The Overlooked Elephant of Object Detection: Open Set

Akshay Raj Dhamija[†], Manuel Günther[†], Jonathan Ventura[‡], and Terrance E. Boult[†]

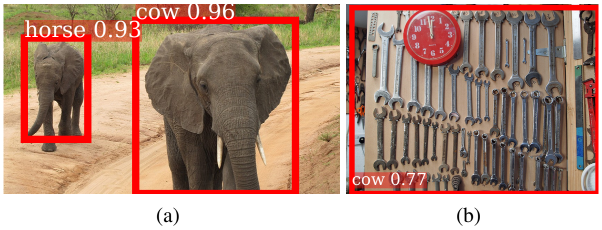
[†]Vision and Security Technology Lab, University of Colorado Colorado Springs
[‡]Department of Computer Science & Software Engineering, California Polytechnic State University
[†]{*adhamija, mgunther, tboult*}*@vast.uccs.edu*, [‡]*jventu09@calpoly.edu*

Abstract

Even though object detection is a popular area of research that has found considerable applications in the real world, it has some fundamental aspects that have never been formally discussed and experimented. One of the core aspects of evaluating object detectors has been the ability to avoid false detections. While major datasets like PASCAL VOC or MSCOCO extensively test the detectors on their ability to avoid false positives, they do not differentiate between their closed-set and open-set performance. Despite systems being trained to reject everything other than the classes of interest, unknown objects from the open world end up being incorrectly detected as known objects, often with very high confidence. This paper is the first to formalize the problem of open-set object detection and propose the first open-set object detection protocol. Moreover, the paper provides a new evaluation metric to analyze the performance of some state-of-the-art detectors and discusses their performance differences.

1. Introduction

Object detection research has a long history in computer vision, dating back more than five decades [31]. The aim of an object detector is to localize all the objects it is trained to identify while neglecting all other regions from random objects or scene backgrounds. Object detection approaches have evolved from feature-based detectors, to sliding window algorithms [27], leading to region proposal methods [7, 6, 24] and anchor box-based approaches [18, 21, 22, 16]. Especially in the past few years, advances in computation speed, the increase of labeled training data and challenges such as the PASCAL Visual Object Categorization (VOC) [2] and Microsoft's Common Objects in Context (MSCOCO) [17] have made the use of deep networks possible, which provide significant improvements to the field.



(a) (b)

Figure 1: THE ELEPHANT IN OBJECT DETECTION *While current state-of-the-art detectors are trained to handle backgrounds, their designs are not well equipped to address unknown objects, which they often incorrectly detect as one of the existing classes with a high confidence. (a) shows results from Faster R-CNN and (b) was produced by RetinaNet, both of which were only trained to detect the 20 classes from PASCAL VOC, which do not include elephants, clocks, scissors or wrenches as present in the above images. As we explore in this paper, different detectors such as Faster R-CNN, RetinaNet and YOLOv2 respond to unknowns differently.*

With the popularity of deep learning techniques, the importance of dataset size has increased. Challenges such as PASCAL VOC increased their training data size between 2007 and 2012, while more recently in 2017 MSCOCO changed its 83k/41k train/val split to 118k/5k, citing the need for more training data by the research community. Increasing the number of training samples can improve generalization and, hence, enable the detectors to better capture variations in a given object.

While the majority of real world detection applications are only interested in a small subset of the object categories provided in these datasets, additional categories seem to be providing a generalization in order not to misclassify a sample as one of the classes of interest. Though detectors trained on smaller academic datasets such as PASCAL VOC seem to perform well on the according test sets, it is frequently ob-

This work is supported in part under Naval Research Laboratory contract N00173-19-P-1222

served that their performances do not translate into the real world. As we shall see, experiments with our new evaluation protocols show that **open-set object detection is far from being solved** – despite training with a ”background” class, which is supposed to reject everything other than the objects of interest. We see that with current designs, unknown objects will often be mapped onto existing classes (see Fig. 1) with high confidence.

Object detectors produce two types of errors: *(a)* False Negatives, i.e., objects of interest are classified as another object or as background, and *(b)* False Positives where a background sample or an unknown object is mistaken as one of the classes of interest. While false negatives may be considered as a shortcoming in the network training or the generalizability of the network or dataset, the same cannot be said for false positives. The network is trained to identify a small set of known objects from the infinite number of object classes in the real world. Even if the network used a ”background” class to reject samples not of interest, it is impossible for a dataset to sample instances from each of the remaining infinite number of undesirable object classes for training. Since these unknown objects are not sampled during training, the expectation that they will be rejected during testing is unrealistic. Though all detectors are somewhat equipped to prevent false positives, the current evaluation protocols used by datasets such as PASCAL VOC and MSCOCO do not sufficiently test a detectors ability to reject unknown objects and, thus, overestimate their real-world performance. In closed-set evaluation protocols, the rejection of random objects directly impacts precision values, but there is no specific differentiation between false positives arising from unknown objects and from random textures in the background. In this work, we focus on understanding the responses of detectors to objects that they were not trained to detect, i.e., we propose and perform open-set evaluation.

Our Contributions: *(a)* In Sec. 2, we categorize current detectors based on their approaches to handle background/unknown objects and generalize our finding to the currently popular detection algorithms. *(b)* In Sec. 3, we formalize object detection as an open-set problem. *(c)* We propose the first open-set object detection protocol that better approximates the real world in Sec. 3.1. *(d)* We propose an evaluation metric for open-set object detection that allows better comparison of performance than mAP. *(e)* In Sec. 4, we highlight the shortcomings of current state-of-the-art object detectors. *(f)* Finally, we attempt to provide an understanding toward choosing an operating point when applying a detector to the real world in Sec. 5.

2. Dividing Detectors by Classifier Type

A core concept common in all object detectors is that they consider a specific region R of an image and attempt to provide the probability p_i for each of the N known classes $C_1, \ldots, C_N$ being present in that area. These specific areas are known by different names such as windows, crops, region proposals or anchor boxes. They may also be generated by different algorithms such as sliding window [27], selective search [7] or region proposal networks [24]. Because there are so many potential regions, it is critical that the systems are good at rejecting regions that do not contain objects of interest. While it is one of the key challenges for object detectors to avoid misdetections in these specific image areas, little research on improving this aspect has been performed. To address detection/classification of objects while rejecting non-object regions or unknown objects, there has been only a small range of designs. We broadly divide these into the following categories:

Multi-Class Classifiers without Background Many early-stage detectors such as OverFeat [27] treated object detection as a sliding window-based image classification problem. These systems are trained to identify objects of N different classes $C_1, \ldots, C_N$ and for each generated window they provide an estimate of the probability p_i for presence of each object category such that $\sum_{i=1}^{N} p_i = 1$. In some approaches, from the various sampled windows R, the window with the maximum classification score is used, which allows to detect only a single object in the image. In others, different crops where the same class is predicted with the maximum score and that have a significant bounding box overlap are combined to provide one detection. While an advancement when first introduced, these systems implicitly assume that all inputs map to one of the known classes, which results in many false detections. Consequently, these approaches are no longer used.

Multi-Class Classifiers with Background Most two-stage detectors such as Fast R-CNN [6] and Faster R-CNN [24] classify a region R into $N+1$ classes. The additional class, called the background class C_b, is trained from non-object windows and p_b is interpreted as representing the probability of R not belonging to any of the N classes such that $p_b + \sum_{i=1}^{N} p_i = 1$. Some one-stage detectors such as SSD [18] also belong to this category. We note that during evaluation all of these systems use each probability independently and do not consider the maximum over $p_b, p_1, \ldots, p_N$ because even for objects of known classes, the background probability p_b is higher than that of the correct class – a supporting experiment is in the supplemental material.

