5-9-2021

Towards open world object detection

K. J. Joseph
Indian Institute of Technology Hyderabad

Salman Khan
Mohamed Bin Zayed University of Artificial Intelligence

Fahad Shahbaz Khan
Mohamed Bin Zayed University of Artificial Intelligence

Vineeth N. Balasubramanian
Indian Institute of Technology Hyderabad

Towards Open World Object Detection

K J Joseph[†‡], Salman Khan[‡⋆], Fahad Shahbaz Khan[‡◇], Vineeth N Balasubramanian[†]
[†]Indian Institute of Technology Hyderabad, India [‡]Mohamed bin Zayed University of AI, UAE
[⋆]Australian National University, Australia [◇]Linköping University, Sweden
{cs17m18p100001,vineethnb}@iith.ac.in, {salman.khan,fahad.khan}@mbzuai.ac.ae

Abstract

Humans have a natural instinct to identify unknown object instances in their environments. The intrinsic curiosity about these unknown instances aids in learning about them, when the corresponding knowledge is eventually available. This motivates us to propose a novel computer vision problem called: 'Open World Object Detection', where a model is tasked to: 1) identify objects that have not been introduced to it as 'unknown', without explicit supervision to do so, and 2) incrementally learn these identified unknown categories without forgetting previously learned classes, when the corresponding labels are progressively received. We formulate the problem, introduce a strong evaluation protocol and provide a novel solution, which we call ORE: Open World Object Detector, based on contrastive clustering and energy based unknown identification. Our experimental evaluation and ablation studies analyse the efficacy of ORE in achieving Open World objectives. As an interesting by-product, we find that identifying and characterising unknown instances helps to reduce confusion in an incremental object detection setting, where we achieve state-of-the-art performance, with no extra methodological effort. We hope that our work will attract further research into this newly identified, yet crucial research direction.[1]

1. Introduction

Deep learning has accelerated progress in Object Detection research [14, 54, 19, 31, 52], where a model is tasked to identify and localise objects in an image. All existing approaches work under a strong assumption that all the classes that are to be detected would be available at training phase. Two challenging scenarios arises when we relax this assumption: 1) A test image might contain objects from unknown classes, which should be classified as *unknown*. 2) As and when information (labels) about such identified unknowns become available, the model should be able to incrementally learn the new class. Research in developmental psychology [41, 36] finds out that the ability to identify what one doesn't know, is key in captivating curiosity.

[1]Source code: https://github.com/JosephKJ/OWOD

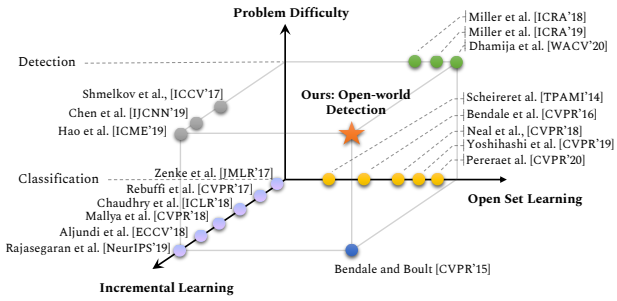


Figure 1: Open World Object Detection (★) is a novel problem that has not been formally defined and addressed so far. Though related to the Open Set and Open World classification, Open World Object Detection offers its own unique challenges, which when addressed, improves the practicality of object detectors.

Such a curiosity fuels the desire to learn new things [9, 16]. This motivates us to propose a new problem where a model should be able to identify instances of unknown objects as unknown and subsequently learns to recognise them when training data progressively arrives, in a *unified* way. We call this problem setting as *Open World Object Detection*.

The number of classes that are annotated in standard vision datasets like Pascal VOC [10] and MS-COCO [32] are very low (20 and 80 respectively) when compared to the infinite number of classes that are present in the open world. Recognising an unknown as an unknown requires strong generalization. Scheirer *et al*. [57] formalise this as *Open Set* classification problem. Henceforth, various methodologies (using 1-vs-rest SVMs and deep learning models) has been formulated to address this challenging setting. Bendale *et al*. [3] extend Open Set to an *Open World* classification setting by additionally updating the image classifier to recognise the identified new unknown classes. Interestingly, as seen in Fig. 1, Open World object detection is unexplored, owing to the difficulty of the problem setting.

The advances in Open Set and Open World image classification cannot be trivially adapted to Open Set and Open World object detection, because of a fundamental difference in the problem setting: *The object detector is trained to detect unknown objects as background.* Instances of many unknown classes would have been already introduced to

the object detector along with known objects. As they are not labelled, these unknown instances would be explicitly learned as background, while training the detection model. Dhamija *et al*. [8] find that even with this extra training signal, the state-of-the-art object detectors result in false positive detections, where the unknown objects end up being classified as one of the known classes, often with very high probability. Miller *et al*. [43] propose to use dropout sampling to get an estimate of the uncertainty of the object detection prediction. This is the only peer-reviewed research work in the open set object detection literature. Our proposed Open World Object Detection goes a step further to incrementally learn the new classes, once they are detected as unknown and an oracle provides labels for the objects of interest among all the unknowns. To the best of our knowledge this has not been tried in the literature.

The Open World Object Detection setting is much more natural than the existing closed-world, static-learning setting. The world is diverse and dynamic in the number, type and configurations of novel classes. It would be naive to assume that all the classes to expect at inference are seen during training. Practical deployments of detection systems in robotics, self-driving cars, plant phenotyping, healthcare and surveillance cannot afford to have complete knowledge on what classes to expect at inference time, while being trained in-house. The most natural and realistic behavior that one can expect from an object detection algorithm deployed in such settings would be to confidently predict an unknown object as unknown, and known objects into the corresponding classes. As and when more information about the identified unknown classes becomes available, the system should be able to incorporate them into its existing knowledge base. This would define a smart object detection system, and ours is an effort towards achieving this goal.
The key contributions of our work are:

- We introduce a novel problem setting, Open World Object Detection, which models the real-world more closely.
- We develop a novel methodology, called ORE, based on contrastive clustering, an unknown-aware proposal network and energy based unknown identification to address the challenges of open world detection.
- We introduce a comprehensive experimental setting, which helps to measure the open world characteristics of an object detector, and benchmark ORE on it against competitive baseline methods.
- As an interesting by-product, the proposed methodology achieves state-of-the-art performance on Incremental Object Detection, even though not primarily designed for it.

2. Related Work

Open Set Classification: The open set setting considers knowledge acquired through training set to be incomplete, thus new unknown classes can be encountered during testing. Scheirer *et al*. [58] developed open set classifiers in a one-vs-rest setting to balance the performance and the risk of labeling a sample far from the known training examples (termed as open space risk). Follow up works [23, 59] extended the open set framework to multi-class classifier setting with probabilistic models to account for the fading away classifier confidences in case of unknown classes.

Bendale and Boult [4] identified unknowns in the feature space of deep networks and used a Weibull distribution to estimate the set risk (called OpenMax classifier). A generative version of OpenMax was proposed in [13] by synthesizing novel class images. Liu *et al*. [35] considered a long-tailed recognition setting where majority, minority and unknown classes coexist. They developed a metric learning framework identify unseen classes as unknown. In similar spirit, several dedicated approaches target on detecting the out of distribution samples [30] or novelties [48]. Recently, self-supervised learning [46] and unsupervised learning with reconstruction [65] have been explored for open set recognition. However, while these works can recognize unknown instances, they cannot dynamically update themselves in an incremental fashion over multiple training episodes. Further, our energy based unknown detection approach has not been explored before.

Open World Classification: [3] first proposed the open world setting for image recognition. Instead of a static classifier trained on a fixed set of classes, they proposed a more flexible setting where knowns and unknowns both coexist. The model can recognize both types of objects and adaptively improve itself when new labels for unknown are provided. Their approach extends Nearest Class Mean classifier to operate in an open world setting by re-calibrating the class probabilities to balance open space risk. [47] studies open world face identity learning while [64] proposed to use an exemplar set of seen classes to match them against a new sample, and rejects it in case of a low match with all previously known classes. However, they don't test on image classification benchmarks and study product classification in e-commerce applications.

Open Set Detection: Dhamija *et al*. [8] formally studied the impact of open set setting on popular object detectors. They noticed that the state of the art object detectors often classify unknown classes with high confidence to seen classes. This is despite the fact that the detectors are explicitly trained with a background class [55, 14, 33] and/or apply one-vs-rest classifiers to model each class [15, 31]. A dedicated body of work [43, 42, 17] focuses on developing measures of (spatial and semantic) uncertainty in object detectors to reject unknown classes. E.g., [43, 42] uses Monte Carlo Dropout [12] sampling in a SSD detector to obtain uncertainty estimates. These methods, however, cannot incrementally adapt their knowledge in a dynamic world.

