UC-OWOD: Unknown-Classified Open World Object Detection

Zhiheng Wu[1,2], Yue Lu[1,2], Xingyu Chen[3], Zhengxing Wu[1,2,⋆], Liwen Kang[1,2], and Junzhi Yu[1,4]

[1] Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
{wuzhiheng2020, luyue2018, zhengxing.wu, kangliwen2020, junzhi.yu}@ia.ac.cn
[3] Xiaobing.AI
chenxingyu@xiaobing.ai
[4] Peking University

Abstract. Open World Object Detection (OWOD) is a challenging computer vision problem that requires detecting unknown objects and gradually learning the identified unknown classes. However, it cannot distinguish unknown instances as multiple unknown classes. In this work, we propose a novel OWOD problem called Unknown-Classified Open World Object Detection (UC-OWOD). UC-OWOD aims to detect unknown instances and classify them into different unknown classes. Besides, we formulate the problem and devise a two-stage object detector to solve UC-OWOD. First, unknown label-aware proposal and unknown-discriminative classification head are used to detect known and unknown objects. Then, similarity-based unknown classification and unknown clustering refinement modules are constructed to distinguish multiple unknown classes. Moreover, two novel evaluation protocols are designed to evaluate unknown-class detection. Abundant experiments and visualizations prove the effectiveness of the proposed method. Code is available at https://github.com/JohnWuzh/UC-OWOD.

Keywords: OWOD, UC-OWOD, Object Detection, Clustering

1 Introduction

Nowadays, deep learning methods have achieved great success in object detection [20,31,47,35,8,10]. Traditional object detection methods are developed under a closed-world assumption, so they can only detect known (labeled) categories [17,46,62]. However, the real world contains many unknown (unlabeled) classes that can hardly be properly handled by conventional detection. Therefore, studying the Open World Object Detection (OWOD) problem for detecting unknown instances is of great significance to facilitate the practical application.

⋆ Corresponding authors.

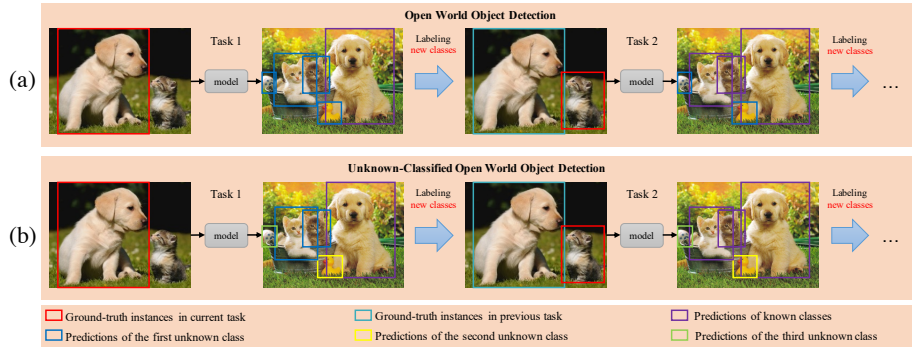

Fig. 1. The comparison between OWOD and UC-OWOD. They can both learn the newly annotated classes by human annotators without forgetting in the next task. (a) OWOD detects unknown objects as a same class. (b) UC-OWOD can detect unknown objects as different classes.

The OWOD problem was pioneered by [24], as shown in Fig. 1 (a). OWOD contains multiple incremental tasks. In each task, OWOD is able to identify all unknown instances as *unknown*. Then, human annotators can gradually assign labels to classes of interest, and the model learns these classes incrementally in the next task. However, beyond distinguishing unknown classes, we also need to determine whether multiple unknown instances belong to the same category. Therefore, there is still a huge difficulty when using OWOD for real-world tasks. For example, in practical applications in robotics [16,28] and self-driving cars [7,53], it is necessary to explore the unknown environment and adopt different strategies for different unknown classes, which requires detection algorithms to confidently localize unknown instances and classify them into different unknown classes.

Most existing open-world detectors are designed for OWOD problem. For example, Open World Object Detector (ORE) [24] can detect unknown classes, but it does not consider the case of classifying unknown objects. More specifically, ORE used pseudo-label supervised training to detect unknown instances. Since pseudo-labels can only be marked as *unknown*, the ORE model cannot be directly used to solve the problem of detecting unknown classes as different classes. Similarly, existing OWOD methods models such as [18,60] follow ORE's spirit, and we are not aware of any previous work that can distinguish multiple unknown classes.

Another difficulty in studying unknown object classification problems is the immature evaluation criterion. Existing metrics only evaluate the degree of confusion between unknown and known classes. They cannot evaluate the situation where two unknown objects of different classes are detected as the same class. But these problems cannot be ignored because they may cause the model to misclassify unknown objects. Therefore, a more reasonable evaluation metric is urgently needed to evaluate the detection accuracy of multiple unknown classes.

Considering the above issues, we propose a novel OWOD problem that is closer to the real-world setting, namely, Unknown-Classified Open World Object Detection (UC-OWOD), which can detect unknown objects as different unknown classes (see Fig. 1 (b)). Meanwhile, we propose a novel framework based on the two-stage detection pipeline to solve this problem. In particular, we design the unknown label-aware proposal (ULP) to construct unknown object ground-truth, the unknown-discriminative classification head (UCH) to mine unknown objects, the similarity-based unknown classification (SUC) to detect unknown objects as different classes, and the unknown clustering refinement (UCR) to refine the classification of unknown objects. To more accurately evaluate the UC-OWOD problem, we propose novel metrics to evaluate the classification and localization performance of unknown instances. A maximum matching is used to assign ground-truth to unknown objects more reasonably. Ultimately, our model achieves the best performance in both existing evaluation metrics and new evaluation metrics. Our main contributions are as follows:

- We introduce a new problem setting, i.e., unknown-classified open world object detection, to inspire future research on real-world object detection.
- We propose a method to solve the UC-OWOD problem based on the unknown label-aware proposal, the unknown-discriminative classification head, the similarity-based unknown classification, and the unknown clustering refinement.
- Novel evaluation metrics for UC-OWOD are proposed, which can evaluate the localization and classification of unknown objects. Extensive experiments are conducted, and the results demonstrate the effectiveness of our method and new metrics for the UC-OWOD problem.

