Detecting the unknown in Object Detection

Dario Fontanel[1], Matteo Tarantino[1], Fabio Cermelli[1,2], Barbara Caputo[1]

[1]Politecnico di Torino, [2]Italian Institute of Technology

dario.fontanel@polito.it

Abstract

Object detection methods have witnessed impressive improvements in the last years thanks to the design of novel neural network architectures and the availability of large scale datasets. However, current methods have a significant limitation: they are able to detect only the classes observed during training time, that are only a subset of all the classes that a detector may encounter in the real world. Furthermore, the presence of unknown classes is often not considered at training time, resulting in methods not even able to detect that an unknown object is present in the image. In this work, we address the problem of detecting unknown objects, known as open-set object detection. We propose a novel training strategy, called UNKAD, able to predict unknown objects without requiring any annotation of them, exploiting non annotated objects that are already present in the background of training images. In particular, exploiting the four-steps training strategy of Faster R-CNN, UNKAD first identifies and pseudo-labels unknown objects and then uses the psuedo-annotation to train an additional unknown class. While UNKAD can directly detect unknown objects, we further combine it with previous unknown detection techniques, showing that it improves their performance at no costs.

1. Introduction

For autonomous system acting in the real world, it is essential to have a clear understanding of their surroundings. To accomplish this purpose, multiple works have concentrated on the task of object detection [2, 9, 10, 12, 21, 22, 24, 29–31, 34], where the goal is to locate the objects inside an image and to assign them a category. Despite the outstanding performance demonstrated by state-of-the-art detection models [12, 24, 29, 31], they still have a critical limitation: they are able to predict only the classes that they have observed during training time, which are defined a priori and annotated in the training dataset. Regardless of how exten-
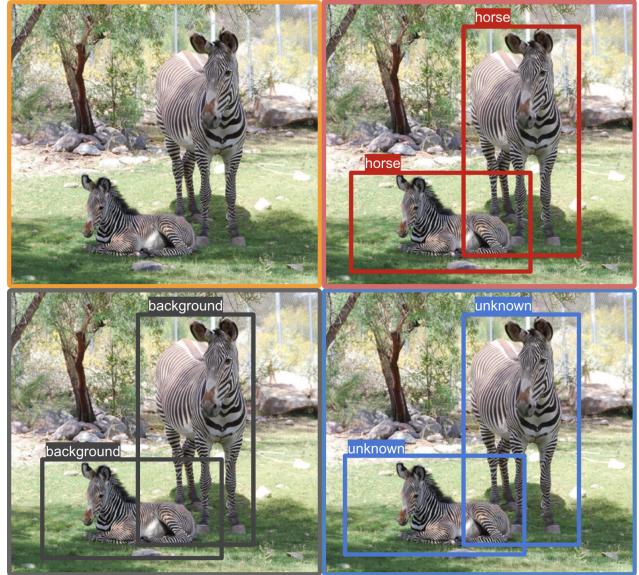

Figure 1. Open-set detection aims to detect previously unseen objects. While previous works consider the unknown objects as background, we are interested in evaluation the capability of detectors to predict unseen object as *unknown*. From left to right, we can see the in the first row a MS-COCO [23] test image and the wrong prediction of Faster R-CNN [31]. In the second one, we can see the background prediction of [5], and the unknown detection made by our framework.

sive the training dataset is, it is practically impossible to capture all the possible objects that system might ever encounter. Thus, in the real world, when current detection models are presented with an *unkown* object they will not detect it, considering it either as a known class or as background. For example, let us consider an autonomous driving car that has been trained on classes that are likely to be encountered in the city, such as pedestrians, other vehicles, trees, etc. However, at some point in time, the car is lead to a country region and a wild animal appears on the road. If we do not enable detection models to recognize un-

known objects, the detectors would not be able to spot the unknown animal and will label it as background, putting at risk the safety of the passengers since the car cannot recognize the obstacle and may crash on it. Ideally, we would like to have object detection models able to detect all the objects in the world ad possibly understand whether an object corresponds to one of the training categories or if it is an unknown object that they have never seen before.

In this work, we focus on this task, known in the literature as open-set object detection, for which an illustration is depicted in Figure 1. Previous works [5] addressed the problem partially, focusing only on limiting performance degradation on known classes when unknown data is encountered. We believe, as demonstrated in the previous example, that considering unknown objects as background is not sufficient to produce systems able to act in the real world as they could not detect obstacles, introducing safety risks. Differently, we propose to explicitly detect unknown objects at test time using a novel approach called UNKAD. It exploits the fact that, in object detection, multiple objects are preset in an image but only few of them are annotated, since the others are not considered relevant. By exploiting the multi-steps training strategy of the Faster R-CNN [31], UNKAD is able to extract pseudo-supervision for the unknown objects, identifying them in the background areas of the training images. Moreover, it exploits the pseudo-supervision on the classification head, introducing an additional *unknown* class that can be predicted, which helps to learn sharper decision boundaries between known and unknown objects. While this training strategy is already able to detect unknown objects directly predicting them in the classifier, we combine it with standard out-of-distribution methods and we show that it is able to improve the performance of them with respect to a standard Faster R-CNN training strategy. We demonstrate the performance of UNKAD on the popular Pascal VOC 2007 [7] and MS-COCO [23] benchmarks.