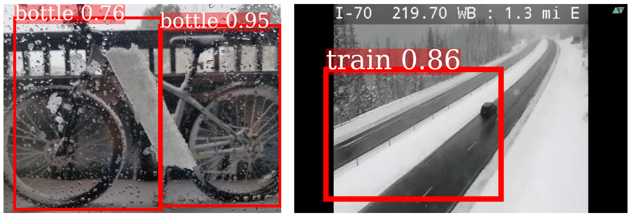
One vs. Rest Classifiers Detection algorithms in this category utilize one-versus-rest classifiers. The idea here is that a region contains the known object or it does not. Hence, the detectors do not explicitly provide a probability p_b for R being not of any known class. But at the same time it is not guaranteed that $\sum_{i=1}^{N} p_i = 1$, and often the models do not even estimate probabilities. Some of the early approaches such as DPM [4], SPPnet [9] and R-CNN [7] fall under this category. For each known class, these models use an SVM-based one-versus-rest classifier to provide the score that the object belongs to this class. Another variation of one-versus-rest classifiers can be seen in the current state-of-the-art one-stage detector RetinaNet [16], which uses binary cross-entropy to identify the presence of a known object. This approach is inspired by the observation that a region may contain more than one object, e.g., a person sitting on a chair with a cat on the lap. In such cases, giving a high probability to just one of the classes would not be justified, but instead the probabilities of several classes could approach 1. Therefore, such detectors give an independent probability to each of the objects and do not force them to sum to 1. If none of the known objects is present, the detector is expected to provide low probabilities to each of the known classes. Unfortunately, one-vs-rest classifiers generally have unbounded open space risk [26] and unknown samples will often confidently be classified as one of the known classes.

Objectness-Based Classifiers One-stage detectors from the YOLO family [21, 22, 23] belong to this category. Before providing a probability score for each of the object categories, these detectors provide an objectness score, i.e., they assess if the region includes any known object. All class scores are considered to be mutually independent and classifiers are trained with either sigmoid or binary cross entropy loss. The authors of [21, 22, 23] present their *objectness score* as a probability of one of the known objects being present in the respective anchor box. However, as we will see later, this objectness score is high in the presence of many other unknown objects as well. Thus, their model is better interpreted as a "generic objectness" score rather than the claimed "known objectness" score.

Discussion While object detection is a problem inherently intended to handle unknown objects by detecting only the known objects, existing systems have not been formally formulated as open-set, and neither have they been evaluated under real-world open-set conditions where really unknown objects need to be ignored. While both training a background class or treating the problem as one-vs-rest classification helps in rejecting some unknowns, neither formulation provides bounded open-space risk as defined in [25]. The remainder of this paper will help to analyze the impact of ignoring the open-space risk possessed by the above families of detection algorithms.

3. Formalizing Open-Set Object Detection

A scenario where a system is tested on instances belonging to classes different from what it was trained on is defined as open-set. Since, by definition, detectors are only supposed to detect objects they were trained to identify while rejecting others, we see object detection as a general open-set problem. While it is easy to draw a parallel to the prior definition of the open-set classification problem [26], we introduce the additional category *mixed unknown*, whose determination is **crucial and unique to the practical open-set object detection problem**. In order to explain the need for this additional subcategory, we explain all the types of object classes that are present in the infinite space of labeled objects Y. These classes can be broadly categorized into [19]:



(a)

(b)

Figure 2: MISDETECTIONS DUE TO UNCOMMON IMAGING OF KNOWN OBJECTS *In (a) we demonstrate how uncommon imaging conditions such as snow or rain can cause misclassification of a known object. Similarly, in (b) we demonstrate how different scenes cause a misdetection of a simple background as a known object, all with high confidence scores.*

- $K = \{\vec{x}_1, \ldots, \vec{x}_M\} \subset Y$: The *known objects* or *objects of interest* that the detector is trained to detect. These can be separated into *known knowns* K_K, the data similar to that used in training, and *unknown knowns* K_U, which can be defined as novel views of known objects and are typically seen in test sets. These views may originate due to environmental conditions, distortions in imaging conditions or deformation of the known object, as provided in Fig. 2. This problem has been a subject of various challenges and datasets [29]. While *unknown knowns* are a part of general open-set object detection, analyzing them is not the core subject of this paper.
- $U = Y \setminus K$: The *unknown objects* of classes the detector needs to reject. Since Y is infinite and K is finite, U is also infinite. The set U is a combination of two subsets:
  1. $U_K \subset U$: The *background*, *garbage*, *undesirable*, or *known unknown* objects. These are the objects the detector should learn to ignore during training, e.g., grass, trees and sky in Fig. 1. Since U is infinitely large, only the small subset U_K can be used during training.
  2. $U_U = Y \setminus (K_K \cup K_U \cup U_K) = U \setminus U_K$: The *unknown unknown* or *previously unseen* objects, which belong to the rest of the infinite space from U. Samples from these object classes are not available during training, but only occur at test time, see Fig. 1 for an example.

The above breakdown provided by Miller *et al.* [19] misses one important aspect required for practical open-set object detection, i.e., the category of *mixed unknown* U_M. In bounding box-based detection datasets, not every pixel in the image is labeled, but known objects K are labeled only with bounding boxes. When creating an open-set protocol, one could identify only certain unknown objects as U_K and certain as U_U, since Y is an infinite space. For example, let's assume in the entire dataset there is a single instance of a person on a walking frame. Since a walking frame is not a labeled object, it does not implicitly belong to either U_K or U_U. If during the dataset splitting, this image ends up in the testing set, the walking frame belongs to U_U but if it ends up in training set it now belongs to U_K. Thus, without labeling all objects of Y in all images, it cannot be determined if the walking frame is U_K or U_U. This means for a bounding box-based true open-set protocol, it is not possible to ensure that no unknown objects are seen during training, hence giving rise to the category of mixed unknowns U_M.

While current detection approaches perform well on datasets such as PASCAL VOC and MSCOCO, their response to unknown unknowns U_U is not specifically studied as the dataset only includes mixed unknowns U_M. In the real world, detectors may be applied in environments with a variety of control levels. For example, the environment for a detector in a warehouse may be highly controlled but that for a self-driving car or a home robot may be unconstrained. For the majority of unconstrained environments, detectors are subjected to images which either do not contain any object (usually captured in U_K) or they contain objects unknown to the detection system (U_U).

3.1. Open-Set Protocol for Object Detection

Though the current object detection protocols proposed in the PASCAL VOC and MSCOCO challenges have been widely accepted by the research community, these protocols do not contain images of random objects. This means that these protocols do not explicitly ensure the inclusion of images with objects from U. One might argue that this is not true in the case of MSCOCO where out of the 5000 validation images 969 do not have objects from any of the 80 object categories. Here, we wish to clarify that though the MSCOCO dataset has 80 object categories, the original dataset had 90 object categories. Rather than excluding all the images containing the additional 10 object categories, the annotations for these objects were simply removed in the protocol. This means that instances of these objects are seen during training, so that the many of the unknown unknowns (U_U) actually are known unknowns (U_K).

Our open-set object detection protocol differs from the traditional protocols in this aspect. In addition to the images from PASCAL VOC 2007 test set, we select the 23008 images from MSCOCO training set that do not contain any of the known PASCAL VOC objects. We use a subset of these MSCOCO images, specifically 4952 for our experiments in Fig. 3 and Fig. 4. This subset is referred to as Wilderness Ratio 1 (WR_1) as described in Sec. 4.2. The remainder of experiments in Sec. 4.2 use all 23008 images from MSCOCO.

Algorithm	**PASCAL VOC**	**Open-Set** WR_1
Faster R-CNN	81.86%	77.09%
RetinaNet	79.29%	73.81%
YOLOv2	75.89%	67.54%
Mask R-CNN	81.70%	77.05%
Dropout Sampling *(Faster R-CNN)*	78.15%	71.07%

Table 1: PERFORMANCE SUMMARIZATION (MAP) *We provide the mean average precision (mAP) values for the detectors used in our experiments. Performance is reported for the standard PASCAL VOC 2007 test set and our open-set protocol WR_1. WR_1 represents a wilderness ratio of 1, which means that for each image from the PASCAL test set we add an image from MSCOCO that does not contain any of the 20 PASCAL objects.*

A very important requirement of our protocol in order to maintain its true open-set nature is to restrict the type of datasets used for training the detectors. Any detector attempting to evaluate using our protocol must not train on any data from MSCOCO, more specifically, any detection dataset that contains instances of the 60 objects that are unique to MSCOCO.

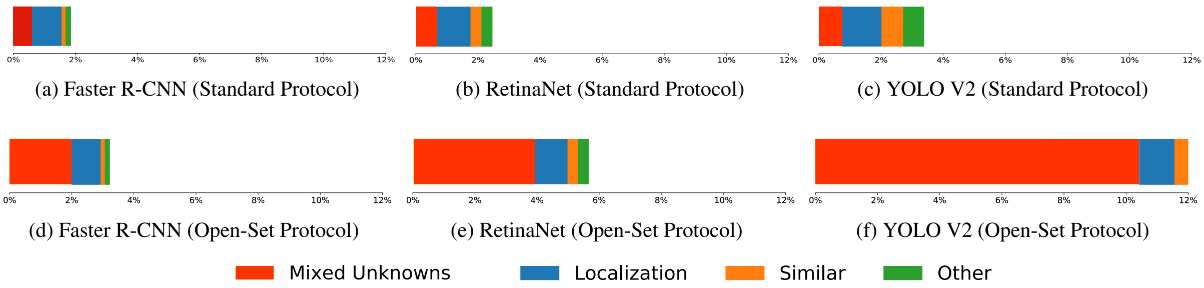
The unsolved problem When using a combination of datasets, since the samples marked in one dataset being considered as unknowns may not be labeled in the other dataset, their absence from training images cannot be guaranteed. This leads to the fact that all unknowns being identified as mixed unknowns U_M. As explained, the presence of mixed unknowns can never be avoided but reducing their number may be one of the primary steps for the research community to make progress in open-set object detection.