3. Open World Object Detection

Let us formalise the definition of Open World Object Detection in this section. At any time t, we consider the set of known object classes as $\mathcal{K}^t = \{1, 2, .., \mathrm{C}\} \subset \mathbb{N}^+$ where $\mathbb{N}^+$ denotes the set of positive integers. In order to realistically model the dynamics of real world, we also assume that their exists a set of unknown classes $\mathcal{U} = \{\mathrm{C}+1, ...\}$, which may be encountered during inference. The known object classes $\mathcal{K}_t$ are assumed to be labeled in the dataset $\mathcal{D}^t = \{\mathbf{X}^t, \mathbf{Y}^t\}$ where $\mathbf{X}$ and $\mathbf{Y}$ denote the input images and labels respectively. The input image set comprises of M training images, $\mathbf{X}^t = \{\boldsymbol{I}_1, \ldots, \boldsymbol{I}_M\}$ and associated object labels for each image forms the label set $\mathbf{Y}^t = \{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_M\}$. Each $\boldsymbol{Y}_i = \{\boldsymbol{y}_1, \boldsymbol{y}_2, .., \boldsymbol{y}_K\}$ encodes a set of K object instances with their class labels and locations i.e., $\boldsymbol{y}_k = [l_k, x_k, y_k, w_k, h_k]$, where $l_k \in \mathcal{K}^t$ and x_k, y_k, w_k, h_k denote the bounding box center coordinates, width and height respectively.

The *Open World Object Detection* setting considers an object detection model $\mathcal{M}_\mathrm{C}$ that is trained to detect all the previously encountered C object classes. Importantly, the model $\mathcal{M}_\mathrm{C}$ is able to identify a test instance belonging to any of the known C classes, and can also recognize a new or unseen class instance by classifying it as an *unknown*, denoted by a label zero (0). The unknown set of instances $\mathbf{U}^t$ can then be forwarded to a human user who can identify n new classes of interest (among a potentially large number of unknowns) and provide their training examples. The learner incrementally adds n new classes and updates itself to produce an updated model $\mathcal{M}_{\mathrm{C}+n}$ without retraining from scratch on the whole dataset. The known class set is also updated $\mathcal{K}_{t+1} = \mathcal{K}_t + \{\mathrm{C}+1, \ldots, \mathrm{C}+n\}$. This cycle continues over the life of the object detector, where it adaptively updates itself with new knowledge. The problem setting is illustrated in the top row of Fig. 2.

4. ORE: Open World Object Detector

A successful approach for Open World Object Detection should be able to identify unknown instances without explicit supervision and defy forgetting of earlier instances when labels of these identified novel instances are presented to the model for knowledge upgradation (without retraining from scratch). We propose a solution, ORE which addresses both these challenges in a unified manner.

Neural networks are universal function approximators [22], which learn a mapping between an input and the output through a series of hidden layers. The latent representation learned in these hidden layers directly controls how each function is realised. We hypothesise that learning clear discrimination between classes in the latent space of object detectors could have two fold effect. *First*, it helps the model to identify how the feature representation of an unknown instance is different from the other known instances, which helps identify an unknown instance as a novelty. *Second*, it facilitates learning feature representations for the new class instances without overlapping with the previous classes in the latent space, which helps towards incrementally learning without forgetting. The key component that helps us realise this is our proposed *contrastive clustering* in the latent space, which we elaborate in Sec. 4.1.

To optimally cluster the unknowns using contrastive clustering, we need to have supervision on what an unknown instance is. It is infeasible to manually annotate even a small subset of the potentially infinite set of unknown classes. To counter this, we propose an auto-labelling mechanism based on the Region Proposal Network [54] to pseudo-label unknown instances, as explained in Sec. 4.2. The inherent separation of auto-labelled unknown instances in the latent space helps our energy based classification head to differentiate between the known and unknown instances. As elucidated in Sec. 4.3, we find that Helmholtz free energy is higher for unknown instances.

Fig. 2 shows the high-level architectural overview of ORE. We choose Faster R-CNN [54] as the base detector as Dhamija *et al*. [8] has found that it has better open set performance when compared against one-stage RetinaNet detector [31] and objectness based YOLO detector [52]. Faster R-CNN [54] is a two stage object detector. In the first stage, a class-agnostic Region Proposal Network (RPN) proposes potential regions which might have an object from the feature maps coming from a shared backbone network. The second stage classifies and adjusts the bounding box coordinates of each of the proposed region. The features that are generated by the residual block in the Region of Interest (RoI) head are contrastively clustered. The RPN and the classification head is adapted to auto-label and identify unknowns respectively. We explain each of these coherent constituent components, in the following subsections:

4.1. Contrastive Clustering

Class separation in the latent space would be an ideal characteristic for an Open World methodology to identify unknowns. A natural way to enforce this would be to model it as a contrastive clustering problem, where instances of same class would be forced to remain close-by, while instances of dissimilar class would be pushed far apart.

For each known class $i \in \mathcal{K}^t$, we maintain a prototype vector $\boldsymbol{p}_i$. Let $\boldsymbol{f}_c \in \mathbb{R}^d$ be a feature vector that is generated by an intermediate layer of the object detector, for an object of class c. We define the contrastive loss as follows:

$$\mathcal{L}_{cont}(\boldsymbol{f}_c) = \sum_{i=0}^{\mathrm{C}} \ell(\boldsymbol{f}_c, \boldsymbol{p}_i), \text{ where,} \tag{1}$$

$$\ell(\boldsymbol{f}_c, \boldsymbol{p}_i) = \begin{cases} \mathcal{D}(\boldsymbol{f}_c, \boldsymbol{p}_i) & i = c \\ \max\{0, \Delta - \mathcal{D}(\boldsymbol{f}_c, \boldsymbol{p}_i)\} & \text{otherwise} \end{cases}$$

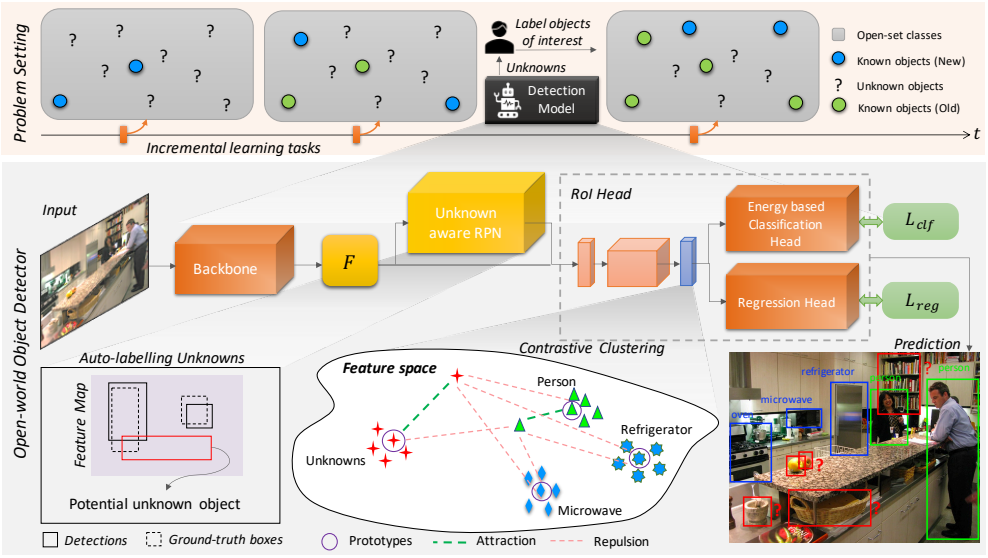


Figure 2: *Approach Overview: Top row:* At each incremental learning step, the model identifies unknown objects (denoted by '?'), which are progressively labelled (as blue circles) and added to the existing knowledge base (green circles). *Bottom row:* Our open world object detection model identifies potential unknown objects using an energy-based classification head and the unknown-aware RPN. Further, we perform contrastive learning in the feature space to learn discriminative clusters and can flexibly add new classes in a continual manner without forgetting the previous classes.

where $\mathcal{D}$ is any distance function and Δ defines how close a similar and dissimilar item can be. Minimizing this loss would ensure the desired class separation in the latent space.