Contributions. To summarize, our contributions are:

- We propose a novel perspective for open-set detection problem, developing a simple yet effective training strategy;

- We introduce out-of-distribution standard approaches into the object detection framework, leading to novel analysis with respect to previous benchmarks;

- Experiments on the widely adopted Pascal VOC and MS-COCO datasets show that standard out-of-distribution methods benefits from our approach by a significant amount.

2. Related work

In this section, we review the foundations of our work, *i.e.* object detection, open-set and open-set object detection.

Object detection. Modern object detection approaches [2, 9, 10, 12, 21, 22, 24, 29–31, 34] are dominated by architectures based on convolutional neural networks that differ on whether or not candidate object proposals are used. We can group these works in two different categories: two-stage approaches [9, 10, 12, 21, 31], that generates object proposals which are classified and regressed by a region-of-interest (RoI) head module, and single-stage approaches [2, 22, 24, 29, 30, 34] that simultaneously output both classification scores and regressed bounding boxes without the need of any object proposal. Despite the outstanding performance achieved on popular benchmarks [7, 23], all of these architectures operate solely in an offline fashion, which means that after the model has been trained, additional knowledge cannot be added. Despite recent efforts to advance and deal with the inclusion of novel classes [16, 27, 32], none of these techniques strictly focus on open-set object detection.

Open-set. In recent years, open-set recognition has sparked a lot of interest in the machine learning community [1, 28, 35]. The seminar work of [1] formalized the open-set recognition task investigating for the first time what might happen when a model is ask to recognize data from categories that it has never seen before. One of the most popular open-set sub-field is out-of-distribution (OOD) detection [13, 15, 20] which aims at discriminating between samples that belong to the training distribution (also known as *in-distribution* or *known* data) and samples that do not (also noted as *out-of-distribution* or *unknown* data).

[13] settled the standard baseline for out-of-distribution detection by applying a threshold over the maximum softmax probability (MSP) to categorize a sample as belonging to known classes or as an unknown one. [8, 17] used Monte Carlo Dropout (MC-Dropout) to estimate the model uncertainty by forwarding the same image through the network multiple times, each time with a different dropout probability. [3] trains an additional neural network to output high confidence values when the prediction of the main model on in-distribution data is correct. This additional branch is then used to detect if the network prediction is reliable or not. Scaling the softmax probabilities by a temperature, for each sample [25] computes the energy which is higher for known samples rather than unobserved ones. ODIN [20] further enhanced MSP by introducing a temperature scaling factor in the softmax function and small perturbations over the test images. Both these hyperparameters are learned on an OOD validation set available during training. As collecting OOD data is not always feasible, in this work we avoid relying on it, leveraging background objects to model

unknown properties.

Open-set object detection. The open-set framework introduces additional challenges once adopted to the object detection task. [26] has been the first to bring open-set object detection to light and [5] further investigated the problem, assessing how detectors performance on known classes varies when evaluated on both known and unknown objects. In this paper, instead, we believe it is critical to evaluate also the capability of object detection models of recognizing objects as unknown. As a result, we need to introduce in the object detection framework out-of-distribution approaches able to distinguish between known and unknown categories, evaluate models' performance on both. Recently, few works [11, 16] made a step further introducing into object detection models the rejection capability, *i.e.* the ability of recognizing an object as unknown. To detect unknown samples, [16] employed a contrastive approach while [11] adopted a transformer-based architecture. Despite the introduction of the rejection capability, it is worth noting that in this work we do not provide comparisons with [16] and [11] as their primarily concerned is with building models capable of expanding their knowledge over time, hence de facto focusing on another task and objective.

3. Method

In this section we first formalize the problem definition and the importance of distinguishing between known e unknown categories (Section 3.1), We then describe UNKAD, showing how to detect unknown objects during training and how to involve them through the learning process (Section 3.2). Finally, in Section 3.3 we will analyze and compare different rejection strategies.

3.1. Preliminaries and problem formulation

The goal of open-set object detection is to detect objects that have not been seen during the training phase [5]. To perform this task, the model is provided with a training dataset $\mathcal{T}_{train} = \{(x, y)\}$, where x is an image and y is its corresponding ground-truth label. As in standard object detection, y contains bounding box annotations for a set of classes, that we indicate as *known* classes $\mathcal{Y}$. We note that, in object detection, the training images contains multiple objects, but not all of them are annotated. We denote all the objects not annotated as *unknown* objects. During testing, we are provided with a dataset $\mathcal{T}_{test}$ containing objects of both $\mathcal{Y}$ and never seen categories (*unknown*).