3.2. Experimental setup

In our experiments, we investigate four object detection networks, i.e., Faster R-CNN, RetinaNet, YOLOv2 and Mask R-CNN. For each of these networks, we use a publicly available ResNet-50 backbone that is pretrained on the ImageNet classification problem. In order to further enhance the performance of the networks, we employ feature pyramid networks (FPN) [15]. While for Faster R-CNN, RetinaNet and Mask R-CNN experiments we used detectron [8] for YOLOv2 [22] we used their implementation along with the configuration and weights provided for the model trained on PASCAL VOC dataset. We also provide the detectron training files along with trained models on our project page,* including all the evaluation scripts used in this paper along with the custom protocol splits.

The performance of all of the networks in terms of mean average precision (mAP) is summarized in Tab. 1. As we can see both in the PASCAL VOC column and in our open-set

*https://github.com/Vastlab/Elephant-of-object-detection



(a) Faster R-CNN (Standard Protocol)
(b) RetinaNet (Standard Protocol)
(c) YOLO V2 (Standard Protocol)
(d) Faster R-CNN (Open-Set Protocol)
(e) RetinaNet (Open-Set Protocol)
(f) YOLO V2 (Open-Set Protocol)

Figure 3: IMPACT OF OPEN-SET ON DETECTION *We diagnose the types of errors made by three object detectors under closed-set and open-set conditions at an operating point of Recall* $= 0.3$. *All detectors are trained to detect the 20 classes from PASCAL VOC 2007-2012. In the top row, detectors are evaluated on the PASCAL VOC 2007 test set, i.e., under closed-set conditions. In the bottom row, we test the same detectors in an open-set condition where, in addition to the PASCAL VOC 2007 test set, the detectors are also presented with an equivalent number of images from MSCOCO that do not contain any of the known (PASCAL) objects, but rather contain some other objects which are labeled in MSCOCO. From the magnitude of the mixed unknowns it may be observed that the detectors rapidly confuse objects they were not trained to identify with known objects with an* $IOU \geq 0.1$. *The response to mixed unknown samples significantly varies across detectors.*

protocol, the mAP results differ only moderately between the employed networks. This would indicate that all networks perform similarly with additional unknown samples. In the next section, we show that this is not the case and, consequently, mAP is not the the most appropriate measure for evaluating open-set object detection.

Existing approaches: There are many approaches that attempt to solve the open-set classification task [10, 11], but the majority of these approaches have been either restricted to smaller datasets such as MNIST or focus on identifying unknown knowns, i.e., novel views of known classes such as adversarial samples, rather than identifying true unknowns. Lately, Miller *et al*. [19] proposed to address open-set object detection by using dropout sampling, which we here apply to our pre-trained Faster R-CNN network. We employ dropout with a default probability of 0.5 to the weights for layers leading to fc7, classification head and regression head. For ROIs from each image we perform 30 forward passes with dropouts, resulting in 30 unique detections per ROI, which are averaged to provide the actual results. As we show in Tab. 1,Fig. 5 and Tab. 2, this method is actually decreasing the performance on unknown samples drastically, and it still overlooks the elephant in object detection.

4. Analyzing Open-Set Object Detectors

4.1. Impact of Unknowns

Since detectors in the real world are deployed at a particular operating point, in our experiments we chose to use the operating point of $Recall = 0.3$. This means that we select confidence thresholds – separately for each detector and each class – such that 30 % of all known object instances in that class are correctly detected. On various other operating points such as $Recall = 0.1$, we report similar results in the supplementary material. We test three different approaches of network-based detectors: Faster R-CNN [24], RetinaNet [16] and YOLOv2 [22], which we trained using the training and validation sets of PASCAL VOC 2007 and 2012. In Fig. 3(a)-(c) we test these networks on the PASCAL VOC 2007 test set. For Fig. 3(d)-(f) we use our WR_1 open-set protocol detailed in Sec. 3.1.

In object detection, background errors are defined as regions having $IOU < 0.1$ with a ground truth object [12] while being classified as one of the known objects. According to our definition of open-set object detection, we interpret these background errors as errors originating from mixed unknowns U_M. When assuming that our labeled samples from MSCOCO do not contain classes from PASCAL VOC, we can further identify the *unknown unknowns* errors. If the detector detects a MSCOCO object as one of the known PASCAL VOC objects with an $IOU \geq 0.1$, it is identified as an *unknown unknowns* error, i.e., a sample from U_U has been mis-identified as being from K. Similarly, in any image if the detector makes a detection which has an $IOU < 0.1$ with objects from both U_U and K, it is considered as *mixed unknowns* error. We call these detections *mixed unknowns* because both PASCAL VOC and MSCOCO do not have a constraint that a member of U_U cannot be present in their images. Therefore, if a detection has an $IOU < 0.1$ with any of the labeled objects, it may still have an $IOU \geq 0.1$ with an object that was not labeled.

We use the definitions provided by Hoiem *et al*. [12] to diagnose the errors made by the detectors. Rather than using pie charts as in [12], we visualize via horizontal bar plots. In order to focus only on the errors, we clip the plots to 12% of the detections, the white region up to 100% depicts correct detections. Since all plots in Fig. 3 are made on a specific operating point of $Recall = 0.3$, they all represent an equal number of true positives, i.e., correctly classified

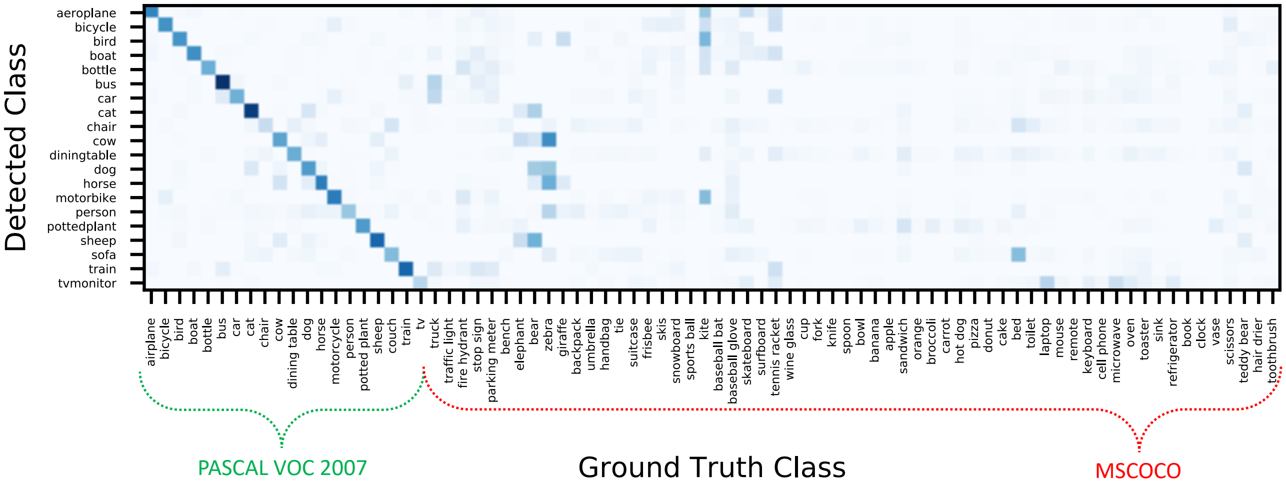


Figure 4: CONFUSION MATRIX FOR KNOWNS AND UNKNOWNS *We analyze the confusion of detections for the Faster R-CNN detector, which was the best performer in Fig. 3. Each detection with an* $IOU \geq 0.1$ *with an object is mapped to its ground-truth object in the confusion matrix. Apart from detecting a lot of known objects on the left side of the plot, the system also detects many unknown objects from MSCOCO as one of the PASCAL VOC classes. Major confusions exist for objects that belong to the same parent family, e.g., animals like elephants, bears and zebras are confused with cats, dogs, horses and sheep. Objects like tennis racket and baseball glove, which are not visually similar to any of the PASCAL objects, are detected as train, car, boat etc. For objects like tie, handbag and backpack we see almost no detection, which we attribute to their unannotated presence in PASCAL VOC training set, which helps detectors to learn to ignore them.*

known samples. It is interesting to see the different response of these detectors to *mixed unknowns* U_M.