2 Related Work

Open Set Recognition and Detection. Open Set Recognition was first defined as a constrained minimization problem [51], and it can submit unknown classes to the algorithm during the testing phase. It was developed to a multi-class classifier by [23,50]. Liu et al. considered a long-tailed recognition environment and developed a metric learning framework to identify unseen classes as unknown classes [33]. Self-supervised learning [41] and unsupervised learning with reconstruction [58] have also been used for open-set recognition. Yue et al. provided a theoretical ground for balancing and improving the seen/unseen classification imbalance [59]. Bendale and Boult proposed a method to adapt deep networks to Open Set Recognition, using OpenMax layers to estimate the probability that the input is from an unknown class [6]. Dhamija et al. first proposed the open-set object detection protocol and formalized the open-set object detection problem [11]. Miller et al. improved object detection performance by extracting label uncertainty under open conditions commonly encountered in robot vision [40]. Some follow-up work also exploited measures of (spatial and semantic) uncertainty in object detectors to reject unknown categories [19]. Miller et al. found that the correct choice of affinity clustering combinations can

greatly improve the effectiveness of classification, spatial uncertainty estimation, and the resulting object detection performance [39]. However, these methods cannot gradually adjust their knowledge in a dynamic world. By contrast, our model can dynamically update known classes based on human-annotated labels.

Open World Recognition and Detection. Compared to open set problems, open world problem has dynamic datasets and can continuously add new known classes like continuous learning [42,13,48,54,26]. Bendale et al. first proposed Open World Recognition and presented a protocol for the evaluation of open world recognition systems [5]. Xu et al. proposed a meta-learning approach to the open world learning problem that uses only examples of instantly seen classes (including newly added classes) for classification and rejection [57]. Joseph et al. presented a new computer vision problem called OWOD [24]. The ORE proposed by them can classify proposals between known and unknown classes, but it relies on a holdout validation set with weak unknown supervision to learn the energy distributions of known and unknown classes. The open-world detection transformer (OW-DETR) improved performance using multi-scale self-attention and deformable receptive fields [18]. Zhao et al. further proposed an OWOD framework including an auxiliary proposal advisor and a class-specific expelling classifier [60]. None of these methods implements the classification of unknown classes. Our work mainly studies the classification of unknown objects.

Constrained Clustering. Constrained clustering is a semi-supervised learning method that involves prior knowledge to assist clustering. The proposed methods for constrained clustering can be divided into three types, i.e., search-based (also known as constraint-based), distance-based (also known as similarity-based), and hybrid (also known as search-and-distance-based) methods [61]. A common technique in search-based methods is to modify the objective function by adding penalty terms for unsatisfied constraints. In distance-based methods, existing clustering methods are usually used, but the distance metric of this method is modified according to prior knowledge. Hybrid methods integrate search-based and distance-based methods. They benefit from the strengths of both and generally perform better than separate methods [12]. Basu et al. allowed the constraints to be violated with violation cost, while optimizing the distance metric [4]. Hsu et al. designed a new loss function to normalize classification with constrained clustering losses, while using other similarity prediction models as pairwise constraints in the clustering process [22]. Lin et al. took pairwise constraints as prior knowledge to guide the clustering process [30]. We use pairwise constraints to optimize the unknown object classification in the model.

3 Unknown-Classified Open World Object Detection

3.1 Problem Formulation

The UC-OWOD problem is defined as follows. There are a set of tasks $\mathcal{T} = \{T_1, T_2, ...\}$. In task T_t, we have a known class set $\mathcal{K}^t = \{1, 2, ..., C\}$ and unknown class set $\mathcal{U}^t = \{C+1, C+2, ...\}$, where C is the number of known classes. The known class set in task T_{t+1} contains that in task T_t, i.e., $\mathcal{K}^t \subset \mathcal{K}^{t+1}$. The

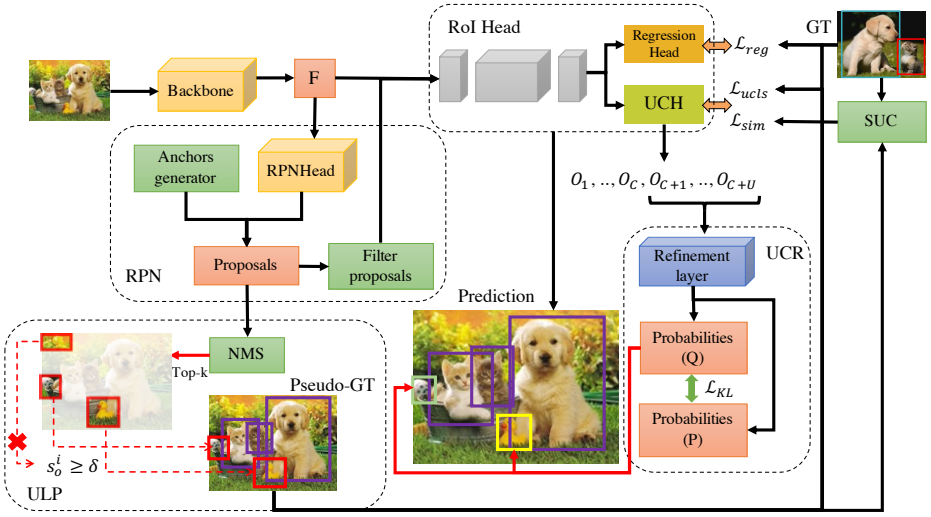

Fig. 2. The architecture of our model. Pseudo-label candidate boxes are filtered according to whether their score s_o^i is greater than the threshold δ. During the training, ULP constructs Pseudo-GT for unknown objects based on the proposals of RPN. According to the regression head and UCH of the model, $\mathcal{L}_{reg}$ and $\mathcal{L}_{ucls}$ are calculated respectively, and $\mathcal{L}_{sim}$ is got by SUC. During the refining, the unknown objects $\{O_{C+1}, ..., O_{C+U}\}$ obtained by UCH are input to UCR to refine clustering, where U is the number of unknown classes.

