize the cross-entropy (CE) loss based on the quality of alignment. If the network makes a mistake, we increase the penalty and if the network is already performing well, we employ a low or negative penalty. Suppose, the penalty $h(\cdot)$ is a function of p and p_* where p_* determines the penalization level, then,

$$L(p) = -\log p + \beta\, h(p, p_*), \tag{2}$$

where, β is a hyper-parameter. Here, $h(p, p_*)$ is given by,

$$h(p, p_*) = \log(1 + p_* - p), \tag{3}$$

where $(p_* - p)$ represents the violation of the expected alignment that controls the margin of the penalty function. We explore two alternatives for selecting p_*, a fixed value in range $0 < p_* \leq 1$ and a dynamically adjusted value based on

$p_* = \max_{i \in C} p_i$. We can get the following scenarios based on the choice of p_* and positive alignment quality p:

- *Expected case, $p{>}p_*$:* Negative penalty to calculate lower loss compared to CE (lower bounded by 0).
- *Moderate case, $p{=}p_*$:* Zero penalty and the calculated loss is equal to regular CE.
- *Extreme case, $p{<}p_*$:* High penalty in comparison to regular CE loss.

Plugging the penalty definition from Eq. 3 to Eq. 2 and enforcing positive loss values $L(p) \in \mathbb{R}^+$, we obtain,

$$L(p) = \max\left[0, -\log \frac{p}{(1+p_*-p)^{\beta}}\right]. \tag{4}$$

After adding an α-balanced modulating factor from focal loss [43], we have,

$$L(p) = \max\Big[0, -\alpha_t(1-p_t)^{\gamma} \log p_t\Big], \text{where, } p_t = \begin{cases} \frac{p}{(1+p_*-p)^{\beta}}, & \text{if } y = 1 \\ 1-p, & \text{otherwise.} \end{cases}$$

Here, β is a parameter that focuses on hard cases and y is the corresponding value from the one-hot encoded ground-truth vector. With β=0, Eq. 5 becomes equivalent to focal loss and with β=0, γ=0, α=1, Eq. 5 becomes CE loss.

Since several objects can co-occur in the same scene, the fine-tuning data can have seen instances. To emphasise rare classes more than the seen ones, we apply our rebalancing loss only on the novel class examples. For a seen anchor, only the focal loss is calculated. Thus, the final loss is,

$$L = \lambda L(s) + (1-\lambda) L(n). \tag{5}$$

For the case of $L(s)$ and $L(n)$, β=0 and β>0 respectively. $L(s)$ and $L(n)$ represent the compatibility scores of seen and novel (few-shot and unseen) classes i.e., $s \in \{1, 2, .., S\}$ and $n \in \{1, 2, .., T\}$.

During inference when a test image is presented, a simple forward pass provides compatibility scores of seen, few-shot and unseen classes for each bounding box. If the score is higher than a threshold, we consider it a correct detection.

Analysis: Based on the quality of alignment, our proposed loss penalizes positive anchors. This scenario helps in the class imbalance problem. Especially, in the extreme case, when the network fails to detect a positive few-shot anchor, we highly penalize our network predictions. It gives extra supervision to the network that it must not make errors on the few-shot classes. In contrast, for the expected and moderate cases, we reduce the loss which avoids the network becoming too confident on few-shot examples. Since, unseen objects are more related to the seen objects as compared to background, a low penalty on confident cases implicitly promotes discovering unseen classes. In effect, this leads to low overfitting on the few-shot classes that helps in achieving good performance on totally unseen classes.

Table 1. ASD results on MSCOCO.

#-Shot	Method	ASD			GASD			
		unseen	FS	HM	seen	unseen	FS	HM
1	Baseline-I	3.74	1.60	2.25	**54.11**	2.04	0.73	1.60
	Baseline-II	8.57	21.39	12.23	51.89	3.79	9.62	7.74
	Ours	**16.57**	**23.50**	**19.44**	51.70	**10.75**	**11.83**	**15.23**
5	Baseline-I	4.16	2.69	3.27	**54.15**	2.35	1.20	2.35
	Baseline-II	8.69	26.19	13.05	51.67	4.85	18.20	10.70
	Ours	**18.22**	**26.31**	**21.53**	51.18	**12.70**	**18.34**	**19.63**
10	Baseline-I	3.45	2.95	3.18	**54.25**	1.89	1.56	2.53
	Baseline-II	7.26	31.14	11.78	51.00	4.12	25.00	9.91
	Ours	**13.21**	**33.52**	**18.95**	51.18	**9.71**	**26.96**	**18.79**

Table 2. Ablation study with 5-shot case.