Mean of feature vectors corresponding to each class is used to create the set of class prototypes: $\mathcal{P} = \{\boldsymbol{p}_0 \cdots \boldsymbol{p}_\mathrm{C}\}$. Maintaining each prototype vector is a crucial component of ORE. As the whole network is trained end-to-end, the class prototypes should also gradually evolve, as the constituent features change gradually (as stochastic gradient descent updates weights by a small step in each iteration). We maintain a fixed-length queue $\boldsymbol{q}_i$, per class for storing the corresponding features. A feature store $\mathcal{F}_{store} = \{\boldsymbol{q}_0 \cdots \boldsymbol{q}_\mathrm{C}\}$, stores the class specific features in the corresponding queues. This is a scalable approach for keeping track of how the feature vectors evolve with training, as the number of feature vectors that are stored is bounded by $\mathrm{C} \times \mathrm{Q}$, where Q is the maximum size of the queue.

Algorithm 1 provides an overview on how class prototypes are managed while computing the clustering loss. We start computing the loss only after a certain number of burn-in iterations (I_b) are completed. This allows the initial feature embeddings to mature themselves to encode class information. Since then, we compute the clustering loss using Eqn. 1. After every I_p iterations, a set of new class prototypes $\mathcal{P}_{new}$ is computed (line 8). Then the existing prototypes $\mathcal{P}$ are updated by weighing $\mathcal{P}$ and $\mathcal{P}_{new}$ with a momentum parameter η. This allows the class prototypes to evolve gradually keeping track of previous context. The computed clustering loss is added to the standard detection loss and back-propagated to learn the network end-to-end.

4.2. Auto-labelling Unknowns with RPN

While computing the clustering loss with Eqn. 1, we contrast the input feature vector $\boldsymbol{f}_c$ against prototype vectors, which include a prototype for unknown objects too ($c \in \{0, 1, .., \mathrm{C}\}$ where 0 refers to the unknown class). This would require unknown object instances to be labelled with `unknown` ground truth class, which is not practically feasible owing to the arduous task of re-annotating all instances of each image in already annotated large-scale datasets.

Algorithm 1 Algorithm COMPUTECLUSTERINGLOSS

Input: Input feature for which loss is computed: $\boldsymbol{f}_c$; Feature store: $\mathcal{F}_{store}$; Current iteration: i; Class prototypes: $\mathcal{P} = \{\boldsymbol{p}_0 \cdots \boldsymbol{p}_\mathrm{C}\}$; Momentum parameter: η.

1: Initialise $\mathcal{P}$ if it is the first iteration.
2: $\mathcal{L}_{cont} \leftarrow 0$
3: **if** $i == I_b$ **then**
4: $\quad \mathcal{P} \leftarrow$ class-wise mean of items in $\mathcal{F}_{Store}$.
5: $\quad \mathcal{L}_{cont} \leftarrow$ Compute using $\boldsymbol{f}_c$, $\mathcal{P}$ and Eqn. 1.
6: **else if** $i > I_b$ **then**
7: $\quad$ **if** $i \% I_p == 0$ **then**
8: $\quad\quad \mathcal{P}_{new} \leftarrow$ class-wise mean of items in $\mathcal{F}_{Store}$.
9: $\quad\quad \mathcal{P} \leftarrow \eta\mathcal{P} + (1-\eta)\mathcal{P}_{new}$
10: $\quad \mathcal{L}_{cont} \leftarrow$ Compute using $\boldsymbol{f}_c$, $\mathcal{P}$ and Eqn. 1.
11: **return** $\mathcal{L}_{cont}$

As a surrogate, we propose to automatically label some of the objects in the image as a potential unknown object. For this, we rely on the fact that Region Proposal Network (RPN) is class agnostic. Given an input image, the RPN generates a set of bounding box predictions for foreground and background instances, along with the corresponding objectness scores. We label those proposals that have high objectness score, but do not overlap with a ground-truth object as a potential unknown object. Simply put, we select the top-k background region proposals, sorted by its objectness scores, as unknown objects. This seemingly simple heuristic achieves good performance as demonstrated in Sec. 5.

4.3. Energy Based Unknown Identifier

Given the features ($\boldsymbol{f} \in F$) in the latent space F and their corresponding labels $l \in L$, we seek to learn an energy function $E(F, L)$. Our formulation is based on the Energy based models (EBMs) [27] that learn a function $E(\cdot)$ to estimates the compatibility between observed variables F and possible set of output variables L using a single output scalar i.e., $E(\boldsymbol{f}) : \mathbb{R}^d \rightarrow \mathbb{R}$. The intrinsic capability of EBMs to assign low energy values to in-distribution data and vice-versa motivates us to use an energy measure to characterize whether a sample is from an unknown class.

Specifically, we use the Helmholtz free energy formulation where energies for all values in L are combined,

$$E(\boldsymbol{f}) = -T \log \int_{l'} \exp\left(-\frac{E(\boldsymbol{f}, l')}{T}\right), \tag{2}$$

where T is the temperature parameter. There exists a simple relation between the network outputs after the softmax layer and the Gibbs distribution of class specific energy values [34]. This can be formulated as,

$$p(l|\boldsymbol{f}) = \frac{\exp(\frac{g_l(\boldsymbol{f})}{T})}{\sum_{i=1}^{\mathrm{C}} \exp(\frac{g_i(\boldsymbol{f})}{T})} = \frac{\exp(-\frac{E(\boldsymbol{f},l)}{T})}{\exp(-\frac{E(\boldsymbol{f})}{T})} \tag{3}$$

where $p(l|\boldsymbol{f})$ is the probability density for a label l, $g_l(\boldsymbol{f})$ is the l^{th} classification logit of the classification head $g(.)$. Using this correspondence, we define free energy of our classification models in terms of their logits as follows:

$$E(\boldsymbol{f}; g) = -T \log \sum_{i=1}^{\mathrm{C}} \exp(\frac{g_i(\boldsymbol{f})}{T}). \tag{4}$$

The above equation provides us a natural way to transform the classification head of the standard Faster R-CNN [54] to an energy function. Due to the clear separation that we enforce in the latent space with the contrastive clustering, we see a clear separation in the energy level of the known class data-points and unknown data-points as illustrated in Fig. 3. In light of this trend, we model the energy distribution of the known and unknown energy values $\xi_{kn}(\boldsymbol{f})$ and $\xi_{unk}(\boldsymbol{f})$, with a set of shifted Weibull distributions. These distributions were found to fit the energy data of a small held out validation set (with both knowns and unknowns instances) very well, when compared to Gamma, Exponential and Normal distributions. The learned distributions can be used to label a prediction as unknown if $\xi_{kn}(\boldsymbol{f}) < \xi_{unk}(\boldsymbol{f})$.

4.4. Alleviating Forgetting

After the identification of unknowns, an important requisite for an open world detector is to be able to learn new classes, when the labeled examples of some of the unknown classes of interest are provided. Importantly, the training data for the previous tasks will not be present at this stage since retraining from scratch is not a feasible solution. Training with only the new class instances will lead to catastrophic forgetting [40, 11] of the previous classes. We note that a number of involved approaches have been developed to alleviate such forgetting, including methods based on parameter regularization [2, 24, 29, 66], exemplar replay [6, 51, 37, 5], dynamically expanding networks [39, 60, 56] and meta-learning [50, 25].

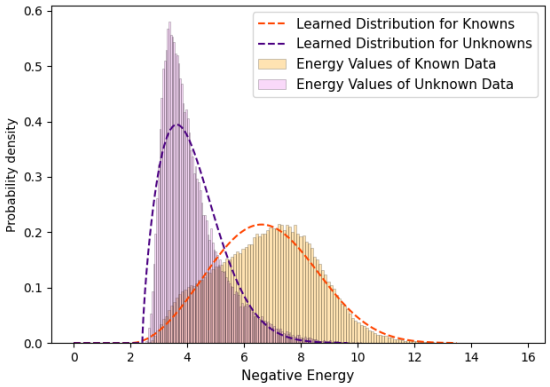


Figure 3: The energy values of the known and unknown datapoints exhibit clear separation as seen above. We fit a Weibull distribution on each of them and use these for identifying unseen known and unknown samples, as explained in Sec. 4.3.

We build on the recent insights from [49, 26, 62] which compare the importance of example replay against other more complex solutions. Specifically, Prabhu *et al*. [49] retrospects the progress made by the complex continual learning methodologies and show that a greedy exemplar selection strategy for replay in incremental learning consistently outperforms the state-of-the-art methods by a large margin. Knoblauch *et al*. [26] develops a theoretical justification for the unwarranted power of replay methods. They prove that an optimal continual learner solves an NP-hard problem and requires infinite memory. The effectiveness of storing few examples and replaying has been found effective in the related few-shot object detection setting by Wang *et al*. [62]. These motivates us to use a relatively simple methodology for ORE to mitigate forgetting i.e., we store a balanced set of exemplars and finetune the model after each incremental step on these. At each point, we ensure that a minimum of N_{ex} instances for each class are present in the exemplar set.