label of the k-th object of the known class dataset $\mathcal{D}^t = \{\mathbf{X}^t, \mathbf{Y}^t\}$ is $\mathbf{y}_k = [l_k, x_k, y_k, w_k, h_k]$, where the class label $l_k \in \mathcal{K}^t$ and x_k, y_k, w_k, h_k denote the bounding box centre coordinates, width and height, respectively. $\mathbf{X}^t$ and $\mathbf{Y}^t$ are the input images and labels, respectively. Instances of unknown classes do not have labels. The object detector $\mathcal{M}_C$ is able to identify test instances belonging to any known class, and can also detect the new or unseen class instances as different unknown classes. Human users can identify u new classes of interest from the unknown set of instances $\mathbf{U}^t$ and provide the corresponding training examples. Update known class set $\mathcal{K}^{t+1} = \mathcal{K}^t \cup \{C+1, ..., C+u\}$. By incrementally adding u new classes in the next task, the learner creates an updated model $\mathcal{M}_{C+u}$ without the need to retrain the model on the entire dataset.

3.2 Overall Architecture

Fig. 2 shows the overall architecture of the proposed method for UC-OWOD. We use Faster R-CNN [47] as the base detector. We introduce (1) ULP and UCH to solve the problem of discovering unknown classes from the background, (2) SUC to detect unknown objects as different classes, and (3) UCR to refine the classification of unknown objects and enhance the robustness of the algorithm. In order to model the differences between unknown objects, we propose a new classification loss. Details will be discussed in the following subsections.

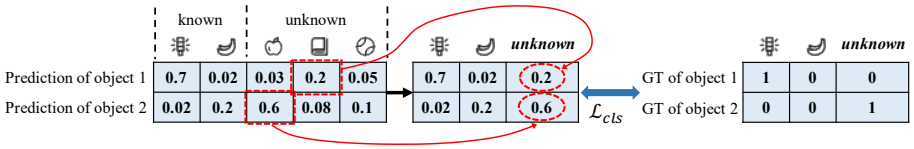

Fig. 3. The diagram of UCH. Traffic light and banana are known classes. Apple, book, and baseball are unknown classes. The unknown class only selects the value with the highest score when calculating the loss.

3.3 Detection of Unknown Objects

Unknown Label-Aware Proposal. Since unknown instances are not labeled, pseudo-labels need to be constructed to train the model's ability to detect unknown classes. We adopt a novel pseudo-labeling strategy, which has better generalization and applicability in detection with multiple unknown classes, as shown in the bottom left of Fig. 2. Based on the fact that the Region Proposal Network (RPN) is class-agnostic, we construct pseudo-labels with bounding box proposals generated by RPN and corresponding objectness scores. First, all proposals are filtered by Non-Maximum Suppression (NMS) to avoid partial overlap between pseudo-labels. Second, we select the filtered top-k background proposals as candidates, which are sorted by their objectness scores. Third, in order to avoid marking the real background regions proposals as *unknown* and make the training results more robust, among the candidates, the proposals with objectness score s_o greater than the threshold δ are used as pseudo-labels, i.e., $\mathbf{y}_{unk} = [\texttt{unknown}, x_i, y_i, w_i, h_i]$ serves as the unknown label-aware proposal.

Unknown-Discriminative Classification Head. To enable the model to locate and classify unknown classes, we introduce multiple unknown classes in the original classification head: $F_{cls} : \mathbb{R}^D \rightarrow \mathbb{R}^{C+U}$, where U is the number of unknown classes. In the training phase, the pseudo-labels are all marked as *unknown*. The original classification strategy cannot classify a variety of unknown objects, so we modify the original classification loss. As shown in Fig. 3, the classification loss of unknown classes is computed by using pseudo labels and the maximum probabilities that are predicted by multiple unknown classes. The new classification loss is constructed as

$$\mathcal{L}_{ucls} = -\frac{1}{N}\sum_{i=1}^{N}(\sum_{j=1}^{C} l_{i,j}\log(p_{i,j}) - l_i^*\log(\max\{p_{i,C+1},...,p_{i,C+U}\})), \quad (1)$$

where N is the number of instances, l is the label of the known class, l^* is the pseudo-label of the unknown class, and p is the predicted probability.

3.4 Similarity-Based Unknown Classification

Clustering unknown classes allow the model to distinguish between different unknown classes. We adopt a pairwise classification loss to measure the similarity

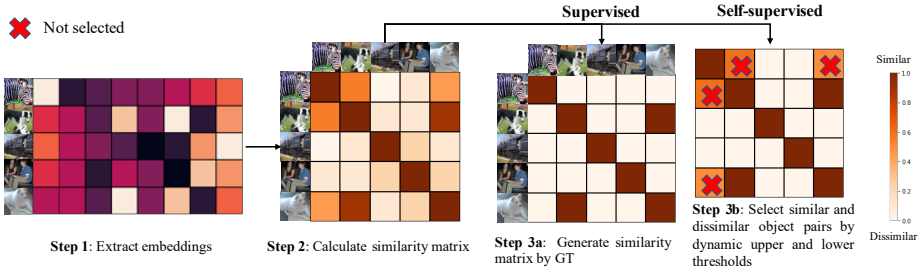

Fig. 4. Build of similarity matrix of embeddings, using supervised method for known classes and self-supervised method for unknown classes.

between samples. By determining whether pairs of samples are similar, our model can classify unknown classes. The outputs E of the UCH, which can represent category information, are used to compute the similarity matrix S:

$$S_{ij} = \frac{E_i E_j^T}{||E_i||||E_j||}, \tag{2}$$

where $||\cdot||$ is L2 norm and $i, j \in \{1, \ldots, n\}$, and n represents the number of proposals. S_{ij} represents the similarity between the i-th proposal and the j-th proposal. As shown in Fig. 4, we use supervised and self-supervised methods successively to optimize the model.