Method	ASD			GASD			
	unseen	FS	HM	seen	unseen	FS	HM
Baseline-I	4.16	2.69	3.27	**54.15**	2.35	1.20	2.35
Baseline-II	8.69	26.19	13.05	51.67	4.85	18.20	10.70
Ours with FL	13.69	23.81	16.61	51.20	9.21	16.34	15.85
Ours with AL	7.03	24.17	10.89	50.74	5.94	17.46	12.23
Ours ($p_* = 0.3$)	17.20	26.85	20.97	51.48	11.84	19.21	19.24
Ours ($p_* = 0.5$)	15.24	24.02	18.65	50.65	10.38	17.06	17.17
Ours ($p_* = 1.0$)	16.17	27.17	20.27	50.58	11.29	19.83	18.90
Ours*	16.60	24.05	19.64	51.32	11.09	16.71	17.70
Ours	**18.22**	**26.31**	**21.53**	51.18	**12.70**	**18.34**	**19.63**

Loss shape analysis: In Fig. 5, we visualize the loss. Fig. 5 (a) shows how the shape of binary cross entropy (CE) changes with different values of β. β controls the penalty $h(.)$ which modifies CE loss. For a fixed p_*=1, increasing β calculates a higher penalty for a wrong prediction. For a fixed margin penalty p_*=0.5 in Fig. 5 (b), a network can predict a lower, equal and higher score than p_*. Correspondingly, it enables the network to calculate a less, equal and higher penalty for expected, moderate and extreme cases, respectively. In contrast, for the dynamic margin penalty case Fig. 5 (c), the predicted score can be at best $p_* = \max_{i \in C} p_i$. Therefore, the extreme case works similarly to the fixed p_* scenario but for the other cases, the network calculates a loss equal to CE/Focal loss. The dynamic p_* estimates the quality of predicted scores based on the current anchor specific situation. E.g., for a given anchor, a small predicted score (e.g., 0.1) for the ground-truth class can be considered as good prediction if all other predictions are < 0.1. It helps to make a good balance between seen, few-shot and unseen predictions because the loss does not unnecessarily tries to maximize the ground-truth class score and thus avoids over-fitting.

3.3 Implementation Details

We implement our framework with a modified version of the RetinaNet architecture proposed in [30] (see Fig. 4(a)). It incorporates the word vectors at the penultimate layers of classification and regression subnets. While performing the base training with focal loss at the first step, we follow the recommended process in [30], where only seen word vectors are used in word processing network, $g(.)$ (see Fig. 4(b)). During the fine-tuning step, we update the base model with newly available data and our proposed loss function. As shown in Fig. 4(c), fine-tuning uses both seen and novel word vectors inside $g(.)$. Note that, in addition to novel class data, the small dataset used for fine-tuning includes some seen instances as well. We train our model for 10 epochs during the fine-tuning stage. After the fine-tuning is finished, our framework can detect seen, few-shot, and unseen classes simultaneously. We use the Adam optimizer for each training stage.

4 Experiments

Datasets: We evaluate our work on the MSCOCO-2014 [44] and PASCAL VOC 2007/12 datasets. For the MSCOCO experiment, we adopt the 65/15

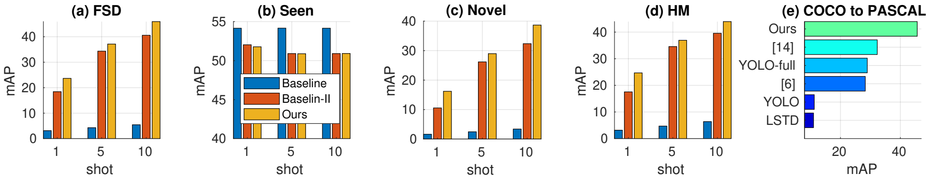


Fig. 6. FSD performance. (a) 1-, 5- and 10-shot detection mAP, (b), (c) and (d) seen, unseen and harmonic mean (HM) of GFSD. (e) 10-shot detection mAP for MSCOCO to PASCAL experiment.