5. Experiments and Results

We propose a comprehensive evaluation protocol to study the performance of an open world detector to identify unknowns, detect known classes and progressively learn new classes when labels are provided for some unknowns.

	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
# test images	4952	1914	1642	1738
# train instances	47223	113741	114452	138996
# test instances	14976	4966	4826	6039

Table 1: The table shows task composition in the proposed Open World evaluation protocol. The semantics of each task and the number of images and instances (objects) across splits are shown.

5.1. Open World Evaluation Protocol

Data split: We group classes into a set of tasks $\mathcal{T} = \{T_1, \cdots T_t, \cdots\}$. All the classes of a specific task will be introduced to the system at a point of time t. While learning T_t, all the classes of $\{T_\tau : \tau{<}t\}$ will be treated as known and $\{T_\tau : \tau{>}t\}$ would be treated as unknown. For a concrete instantiation of this protocol, we consider classes from Pascal VOC [10] and MS-COCO [32]. We group all VOC classes and data as the first task T_1. The remaining 60 classes of MS-COCO [32] are grouped into three successive tasks with semantic drifts (see Tab. 1). All images which correspond to the above split from Pascal VOC and MS-COCO train-sets form the training data. For evaluation, we use the Pascal VOC test split and MS-COCO val split. 1k images from training data of each task is kept aside for validation. Data splits and codes can be found at https://github.com/JosephKJ/OWOD.

Evaluation metrics: Since an unknown object easily gets confused as a known object, we use the Wilderness Impact (WI) metric [8] to explicitly characterises this behaviour.

$$\text{Wilderness Impact}\,(WI) = \frac{P_{\mathcal{K}}}{P_{\mathcal{K}\cup\mathcal{U}}} - 1, \qquad (5)$$

where $P_{\mathcal{K}}$ refers to the precision of the model when evaluated on known classes and $P_{\mathcal{K}\cup\mathcal{U}}$ is the precision when evaluated on known and unknown classes, measured at a recall level R (0.8 in all experiments). Ideally, WI should be less as the precision must not drop when unknown objects are added to the test set. Besides WI, we also use Absolute Open-Set Error (A-OSE) [43] to report the number count of unknown objects that get wrongly classified as any of the known class. Both WI and A-OSE implicitly measure how effective the model is in handling unknown objects.

In order to quantify incremental learning capability of the model in the presence of new labeled classes, we measure the mean Average Precision (mAP) at IoU threshold of 0.5 (consistent with the existing literature [61, 45]).

5.2. Implementation Details

ORE re-purposes the standard Faster R-CNN [54] object detector with a ResNet-50 [20] backbone. To handle variable number of classes in the classification head, following incremental classification methods [50, 25, 6, 37], we assume a bound on the maximum number of classes to expect, and modify the loss to take into account only the classes of interest. This is done by setting the classification logits of the unseen classes to a large negative value (v), thus making their contribution to softmax negligible ($e^{-v} \rightarrow 0$).

The 2048-dim feature vector which comes from the last residual block in the RoI Head is used for contrastive clustering. The contrastive loss (defined in Eqn. 1) is added to the standard Faster R-CNN classification and localization losses and jointly optimised for. While learning a task T_i, only the classes that are part of T_i will be labelled. While testing T_i, all the classes that were previously introduced are labelled along with classes in T_i, and all classes of future tasks will be labelled '`unknown`'. For the exemplar replay, we empirically choose $N_{ex} = 50$. We do a sensitivity analysis on the size of the exemplar memory in Sec. 6. Further implementation details are provided in supplementary.

5.3. Open World Object Detection Results

Table 2 shows how ORE compares against Faster R-CNN on the proposed open world evaluation protocol. An 'Oracle' detector has access to all known and unknown labels at any point, and serves as a reference. After learning each task, WI and A-OSE metrics are used to quantify how unknown instances are confused with any of the known classes. We see that ORE has significantly lower WI and A-OSE scores, owing to an explicit modeling of the unknown. When unknown classes are progressively labelled in Task 2, we see that the performance of the baseline detector on the known set of classes (quantified via mAP) significantly deteriorates from 56.16% to 4.076%. The proposed balanced finetuning is able to restore the previous class performance to a respectable level (51.09%) at the cost of increased WI and A-OSE, whereas ORE is able to achieve both goals: detect known classes and reduce the effect of unknown comprehensively. Similar trend is seen when Task 3 classes are added. WI and A-OSE scores cannot be measured for Task 4 because of the absence of any unknown ground-truths. We report qualitative results in Fig. 4 and supplementary section, along with failure case analysis. We conduct extensive sensitivity analysis in Sec. 6 and supplementary section.

5.4. Incremental Object Detection Results

We find an interesting consequence of the ability of ORE to distinctly model unknown objects: it performs favorably well on the incremental object detection (iOD) task against the state-of-the-art (Tab. 3). This is because, ORE reduces the confusion of an unknown object being classified as a known object, which lets the detector incrementally learn the true foreground objects. We use the standard protocol [61, 45] used in the iOD domain to evaluate ORE, where group of classes (10, 5 and the last class) from Pascal VOC

Task IDs (→)	Task 1			Task 2					Task 3					Task 4		
	WI	A-OSE	mAP (↑)	WI	A-OSE	mAP (↑)			WI	A-OSE	mAP (↑)			mAP (↑)		
	(↓)	(↓)	Current known	(↓)	(↓)	Previously known	Current known	Both	(↓)	(↓)	Previously known	Current known	Both	Previously known	Current known	Both
Oracle	0.02004	7080	57.76	0.0066	6717	54.99	30.31	42.65	0.0038	4237	40.23	21.51	30.87	32.52	19.27	31.71
Faster-RCNN	0.06991	13396	56.16	0.0371	12291	4.076	25.74	14.91	0.0213	9174	6.96	13.481	9.138	2.04	13.68	4.95
Faster-RCNN + Finetuning	Not applicable as incremental component is not present in Task 1			0.0375	12497	51.09	23.84	37.47	0.0279	9622	35.69	11.53	27.64	29.53	12.78	25.34
ORE	**0.02193**	**8234**	**56.34**	**0.0154**	**7772**	52.37	25.58	**38.98**	**0.0081**	**6634**	37.77	12.41	**29.32**	30.01	13.44	**26.66**

Table 2: Here we showcase how ORE performs on Open World Object Detection. Wilderness Impact (WI) and Average Open Set Error (A-OSE) quantify how ORE handles the unknown classes (gray background), whereas Mean Average Precision (mAP) measures how well it detects the known classes (white background). We see that ORE consistently outperforms the Faster R-CNN based baseline on all the metrics. Kindly refer to Sec. 5.3 for more detailed analysis and explanation for the evaluation metrics.

