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. Introduction

Deep learning has accelerated progress in Object Detection research [13, 53, 18, 30, 51], 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 arise 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 [40, 35] finds out that the ability to identify what one doesn’t know, is key in captivating curiosity.

†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 [8, 15]. This motivates us to propose a new problem where a model should be able to identify instances of unknown objects as unknown and subsequently learn 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 [9] and MS-COCO [31] 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. [56] formalises 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. [2] extends 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. [7] finds that even with this extra training signal, the state-of-the-art object detectors results 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. [42] proposes 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. Fig. 1 shows a taxonomy of existing research work in this space.

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. [57] 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 [22, 58] 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 [3] 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 [12] by synthesizing novel class images. Liu et al. [34] 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 [29] or novelties [47]. Recently, self-supervised learning [45] and unsupervised learning with reconstruction [64] 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: [2] 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. [46] studies open world face identity learning while [63] 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. [7] 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 [54, 13, 32] and/or apply one-vs-rest classifiers to model each class [14, 30]. A dedicated body of work [42, 41, 16] focuses on developing measures of (spatial and semantic) uncertainty in object detectors to reject unknown classes. E.g., [42, 41] uses Monte Carlo Dropout [11] 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 $K_t^t = \{1, 2, \ldots, C\} \subset \mathbb{N}^+$ 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 $U = \{C + 1, \ldots\}$, which may be encountered during inference. The known object classes $K_t$ are assumed to be labeled in the dataset $\mathcal{D}^t = \{X^t, Y^t\}$ where $X$ and $Y$ denote the input images and labels respectively. The input image set comprises of $M$ training images, $X^t = \{I_1, \ldots, I_M\}$ and associated object labels for each image forms the label set $Y^t = \{Y_1, \ldots, Y_M\}$. Each $Y_i = \{y_1, y_2, \ldots, y_K\}$ encodes a set of $K$ object instances with their class labels and locations i.e., $y_k = [l_k, x_k, y_k, w_k, h_k]$, where $l_k \in K_t$ 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 $\mathcal{M}_C$ that is trained to detect all the previously encountered $C$ object classes. Importantly, the model $\mathcal{M}_C$ is able to identify a test instance by classifying it as an unknown, denoted by a label zero (0). The unknown set of instances $U^t$ 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 $\mathcal{M}_{C+n}$ without retraining from scratch on the whole dataset. The known class set is also updated $K_{t+1} = K_t + \{C + 1, \ldots, C + n\}$. 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 defray 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 [21], 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 [53] 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 [53] as the base detector as Dhamija et al. [7] has found that it has better open set performance when compared against one-stage RetinaNet detector [30] and objectness based YOLO detector [51]. Faster R-CNN [53] 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 $i \in K^t$, we maintain a prototype vector $p_i$. Let $f_i \in \mathbb{R}^d$ 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:

$$
\mathcal{L}_{cont}(f_c) = \sum_{i=0}^{C} \ell(f_c, p_i),
$$

where,

$$
\ell(f_c, p_i) = \begin{cases} 
\mathcal{D}(f_c, p_i) & i = c \\
\max\{0, \Delta - \mathcal{D}(f_c, p_i)\} & \text{otherwise}
\end{cases}
$$
where \( \mathcal{D} \) is any distance function and \( \Delta \) 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: \( \mathcal{P} = \{p_0 \cdots p_C\} \). 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 \( q_c \), per class for storing the corresponding features. A feature store \( \mathcal{F}_{\text{store}} = \{q_0 \cdots q_C\} \), 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 \( C \times Q \), 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 \( \mathcal{P}_{\text{new}} \) is computed (line 8). Then the existing prototypes \( \mathcal{P} \) are updated by weighing \( \mathcal{P} \) and \( \mathcal{P}_{\text{new}} \) with a momentum parameter \( \eta \). 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 \( f_c \) against prototype vec-

tors, which include a prototype for unknown objects too (\( c \in \{0, 1, \ldots, C\} \) 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.

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 \( \mathbf{f} \in F \) 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) [26] 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., \( E(\mathbf{f}) : \mathbb{R}^d \rightarrow \mathbb{R} \). 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,

\[
E(\mathbf{f}) = -T \log \int_L \exp \left( \frac{-E(\mathbf{f}, l)}{T} \right),
\]

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 [33]. This can be formulated as,

\[
p(l|\mathbf{f}) = \frac{\exp \left( \frac{E(\mathbf{f}, l)}{T} \right)}{\sum_{i=1}^C \exp \left( \frac{E(\mathbf{f}, l_i)}{T} \right)} = \frac{\exp(-E(\mathbf{f}, l))}{\exp(-E(\mathbf{f}, l))} \tag{3}
\]

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

\[
E(\mathbf{f}; g) = -T \log \sum_{i=1}^C \exp \left( \frac{g_i(\mathbf{f})}{T} \right). \tag{4}
\]

The above equation provides us a natural way to transform the classification head of the standard Faster R-CNN [53] 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 \( \xi_{kn}(\mathbf{f}) \) and \( \xi_{unk}(\mathbf{f}) \), with a set of shifted Weibull distributions. These distributions were found to fit the energy data of a held out validation set very well, when compared to Gamma, Exponential and Normal distributions. The learned distributions can be used to label a prediction as unknown if \( \xi_{kn}(\mathbf{f}) < \xi_{unk}(\mathbf{f}) \).