It can be observed from Fig. 3(d) that while Faster R-CNN is the state-of-the-art two-stage object detector, its percentage of false positives almost doubles when compared to the closed-set performance in Fig. 3(a), attributed to the errors from mixed unknowns. When we compare Fig. 3(a) and (b) the performance seems almost equivalent as has been claimed in [16], but the response of the two state-of-the-art detectors to mixed unknowns varies drastically. It seems that RetinaNet is much more susceptible to these errors than the two stage detector, and this effect may be attributed to the one vs. rest loss function, since the two detectors are based on the same network architecture. The earlier single shot detector YOLOv1 [21] was known to make localization errors but was also claimed to be much less prone to background errors than detectors such as Fast R-CNN [6]. Surprisingly both YOLOv1 and YOLOv2 are the worst performers when evaluated under our open-set protocol. Since YOLOv2 is an advancement of YOLOv1 and performance of YOLOv1 was worse, we only provide results for YOLOv2 in Fig. 3. As discussed in Sec. 2, the poor performance of YOLOv2 to mixed unknowns may be attributed to its objectness score, which rather than providing the probability of an object from K being present, provides a probability of any object from Y being present.

Since we have inferred from Fig. 3 that unknown unknowns are frequently detected as one of the known objects, we further attempt to understand which objects tend to get confused with each other. In Fig. 4, we plot a confusion matrix of all the detections from Faster R-CNN that do not belong to the mixed unknown error category, i.e., they have an $IOU \geq 0.1$ with either U_U or K. It may be observed from Fig. 4 that most of the confusion is present among objects from the same parent family such as animals, furniture, appliances etc. Some objects that are not visually similar to any of the PASCAL objects, such as tennis racket, baseball glove or sandwich, also get detected as train, car, boat, aeroplane, dining table or potted plant. For objects that can be commonly found on a person, which is one of the objects with most instances in PASCAL such as tie, handbag, backpack, spoon and fork we see almost no detections. This may be attributed to the presence of these objects in PASCAL VOC training set, where they were identified as background and, hence, the detectors learned to avoid detecting them.

4.2. Detection and the wilderness

As we have observed from Fig. 3, unknown objects can have a very significant impact on a detector's performance. When object detection systems are deployed in the wild for real-world applications like robotics, detection is performed on frames in a video sequence. The majority of these frames may not contain any of the known objects K_U; rather, they may contain objects that either the system was trained to ignore as background U_K, or unknown unknown objects U_U that the detector was not trained to handle. While deploying such a system, an operating point is chosen either based on the detector's performance for one of the academic datasets or by applying it to a small subset of images from the application it is targeted toward. In neither of these cases,

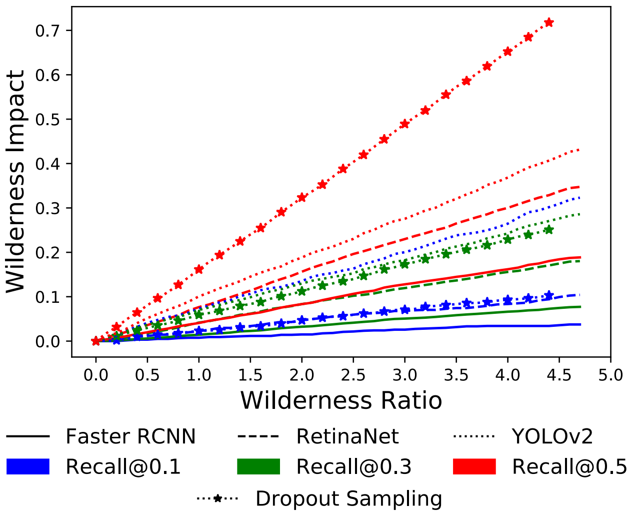


Figure 5: WILDERNESS IMPACT *The frequency with which unknown samples are presented to a detector can greatly impact the detectors performance. For an ideal detector, the impact of all levels of wilderness should be* 0*, i.e., the precision for closed-set and open-set testing should be identical. It may be observed that detectors are more significantly impacted when operating on higher recalls. Also, single shot detectors (RetinaNet, YOLOv2) are impacted more than the two stage detector (Faster R-CNN). The dropout sampling approach leads to much worse results.*

Algorithm	**Average Wilderness Impact**		
	Recall **.1**	Recall **.3**	Recall **.5**
Faster R-CNN	0.0195	0.0382	0.0971
RetinaNet	0.0521	0.0932	0.1785
YOLOv2	0.1603	0.1468	0.2192
Dropout Sampling *(Faster R-CNN)*	0.0511	0.1255	0.3569

Table 2: SUMMARIZING WILDERNESS IMPACT *We provide the average wilderness impact (AWI) of various detectors tested on several levels of wilderness at various recalls. Results for dropout sampling indicate the performance drop by using that approach.*

the impact of unknown unknowns is explicitly considered. The frequency with which these unknowns are encountered largely depends on the environment in which the detector is applied. Scheirer *et al.* [25] define a measure called openness, but it uses the number of known and unknown classes and ignores the frequency of unknowns. We formalize a new measure that captures the frequency of frames that may have unknowns, which we call *Wilderness*. We define the *Wilderness Ratio* as:

$$\text{Wilderness Ratio} = \frac{\#\text{ images with } U_M}{\#\text{ images with } K}.$$

In order to understand the variation in performance of a system when subjected to mixed unknowns as compared to being tested in closed-set conditions, we design an experiment in which the number of images with unknowns is increased by 10% of the images containing knowns, hence increasing the wilderness. For the images containing only unknowns, we use the MSCOCO split as described in Sec. 3.1. In order to understand the performance impact of these open-set conditions, we evaluate the performance of three detectors on various operating points at several levels of wilderness in Fig. 5. Our evaluations highlight the impact of the number of unknowns on a detectors performance.

To understand the impact of wilderness, we study the ratio of the precision values under closed-set and open-set conditions. Since for an ideal detector, the impact of wilderness should be 0, i.e., the precision under open-set conditions should be same as under closed set conditions, we subtract one:

$$\text{Wilderness Impact} = \frac{\text{Precision in closed-set}}{\text{Precision in open-set}} - 1$$

Simplifying the above equation, we compute the Wilderness Impact as:

$$\left\{\frac{TP_c}{TP_c + FP_c} \Big/ \frac{TP_c}{TP_c + FP_c + FP_o}\right\} - 1 = \frac{FP_o}{TP_c + FP_c}$$

where TP_c is the number of true positive detections from the PASCAL images, since there cannot be any true positives from the images responsible for wilderness we do not have a TP_o. The false positives resulting from the PASCAL images are denoted as FP_c while any detections made from the wilderness images are denoted as FP_o. As observed in Fig. 5, the wilderness impact for a detector increases as its operating point is changed to represent a higher recall. Moreover, it is clearly visible that single stage detectors such as RetinaNet and YOLOv2 are much more impacted by wilderness than the two stage detector Faster R-CNN. In order to consolidate the performance of a detector across various levels of wilderness, we suggest the measure of Average Wilderness Impact (AWI). For the wilderness impact curve, smaller AWI values represent better detectors. In order to calculate the AWI, we use the average of the wilderness impact values at various levels of wilderness. The AWI values for various detectors are consolidated in Tab. 2.

5. Making choices

In order to deploy a detector for a specific application, various factors need to be taken into consideration, i.e., choosing the object detector to be applied and its operating point.

5.1. Selecting Detector

Mean average precision (mAP) is the preferred choice for evaluating a detectors performance on various detection datasets. Since mAP provides a single number across all object classes, it makes comparison of detectors easy in order to decide the new state of the art in the field. As it has been observed in Tab. 1, while object detectors can have comparable mAPs in closed-set conditions on the standard academic datasets, their performance on open-set conditions may vary considerably. For example, though Faster R-CNN and RetinaNet both have a comparable performance on PASCAL VOC 2007 test set, when tested along with the same number of images that did not contain any of the known objects the mAP of Faster RCNN drops by close to 4 % and that of RetinaNet drops by 6 %. On the other hand, we can see from Tab. 2 that Faster R-CNN is much more stable when presented with unknown samples, which is not reflected in mAP. This observation leads us to believe that, though mAP provides a good measure for comparing performance in closed set, it alone cannot be used to provide performance measures in open-set conditions and better measures in the direction of the (average) wilderness impact need to be constituted.