Supervised Method. We treat labeled data as prior knowledge and use it to guide similar relationships between different unknown instances. In supervised methods, since the relationship between unknown instances is not known, we only use known-known instance pairs, unknown-known instance pairs, known-background instance pairs, and unknown-background instance pairs. We can construct the label matrix M as

$$M_{ij} = \begin{cases} 1, \text{ if } l_i = l_j \text{ and } l_i, l_j \notin \mathcal{U}, \\ 0, \text{ if } l_i \neq l_j, \\ \text{Not selected, otherwise,} \end{cases} \tag{3}$$

where l_i is the class label of the i-th instance, $i, j \in \{1, \ldots, n\}$, and $\mathcal{U}$ is the set of unknown classes. Known instances with ground-truth are utilized to reduce errors. Therefore, we construct a similarity loss $\mathcal{L}_{sim}$ with labels M and similarity S as

$$\mathcal{L}_{sim}(M_{ij}, S_{ij}) = -M_{ij} \log(S_{ij}) - (1 - M_{ij}) \log(1 - S_{ij}). \tag{4}$$

Self-Supervised Method. We use thresholds to determine whether unknown instance pairs are similar. $TH(\lambda)$ and $TL(\lambda)$ are dynamic upper and lower thresholds applied to the similarity matrix S to obtain the self-labeled matrix $\tilde{M}$, where λ is an adaptive parameter that controls sample selection. Those

unknown instance pairs that have similarities between $TH(\lambda)$ and $TL(\lambda)$ are excluded from the training phase. $\tilde{M}$ is defined as follows:

$$\tilde{M}_{ij} = \begin{cases} 1, & \text{if } l_i, l_j \in \mathcal{U} \text{ and } S_{ij} > TH(\lambda), \\ -1, & \text{if } l_i, l_j \in \mathcal{U} \text{ and } S_{ij} < TL(\lambda), \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Then, we construct the label matrix $\hat{M}$ with the self-labeled matrix $\tilde{M}$ and the class labels l as

$$\hat{M}_{ij} = \begin{cases} 1, \text{ if } l_i = l_j \text{ and } l_i, l_j \notin \mathcal{U}, \text{ or } \tilde{M}_{ij} > 0, \\ 0, \text{ if } l_i \neq l_j \text{ or } \tilde{M}_{ij} < 0, \\ \text{Not selected, otherwise.} \end{cases} \tag{6}$$

The similarity loss $\hat{\mathcal{L}}_{sim}$ is computed by the similarity matrix S and the label matrix $\hat{M}$:

$$\hat{\mathcal{L}}_{sim}(\hat{M}_{ij}, S_{ij}) = -\hat{M}_{ij} \log(S_{ij}) - (1 - \hat{M}_{ij}) \log(1 - S_{ij}) + \mathcal{L}_{ul}(\lambda), \tag{7}$$

where the penalty term $\mathcal{L}_{ul}(\lambda)$ for the number of samples is given as

$$\mathcal{L}_{ul}(\lambda) = TH(\lambda) - TL(\lambda). \tag{8}$$

The adaptive parameter λ updated by:

$$\lambda := \lambda - \eta \cdot \frac{\partial \mathcal{L}_{ul}(\lambda)}{\partial \lambda}, \tag{9}$$

where η is the learning rate of λ. More and more instance pairs participate in the training phase as λ is updating. To obtain clustering-friendly representations, we train the model from easily classified unknown instance pairs to hardly classified unknown instance pairs iteratively as the thresholds change. The iterative process is terminated when $TH(\lambda) \leq TL(\lambda)$.

3.5 Unknown Clustering Refinement

To enhance the robustness of the proposed algorithm, we apply the soft assignment method [56] to improve the unknown classification based on the previous network output. UCR uses clustering to improve the separability of unknown objects. In the first step, according to the output of UCH, the embedding E of the unknown class and the cluster centroid Φ of the unknown class are obtained. And we compute a soft assignment between E_i and Φ_j saved in the refinement layer while using the Student's t-distribution [36] as the kernel:

$$P_{ij} = \frac{(1 + ||E_i - \Phi_j||^2)^{-1}}{\sum_k (1 + ||E_i - \Phi_k||^2)^{-1}}, \tag{10}$$

where P_{ij} can be interpreted as the probability (soft assignment) of assigning instance i to cluster j. In the second step, the auxiliary target distribution Q is used to refine the clusters based on their high confidence assignments:

$$Q_{ij} = \frac{P_{ij}^2/F_i}{\sum_k P_{ik}^2/F_k}, \tag{11}$$

where $F_i = \sum_i P_{ij}$ is soft cluster frequencies. The quadratic term of the auxiliary target distribution can emphasize high confidence assignments. Therefore, with the assistance of the auxiliary target distribution, the model can gradually learn good clustering structure and improve clustering purity. Then, we minimize the Kullback-Leibler (KL) divergence loss between the soft assignments P and the auxiliary distribution Q to refine clustering:

$$\mathcal{L}_{KL} = \mathrm{KL}(Q||P) = \sum_i \sum_j Q_{ij} \log \frac{Q_{ij}}{P_{ij}}. \tag{12}$$