10 + 10 setting	aero	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv	mAP
All 20	68.5	77.2	74.2	55.6	59.7	76.5	83.1	81.5	52.1	79.8	55.1	80.9	80.1	76.8	80.5	47.1	73.1	61.2	76.9	70.3	70.51
First 10	79.3	79.7	70.2	56.4	62.4	79.6	88.6	76.6	50.1	68.9	0	0	0	0	0	0	0	0	0	0	35.59
New 10	7.9	0.3	5.1	3.4	0	0	0.2	2.3	0.1	3.3	65	69.3	81.3	76.4	83.1	47.2	67.1	68.4	76.5	69.2	36.31
ILOD [61]	69.9	70.4	69.4	54.3	48	68.7	78.9	68.4	45.5	58.1	59.7	72.7	73.5	73.2	66.3	29.5	63.4	61.6	69.3	62.2	63.15
ILOD + Faster R-CNN	70.5	75.6	68.9	59.1	56.6	67.6	78.6	75.4	50.3	70.8	43.2	68.1	66.2	65.1	66.5	24.3	61.3	46.6	58.1	49.9	61.14
Faster ILOD [45]	72.8	75.7	71.2	60.5	61.7	70.4	83.3	76.6	53.1	72.3	36.7	70.9	66.8	67.6	66.1	24.7	63.1	48.1	57.1	43.6	62.16
ORE - (CC + EBUI)	53.3	69.2	62.4	51.8	52.9	73.6	83.7	71.7	42.8	66.8	46.8	59.9	65.5	66.1	68.6	29.8	55.1	51.6	65.3	51.5	59.42
ORE	63.5	70.9	58.9	42.9	34.1	76.2	80.7	76.3	34.1	66.1	56.1	70.4	80.2	72.3	81.8	42.7	71.6	68.1	77	67.7	**64.58**
15 + 5 setting	aero	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv	mAP
First 15	74.2	79.1	71.3	60.3	60	80.2	88.1	80.2	48.8	74.6	61	76	85.3	78.2	83.4	0	0	0	0	0	55.03
New 5	3.7	0.5	6.3	4.6	0.9	0	8.8	3.9	0	0.4	0	0	16.4	0.7	0	41	55.7	49.2	59.1	67.8	15.95
ILOD [61]	70.5	79.2	68.8	59.1	53.2	75.4	79.4	78.8	46.6	59.4	59	75.8	71.8	78.6	69.6	33.7	61.5	63.1	71.7	62.2	65.87
ILOD + Faster R-CNN	63.5	76.3	70.7	53.1	55.8	67.1	81.5	80.3	49.6	73.8	62.1	77.1	79.7	74.2	73.9	37.1	59.1	61.7	68.6	61.3	66.35
Faster ILOD [45]	66.5	78.1	71.8	54.6	61.4	68.4	82.6	82.7	52.1	74.3	63.1	78.6	80.5	78.4	80.4	36.7	61.7	59.3	67.9	59.1	67.94
ORE - (CC + EBUI)	65.1	74.6	57.9	39.5	36.7	75.1	80	73.3	37.1	69.8	48.8	69	77.5	72.8	76.5	34.4	62.6	56.5	80.3	65.7	62.66
ORE	75.4	81	67.1	51.9	55.7	77.2	85.6	81.7	46.1	76.2	55.4	76.7	86.2	78.5	82.1	32.8	63.6	54.7	77.7	64.6	**68.51**
19 + 1 setting	aero	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv	mAP
First 19	77.8	81.7	69.3	51.6	55.3	74.5	86.3	80.2	49.3	82	63.6	76.8	80.9	77.5	82.4	42.9	73.9	70.4	70.4	0	67.34
Last 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64	3.2
ILOD [61]	69.4	79.3	69.5	57.4	45.4	78.4	79.1	80.5	45.7	76.3	64.8	77.2	80.8	77.5	70.1	42.3	67.5	64.4	76.7	62.7	68.25
ILOD + Faster R-CNN	60.9	74.6	70.8	56	51.3	70.7	81.7	81.5	49.45	78.3	58.3	79.5	79.1	74.8	75.7	42.8	74.7	61.2	67.2	65.1	67.72
Faster ILOD [45]	64.2	74.7	73.2	55.5	53.7	70.8	82.9	82.6	51.6	79.7	58.7	78.8	81.8	75.3	77.4	43.1	73.8	61.7	69.8	61.1	68.56
ORE - (CC + EBUI)	60.7	78.6	61.8	45	43.2	75.1	82.5	75.5	42.4	75.1	56.7	72.9	80.8	75.4	77.7	37.8	72.3	64.5	70.7	49.9	64.93
ORE	67.3	76.8	60	48.4	58.8	81.1	86.5	75.8	41.5	79.6	54.6	72.8	85.9	81.7	82.4	44.8	75.8	68.2	75.7	60.1	**68.89**

Table 3: We compare ORE against state-of-the-art incremental Object Detectors on three different settings. 10, 5 and the last class from the Pascal VOC 2007 [10] dataset are introduced to a detector trained on 10, 15 and 19 classes respectively (shown in blue background). ORE is able to perform favourably on all the settings with no methodological change. Kindly refer to Sec. 5.4 for more details.

2007 [10] are incrementally learned by a detector trained on the remaining set of classes. Remarkably, ORE is used as it is, without any change to the methodology introduced in Sec. 4. We ablate contrastive clustering (CC) and energy based unknown identification (EBUI) to find that it results in reduced performance than standard ORE.

6. Discussions and Analysis

6.1 Ablating ORE Components: To study the contribution of each of the components in ORE, we design careful ablation experiments (Tab. 4). We consider the setting where Task 1 is introduced to the model. The auto-labelling methodology (referred to as ALU), combined with energy based unknown identification (EBUI) performs better together (row 5) than using either of them separately (row 3 and 4). Adding contrastive clustering (CC) to this configuration, gives the best performance in handling unknown (row 7), measured in terms of WI and A-OSE. There is no severe performance drop in known classes detection (mAP metric) as a side effect of unknown identification. In row 6, we see that EBUI is a critical component whose absence increases WI and A-OSE scores. Thus, each component in ORE has a critical role to play for unknown identification.

Row ID	CC	ALU	EBUI	WI (↓)	A-OSE (↓)	mAP (↑)
1	Oracle			0.02004	7080	57.76
2	×	×	×	0.06991	13396	56.16
3	×	×	✓	0.05932	12822	56.21
4	×	✓	×	0.05542	12111	56.09
5	×	✓	✓	0.04539	9011	55.95
6	✓	✓	×	0.05614	12064	**56.36**
7	✓	✓	✓	**0.02193**	**8234**	56.34

Table 4: We carefully ablate each of the constituent component of ORE. CC, ALU and EBUI refers to 'Contrastive Clustering', 'Auto-labelling of Unknowns' and 'Energy Based Unknown Identifier' respectively. Kindly refer to Sec. 6.1 for more details.

N_{ex}	WI	A-OSE	mAP (↑)		
	(↓)	(↓)	Previously known	Current known	Both
0	0.0406	9268	8.74	26.81	17.77
10	0.0237	8211	46.78	24.32	35.55
20	0.0202	8092	48.83	25.42	37.13
50	0.0154	7772	52.37	25.58	38.98
100	0.0410	11065	52.29	26.21	39.24
200	0.0385	10474	53.41	26.35	39.88
400	0.0396	11461	53.18	26.09	39.64

Table 5: The table shows sensitivity analysis. Increasing N_{ex} by a large value hurts performance on unknown, while a small set of images are essential to mitigate forgetting (best row in green).

6.2 Sensitivity Analysis on Exemplar Memory Size: Our balanced finetuning strategy requires storing exemplar images with at least N_{ex} instances per class. We vary N_{ex} while learning Task 2 and report the results in Table 5. We find that balanced finetuning is very effective in improving the accuracy of previously known class, even with just having minimum 10 instances per class. However, we find that increasing N_{ex} to large values does-not help and at the same time adversely affect how unknowns are handled (evident from WI and A-OSE scores). Hence, by validation, we set N_{ex} to 50 in all our experiments, which is a sweet spot that balances performance on known and unknown classes.

6.3 Comparison with an Open Set Detector: The mAP values of the detector when it is evaluated on closed set data (trained and tested on Pascal VOC 2007) and open set data (test set contains equal number of unknown images from MS-COCO) helps to measure how the detector handles unknown instances. Ideally, there should not be a performance drop. We compare ORE against the recent open set detector proposed by Miller *et al*. [43]. We find from Tab. 6 that drop in performance of ORE is much lower than [43] owing to the effective modelling of the unknown instances.

6.4 Clustering loss and t-SNE [38] visualization: We visualise the quality of clusters that are formed while training with the contrastive clustering loss (Eqn. 1) for Task 1. We see nicely formed clusters in Fig. 5 (a). Each number in the legend correspond to the 20 classes introduced in Task 1. Label 20 denotes unknown class. Importantly, we see

Evaluated on →	VOC 2007	VOC 2007 + COCO (WR1)
Standard Faster R-CNN	81.86	77.09
Dropout Sampling [43]	78.15	71.07
ORE	**81.31**	**78.16**

Table 6: Performance comparison with an Open Set object detector. ORE is able to reduce the fall in mAP values considerably.

Figure 4: Predictions from ORE after being trained on Task 1. 'elephant', 'apple', 'banana', 'zebra' and 'giraffe' have not been introduced to the model, and hence are successfully classified as 'unknown'. The approach misclassifies one of the 'giraffe' as a 'horse', showing the limitation of ORE.

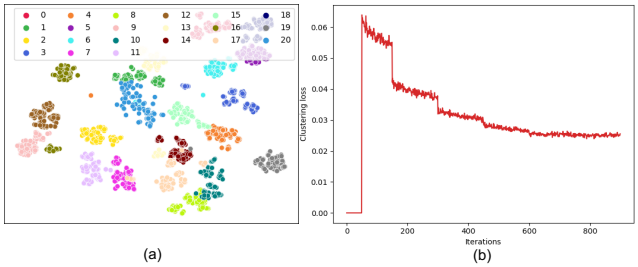


Figure 5: (a) Distinct clusters in the latent space. (b) Our contrastive loss which ensures such a clustering steadily converges.

that the unknown instances also gets clustered, which reinforces the quality of the auto-labelled unknowns used in contrastive clustering. In Fig. 5 (b), we plot the contrastive clustering loss against training iterations, where we see a gradual decrease, indicative of good convergence.