Focusing on R-CNN architectures [9, 31], our aim is to learn a function $\mathcal{F}$ mapping an image x to its known and unknown predictions at bounding box level. We consider $\mathcal{F}$ as built upon four key components. The first one is the feature extractor $\mathcal{F}_{FE}$ that is responsible for producing a feature map for each image x. The second one is

a region proposal network $\mathcal{F}_{RPN}$ that is fed with the feature map of the input image and produces a set of rectangular object proposals associated with a binary objectness score. In particular, it firstly projects the input feature map into a lower dimensional space by means of an intermediate projection layer and then the projected features are fed into two separate convolutional layers, one responsible for regressing the bounding box and the other responsible to output an objectness score ω. The set of object proposals is then applied to the feature map and pooled, producing $\mathcal{R}$ regions of interest (RoIs). The third component is the classification head and aims to classify the objects and regress the correct bounding box. It is composed by two functions: a class-scoring function $\mathcal{F}_\psi : \mathcal{R} \rightarrow \mathbb{R}^{\mathcal{R} \times |\mathcal{Y}|}$ that maps $\mathcal{R}$ RoIs to a box-level class scores, and a function $\mathcal{F}_\rho$ that regresses, for each RoIs and class, the precise bounding box coordinates, *i.e.* $\mathcal{F}_\rho : \mathcal{R} \rightarrow \mathbb{R}^{\mathcal{R} \times 4|\mathcal{Y}|}$. Finally, the last component is the unknown detection function $\mathcal{F}_\phi : [0, 1]^{\mathcal{R} \times |\mathcal{K}_{known}|} \rightarrow \mathbb{R}^{\mathcal{R}}$ that a binary score indicating if the RoI is unknown ($\mathcal{F}_\phi = 1$) or not ($\mathcal{F}_\phi = 0$).

3.2. Detecting the unknown

In this work we take inspiration from out-of-distribution detection approaches [6, 14, 18, 19, 33] that leverage external OOD data as extra supervision to learn richer decision boundaries between known and unknown samples. As collecting external unknown datasets is not always feasible, and most of the time human intervention is required to at least partially annotate them, our intuition is that we can leverage as OOD data a portion of the objects labelled as background. This is uniquely possible due to the object detection framework, that provides labels only for the classes of interest in $\mathcal{Y}$, leaving all the other objects that are not meant to be learned not annotated.

Towards this end, we train our $\mathcal{F}$ detector module in a four alternate steps strategy, adapting the methodology proposed in Faster R-CNN [31] and proposing **UNKAD** (**UNK**nown **A**ware **D**etection), that extends the model classification ability towards unknown objects. In particular, in the first and the third steps the $\mathcal{F}_{RPN}$ learns to extracts class-agnostic RoIs, while in the second and the fourth the detector $\mathcal{F}$ learns to classify known classes $\mathcal{Y}$ and the unknown. An illustration of UNKAD four steps training strategy is reported in Fig. 2.

Discovering the unknown. The intuition behind UNKAD is that unknown objects are already present in the training images and we can learn to recognize them by using the RPN class-agnostic ability to detect objects. In particular, the RPN exploits the ground truth labels to find and predict the RoI, *i.e.* n region where is likely to find an object. During preliminary experiments (see Section 4.3), we found that the RPN was able to detect, other than the anno-

tated objects, the objects in the image background with high confidence. Arguing that its class-agnostic nature helps to find any object in the image, we propose to pseudo-label the objects detected in the background as *unknown* objects. In order to generate pseudo-labels during training, we propose to use a simple thresholding mechanism. In particular, we define a threshold to establish whether a RoI is a region containing an object (known or unknown) or not. The threshold τ_{obj} is derived from data, and it is computed as:

$$\tau_{obj} = \mu + \lambda \cdot \sigma, \tag{1}$$

where

$$\mu = \frac{\sum_i \omega_i}{|\mathcal{R}_{FG}|}, \tag{2}$$

$$\sigma = \sqrt{\frac{\sum_i (\omega_i - \mu)^2}{|\mathcal{R}_{FG}|}}, \tag{3}$$

and μ and σ represent respectively the mean and the standard deviation of foreground-layer activations, $\mathcal{R}_{FG}$ is the set of RoIs that either (i) has the highest IoU, or (ii) has an IoU overlap with with any ground-truth box higher than 0.7. We recall that ω_i represents the objectness score for the ith region fed to $\mathcal{F}_{RPN}$. λ is hyperparameter set to 1.

To obtain the pseudo-labels, we first select all the RoIs having an objectness score greater than τ_{obj}. Then, we select all the RoIs that do not match with a ground truth annotation (*i.e.* they do not overlap more than 0.3 IoU with it) ad we associate them to the unknown class u.

Once we obtain pseudo-labels on the unknown objects that are present in the training images, we exploit them to train the classification head to detect every unknown object.

Learning to predict unknown As our primary goal is to learn an unknown detector through pseudo-supervision, we add to the final classification layer $\mathcal{F}_\phi$ an additional class that leverages the unknown pseudo-labels generated by $\mathcal{F}_{RPN}$. More precisely, following the alternate training strategy of Faster R-CNN [9, 31], at the end of the first and the third steps, we perform the pseudo-labelling phase. In the second and the fourth steps, $\mathcal{F}_\rho$ and $\mathcal{F}_\phi$ are learned, leveraging both annotated and pseudo-annotated data. By using the pseudo-annotations as ground truth, $\mathcal{F}$ learns to separate at training time the unknown category from all the known classes.