3.6 Training and Refining

Training. Our model is trained end-to-end with the following loss function:

$$\mathcal{L}_{tra} = \alpha_1 \mathcal{L}_{rpn} + \alpha_2 \mathcal{L}_{ucls} + \alpha_3 \mathcal{L}_{reg} + \alpha_4 \mathcal{L}_{sim}, \tag{13}$$

where $\mathcal{L}_{rpn}$ and $\mathcal{L}_{reg}$ denote the loss terms for RPN and bounding box regression, respectively. In detail, $\mathcal{L}_{rpn}$ is formulated using the standard RPN loss [47], $\mathcal{L}_{reg}$ is the standard ℓ_1 regression loss. $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ denote weight factors. When the model is only trained with the current class of task T_t, it will catastrophically forget the information learned in the previous task [38,15]. Comparing existing solutions, i.e. parameter regularization [2,29], exemplar replay [45,9], dynamically expanding networks [37,49,52], and meta-learning [44,25], we choose a relatively simple few-example replay method [55,43,24]. The model is finetuned using a set of stored examples for each known class after learning the task T_t.
Refining. In the clustering refinement stage for unknown objects, the main purpose is to improve the classification of unknown objects. We only use the KL divergence loss for training on unknown objects:

$$\mathcal{L}_{ref} = \mathcal{L}_{KL}. \tag{14}$$

4 Experiments

4.1 Preparation

Datasets. We evaluate our model for the UC-OWOD problem on the set of tasks $\mathcal{T} = \{T_1, T_2, \cdots\}$. Classes in T_λ are introduced when $t = \lambda$. For the task T_t, all introduced classes in $\{T_\tau : \tau \leq t\}$ are *known* and classes in $\{T_\tau : \tau > t\}$ are *unknown*. As shown in Table 1, we construct 4 tasks with 20 classes in each

Table 1. Datasets for each task. Table shows the semantics and the number of images and instances each task contains.

Task	Task 1	Task 2	Task 3	Task 4
Semantic split	VOC Classes	Outdoor, Accessories Appliance, Truck	Sports Food	Electronic, Indoor Kitchen, Furniture
Training images	16551	45520	39402	40260
Training instances	47223	113741	114452	138996
Test images	10246			
Test instances	61707			

task using the Pascal VOC [14] and MS-COCO [32] datasets. The task T_1 consists of all VOC classes and data, which do not contain any information about unknown instances. This allows the model to be tested without any *unknown* information during the training phase. The remaining 60 classes of MS-COCO are divided into three parts, i.e., T_2, T_3, and T_4. Although the training images in T_2 and T_3 do not have labels of unknown instances, they contain unknown instances, which can test the effect of the model in this situation. In every task, the evaluation data consists of Pascal VOC test split and MS-COCO validation split.

Evaluation Metrics. For the overall evaluation of unknown classes, we use two evaluation metrics, i.e., Absolute Open-Set Error (A-OSE) [40,24] and Wilderness Impact (WI) [11,24]. A-OSE is the number of unknown objects misclassified into *known*. WI is calculated by true positive proposals $TP_{\mathcal{K}}$ and false positive proposals $FP_{\mathcal{K}}$ of current *known*:

$$\text{WI} = \frac{\text{A-OSE}}{TP_{\mathcal{K}} + FP_{\mathcal{K}}}. \tag{15}$$

For the refinement of unknown classes, there is no label-prediction pair, so mean average precision (mAP) does not work. We are also not aware of any other metric that can handle the evaluate multiple unknown categories. Inspired by the clustering evaluation metric, i.e., clustering accuracy [1], we introduce a novel evaluation metric, unknown mean average precision (UC-mAP), to evaluate the detection of unknown classes. Therefore, UC-mAP is the mAP with automatic category matching:

$$\text{UC-mAP}(\mathcal{Y}_{gt}, \mathcal{Y}_{pre}) = \max_{perm \in \mathcal{P}} \text{mAP}(perm(\mathcal{Y}_{pre}), \mathcal{Y}_{gt}), \tag{16}$$

where $\mathcal{P}$ is the set of all permutations in of 1 to U, U is the number of unknown classes, $\mathcal{Y}_{pre}$ is the predicted value, and $\mathcal{Y}_{gt}$ is the ground-truth. The best match uses the Hungarian algorithm [27] for fast computation. The model is also better if it can detect some new instances which are unlabeled in the MS-COCO dataset, but traditional mAP metrics are very sensitive to missing annotations

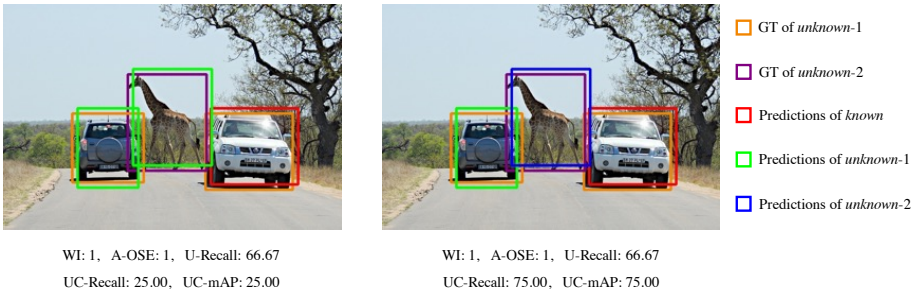

Fig. 5. *Car* is *unknown*-1 and *giraffe* is *unknown*-2. Both images mis-detect the car on the right as *known*, and the left image mis-detects the giraffe as *unknown*-1.

Table 2. Validation results of UC-mAP and UC-Recall. Label-free UC-mAP can achieve the same evaluation results as label-based mAP, as does Recall.