7. Conclusion

The vibrant object detection community has pushed the performance benchmarks on standard datasets by a large margin. The closed-set nature of these datasets and evaluation protocols, hampers further progress. We introduce Open World Object Detection, where the object detector is able to label an unknown object as unknown and gradually learn the unknown as the model gets exposed to new labels. Our key novelties include an energy-based classifier for unknown detection and a contrastive clustering approach for open world learning. We hope that our work will kindle further research along this important and open direction.

Acknowledgements

We thank TCS for supporting KJJ through its PhD fellowship; MBZUAI for a start-up grant; VR starting grant (2016-05543) and DST, Govt of India, for partly supporting this work through IMPRINT program (IMP/2019/000250). We thank our anonymous reviewers for their valuable feedback.

References

[1] Manoj Acharya, Tyler L. Hayes, and Christopher Kanan. Rodeo: Replay for online object detection. In *The British Machine Vision Conference*, 2020. 13

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018. 5

[3] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902, 2015. 1, 2

[4] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016. 2

[5] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248, 2018. 5

[6] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019. 5, 6

[7] Li Chen, Chunyan Yu, and Lvcai Chen. A new knowledge distillation for incremental object detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019. 13

[8] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1021–1030, 2020. 2, 3, 6

[9] Susan Engel. Children's need to know: Curiosity in schools. *Harvard educational review*, 81(4):625–645, 2011. 1

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 1, 6, 7

[11] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 5

[12] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016. 2

[13] Zongyuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. In *British Machine Vision Conference 2017*. British Machine Vision Association and Society for Pattern Recognition, 2017. 2

[14] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1, 2

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[16] Brian Grazer and Charles Fishman. *A curious mind: The secret to a bigger life*. Simon and Schuster, 2016. 1

[17] David Hall, Feras Dayoub, John Skinner, Haoyang Zhang, Dimity Miller, Peter Corke, Gustavo Carneiro, Anelia Angelova, and Niko Sünderhauf. Probabilistic object detection: Definition and evaluation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1031–1040, 2020. 2

[18] Yu Hao, Yanwei Fu, Yu-Gang Jiang, and Qi Tian. An end-to-end architecture for class-incremental object detection with knowledge distillation. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2019. 13

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[21] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 13

[22] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. 3

[23] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*, pages 393–409. Springer, 2014. 2

[24] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 5

[25] Joseph KJ and Vineeth Nallure Balasubramanian. Meta-consolidation for continual learning. *Advances in Neural Information Processing Systems*, 33, 2020. 5, 6

[26] Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is np-hard. *arXiv preprint arXiv:2006.05188*, 2020. 5

[27] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 5

[28] Dawei Li, Serafettin Tasci, Shalini Ghosh, Jingwen Zhu, Junting Zhang, and Larry Heck. Rilod: near real-time incremental learning for object detection at the edge. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 113–126, 2019. 13

[29] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. 5

[30] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. 2

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1, 2, 3

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 1, 6

[33] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[34] Weitang Liu, Xiaoyun Wang, John Owens, and Sharon Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33, 2020. 5

[35] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 2

[36] Mario Livio. *Why?: What makes us curious*. Simon and Schuster, 2017. 1

[37] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017. 5, 6

[38] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 8

[39] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 5

[40] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 5

[41] John A Meacham. Wisdom and the context of knowledge: Knowing that one doesn't know. *On the development of developmental psychology*, 8:111–134, 1983. 1

[42] Dimity Miller, Feras Dayoub, Michael Milford, and Niko Sünderhauf. Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2348–2354. IEEE, 2019. 2

[43] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018. 2, 6, 8

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 13

[45] Can Peng, Kun Zhao, and Brian C. Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern Recognition Letters*, 140:109 – 115, 2020. 6, 7, 13

[46] Pramuditha Perera, Vlad I. Morariu, Rajiv Jain, Varun Manjunatha, Curtis Wigington, Vicente Ordonez, and Vishal M. Patel. Generative-discriminative feature representations for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[47] Federico Pernici, Federico Bartoli, Matteo Bruni, and Alberto Del Bimbo. Memory based online learning of deep representations from video streams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2324–2334, 2018. 2

[48] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In *Advances in neural information processing systems*, pages 6822–6833, 2018. 2

[49] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European Conference on Computer Vision*, pages 524–540. Springer, 2020. 5

[50] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. itaml: An incremental task-agnostic meta-learning approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13588–13597, 2020. 5, 6

[51] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542. IEEE, 2017. 5

[52] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 3

[53] Matthew Reid. Matthewreid854/reliability: v0.5.4, Nov. 2020. 13

[54] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 3, 5, 6

[55] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 2

[56] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 5

[57] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2012. 1

[58] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(35):1757–1772, 2013. 2

[59] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324, 2014. 2

[60] Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. *arXiv preprint arXiv:1801.01423*, 2018. 5

[61] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3400–3409, 2017. 6, 7, 13

[62] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020. 5

[63] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019. 13

[64] Hu Xu, Bing Liu, Lei Shu, and P Yu. Open-world learning and application to product classification. In *The World Wide Web Conference*, pages 3413–3419, 2019. 2

[65] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[66] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org, 2017. 5

Supplementary Material

In this supplementary material, we provide additional details which we could not include in the main paper due to space constraints, including experimental analysis, implementation details, discussion and results that help us develop further insights to the proposed Open World Object Detection approach. We discuss:

- Sensitivity analysis on queue size of Feature Store, the momentum parameter η, margin in clustering loss Δ and temperature parameter in energy computation.
- Additional details on contrastive clustering
- More specific implementation details.
- Discussion regarding failure cases.
- Related works in incremental object detection.
- Some qualitative results of ORE.

A. Varying the Queue Size of $\mathcal{F}_{Store}$

In Sec. 4.1, we explain how class specific queues $\boldsymbol{q}_i$ are used to store the feature vectors, which are used to compute the class prototypes. A hyper-parameter Q controls the size of each $\boldsymbol{q}_i$. Here we vary Q, while learning Task 1, and report the results in Tab. 7. We observe relatively similar performance, across experiments with different Q values. This can be attributed to the fact that after a prototype is defined, it gets periodically updated with newly observed features, thus effectively evolving itself. Hence, the actual number of features used to compute those prototypes ($\mathcal{P}$ and $\mathcal{P}_{new}$) is not very significant. We use Q $= 20$ for all the experiments.

Q	WI (↓)	A-OSE (↓)	mAP (↑)
5	0.02402	8123	56.01
10	0.02523	8126	56.02
20	0.02193	8234	56.34
30	0.02688	8487	55.78
50	0.02623	8578	56.22

Table 7: We find that varying the number of features that are used to compute the class prototype does not have a huge impact on the performance.

B. Sensitivity Analysis on η

The momentum parameter η controls how rapidly the class prototypes are updated, as elaborated in Algorithm 1. Larger values of η imply smaller effect of the newly computed prototypes on the current class prototypes. We find from Tab. 8 that performance improves when prototypes are updated slowly (larger values of η). This result is intuitive, as slowly changing the cluster centers helps stabilize contrastive learning.

η	WI (↓)	A-OSE (↓)	mAP (↑)
0.4	0.05926	9476	55.96
0.6	0.04977	9095	55.56
0.8	0.02945	8375	55.73
0.9	0.02193	8234	56.34

Table 8: We see that higher values of η gives better performance, implying that gradual evolution of class prototypes improves contrastive clustering.

C. Varying the Margin (Δ) in $\mathcal{L}_{cont}$

The margin parameter Δ in the contrastive clustering loss $\mathcal{L}_{cont}$ (Eqn. 1) defines the minimum distance that an input feature vector should keep from dissimilar class prototypes in the latent space. As we see in Tab. 9, increasing the margin while learning the first task, increases the performance on the known classes and how unknown classes are handled. This would imply that larger separation in the latent space is beneficial for ORE.

Δ	WI (↓)	A-OSE (↓)	mAP (↑)
5	0.04094	9300	55.73
10	0.02193	8234	56.34
15	0.01049	8088	56.65

Table 9: Increasing the margin Δ, improves the performance on known and unknown classes, concurring with our assumption that separation in the latent space is beneficial for ORE.

D. Varying the Temperature (T) in Eqn. 4

We fixed the temperature parameter (T) in Eqn. 4 to 1 in all the experiments. Softening the energies a bit more to $T = 2$, gives slight improvement in unknown detection, however increasing it further hurts as evident from Tab. 10.