5.2. Selecting Operating Point

In theory, for selecting an operating point, one can either use a threshold for a specific recall or a specific precision. While a precision-recall (PR) curve can be used in order to decide an operating point, its non-monotonic nature adds complexity. Often, PR curves are artificially made monotonic by updating the precision value at a recall r' with the maximum precision values for $r' \leq r$ [3]. This means that if a PR curve is used to obtain the operating point based on precision, the precision we are hoping for is not the one we would get.

Since, in practice, one attempts to obtain a balance between precision and recall rather than attempting to obtain either a high precision or a high recall, a need for a more complex evaluation metric arises. One such evaluation metric is the F_β score, which is defined as the weighted harmonic mean of precision and recall values:

$$F_\beta = (1 + \beta^2)\frac{precision * recall}{recall + \beta^2 precision}$$

Since F_β is just a combination of the precision and recall values and does not need a value for true negatives, which are needed by metrics such as accuracy and are not available for a detection problem, it naturally has become the second choice for the detection community [14, 1]. F_β can provide an excellent operating point, but it requires one to weight precision and recall, and often equal weights (F_1) are presumed. Though F_β may be used to decide an operating point for a detector, it too is unable to address the performance of detector under open-set conditions – see supplemental material for quantitative data.

6. Conclusion

This paper's primary goal is to provide an understanding of object detector performances in the real world. In order to achieve this goal, we have formalized object detection as an open-set problem. Though Miller *et al.* [19] also attempted to approach object detection as an open-set problem, they did not provide a formalization of the problem and its deep impact on the applicability of object detectors in the real world. Our open-set evaluation protocol enables researchers to estimate the performance of any object detector under real-world conditions. Rather than simply containing a fixed number of unknowns, this protocol varies the frequency of unknown inputs in what we call the wilderness ratio. This varying frequency allows to simulate a detectors performance in environments with varied levels of control over the input to the detector. Because an operating system cannot know the wilderness ratio in which it might operate, we introduce the novel Average Wilderness Impact (AWI) measure to quantify an algorithms sensitivity to unknown unknowns over a range of wilderness.

We investigated the open-set performance of three object detection networks that all have different approaches to handle the background. While the mAP of all these networks are similar in both closed-set and open-set evaluation, we found that the algorithms handle unknown objects very differently. The state-of-the-art two-stage multi-class detector Faster R-CNN, which uses an additional background class to assemble known unknown samples in a separate region of the feature space, has the lowest AWI showing it is the least influenced by samples of unknown objects. The state-of-the-art one-stage detector RetinaNet provides comparable performance in closed-set conditions, but has difficulty in rejecting unknown objects due to its one-versus-rest classifier. Finally, the objectness-based YOLO detectors have a high AWI and may not be ready to handle unknown objects well. We attribute this to the fact that their objectness score is not just high for known but also for unknown objects. This makes rejecting unknowns based on the objectness score difficult. Thus, we believe that the type of classifier used by the detectors to identify backgrounds versus knowns highly impacts their performance in the open-set protocol.

There has been significant progress in zero-shot, one-shot, few-shot, and incremental learning [13, 30, 5, 20, 28], which can be applied to object detection. However, if detectors incorrectly but confidently classify unknowns, there is no reason for a system to consider learning these objects as new classes. Even if the system was robust to unknowns and was to simply ignore unknown objects as ”background”, it could not learn them as new objects. Therefore, we consider it important that detection systems eventually learn to create a separation between background and unknown objects, enabling new objects to be identified. Currently, there is no such architecture and a design is left for future work. This paper lays the ground for research to eventually progress in this direction while providing new open-set evaluation protocols and metrics as the first steps. We hope that these steps guide object detection research toward detectors that are robust even beyond academic datasets.

References

[1] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. In *European Conference on Computer Vision (ECCV)*. Springer, 2012. 8

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 88(2), 2010. 1

[3] M. Everingham and J. Winn. The PASCAL Visual Object Classes challenge 2012 (VOC2012) development kit. Technical report, Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL), 2011. 8

[4] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010. 2

[5] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 8

[6] R. Girshick. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*. IEEE, 2015. 1, 2, 6

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014. 1, 2

[8] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. `http://github.com/facebookresearch/detectron`, 2018. 4

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(9), 2015. 2

[10] M. Hein, M. Andriushchenko, and J. Bitterwolf. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019. 5

[11] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017. 5

[12] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *European Conference on Computer Vision (ECCV)*. Springer, 2012. 5

[13] X. Li, Y. Guo, and D. Schuurmans. Semi-supervised zero-shot classification with label representation learning. In *International Conference on Computer Vision (ICCV)*. IEEE, 2015. 8

[14] D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3D object detection with RGBD cameras. In *International Conference on Computer Vision (ICCV)*. IEEE, 2013. 8

[15] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 4

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 1, 3, 5, 6

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*. Springer, 2016. 1, 2

[19] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2018. 3, 5, 8

[20] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. iCaRL: Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 8

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. 1, 3, 6

[22] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. In *Conference on Computer Vision and Pattern Recognition*. IEEE, 2017. 1, 3, 4, 5

[23] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv:1804.02767*, 2018. 3

[24] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*. Neural Information Processing Systems Foundation, 2015. 1, 2, 5

[25] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(7), 2013. 3, 7

[26] W. J. Scheirer, L. P. Jain, and T. E. Boult. Probability models for open set recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(11), 2014. 3

[27] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv:1312.6229*, 2013. 1, 2

[28] K. Shmelkov, C. Schmid, and K. Alahari. Incremental learning of object detectors without catastrophic forgetting. In *International Conference on Computer Vision (ICCV)*. IEEE, 2017. 8

[29] R. G. Vidal, S. Banerjee, K. Grm, V. Struc, and W. J. Scheirer. UGˆ2: A video benchmark for assessing the impact of image restoration and enhancement on automatic visual recognition. In *Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018. 3

[30] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning – a comprehensive evaluation of the good, the bad and the ugly. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 8

[31] Y. Yakimovsky. Boundary and object detection in real world images. *Journal of the Association for Computing Machinery (JACM)*, 23(4), 1976. 1