	Task 1	Task 2	Task 3	Task 4
mAP / Recall	56.43 / 75.40	46.14 / 68.02	28.03 / 53.26	26.72 / 50.88
UC-mAP / UC-Recall	56.43 / 75.40	46.14 / 68.02	28.03 / 53.26	26.72 / 50.88

and treat such detections as false positives. Therefore, we also use the unknown class Recall [34,3,18] after maximum matching as the evaluation metric, i.e., UC-Recall.

Implementation Details. Our model is based on the standard Faster R-CNN [47] object detector with ResNet-50 [21] backbone. We set the total number of unknown and known classes to 80, which corresponds to the MS-COCO dataset. As described earlier, in the classification loss, we only learn the unknown class with the highest prediction probability. This is achieved by setting the logits of the invisible classes to a large negative value (v) so that their contribution to the softmax is negligible ($e^{-v} \rightarrow 0$). We set $TH(\lambda) = 0.95 - \lambda$, $TL(\lambda) = 0.455 + 0.1\lambda$, $\alpha_1 = \alpha_2 = \alpha_3 = 1, \alpha_4 = 0.5$, and learning rate is 0.01. When refining, we fix the layers before the refinement layer and use a learning rate of 0.1. The initial cluster centroids of unknown classes are obtained using K-means. Because the refinement phase relies on the unknown object information in the training set, we only use UCR for task 2 and task 3.

Validity of UC-mAP and UC-Recall. We analyze the evaluation results of WI, A-OSE, U-Recall, UC-Recall and UC-mAP in different situations (see Fig. 5). All metrics reflect the situation where unknown objects are misclassified as known. WI, A-OSE, and U-Recall [18] cannot determine whether *unknown*-1 and *unknown*-2 are wrongly classified into the same class, but UC-Recall and UC-mAP may result in higher scores under correct detection. UC-Recall and UC-mAP are further evaluated with known classes of *Oracle* detectors, which can access to all known and unknown labels at any task (see Table 2). We can

Table 3. The performance of our model on known classes. PK means the mAP of previously known instances and CK means the mAP of current known instances.

	Task 1	Task 2	Task 3	Task 4
PK (↑) / CK (↑)	- / 50.66	33.13 / 30.54	28.80 / 16.34	25.57 / 15.88

Table 4. The performance of our model on UC-OWOD. WI, A-OSE, UC-mAP and UC-Recall quantify how the model handles unknown classes.

Task 1		Oracle	Faster-RCNN	Faster-RCNN +Finetuning	ORE	Ours	Ours+UCR
UC-mAP	(↑)	0	0	-	0.0133	**0.1344**	-
WI	(↓)	-	0.0188	-	0.0155	**0.0136**	-
A-OSE	(↓)	-	13300	-	10672	**9294**	-
UC-Recall	(↑)	-	0	-	0.7772	**2.3915**	-

Task 2		Oracle	Faster-RCNN	Faster-RCNN +Finetuning	ORE	Ours	Ours+UCR
UC-mAP	(↑)	15.50	0	0	0.0065	0.0862	**0.1694**
WI	(↓)	0.0022	0.0069	0.0140	0.0153	0.0116	0.0117
A-OSE	(↓)	6050	4582	7169	10376	5602	5602
UC-Recall	(↑)	40.45	0	0	0.0371	2.6926	**3.4431**

Task 3		Oracle	Faster-RCNN	Faster-RCNN +Finetuning	ORE	Ours	Ours+UCR
UC-mAP	(↑)	10.61	0	0	0.0070	0.0249	**0.0744**
WI	(↓)	0.0042	0.0241	0.0099	0.0086	0.0073	**0.0073**
A-OSE	(↓)	4857	4841	9181	7544	3801	**3801**
UC-Recall	(↑)	28.54	0	0	0.8833	4.8077	**8.7303**

see that UC-mAP/UC-Recall are equivalent to mAP/Recall when the model is trained with the corresponding labels.

4.2 Results and Analysis

As shown in Table 3, our model is able to avoid catastrophic forgetting of previous classes. To better analyze the performance on the UC-OWOD problem, we compare our model with Faster-RCNN and ORE, whose performance on unknown object detection is shown in Table 4. Due to limited space, full experimental data are given in Supplementary Materials. WI and A-OSE metrics are used to quantify the degree of confusion between unknown instances and any known classes. The UC-Recall metric is used to quantify the ability of the model to retrieve unknown object instances. The UC-mAP metric is used to quantify

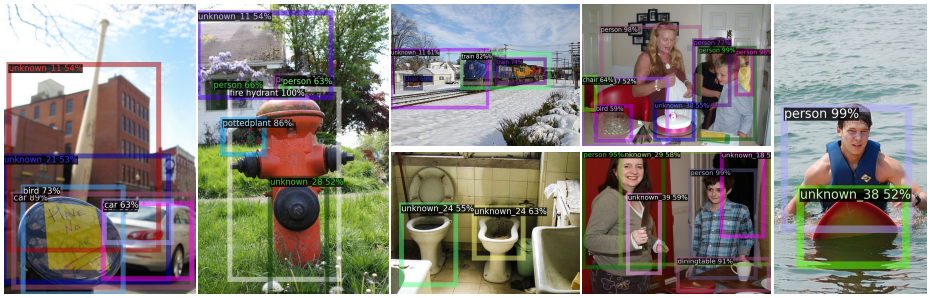

Fig. 6. Qualitative results of our model. *unknown_x* represents an unknown object of the x-th class. Our model detects the *house* as *unknown*-11 and is able to distinguish it from other unknown classes in the same image. This means that our model cannot only detect the categories annotated in the MS-COCO dataset, but also mine new categories and distinguish them from other categories. Some other unknown classes are also shown, i.e., *toilet* as *unknown*-24, *knife* as *unknown*-39 and so on. The last column shows a failure case that misclassify *surfboard* as *unknown*-38 which was actually *cake*. The Supplementary Materials contain more visualised results.