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 [39, 10] 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 [1, 23, 28, 65], exemplar replay [5, 50, 36, 4], dynamically expanding networks [38, 59, 55] and meta-learning [49, 24].

We build on the recent insights from [48, 25, 61] which compare the importance of example replay against other more complex solutions. Specifically, Prabhu et al. [48] retrospect 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. [25] 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. [61]. 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.

5.1. Open World Evaluation Protocol

Data split: We group classes into a set of tasks \( T = \{ T_1, \cdots, T_t, \cdots \} \). All the classes of a specific task will be introduced to the system at a point of time \( t \). While learn-
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 \cite{60, 44}).

5.2. Implementation Details

ORE re-purposes the standard Faster R-CNN \cite{53} object detector with a ResNet-50 \cite{19} backbone. To handle variable number of classes in the classification head, following incremental classification methods \cite{49, 24, 5, 36}, 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 \cite{60, 44} used in the iOD domain to evaluate ORE, where group of classes (10, 5 and the last class) from Pascal VOC 2007 \cite{9} 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.

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.

<table>
<thead>
<tr>
<th>Semantic split</th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOC Classes</td>
<td>Outdoor, Accessories, Truck</td>
<td>Sports, Food</td>
<td>Electronic, Indoor, Kitchen, Furniture</td>
<td></td>
</tr>
<tr>
<td># training images</td>
<td>16551</td>
<td>45520</td>
<td>39402</td>
<td>40260</td>
</tr>
<tr>
<td># test images</td>
<td>4952</td>
<td>1914</td>
<td>1642</td>
<td>1738</td>
</tr>
<tr>
<td># train instances</td>
<td>47223</td>
<td>113741</td>
<td>114452</td>
<td>138996</td>
</tr>
<tr>
<td># test instances</td>
<td>14976</td>
<td>4966</td>
<td>4826</td>
<td>6039</td>
</tr>
</tbody>
</table>

Evaluation metrics: Since an unknown object easily gets confused as a known object, we use the Wilderness Impact (WI) metric \cite{7} to explicitly characterises this behaviour.

\[
\text{Wilderness Impact (WI)} = \frac{P_K}{P_{K∪U}} - 1, \tag{5}
\]

where $P_K$ refers to the precision of the model when evaluated on known classes and $P_{K∪U}$ 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) \cite{42} 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 \cite{60, 44}).
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 pipeline on all the metrics. Kindly refer to Sec. 5.3 for more detailed analysis and explanation for the evaluation metrics.

<table>
<thead>
<tr>
<th>WI</th>
<th>A-OSE</th>
<th>mAP (↑)</th>
<th>WI</th>
<th>A-OSE</th>
<th>mAP (↑)</th>
<th>WI</th>
<th>A-OSE</th>
<th>mAP (↑)</th>
<th>WI</th>
<th>A-OSE</th>
<th>mAP (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current known</td>
<td>Previous known</td>
<td>Both</td>
<td>Current known</td>
<td>Previous known</td>
<td>Both</td>
<td>Current known</td>
<td>Previous known</td>
<td>Both</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle</td>
<td>0.02004</td>
<td>7080</td>
<td>57.76</td>
<td>0.0066</td>
<td>6717</td>
<td>54.99</td>
<td>30.31</td>
<td>42.65</td>
<td>0.0038</td>
<td>4237</td>
<td>40.23</td>
</tr>
<tr>
<td>Faster-RCNN</td>
<td>0.06991</td>
<td>13396</td>
<td>56.16</td>
<td>0.0371</td>
<td>12291</td>
<td>4.076</td>
<td>25.74</td>
<td>14.91</td>
<td>0.0213</td>
<td>9174</td>
<td>6.96</td>
</tr>
<tr>
<td>Faster-RCNN + Finetuning</td>
<td>Not applicable as incremental component is not present in Task 1</td>
<td>0.0375</td>
<td>12497</td>
<td>51.09</td>
<td>23.84</td>
<td>37.47</td>
<td>0.0279</td>
<td>9622</td>
<td>35.69</td>
<td>11.53</td>
<td>27.64</td>
</tr>
<tr>
<td>ORE</td>
<td>0.02193</td>
<td>8234</td>
<td>56.34</td>
<td>0.0154</td>
<td>7772</td>
<td>52.37</td>
<td>25.58</td>
<td>38.98</td>
<td>0.0081</td>
<td>6634</td>
<td>37.77</td>
</tr>
</tbody>
</table>

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 [9] 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.