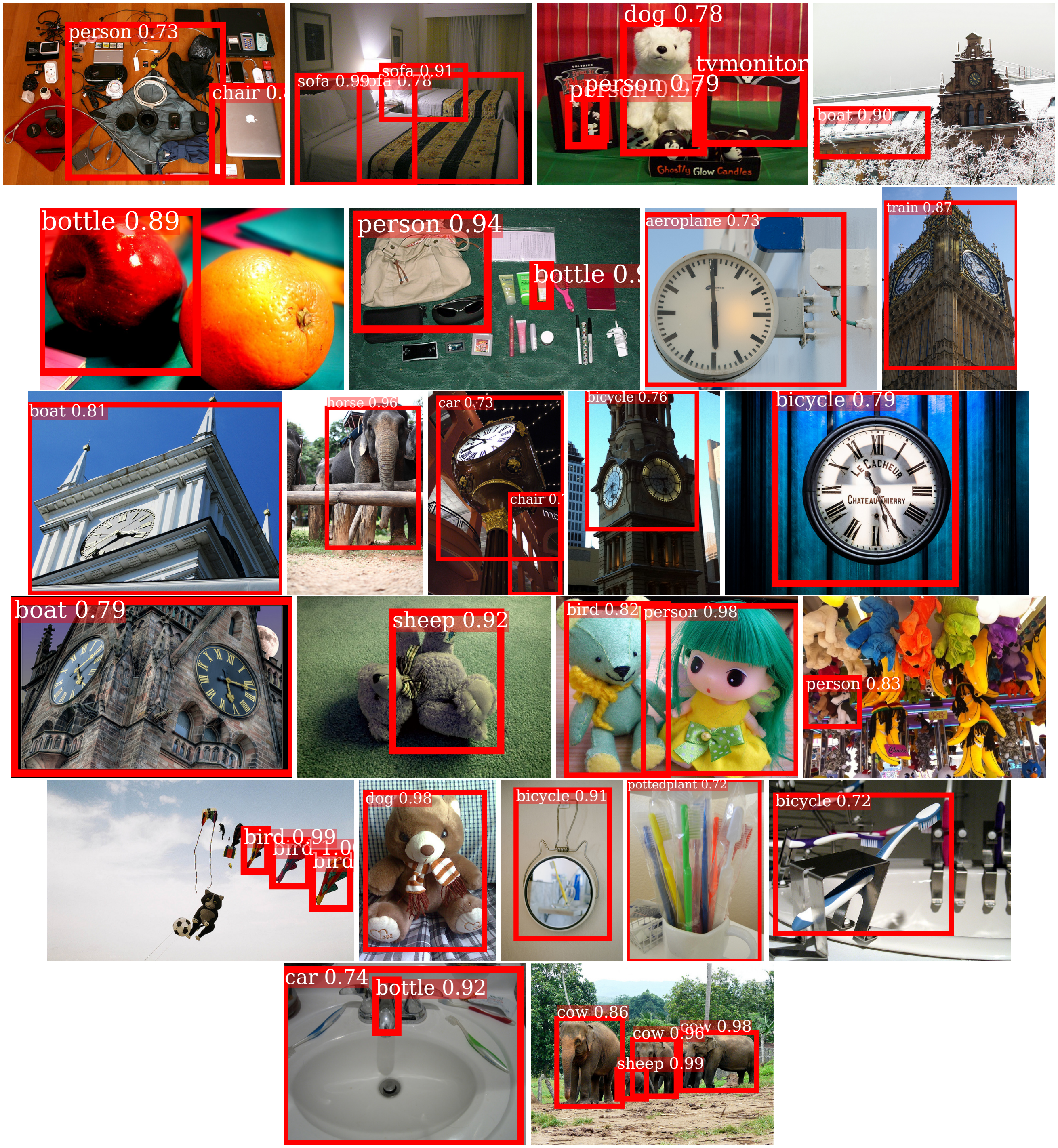


Figure 6: MISDETECTIONS FROM FASTER R-CNN *These results are from the Faster R-CNN network used in our paper. The network was trained on the PASCAL VOC 2007-2012 training and validation set while the above images belong to MSCOCO dataset. Ideally, the network was supposed to reject all the above detections as background.*

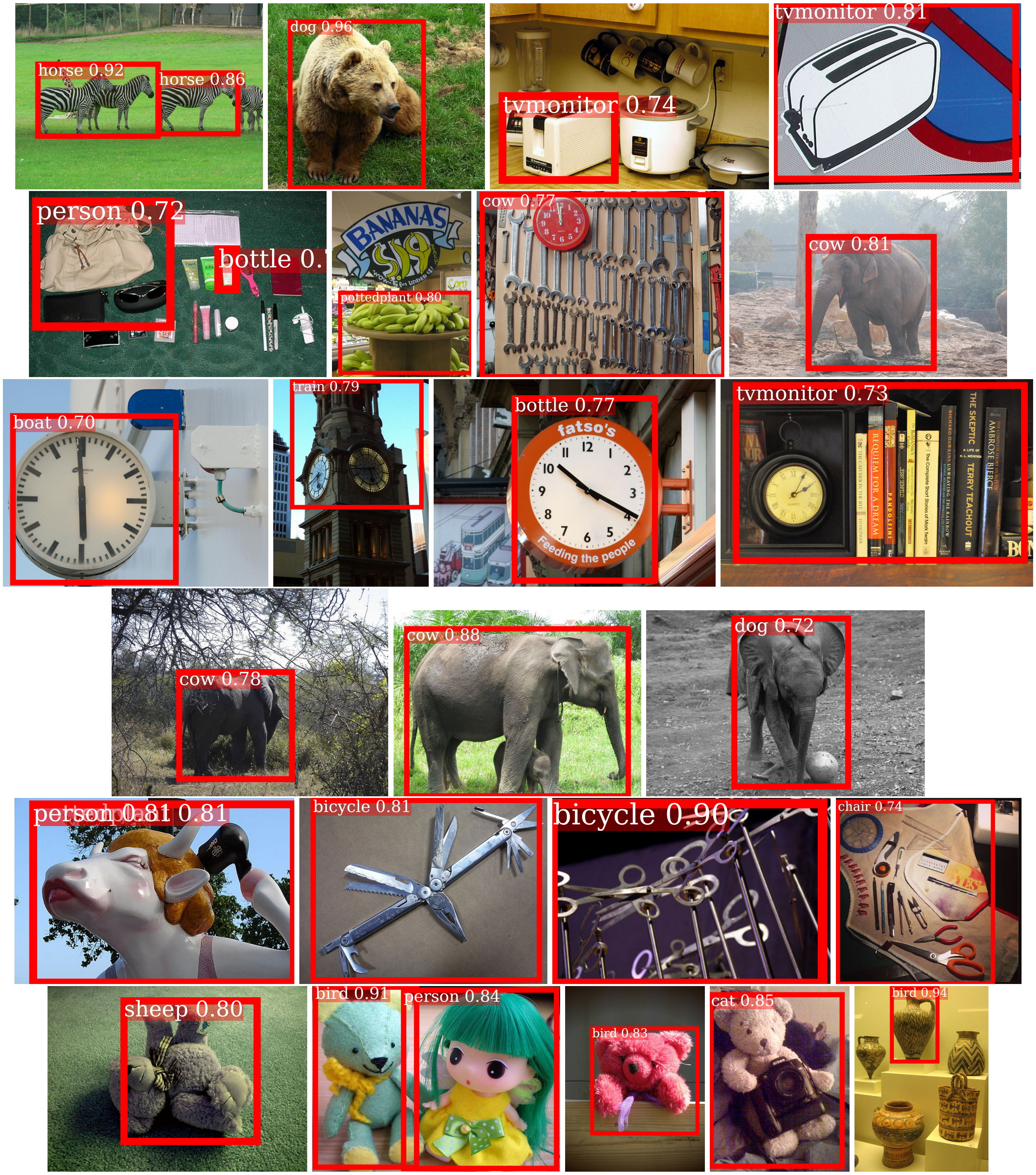
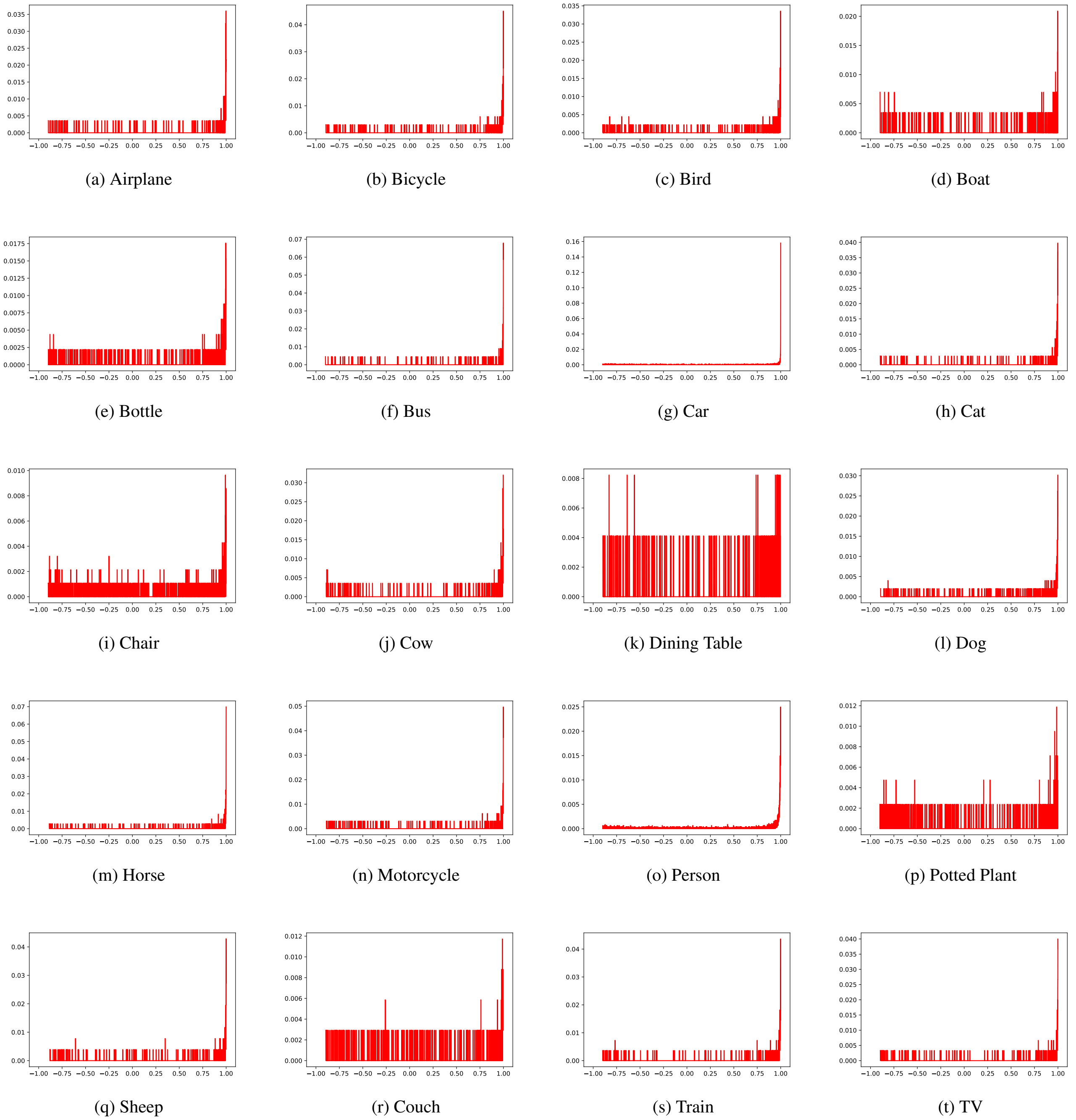