seen/unseen split setting used in [30, 31]. In both ZSD and FSD experiments, we consider unseen classes as the novel ones. However, in ASD experiments, we further split 15 novel classes into 8 few-shot and 7 unseen classes. We use the same 62,300 images during training where no single instance of novel classes is present. For testing ASD, we use 10,098 images where each image contains at least one novel object. However for GASD testing, we use the entire validation set of 40,504 images. We randomly choose additional images with a few (1/5/10) annotated bounding boxes for each novel category while performing FSD/ASD on MSCOCO. These images may contain some seen objects as well. For the PASCAL VOC 2007/12 experiment, we adopt three 15/5 seen/novel split settings from [37]. As recommended, we use train+val sets from PASCAL VOC 2007 and 2012 as training data and test-set from PASCAL VOC 2007 for evaluation. For fine-tuning, we use the images provided by Kang *et al.* [37] as few-shot data. For both MSCOCO and PASCAL VOC classes and vocabulary texts, we use 300-dimensional and ℓ_2 normalized word2vec vectors [5]. We have used same set of 4717 vocabulary atoms as used in [30] which are originally taken from [45].

Evaluation criteria: For FSD and ASD, we evaluate our method with mean average precision (mAP). To report GFSD and GASD, we calculate the harmonic mean of the individual seen and novel class mAP. For ZSD, we report also recall@100 (RE) results as recommended in [22].

Validation experiment: α, β, γ and λ are the hyper-parameter of our model. Among them, α and γ are specific to focal loss. Thus, we fix the recommend value $\alpha = 0.25$ and $\gamma = 2$ following [43]. We validate β and λ by creating a validation dataset based on splitting 65 seen classes into 55 seen and 10 novel classes. From the validation experiment, we select $\beta = 5$ and $\lambda = 0.1$. Detailed validation results are presented in the supplementary material.

Baseline methods: Here, we introduce our baseline methods.
• *Baseline-I:* A RetinaNet architecture where fixed semantics are used in semantic processing pipeline i.e. $g(\boldsymbol{W}_s) = \boldsymbol{W}_s$ and the training is done with the basic focal loss. The fine-tuning step uses all seen and novel class data together.
• *Baseline-II:* The second baseline approach is identical to Baseline-I, except that the fine-tuning step uses novel class data and a few examples of seen.
Finally, *Ours* denote the complete approach where the RetinaNet architecture

Table 3. Base class (of Novel set 1) mAP on Pascal VOC 2007 test set.

Method	aero	bike	boat	bottle	car	cat	chair	table	dog	horse	person	plant	sheep	train	tv	mean
LSTD [46]	74.8	68.7	**57.1**	44.1	**78.0**	83.4	46.9	64.0	**78.7**	**79.1**	70.1	39.2	58.1	**79.8**	71.9	66.3
Kang *et al.*[37]	73.6	**73.1**	56.7	41.6	76.1	78.7	42.6	**66.8**	72.0	77.7	68.5	42.0	57.1	74.7	70.7	64.8
Ours	**80.4**	52.8	50.2	**55.9**	76.9	**85.1**	**49.8**	54.0	76.8	72.7	**81.1**	**44.8**	**61.7**	79.0	**76.8**	**66.5**

is trained with adaptive prototypes in the semantic processing pipeline i.e. $g(\boldsymbol{W}_s) = \delta(\boldsymbol{W}_s\boldsymbol{M}\boldsymbol{D})$ and the training is done with our proposed loss.

4.1 ASD Performance

Here, we discuss the ASD and GASD performance with the 65/8/7 split of MSCOCO. For ASD, we show the performance of novel classes (i.e. unseen and few-shot classes) and the harmonic mean of individual performances. For GASD, we report separate seen, few-shot, unseen mAP and their harmonic mean mAP to show the overall performance.

Main results: In Table 1, we report the our main results and comparisons with the baselines. Our observations are as follows: **(1)** Using more few-shot examples generally helps. However the effect of higher shots on the unseen performance is not always positive since more instances of few-shot classes can bias the model towards them. **(2)** Except Baseline-1, few-shot mAP is always better than unseen mAP because few-shot examples with our proposed loss improve the alignment with respective class semantics. In the Baseline-I case, as all seen and few-shot data is used together, the network overfits to seen classes. **(3)** Our seen class performance in GASD remains good across different shots. This denotes that the network does not forget seen classes when trained on novel ones. Seen classes get the maximum performance for the Baseline-I due to overfitting, thereby giving poor performance on novel classes. **(4)** Across different shots, Baseline-II beats the Baseline-I method as it is less prone to overfitting. With the proposed adaptive semantic prototypes and our loss function, we beat Baseline-II. **(5)** The improvement for unseen mAP is greater than few-shot or seen mAP irrespective of the number of shots, ASD, or GASD tasks. It tells us that our loss formulation not only tackles the class imbalance of few-shot classes but also promotes detection of unseen classes. In Fig. 7 and Fig. 1 of the supplementary material, we show qualitative results for GASD.