3.3. Rejection strategies

In this section we present different strategies to predict a RoI as an unknown object, *i.e.* different implementation of the $\mathcal{F}_\phi$ function. In particular, we assume that, at inference time, we obtain $\mathcal{R}$ RoIs from the RPN, which are passed to the classification head to obtain the final classification scores s using $\mathcal{F}_\psi$.
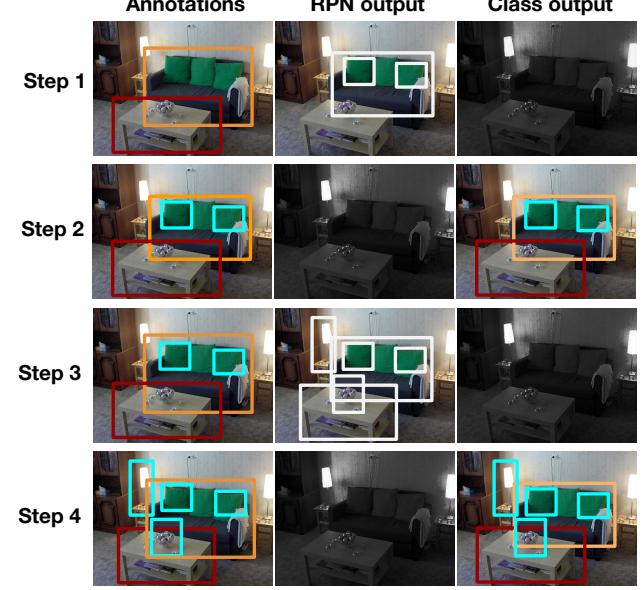

Figure 2. Illustration of the UNKAD training procedure. In the first step, the RPN learns to predict class-agnostic RoIs (white boxes) containing objects. It then pseudo-labels unknown objects and, in step 2, trains the classifier with the additional unknown class. In step 3, the RPN is trained again, considering both known and unknown objects. Finally, in step 4 the final model able to detect both known and unknown objects is obtained.

Direct prediction. The standard approach of localizing and classifying objects in closed-set scenarios is to localize the object and assign it the class with the highest class probability. Since UNKAD extends the classification logits also to the unknown class, we may apply the same principle and predict a sample as unknown only if has the unknown class has the highest score. Formally, given an image and a RoI r, we obtain the set s of class scores, including also the score for the u class, and we compute $\mathcal{F}_\phi$ as:

$$\mathcal{F}_\phi(r) = \begin{cases} 1 & \text{if } \hat{y} = \text{u} \\ 1 & \text{if } \hat{y} = \text{b} \ \& \ \omega_r > \tau_{obj} \\ 0 & otherwise \end{cases} \tag{4}$$

where $\hat{y} = \arg\max_{c \in \mathcal{Y} \cup \text{u}} s_c$ is the class with highest score, ω_r is the RPN objetness score for r, and b indicates the background class. We note that we added the second case to avoid missing potential unknown objects.

Maximum Softmax Probability (MSP). Maximum Softmax Probability [13] leverages the highest probability value assigned to any known class in $\mathcal{Y}$ as a measure of uncertainty. For an image, given a RoI r and the class scores s for it, MSP computes the value for the unknown class $\mathcal{F}_\phi(r)$ as:

$$\mathcal{F}_\phi(r) = \max_{c \in \mathcal{Y} \cup \text{b}} \frac{e^{s_c}}{\sum_{k \in \mathcal{Y} \cup \text{b}} e^{s_k}} \leq \tau_{MSP}, \tag{5}$$

4

where τ_{MSP} is a user defined threshold that we set to 0.5 and b indicates the background class. Intuitively, if a class or background is predicted with a probability inferior to τ_{MSP}, MSP identifies the RoI as an unknown object.

Energy-based classifier (Energy). The energy-based scoring function [25] adopts a completely different perspective from traditional classifiers. Instead of computing class probabilities, it maps each samples to a scalar value, *i.e.* the energy. Given an RoI r of a image, $\mathcal{F}_\phi(r)$ is computed as:

$$\mathcal{F}_\phi(r) = -T \cdot \log \sum_{c=1}^{\mathcal{Y} \cup \text{b}} e^{\frac{s_c}{T}} \quad \leq \tau_{EN}, \tag{6}$$

where T is the temperature hyperparameter, τ_{EN} is a user defined threshold set to -3, and s_c is the classification score for class c. To align with the convention introduced by [25], we keep the negative notation for the energy score, which is higher for known classes and lower for unknown ones.

ODIN. To enhance MSP performances, ODIN [20] adopted temperature scaling and input perturbation. In detail, before feeding the scores to the softmax function, ODIN scales each class scores by a temperature parameter T. In addition, it also pre-processes each input image x by introducing a small perturbation, that we adapt in the context of object detection as:

$$\tilde{x} = x - \epsilon \cdot sgn(-\nabla_x \cdot \frac{1}{\mathcal{R}} \sum_{r=1}^{\mathcal{R}} \log(\max_{c \in \mathcal{Y} \cup \text{b}} p_c)), \quad (7)$$

where ϵ is the perturbation magnitude, ∇_x is the gradient vector with respect to x, and p is the softmax of the scores s, *i.e.* $p = \text{softmax}(s)$.