Figure 7: MISDETECTION FROM RETINANET *These results are from the RetinaNet network used in our paper. The network was trained on the PASCAL VOC 2007-2012 training and validation set while the above images belong to MSCOCO dataset. Ideally the network was supposed to reject all the above detections as background.*



(a) Airplane

(b) Bicycle

(c) Bird

(d) Boat

(e) Bottle

(f) Bus

(g) Car

(h) Cat

(i) Chair

(j) Cow

(k) Dining Table

(l) Dog

(m) Horse

(n) Motorcycle

(o) Person

(p) Potted Plant

(q) Sheep

(r) Couch

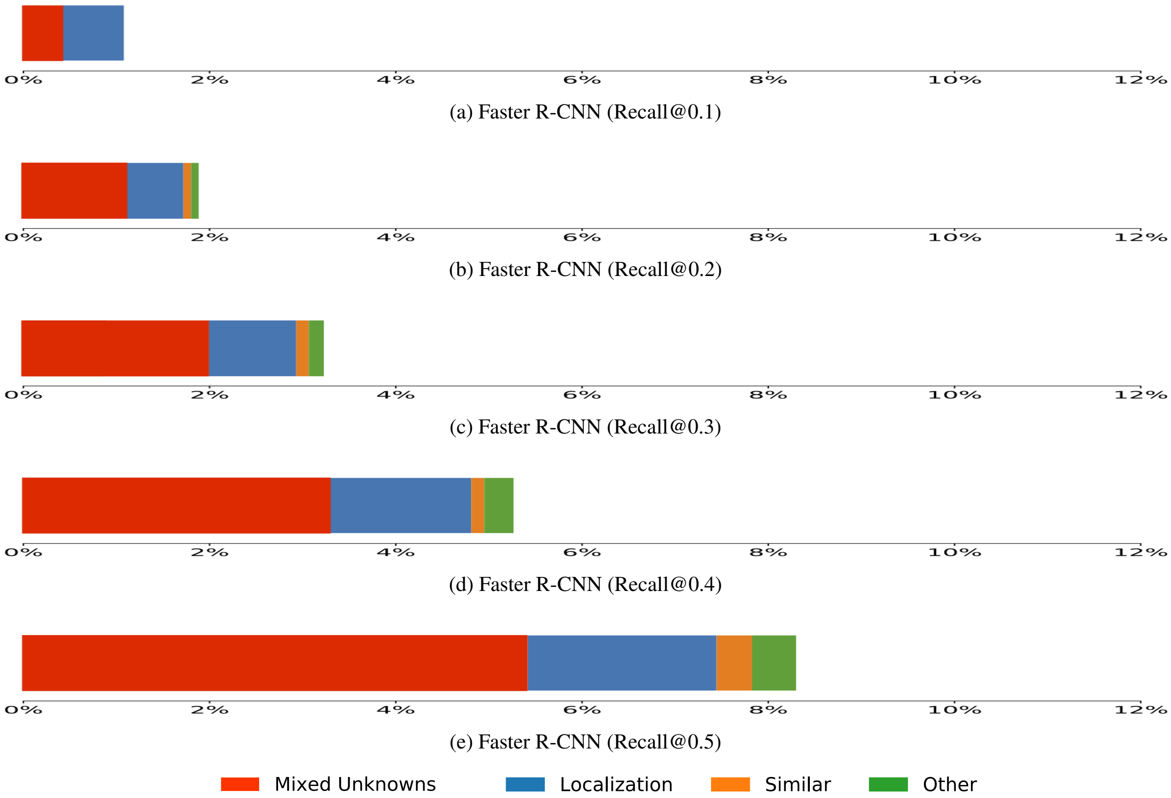
(s) Train

(t) TV

Figure 8: OBJECT VS BACKGROUND PROBABILITY DISTRIBUTION *One might think that for open set detections it would be sufficient to reject when the background class is greater than any other class. Unfortunately, this fails miserably. To see why, we compare the distribution of probability scores of correct detections (p_i) to background probability (p_b) for Faster R-CNN on the PASCAL VOC 2007 test set. This figure shows the histogram of ($p_i - p_b$), with the x-axis ranging from -1 ($p_i = 0, p_b = 1$) to 1 ($p_i = 1, p_b = 0$). As seen in the histograms, for a significant number of detections the difference is negative, i.e., the probability score for an ROI being background is higher than the probability for the actual object in the ROI. We also summarize the Mean and Standard Deviation values for each of the classes in Table 1. As explained in Section 2 of the main paper (Multi-Class Classifiers with Background), this skew indicates that we should not perform the evaluation as for a classification problem.*

Class	**Mean**	**Standard Deviation**
aeroplane	0.66	0.598
bicycle	0.6842	0.539
bird	0.6129	0.6284
boat	0.3873	0.6842
bottle	0.4747	0.6718
bus	0.6737	0.5411
car	0.6555	0.5947
cat	0.7638	0.4953
chair	0.2317	0.6881
cow	0.6331	0.6023
diningtable	0.264	0.6867
dog	0.7326	0.4882
horse	0.7319	0.5088
motorbike	0.6125	0.6161
person	0.6157	0.611
pottedplant	0.2321	0.7348
sheep	0.5874	0.6021
sofa	0.3421	0.6321
train	0.6825	0.5756
tvmonitor	0.653	0.5939

Table 3: OBJECT VS BACKGROUND PROBABILITY DISTRIBUTION



(a) Faster R-CNN (Recall@0.1)

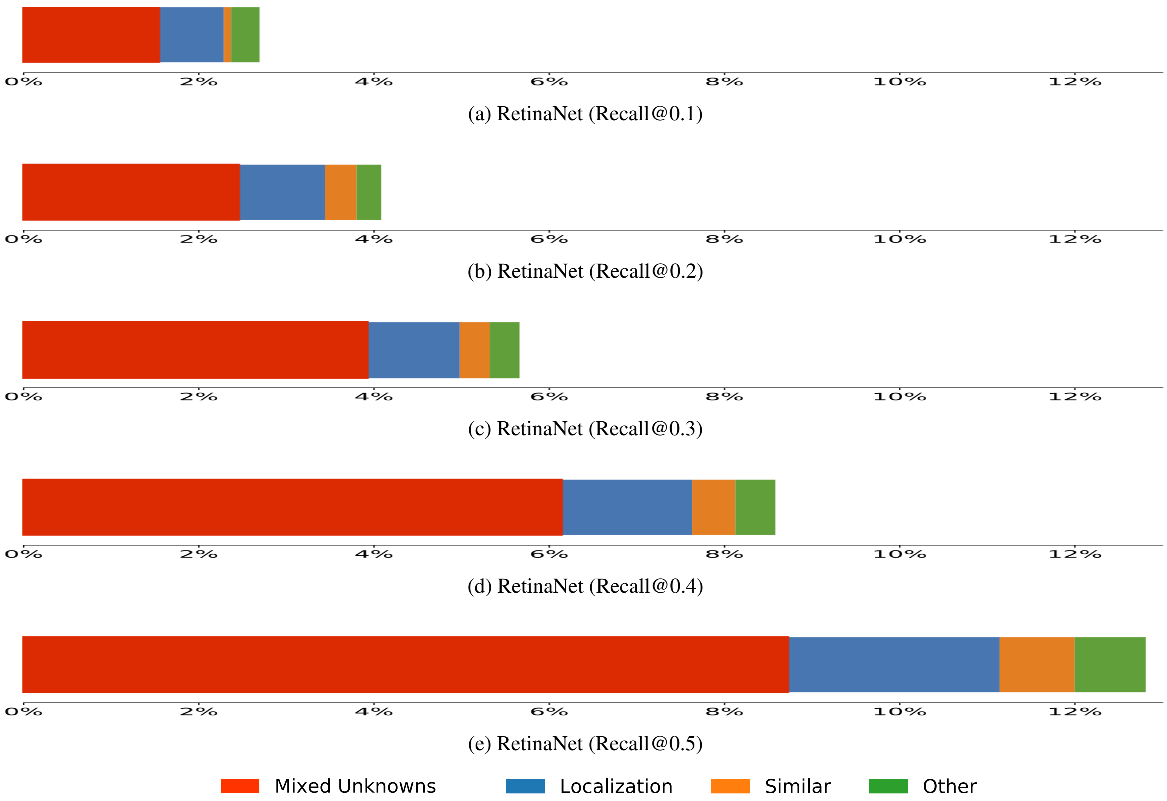
(b) Faster R-CNN (Recall@0.2)