Ablation studies: In Table 2, we report ablation experiments on MSCOCO dataset with with alternate versions of our approach. Baseline-I and Baseline-II operate with fixed semantics. For the rest of the cases, we use our adaptive semantic-prototype approach (Sec. 3.2) to update the word vectors. Here, we first use a basic focal loss [43] (FL) to train the network. This approach outperforms both baselines because of the adaptable semantic prototypes. Then, we try two variants of FL: Anchor Loss [47] (AL) and a modified anchor loss with our loss penalty definition for few-shot classes. We notice that these variations do not

Table 4. FSD mAP of novel classes on Pascal VOC 2007 test set.

	Novel Set 1						Novel Set 2						Novel Set 3					
Method	bird	bus	cow	mbike	sofa	mean	aero	bottle	cow	horse	sofa	mean	boat	cat	mbike	sheep	sofa	mean
LSTD [46]	23.1	22.6	15.9	0.4	0.0	12.4	12.6	0.7	11.3	0.4	0.0	5.0	0.0	36.6	21.4	16.9	0.0	15.0
Kang *et al.*[37]	26.1	19.1	**40.7**	20.4	**27.1**	26.7	29.4	**4.6**	34.9	6.8	**37.9**	22.7	**11.7**	48.2	17.4	34.7	**30.1**	28.4
Ours	**37.4**	**23.0**	23.7	**37.0**	25.0	**29.2**	**32.0**	1.1	**39.2**	**55.5**	25.1	**30.6**	4.4	**66.9**	**43.7**	**49.0**	25.5	**37.9**

work well in both ASD and GASD cases because AL penalizes negative anchors that network confuses with positive ones. This idea is beneficial for traditional recognition cases, but unsuitable for ZSD/FSD/ASD scenarios. This is because a negative anchor may contain an unseen object which is closely related to seen or few-shot semantics, and we do not want to suppress the anchor even though it predicts similar scores as the positive ones. Next, we apply our loss by fixing p_* to a constant value e.g., 0.3, 0.5, and 1. These trials outperform both baselines and FL based methods since the network emphasizes few-shot examples based on the quality of the visual-semantic alignment. Finally, alongside the adaptive semantics, we apply our final loss formulation which dynamically selects p_*. Our loss beats all previously described settings because it brings better emphasis on novel classes. Notably, we also experiment with the Our* case that applies our loss to all predictions (instead of just the novel scores) i.e., $\beta > 0$ for all classes. However, it does not perform as well as Ours potentially because the representations suitable for seen classes are already learnt well.

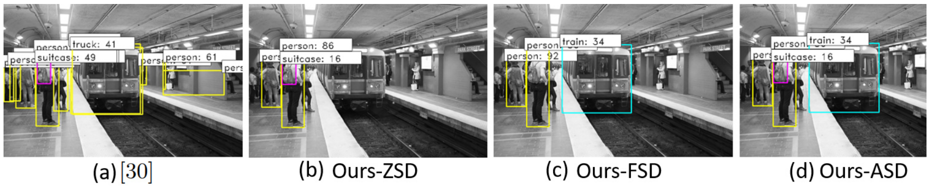
4.2 FSD Performance

If U=0, our ASD framework becomes a few-shot detection problem. In this paper, we experiment on FSD with the following three different dataset setups.

MSCOCO: Here we consider all 15 novel classes of the MSCOCO split [30] as few-shot classes. We report mAP results of 1, 5 and 10-shot detection tasks of Baseline-I, Baseline-II, and Ours model in Fig. 6 (a-d). Besides, we report generalized FSD results. Overall, FSD performance improves with more examples. When trained with the adaptive semantic prototypes and rebalancing loss, our model successfully outperforms both baselines.