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.

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 hav-
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.

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).

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

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. [42]. We find from Tab. 6 that drop in performance of ORE is much lower than [42] owing to the effective modelling of the unknown instances.

6.4 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.

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.

6.5 Clustering loss and t-SNE [37] 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 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.
References


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


[40] 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


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


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 \( \eta \), margin in clustering loss \( \Delta \) 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 \( \mathcal{F}_{\text{Store}} \)

In Sec. 4.1, we explain how class specific queues \( q_i \) are used to store the feature vectors, which are used to compute the class prototypes. A hyper-parameter \( Q \) controls the size of each \( q_i \). 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 \( (\mathcal{P} \text{ and } \mathcal{P}_{\text{new}}) \) is not very significant. We use \( Q = 20 \) for all the experiments.

<table>
<thead>
<tr>
<th>( Q )</th>
<th>W( I ) (↑)</th>
<th>A-OSE (↓)</th>
<th>mAP (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.02402</td>
<td>8123</td>
<td>56.01</td>
</tr>
<tr>
<td>10</td>
<td>0.02523</td>
<td>8126</td>
<td>56.02</td>
</tr>
<tr>
<td>20</td>
<td>0.02193</td>
<td>8234</td>
<td>56.34</td>
</tr>
<tr>
<td>30</td>
<td>0.02688</td>
<td>8487</td>
<td>55.78</td>
</tr>
<tr>
<td>50</td>
<td>0.02623</td>
<td>8578</td>
<td>56.22</td>
</tr>
</tbody>
</table>

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 \( \eta \)

The momentum parameter \( \eta \) controls how rapidly the class prototypes are updated, as elaborated in Algorithm 1. Larger values of \( \eta \) 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 \( \eta \)). This result is intuitive, as slowly changing the cluster centers helps stabilize contrastive learning.

<table>
<thead>
<tr>
<th>( \eta )</th>
<th>W( I ) (↑)</th>
<th>A-OSE (↓)</th>
<th>mAP (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.05926</td>
<td>9476</td>
<td>55.96</td>
</tr>
<tr>
<td>0.6</td>
<td>0.04977</td>
<td>9095</td>
<td>55.56</td>
</tr>
<tr>
<td>0.8</td>
<td>0.02945</td>
<td>8375</td>
<td>55.73</td>
</tr>
<tr>
<td>0.9</td>
<td>0.02193</td>
<td>8234</td>
<td>56.34</td>
</tr>
</tbody>
</table>

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

C. Varying the Margin (\( \Delta \)) in \( \mathcal{L}_{\text{cont}} \)

The margin parameter \( \Delta \) in the contrastive clustering loss \( \mathcal{L}_{\text{cont}} \) (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.

<table>
<thead>
<tr>
<th>( \Delta )</th>
<th>W( I ) (↑)</th>
<th>A-OSE (↓)</th>
<th>mAP (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.04094</td>
<td>9300</td>
<td>55.73</td>
</tr>
<tr>
<td>10</td>
<td>0.02193</td>
<td>8234</td>
<td>56.34</td>
</tr>
<tr>
<td>15</td>
<td>0.01049</td>
<td>8088</td>
<td>56.65</td>
</tr>
</tbody>
</table>

Table 9: Increasing the margin \( \Delta \), 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.

<table>
<thead>
<tr>
<th>( T )</th>
<th>W( I ) (↑)</th>
<th>A-OSE (↓)</th>
<th>mAP (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0219</td>
<td>8234</td>
<td>56.34</td>
</tr>
<tr>
<td>2</td>
<td>0.0214</td>
<td>8057</td>
<td>55.68</td>
</tr>
<tr>
<td>3</td>
<td>0.0411</td>
<td>11266</td>
<td>55.51</td>
</tr>
<tr>
<td>5</td>
<td>0.0836</td>
<td>12063</td>
<td>56.25</td>
</tr>
<tr>
<td>10</td>
<td>0.0835</td>
<td>12064</td>
<td>56.31</td>
</tr>
</tbody>
</table>

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.
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.

**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 \cite{60, 27, 17, 6} use knowledge distillation \cite{20} as a regularization measure to avoid forgetting old class information while training on new classes. Specifically, Shmelkov et al. \cite{60} 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. \cite{6} and Li et al. \cite{27} also distilled the intermediate network features. Hao et al. \cite{17} builds on Faster R-CNN and uses a student-teacher framework for RPN adaptation. Recently, Peng et al. \cite{44} 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. Using Softmax based Unknown Identifier**

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

**I. 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.

**J. Discussion Regarding Failure Cases**

Occlusions and crowding of objects are cases where our method tends to get confused \cite{external-storage, walkman} and bag not detected as unknown in Figs. 11, 13). Difficult viewpoints (such as backside) also lead to some misclassifications \cite{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.

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).
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.

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.

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).
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 afterwards. 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.

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