Table 5. Ablation experimental results of our model.

ID	UCH	SUC	WI (↓)	A-OSE (↓)	UC-Recall (↑)	UC-mAP (↑)
1	✗	✗	0.0213	16453	0	0
2	✗	✔	0.0247	16667	0	0
3	✔	✗	0.0155	**9185**	1.4796	0.0176
4	✔	✔	**0.0136**	9294	**2.3915**	**0.1343**

the average level of detection of all unknown classes by the model. Under the setting of UC-OWOD, Faster-RCNN and Faster-RCNN +Finetuning do not have the ability to detect unknown instances, and finetuning will result in lower scores for WI and A-OSE. In all tasks, we achieve better results than ORE on measures about unknown classes. The ability of the model to detect unknown objects is significantly improved after adding UCR. Fig. 6 and Supplementary Materials show qualitative results on example images.

4.3 Ablation Study

Ablation of Components. We design ablation experiments to study the contributions of UCH and SUC in the model (see Table 5). When UCH and SUC (row 1 and row 2) are missing, the model loses its ability to detect unknown classes. Adding only SUC (row 2) will not improve the model's ability to detect unknown classes. Only the absence of SUC (row 3) affects the classification ability for unknown classes, but the model performs best at the detection of known classes. Hence, the scores of WI, UC-Recall, and UC-mAP are worse

Table 6. Sensitivity analysis on hyperparameters.

ID	NMS threshold	δ	Number of pseudo-GT	WI (↓)	A-OSE (↓)	UC-Recall (↑)	UC-mAP (↑)
1	0.3	0.3	1	0.0146	12649	0.9314	0.0375
2	0.3	0.3	5	0.0136	**9294**	**2.3915**	**0.1343**
3	0.3	0.7	1	**0.0101**	11323	1.9017	0.0995
4	0.3	0.7	5	0.0143	10161	1.6730	0.1212
5	0.7	0.3	1	0.0203	12243	0.7222	0.0037
6	0.7	0.3	5	0.0202	12780	0.7923	0.0120
7	0.7	0.7	1	0.0240	13032	0.1399	0.0004
8	0.7	0.7	5	0.0141	12156	0.8827	0.0026

than those that have both UCH and SUC (row 4). Therefore, the best performance is achieved when both components are present.

Sensitivity Analysis on Hyperparameters. As shown in Table 6, we analyze the detection performance of the model under different hyperparameter settings. When the NMS threshold is large, the recall rate for unknown classes is low, because the model may set the region with a high degree of coincidence with the known class label as a pseudo label. The model can only locate known instance regions, but cannot locate unknown instance regions. When the value of δ is large, the model tends to label fewer unknown classes, resulting in poorer detection performance of the model for unknown classes. Similarly, when the Number of pseudo-GT is set to 1, the model will be less effective due to fewer unknown classes being labeled. We chose the hyperparameter settings with the better scores for WI, A-OSE, UC-Recall, and UC-mAP, i.e., NMS threshold is 0.3, δ is 0.3, and Number of pseudo-GT is 5.

5 Conclusions and Future Work

In this work, we have proposed a novel problem UC-OWOD on the basis of OWOD, which is closer to the real world. The UC-OWOD requires detecting unknown objects as different unknown classes. We also establish evaluation protocols for this issue. In addition, we propose a new method including ULP, UCH, SUC, and UCR. Abundant experiments demonstrate the effectiveness of our method on the UC-OWOD problem and also verify the rationality of our metrics. In future work, we hope to apply our method to some real-world online tasks and achieve open-world automatic annotation.

Acknowledgements. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1310300 and in part by the National Natural Science Foundation of China under Grant 62022090.

References

1. Ahmadinejad, N., Liu, L.: J-score: A robust measure of clustering accuracy. arXiv preprint arXiv:2109.01306 (2021) 4.1
2. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 139–154 (2018) 3.6
3. Bansal, A., Sikka, K., Sharma, G., Chellappa, R., Divakaran, A.: Zero-shot object detection. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 384–400 (2018) 4.1
4. Basu, S., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: Proceedings of the 2004 SIAM international conference on data mining. pp. 333–344 (2004) 2
5. Bendale, A., Boult, T.: Towards open world recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1893–1902 (2015) 2
6. Bendale, A., Boult, T.E.: Towards open set deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1563–1572 (2016) 2
7. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11621–11631 (2020) 1
8. Cao, J., Cholakkal, H., Anwer, R.M., Khan, F.S., Pang, Y., Shao, L.: D2det: Towards high quality object detection and instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 1
9. Castro, F.M., Marin-Jimenez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018) 3.6
10. Chen, X., Yu, J., Kong, S., Wu, Z., Wen, L.: Joint anchor-feature refinement for real-time accurate object detection in images and videos. IEEE Transactions on Circuits and Systems for Video Technology **31**(2), 594–607 (2020) 1
11. Dhamija, A., Gunther, M., Ventura, J., Boult, T.: The overlooked elephant of object detection: Open set. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1021–1030 (2020) 2, 4.1
12. Dinler, D., Tural, M.K.: A survey of constrained clustering. In: Unsupervised learning algorithms, pp. 207–235 (2016) 2
13. Dong, N., Zhang, Y., Ding, M., Lee, G.H.: Bridging non co-occurrence with unlabeled in-the-wild data for incremental object detection. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS). pp. 30492–30503 (2021) 2
14. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision (IJCV) **88**(2), 303–338 (2010) 4.1
15. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences **3**(4), 128–135 (1999) 3.6
16. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013) 1