As for MSP [13], given a perturbed image $\tilde{x}$ and the class scores s computed on the RoI r, ODIN computes $\mathcal{F}_\phi(r)$ as:

$$\mathcal{F}_\phi(r) = \max_{c \in \mathcal{Y}} \frac{e^{s_c}}{\sum_{k \in \mathcal{Y}} e^{s_k}} \leq \tau_{ODIN}, \tag{8}$$

τ_{ODIN} is a user defined threshold set to 0.4.

4. Experiments

4.1. Experimental Protocol

Datasets. We conduct our experiments on the popular Pascal VOC 2007 [7] and MS-COCO [23] datasets. Pascal VOC 2007 is a widely used benchmark that includes 20 foreground object classes and consists in 5K images for training, considering both training and the validation splits, and 5K for testing. MS-COCO is a large scale dataset that provide 80K images for training, 20K for validation and for the test. It contains annotation for 80 object classes of which 20 are in common with Pascal VOC 2007. For the open-set

evaluation, we follow the split defined as WR_1 in [5]. In particular, the test set of Pascal VOC 2007 is used for the evaluation on the $\mathcal{Y}$ classes, while 4952 images from MS-COCO training set that do not contain any Pascal VOC class are selected for the evaluation on the u class (see. Figure 3), resulting in a total of nearly 10k test images.

Baseline. We compare our proposed training strategy with the 4-steps Faster R-CNN [31] training procedure. We implement both strategies upon multiple open-set and out-of-distribution detection strategies. In particular, we implement the training strategy described in Sec. 3.3: direct prediction, MSP [13], Energy [25] and ODIN [20]. We note that it is not possible to use direct prediction on the standard Faster R-CNN training since it does not provide the unknown class in the classification head.

Architectures. Each of the evaluated method share the same Faster R-CNN architecture with a ResNet-50 backbone initialized with ImageNet pretrained weights [4]. We train the network using SGD optimizer with batch size equal to 4, learning rate set to 10^{-3} and decreased after 75% of iterations by a factor of 0.1. The weight decay is set equal to 10^{-4} and SGD momentum to 0.9. The number of training iterations is set to $40k$ for both the 1st and the 2nd training step, while it is decreased down to $10k$ in the 3rd and 4th step. We resize images to 800×1333 in both training and testing, while during training we also performed random crop and random horizontal and vertical flip operations.

Metrics. To assess the impact of the unknown objects on the performance of standard object detection models, [5] introduced the metric called *Wilderness Impact (WI)*, defined as:

$$WI = \frac{\text{Precision Closed-Set}}{\text{Precision Open-Set}} - 1 =$$
$$= \frac{TP_c}{TP_c + FP_c} \cdot \frac{TP_c + FP_c + TP_o + FP_o}{TP_c + TP_o} - 1, \tag{9}$$

where TP_c, FP_c, indicate the true positives and false positives of known classes, while TP_o, FP_o the true and false positive on unkowns. However, in [5] the metric was simplified since it did not considered the rejection option, *i.e.* the models cannot predict the unknown class, thus $TP_o = 0$. They considered the following metric:

$$WI_{no\_rej} = \frac{FP_o}{TP_c + FP_c}. \tag{10}$$

We note that WI_{no_rej} does not consider the detection performance on unknown classes, but it considers only the performance on the known classes. In particular, classifying all the unknown as background objects would get the optimal
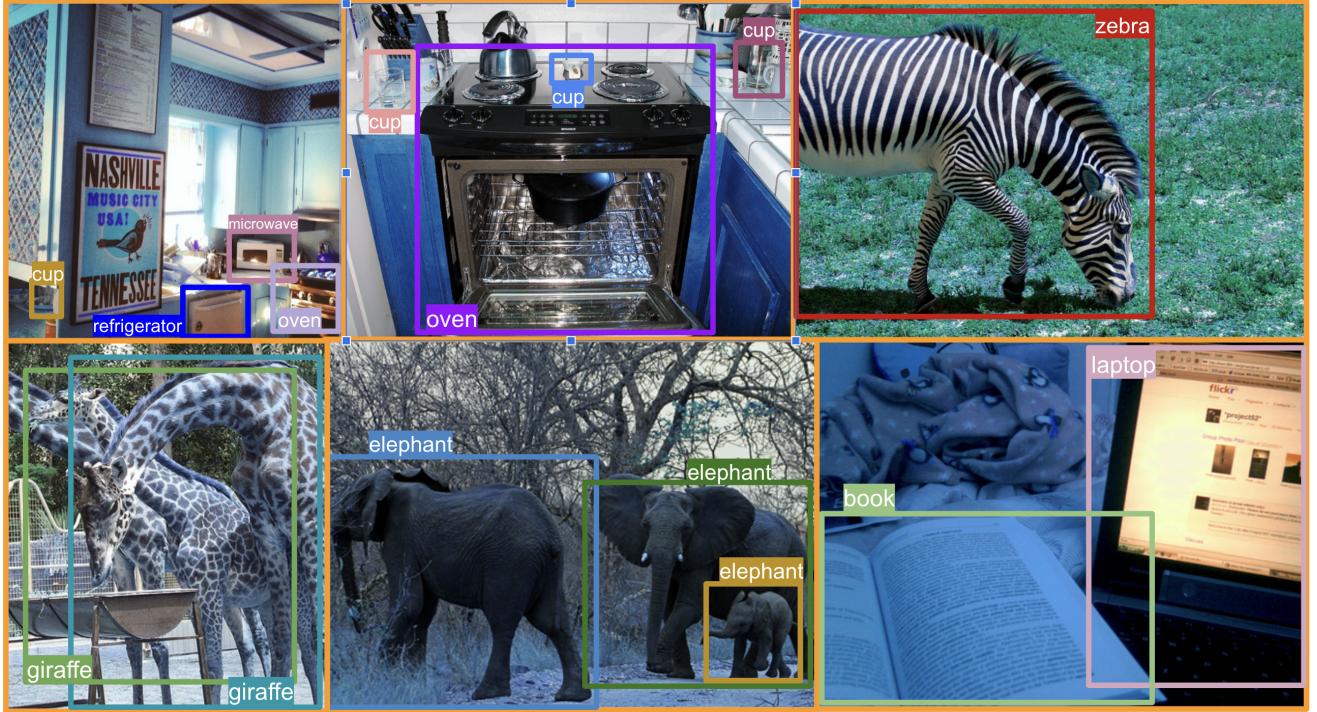
Figure 3. Extract of unknown images taken from [23].