T	WI(↓)	A-OSE(↓)	mAP(↑)
1	0.0219	8234	56.34
2	0.0214	8057	55.68
3	0.0411	11266	55.51
5	0.0836	12063	56.25
10	0.0835	12064	56.31

Table 10: There is a nice ballpark for temperature parameter between $T = 1$ and $T = 2$, which gives the optimal performance.

E. More Details on Contrastive Clustering

The motivation for using contrastive clustering to ensure separation in the latent space is two-fold: 1) it enables the model to cluster unknowns separately from known instances, thus boosting unknown identification; 2) it ensures instances of each class are well-separated from other classes, alleviating the forgetting issue.

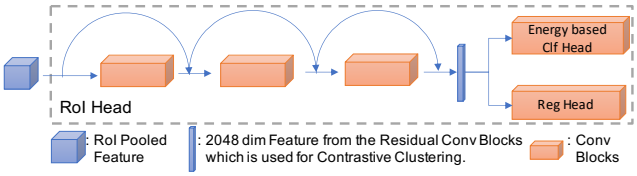


Figure 6: RoI head architecture, showing 2048-dim feature vector used for contrastive clustering.

The 2048-dim feature vector that comes out from residual blocks of RoI head (Fig 6) is contrastively clustered. The contrastive loss is added to the Faster R-CNN loss and the entire network is trained end-to-end. Thus all parts of the network before and including the residual block in the RoI head in the Faster R-CNN pipeline will get updated with the gradients from the contrastive clustering loss.

F. Further Implementation Details

We complete the discussion related to the implementation details, that we had in Sec. 5.2 here. We ran our experiments on a server with 8 Nvidia V100 GPUs with an effective batch size of 8. We use SGD with a learning rate of 0.01. Each task is learned for 8 epochs ($\sim$ 50k iterations). The queue size of the feature store is set to 20. We initiate clustering after 1k iterations and update the cluster prototypes after each 3k iterations with a momentum parameter of 0.99. Euclidean distance is used as the distance function $\mathcal{D}$ in Eqn. 1. The margin (Δ) is set as 10. For auto-labelling the unknowns in the RPN, we pick the top-1 background proposal, sorted by its objectness score. The temperature parameter in the energy based classification head is set to 1. The code is implemented in PyTorch [44] using Detectron 2 [63]. Reliability library [53] was used for modelling the energy distributions. We release all our codes publicly for foster reproducible research: https://github.com/JosephKJ/OWOD.

G. Related Work on Incremental Object Detection

The class-incremental object detection (iOD) setting considers classes to be observed incrementally over time and that the learner must adapt without retraining on old classes from scratch. The prevalent approaches [61, 28, 18, 7] use knowledge distillation [21] as a regularization measure to avoid forgetting old class information while training on new classes. Specifically, Shmelkov *et al*. [61] repurpose Fast R-CNN for incremental learning by distilling classification and regression outputs from a previous stage model. Beside distilling model outputs, Chen *et al*. [7] and Li *et al*. [28] also distilled the intermediate network features. Hao *et al*. [18] builds on Faster R-CNN and uses a student-teacher framework for RPN adaptation. Acharya *et al*. [1] proposes a replay mechanism for online detection. Recently, Peng *et al*. [45] introduces an adaptive distillation technique into Faster R-CNN. Their methodology is the current state-of-the-art in iOD. These methods cannot however work in an Open World environment, which is the focus of this work, and are unable to identify unknown objects.

H. Time and Storage Expense:

The training and inference of ORE takes an additional 0.1349 sec/iter and 0.009 sec/iter than standard Faster R-CNN. The storage expense for maintaining F_{Store} is negligible, and the exemplar memory (for $N_{ex} = 50$) takes approximately 34 MB.

I. Using Softmax based Unknown Identifier

We modified the unknown identification criteria to *max(softmax(logits))* $<$ t. For t = $\{0.3, 0.5, 0.7\}$: A-OSE, WI and mAP (mean and std-dev) are 11815 ± 352.13, 0.0436 ± 0.009 and 55.22 ± 0.02. This is inferior to ORE.

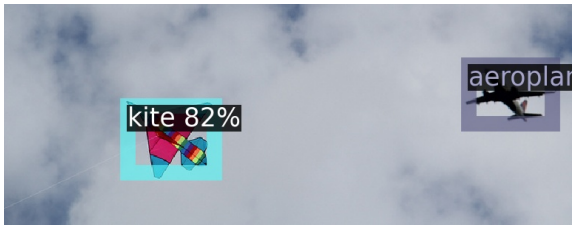
J. Qualitative Results

We show qualitative results of ORE in Fig. 8 through Fig. 13. We see that ORE is able to identify a variety of unknown instances and incrementally learn them, using the proposed contrastive clustering and energy-based unknown identification methodology. Sub-figure (a) in all these images shows the identified unknown instances along with the the other instances known to the detector. The corresponding sub-figure (b), shows the detections from the same detector after the new classes are incrementally added.

K. Discussion Regarding Failure Cases

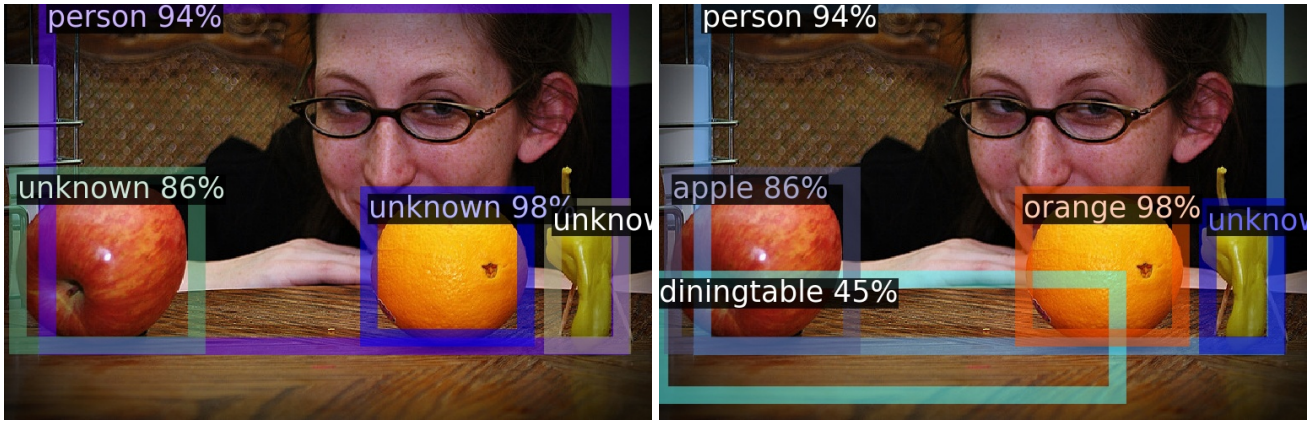
Occlusions and crowding of objects are cases where our method tends to get confused (`external-storage`, `walkman` and `bag` not detected as `unknown` in Figs. 11, 13). Difficult viewpoints (such as backside) also lead to some misclassifications (`giraffe`→`horse` in Figs. 4, 12). We have also noticed that detecting small *unknown* objects co-occurring with larger known objects is hard. As ORE is the first effort in this direction, we hope these identified shortcomings would be basis of further research.

(a)



(b)

Figure 7: ORE trained on just Task 1, successfully localises a `kite` as an unknown in sub-figure (a), while after learning about `kite` in Task 3, it incrementally learns to detect both `kite` and `aeroplane` in sub-figure (b).