17. Girshick, R.B.: Fast R-CNN. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1440–1448 (2015) 1
18. Gupta, A., Narayan, S., Joseph, K., Khan, S., Khan, F.S., Shah, M.: OW-DETR: Open-world detection transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9235–9244 (2022) 1, 2, 4.1, 4.1
19. Hall, D., Dayoub, F., Skinner, J., Zhang, H., Miller, D., Corke, P., Carneiro, G., Angelova, A., Sünderhauf, N.: Probabilistic object detection: Definition and evaluation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1031–1040 (2020) 2
20. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2961–2969 (2017) 1
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016) 4.1
22. Hsu, Y.C., Lv, Z., Kira, Z.: Learning to cluster in order to transfer across domains and tasks. In: Proceedings of International Conference on Learning Representations (ICLR) (2018) 2
23. Jain, L.P., Scheirer, W.J., Boult, T.E.: Multi-class open set recognition using probability of inclusion. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 393–409 (2014) 2
24. Joseph, K., Khan, S., Khan, F.S., Balasubramanian, V.N.: Towards open world object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5830–5840 (2021) 1, 2, 3.6, 4.1
25. K J, J., N Balasubramanian, V.: Meta-consolidation for continual learning. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS). pp. 14374–14386 (2020) 3.6
26. Kj, J., Rajasegaran, J., Khan, S., Khan, F.S., Balasubramanian, V.N.: Incremental object detection via meta-learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2021) 2
27. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955) 4.1
28. Lenz, I., Lee, H., Saxena, A.: Deep learning for detecting robotic grasps. The International Journal of Robotics Research **34**(4-5), 705–724 (2015) 1
29. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **40**(12), 2935–2947 (2018) 3.6
30. Lin, T.E., Xu, H., Zhang, H.: Discovering new intents via constrained deep adaptive clustering with cluster refinement. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). pp. 8360–8367 (2020) 2
31. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2980–2988 (2017) 1
32. 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: Proceedings of the European Conference on Computer Vision (ECCV). pp. 740–755 (2014) 4.1
33. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2537–2546 (2019) 2

34. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 852–869 (2016) 4.1
35. Lu, Y., Chen, X., Wu, Z., Yu, J.: Decoupled metric network for single-stage few-shot object detection. IEEE Transactions on Cybernetics pp. 1–12 (2022) 1
36. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research (JMLR) **9**(11) (2008) 3.5
37. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7765–7773 (2018) 3.6
38. McCloskey, M., Cohen, N.: Catastrophic interference in connectionist networks: The sequential learning problem. Psychology of Learning and Motivation **24**, 109–165 (1989) 3.6
39. Miller, D., Dayoub, F., Milford, M., Sünderhauf, N.: Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). pp. 2348–2354 (2019) 2
40. Miller, D., Nicholson, L., Dayoub, F., Sünderhauf, N.: Dropout sampling for robust object detection in open-set conditions. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). pp. 3243–3249 (2018) 2, 4.1
41. Perera, P., Morariu, V.I., Jain, R., Manjunatha, V., Wigington, C., Ordonez, V., Patel, V.M.: Generative-discriminative feature representations for open-set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11814–11823 (2020) 2
42. Perez-Rua, J.M., Zhu, X., Hospedales, T.M., Xiang, T.: Incremental few-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13846–13855 (2020) 2
43. Prabhu, A., Torr, P., Dokania, P.: Gdumb: A simple approach that questions our progress in continual learning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 524–540 (2020) 3.6
44. Rajasegaran, J., Khan, S., Hayat, M., Khan, F.S., Shah, M.: itaml: An incremental task-agnostic meta-learning approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13588–13597 (2020) 3.6
45. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2001–2010 (2017) 3.6
46. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 779–788 (2016) 1
47. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS). pp. 91–99 (2015) 1, 3.2, 3.6, 4.1
48. Rostami, M., Spinoulas, L., Hussein, M., Mathai, J., Abd-Almageed, W.: Detection and continual learning of novel face presentation attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 14851–14860 (2021) 2
49. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016) 3.6

50. Scheirer, W.J., Jain, L.P., Boult, T.E.: Probability models for open set recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **36**(11), 2317–2324 (2014) 2
51. Scheirer, W.J., de Rezende Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **35**(7), 1757–1772 (2013) 2
52. Serra, J., Suris, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. In: Proceedings of International Conference on Machine Learning (ICML). pp. 4548–4557 (2018) 3.6
53. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2446–2454 (2020) 1
54. Wang, J., Wang, X., Shang-Guan, Y., Gupta, A.: Wanderlust: Online continual object detection in the real world. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10829–10838 (2021) 2
55. Wang, X., Huang, T.E., Gonzalez, J., Darrell, T., Yu, F.: Frustratingly simple few-shot object detection. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 9919–9928 (2020) 3.6
56. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: Proceedings of International Conference on Machine Learning (ICML). pp. 478–487 (2016) 3.5
57. Xu, H., Liu, B., Shu, L., Yu, P.: Open-world learning and application to product classification. In: Proceedings of the World Wide Web Conference (WWW). pp. 3413–3419 (2019) 2
58. Yoshihashi, R., Shao, W., Kawakami, R., You, S., Iida, M., Naemura, T.: Classification-reconstruction learning for open-set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4016–4025 (2019) 2
59. Yue, Z., Wang, T., Sun, Q., Hua, X.S., Zhang, H.: Counterfactual zero-shot and open-set visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15404–15414 (2021) 2
60. Zhao, X., Liu, X., Shen, Y., Ma, Y., Qiao, Y., Wang, D.: Revisiting open world object detection. arXiv preprint arXiv:2201.00471 (2022) 1, 2
61. Zhigang, C., Xuan, L., Fan, Y.: Constrained k-means with external information. In: Proceedings of 2013 8th International conference on computer science & education. pp. 490–493 (2013) 2
62. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. In: Proceedings of International Conference on Learning Representations (ICLR) (2021) 1