Table 1. Evaluation of out-of-distribution detection methods adopting standard and UNKAD (ours) training strategies on Pascal VOC [7] and MS-COCO [23] datasets. Results on $\mathcal{Y}$ are evaluated through mAP and WI_{no_rej} [5] metrics, while results on u are evaluated computing WI [5] and the recall, precision and F1 scores on unknown objects.

Training	Rejection	mAP%↑	WI$_{no\_rej}$ ↓	WI↓	U$_{Recall}$ ↑	U$_{Precision}$ ↑	U$_{F1}$ ↑
Standard	Without Rejection [31]	67.29	1.63	1.63	0.00	0.00	0.00
	MSP [13]	67.36	1.58	-17.52	2.46	4.20	3.10
	Energy [25]	51.01	0.78	**-30.39**	2.40	0.41	0.70
	ODIN [20]	67.22	1.58	-20.43	3.81	1.38	2.02
UNKAD	Without Rejection [31]	**67.75**	1.50	1.50	0.00	0.00	0.00
	MSP [13]	67.74	1.50	-19.22	3.22	**4.57**	**3.78**
	Energy [25]	51.75	**0.68**	-29.82	2.34	0.30	0.53
	ODIN [20]	67.63	1.49	-21.56	4.85	1.62	2.43
	Direct Prediction	**67.75**	1.48	-21.91	**5.19**	2.83	3.67

score since $FP_o = 0$. Despite the second metric has been used by recent works [5,16], we consider more suited to our task the WI metric since it took also into account the true positive on unknown objects.

Moreover, since in our work we are explicitly interested in evaluating the models on unknown objects, we report the recall, precision and F1 metrics considering only the unknown class. We define them as:

$$U_{Recall} = \frac{TP_o}{TP_o + FN_o}, \quad (11)$$

$$U_{Precision} = \frac{TP_o}{TP_o + FP_o}. \quad (12)$$

The more the model tends to predict each sample as belonging to the unknown category, the higher the recall will be, but the lower the precision will be as the number of false positive tends to increase. For this reason, we introduce the U_{F1} metric which summarizes both U_{Recall} and $U_{Precision}$. U_{F1} is maximized if and only if both U_{Recall}

6

and $U_{Precision}$ are maximized. It is defined as:

$$U_{F1} = 2 \cdot \frac{U_{Recall} \cdot U_{Precision}}{U_{Recall} + U_{Precision}}. \qquad (13)$$

Finally, in addition to the open-set metrics, we also report the *mAP* to evaluate the model performances on $\mathcal{Y}$ classes, defined as:

$$mAP = \frac{1}{|\mathcal{Y}|} \sum_{c \in \mathcal{Y}} AP_c, \qquad (14)$$

where AP_c is the average precision for the class c computed at different recall levels.

4.2. Unknown detection results

Tab. 1 reports the comparison among the standard Faster R-CNN training and UNKAD, considering multiple unknown detection strategies. As the table shows, our approach improves detection of both known and unknown objects. Starting from the former, standard Faster R-CNN architecture with no rejection capability benefits from our approach, achieving up to to 67.75% mAP. This behaviour is also confirmed by the WI improvement from 1.63 to 1.5 (the lower the better), indicating that UNKAD distinguishes better known and unknown objects. The same behavior on the known classes is evident when employing a rejection strategy: considering the WI metric, MSP improves from -17.52 to -19.22 and ODIN from -20.43 to -21.56, while Energy obtain similar results (-30.39 vs -29.82). We note that, while the Energy approach is the best on the WI and WI_{no_rej} metric, it obtains very low mAP. We ascribe this behavior to the high unknown score that Energy assigns to samples, leading to the rejection of most of the known objects. Moreover, the contrast among the WI and mAP metrics reveals that WI is not well-suited to evaluate methods on the open-set task, since it does not consider the overall model performance but only the ratio among closed- and open-set precision.