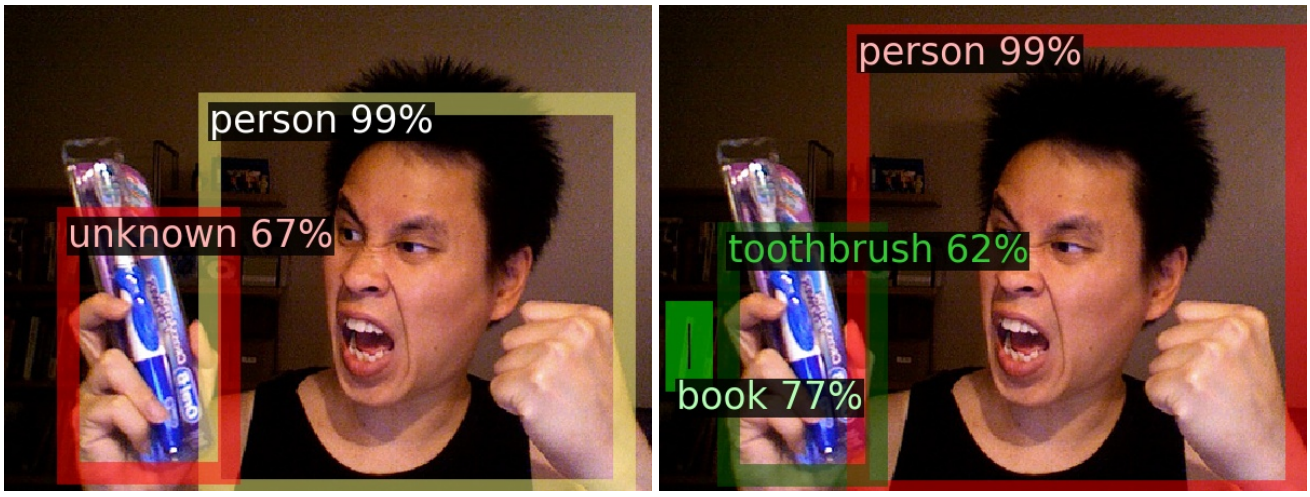


(a) (b)

Figure 8: The sub-figure (a) is the result produced by ORE after learning Task 2. As Task 3 classes like `apple` and `orange` has not been introduced, ORE identifies it and correctly labels them as `unknown`. After learning Task 3, these instances are labelled correctly in sub-figure (b). An unidentified class instance still remains, and ORE successfully detects it as `unknown`.

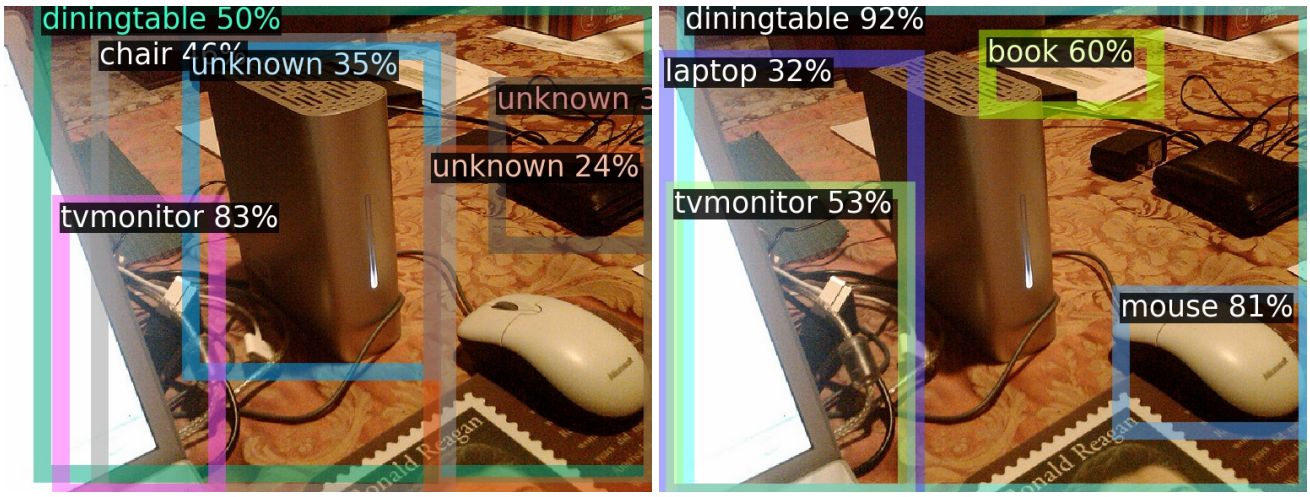
(a) (b)

Figure 9: The `clock` class is eventually learned as part of Task 4 (in sub-figure (b)), after being initially identified as `unknown` (in sub-figure (a)). ORE exhibits the true characteristics of an Open World detector, where it is able to incrementally learn an identified unknown.



(a) (b)

Figure 10: `toothbrush` and `book` are indoor objects introduced as part of Task 4. The detector trained till Task 3, identifies `toothbrush` as an unknown objects in sub-figure (a) and eventually learn it as part of Task 4, without forgetting how to identify `person` in sub-figure (b).

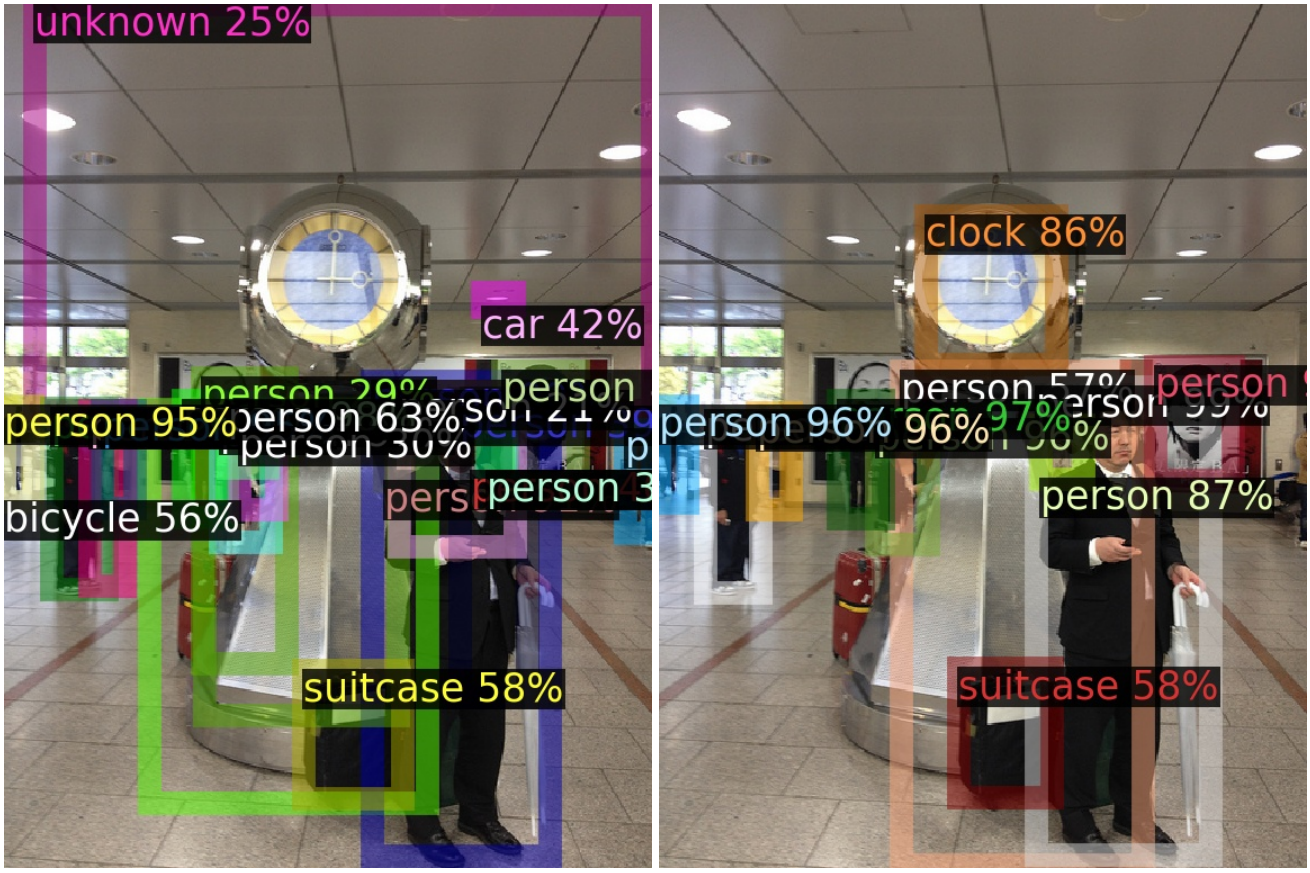


(a) (b)

Figure 11: Several items next to a `laptop` on top of a table are identified as `unknown`, after learning Task 1. `laptop`, `book` and `mouse` are introduced as part of Task 4, and hence are detected afterwords. `external-storage` and `walkman` (both are never introduced) were identified as `unknown` initially, but has not been detected after learning Task 4, and is one of the failure cases of ORE.

(a) (b)

Figure 12: `suitcase` which was identified as `unknown` is eventually learned in Task 2, along with a false positive detection of `chair`.



(a) (b)

Figure 13: In this highly cluttered scene, the `unknown` instance `clock` is identified, but is not localised well, after learning Task 2. After learning Task 4, ORE detects `clock`, along with reducing false positive detections of `car` and `bicycle`. The red `suitcase` is not labelled after learning either of the tasks, and hence is a failure case.