(c) Faster R-CNN (Recall@0.3)

(d) Faster R-CNN (Recall@0.4)

(e) Faster R-CNN (Recall@0.5)

Figure 9: IMPACT OF UNKNOWNS AT WR1 ON FASTER RCNN *The errors are plot upto 12% the white region upto 100% representing correct detections. From the magnitude of the unknown unknowns it may be observed that the detectors rapidly confuse objects they were not trained to identify with known objects with an* $IOU \geq 0.1$.



(a) RetinaNet (Recall@0.1)

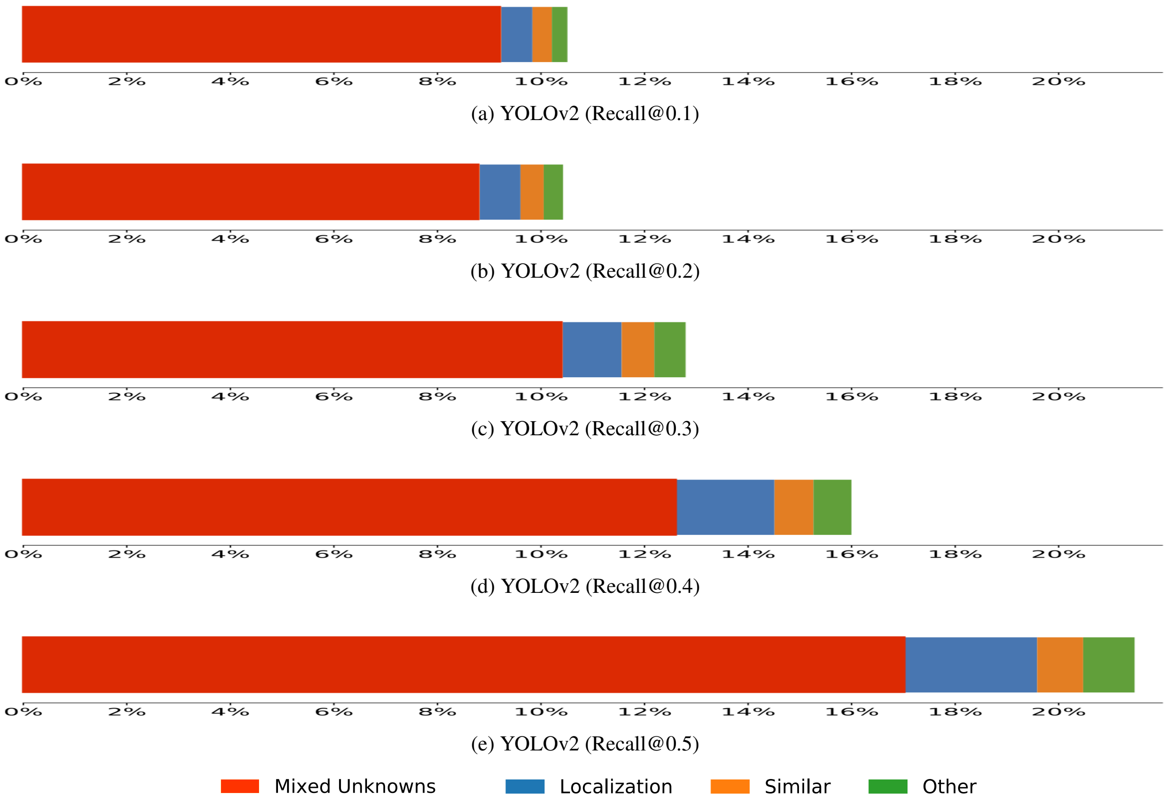
(b) RetinaNet (Recall@0.2)

(c) RetinaNet (Recall@0.3)

(d) RetinaNet (Recall@0.4)

(e) RetinaNet (Recall@0.5)

Figure 10: Impact of Unknowns at WR1 on RetinaNet *The errors are plot upto 13% the white region upto 100% representing correct detections.*



(a) YOLOv2 (Recall@0.1)

(b) YOLOv2 (Recall@0.2)

(c) YOLOv2 (Recall@0.3)

(d) YOLOv2 (Recall@0.4)

(e) YOLOv2 (Recall@0.5)

Figure 11: Impact of Unknowns at WR1 on YOLOv2 *The errors are plot upto 22% the white region upto 100% representing correct detections.*

WR	β=0.5				β=1.0				β=1.5			
	F Score	Prec.	Recall	Thre.	F Score	Prec.	Recall	Thre.	F Score	Prec.	Recall	Thre.
0.0	0.85	87.91	77.46	0.62	0.83	79.25	86.07	0.49	0.84	78.52	86.89	0.48
0.25	0.81	87.38	73.77	0.65	0.80	87.38	73.77	0.65	0.81	69.51	86.89	0.48
0.5	0.75	84.39	70.90	0.69	0.77	81.08	73.77	0.65	0.77	65.19	84.43	0.51
0.75	0.68	82.23	66.39	0.72	0.74	76.89	70.90	0.69	0.75	68.98	77.46	0.62
1.0	0.65	70.87	73.77	0.65	0.72	70.87	73.77	0.65	0.73	65.85	77.46	0.62
1.25	0.63	71.19	70.90	0.69	0.71	71.19	70.90	0.69	0.72	62.17	77.46	0.62
1.5	0.61	75.36	65.16	0.73	0.70	75.36	65.16	0.73	0.71	64.75	73.77	0.65
1.75	0.58	71.30	65.16	0.73	0.68	71.30	65.16	0.73	0.69	60.40	73.77	0.65
2.0	0.57	69.43	65.16	0.73	0.67	69.43	65.16	0.73	0.68	57.88	73.77	0.65
2.25	0.54	72.06	60.25	0.77	0.66	67.83	63.93	0.74	0.67	59.45	70.90	0.69
2.5	0.52	69.34	60.25	0.77	0.64	69.34	60.25	0.77	0.65	55.45	70.90	0.69
2.75	0.49	65.04	60.25	0.77	0.63	65.04	60.25	0.77	0.64	51.95	70.90	0.69
3.0	0.49	64.47	60.25	0.77	0.62	64.47	60.25	0.77	0.63	50.44	70.90	0.69
3.25	0.47	62.82	60.25	0.77	0.62	62.82	60.25	0.77	0.62	57.45	64.75	0.74
3.5	0.46	61.51	60.25	0.77	0.61	61.51	60.25	0.77	0.62	56.23	64.75	0.74
3.75	0.45	59.51	60.25	0.77	0.60	59.51	60.25	0.77	0.61	54.30	64.75	0.74
4.0	0.44	58.57	60.25	0.77	0.59	58.57	60.25	0.77	0.61	52.84	64.75	0.74
4.25	0.42	55.89	60.25	0.77	0.58	56.64	59.43	0.77	0.60	50.64	64.75	0.74

Table 4: Using F_β to select operating point for Cow *We show various values of β that might be used to choose an operating point for RetinaNet in order to detect the cow class. For a provided set of detections, the threshold that provides the maximum value of F_β is chosen. For each such operating point we provide the threshold, F_β score, precision and recall values. The operating point may also be chosen at various levels of open-set conditions or wilderness ratios (WR). It may be observed from the above table that the thresholds selected for a specific value of β at a certain wilderness ratio do not generalize to other levels of wilderness.*