MSCOCO to PASCAL: It is a cross-dataset experiment. Following [37], we use 20 PASCAL VOC classes (overlapped with MSCOCO) as the few-shot classes and the remaining 60 MSCOCO classes as seen. This setting performs base training on MSCOCO seen classes and fine-tunes the base model using the 10-shot examples of the PASCAL VOC dataset. Finally, the PASCAL VOC 2007 test set is used to evaluate FSD. In Fig. 6(e), our method outperforms others including a recent approach [37].

PASCAL VOC 2007/12: Using three 15/5 novel-splits proposed in [37], we compare FSD performance of our work with Kang *et al.* [37] and LSTD [46] in Tables 3 and 4. We achieve better performance than them in both novel and base class detection with 3-shot detection settings.



(a) [30] (b) Ours-ZSD (c) Ours-FSD (d) Ours-ASD

Fig. 7. Qualitative comparison with [30] and Our method. Object bounding boxes: Yellow (seen), blue (few-shot) and pink (unseen). *(best viewed with zoom)*

Table 5. ZSD results on MS-COCO dataset. Our rebalancing loss used in the fine-tuning stage (applied on seen data) leads to improved results.

Method		**GZSD**		
	ZSD	Seen	Unseen	HM
Split in [22] (↓)	(mAP/RE)	(mAP/RE)	(mAP/RE)	(mAP/RE)
SB [22]	0.70/24.39	-	-	-
DSES [22]	0.54/27.19	-/15.02	-/15.32	-/15.17
Baseline	5.91/18.67	36.57/42.21	2.64/17.60	4.93/24.84
Ours	**7.78/32.83**	34.50/41.66	**3.06/27.34**	**5.63/33.01**

4.3 ZSD Performance

For ZSD case, after the base training, we do not have any more data to fine-tune. Thus, we perform the second step of fine-tuning with the same data used in the first step but apply our loss instead of the focal loss. As Q=0, we consider each seen class as a few-shot class during the second step. It emphasizes all seen classes in the same way. But, based on the dynamic choice of p_*, the network penalizes a bad prediction by calculating high loss and compensates a good prediction with no penalty. We notice that it helps to improve ZSD performance. We apply this process with the 48/17 split setting of [22] on MSCOCO. We report the mAP and Recall (RE) scores of this experiment in Table 5. With the recommended setting of [22], our work outperforms other methods in both ZSD and GZSD.

5 Conclusion

In this paper, we propose a unified any-shot detection approach where novel classes include both unseen and few-shot objects. Traditional approaches consider solving zero and few-shot tasks separately, whereas our approach encapsulates both tasks into a common framework. This approach does not forget the base training while learning novel classes, which helps to perform generalized ASD. Moreover, we propose a new loss function to learn new tasks. This loss penalizes the wrong prediction of a novel class more than the seen classes. We evaluate the proposed ASD tasks on the challenging MSCOCO and PASCAL VOC datasets. Besides, we compare ZSD and FSD performance of our approach with established state-of-the-art methods. Our first ASD framework delivers strong performance on ZSD, FSD, and ASD tasks.

References

1. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. IEEE Transactions on Pattern Analysis and Machine Intelligence **41** (2019) 2251–2265
2. Chen, H., Luo, Y., Cao, L., Zhang, B., Guo, G., Wang, C., Li, J., Ji, R.: Generalized zero-shot vehicle detection in remote sensing imagery via coarse-to-fine framework. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization (2019) 687–693
3. Chao, W.L., Changpinyo, S., Gong, B., Sha, F.: An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In Leibe, B., Matas, J., Sebe, N., Welling, M., eds.: Computer Vision – ECCV 2016, Cham, Springer International Publishing (2016) 52–68
4. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR, IEEE (2009) 1778–1785
5. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. NIPS'13, USA, Curran Associates Inc. (2013) 3111–3119
6. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP. (2014) 1532–1543
7. Song, J., Shen, C., Yang, Y., Liu, Y., Song, M.: Transductive unbiased embedding for zero-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
8. Zhao, A., Ding, M., Guan, J., Lu, Z., Xiang, T., Wen, J.R.: Domain-invariant projection learning for zero-shot recognition. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., eds.: Advances in Neural Information Processing Systems 31. Curran Associates, Inc. (2018) 1019–1030
9. Kodirov, E., Xiang, T., Fu, Z., Gong, S.: Unsupervised domain adaptation for zero-shot learning. In: The IEEE International Conference on Computer Vision (ICCV). (2015)
10. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
11. Al-Halah, Z., Tapaswi, M., Stiefelhagen, R.: Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
12. Al-Halah, Z., Stiefelhagen, R.: Automatic discovery, association estimation and learning of semantic attributes for a thousand categories. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
13. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C., Huang, J.B.: A closer look at few-shot classification. In: International Conference on Learning Representations. (2019)
14. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS'16, USA, Curran Associates Inc. (2016) 3637–3645
15. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems. (2017) 4077–4087

16. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: In International Conference on Learning Representations (ICLR). (2017)
17. Qi, H., Brown, M., Lowe, D.G.: Low-shot learning with imprinted weights. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
18. Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., Akata, Z.: Generalized zero- and few-shot learning via aligned variational autoencoders. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
19. Xian, Y., Sharma, S., Schiele, B., Akata, Z.: F-vaegan-d2: A feature generating framework for any-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
20. Rahman, S., Khan, S., Porikli, F.: A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. IEEE Transactions on Image Processing **27** (2018) 5652–5667
21. Tsai, Y.H., Huang, L., Salakhutdinov, R.: Learning robust visual-semantic embeddings. In: The IEEE International Conference on Computer Vision (ICCV). (2017)
22. Bansal, A., Sikka, K., Sharma, G., Chellappa, R., Divakaran, A.: Zero-shot object detection. In: The European Conference on Computer Vision (ECCV). (2018)
23. Demirel, B., Cinbis, R.G., Ikizler-Cinbis, N.: Zero-shot object detection by hybrid region embedding. In: British Machine Vision Conference (BMVC). (2018)
24. Zhu, P., Wang, H., Saligrama, V.: Zero shot detection. IEEE Transactions on Circuits and Systems for Video Technology (2019) 1–1
25. Rahman, S., Khan, S., Porikli, F.: Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In: Computer Vision – ACCV 2018, Cham, Springer International Publishing (2019) 547–563
26. Rahman, S., Khan, S.H., Porikli, F.: Zero-shot object detection: Joint recognition and localization of novel concepts. International Journal of Computer Vision **128** (2020) 2979–2999
27. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., eds.: ECCV, Cham, Springer International Publishing (2014) 391–405
28. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2017) 1137–1149
30. Rahman, S., Khan, S., Barnes, N.: Polarity loss for zero-shot object detection. arXiv preprint arXiv:1811.08982 (2018)
31. Rahman, S., Khan, S., Barnes, N.: Transductive learning for zero-shot object detection. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
32. Li, Z., Yao, L., Zhang, X., Wang, X., Kanhere, S., Zhang, H.: Zero-shot object detection with textual descriptions. Proceedings of the AAAI Conference on Artificial Intelligence **33** (2019) 8690–8697
33. Dong, X., Zheng, L., Ma, F., Yang, Y., Meng, D.: Few-example object detection with model communication. IEEE Transactions on Pattern Analysis and Machine Intelligence **41** (2019) 1641–1654
34. Wang, Y.X., Hebert, M.: Model recommendation: Generating object detectors from few samples. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015)

35. Chen, H., Wang, Y., Wang, G., Qiao, Y.: LSTD: A low-shot transfer detector for object detection. In McIlraith, S.A., Weinberger, K.Q., eds.: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, AAAI Press (2018) 2836–2843
36. Karlinsky, L., Shtok, J., Harary, S., Schwartz, E., Aides, A., Feris, R., Giryes, R., Bronstein, A.M.: Repmet: Representative-based metric learning for classification and few-shot object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
37. Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T.: Few-shot object detection via feature reweighting. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
38. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence **40** (2018) 2935–2947
39. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, USA, Curran Associates Inc. (2017) 742–751
40. Shmelkov, K., Schmid, C., Alahari, K.: Incremental learning of object detectors without catastrophic forgetting. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 3400–3409
41. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
42. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
43. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)
44. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV, Springer (2014) 740–755
45. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.T.: Nus-wide: A real-world web image database from national university of singapore. In: CIVR, Santorini, Greece. (July 8-10, 2009)
46. Chen, H., Wang, Y., Wang, G., Qiao, Y.: Lstd: A low-shot transfer detector for object detection. In: Thirty-Second AAAI Conference on Artificial Intelligence. (2018)
47. Ryou, S., Jeong, S.G., Perona, P.: Anchor loss: Modulating loss scale based on prediction difficulty. In: The IEEE International Conference on Computer Vision (ICCV). (2019)