While improving the results on known classes is important, our goal is mainly detect unknown objects. Considering the U-$F1$ score, we note that UNKAD increases the results of MSP from 3.10% to 3.78% and from 2.02% to 2.43% for ODIN [20]. This remarks the impressive ability of our training strategy, that improves the detection ability of the model without introducing additional costs, pseudo-labeling unknown objects in the background of the images. We acknowledge that our training procedure slightly hamper the Energy performance. However, we remark that Energy already assigns a very high score for the objects, as demonstrated by the very low $U_{Precision}$ achieved by it. We also note that only Energy shows this behavior, while both MSP and ODIN, when using UNKAD, improve the results on both $U_{Precision}$ and U_{Recall} metrics. Finally, comparing the simple direct prediction approach with the others,

Table 2. Ablation study of the direct approach rejection strategy when adopting UNKAD. We compute mAP and U_{F1} metrics without rejection, using the direct prediction of the final classifier, exploiting the τ_{obj} for unknown predictions and combining the two strategies.

No rejection	$\hat{y} = $ u	τ_{obj}	mAP↑	U_{F1} ↑
✓			67.75	0.00
	✓		67.73	3.46
		✓	67.72	3.41
	✓	✓	**67.75**	**3.67**

we see that it achieves comparable or even better performance. In particular, it achieves the best performance on U_{Recall} without suffering performance drop, as indicated by the mAP performance. However, the $U_{Precision}$ is lower than other approaches (-1.74% w.r.t. MSP).

4.3. Ablation studies

Direct prediction rejection strategy. In Table 2 we report a detailed analysis on the direct prediction rejection strategy available when using our UNKAD approach. It is composed by two key components: the additional unknown class on the final classification layer, able to predict the unknown class for each RoI; and the objectness threshold τ_{obj}. As we can see from the second row, the additional unknown class allows to maintain the same 67.75% mAP performance over the known classes of the standard Faster R-CNN with no rejection capability; while it also achieves 3.46% of U-$F1$. In the third row, we reported instead the results achieved by directly applying τ_{obj} to background RoIs. In particular, for each RoI predicted as background with the highest probability among all classes, if the $\mathcal{F}_{RPN}$ emits an objectness score higher than τ_{obj} it is then considered unknown. Combining the two strategies together allows to achieve the best results, as shown in the last row of the table. The overall rejection procedure maintains the highest mAP on known classes, while also increases the U-$F1$ up to 3.67% and decreases the WI down to 1.48.

RPN unknown detection. Although $\mathcal{F}_{RPN}$ is a class-agnostic detector by design [9, 31], an open question is whether it is able to recognize even objects on which it has not been explicitly trained on, as it is essential for the unknown pseudo-labeling procedure. To this end, we formulated the AVG_{obj} metric as a quantitative measure of the $\mathcal{F}_{RPN}$ ability to identify objects within an image in both closed- and open-set scenarios. It is formulated as follows:

$$AVG_{obj} = \frac{1}{|\mathcal{P}_{fg}|} \sum_{p \in \mathcal{P}_{fg}} f_{fg}(p) \qquad (15)$$

where $\mathcal{P}_{fg}$ is the set of ground truth foreground proposals
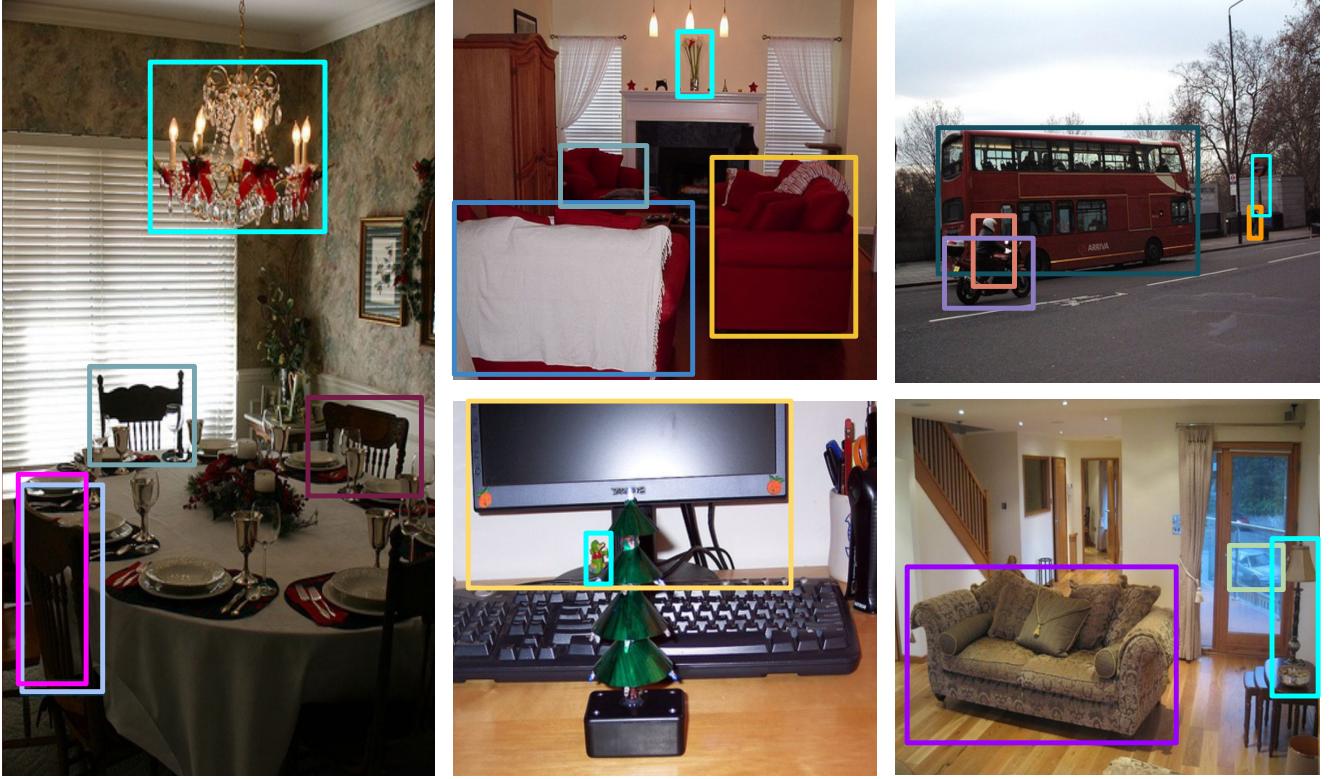
Figure 4. **Qualitative** results of $\mathcal{F}_{RPN}$ detections on Pascal VOC [7]. The cyan boxes indicate a detection on unknown objects, while the other ones indicate a detection on known classes. Best viewed in color.

Table 3. Ablation study of $\mathcal{F}_{RPN}$ ability to identify objects in closed- and open-set scenarios.

Standard	UNKAD	known	unknown	AVG_{obj}
✓		✓		0.98
✓			✓	0.97
	✓	✓		**0.99**
	✓		✓	0.98

and $f_{fp}(p)$ is the probability that the proposal p is actually considered foreground.

We report in Tab. 3 the evaluation of standard training procedure and UNKAD under the AVG_{obj} computed on both known and unknown objects. Our approach achieves up to 0.99 on known and 0.98 on unknown (being 1 the upper bound), surpassing the standard procedure in both the evaluations. Achieving comparable performance in both closed- and open-set scenarios proves our intuition that $\mathcal{F}_{RPN}$ is able to precisely detect objects despite their belonging to the training distribution or not. It is worth noting that UNKAD increases the confidence on considering proposals as foreground ones on the known classes, as shown in the comparison between the first row and the third one.

5. Conclusions

In this work, we proposed a novel training strategy, called UNKAD, to improve open-set object detection performance. UNKAD relies on the assumption that, during training, the images contain multiple non-annotated objects. Instead of requiring an explicit annotation for them, it automatically detects and pseudo-labels them, exploiting the four-steps Faster R-CNN training procedure. In particular, in the first step, it trains the class-agnostic RPN to detect objects using the ground truth annotations. Then, it pseudo-labels as unknown all the objects in the dataset with a high objectness score that do not match a ground truth annotation. The pseudo-labels are then used as pseudo ground-truths to train the classification head. In the third and fourth training steps, the knowledge on the unknowns is further consolidated, obtaining the final model.

We demonstrate that UNKAD is able to directly detect the unknown classes and it also improves the performance of previous training strategies with no additional costs on the Pascal VOC and MS-COCO datasets. Indeed, the unknown detection performance is still far from a system ready to operate in the wild and we hope that our work establishes a new baseline to push forward the state of the art in this research field.

References

[1] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR-15*. 2

[2] Jiale Cao, Yanwei Pang, Jungong Han, and Xuelong Li. Hierarchical shot detector. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9705–9714, 2019. 1, 2

[3] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *Adv. Neural Inform. Process. Syst.*, pages 2902–2913, 2019. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[5] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1021–1030, 2020. 1, 2, 3, 5, 6

[6] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. *Advances in Neural Information Processing Systems*, 31, 2018. 3

[7] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2009. 2, 5, 6, 8

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

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1, 2, 3, 4, 7

[10] 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. 1, 2

[11] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Owdetr: Open-world detection transformer. *arXiv preprint arXiv:2112.01513*, 2021. 3

[12] 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, 2

[13] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016. 2, 4, 5, 6

[14] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018. 3

[15] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10951–10960, 2020. 2

[16] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5830–5840, 2021. 2, 3, 6

[17] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Adv. Neural Inform. Process. Syst.*, pages 5574–5584, 2017. 2

[18] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Adv. Neural Inform. Process. Syst.*, pages 7167–7177, 2018. 3

[19] Yi Li and Nuno Vasconcelos. Background data resampling for outlier-aware classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13218–13227, 2020. 3

[20] 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, 5, 6, 7

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 2

[22] 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

[23] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 2, 5, 6

[24] 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. 1, 2

[25] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020. 2, 5, 6

[26] 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 3243–3249. IEEE, 2018. 3

[27] 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. 2

[28] 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*, pages 11814–11823, 2020. 2

[29] 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, 2

[30] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1, 2

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1, 2, 3, 4, 5, 6, 7

[32] 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. 2

[33] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 550–564, 2018. 3

[34] Tiancai Wang, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Learning rich features at high-speed for single-shot object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1971–1980, 2019. 1, 2

[35] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2021. 2