Query-Based Hard-Image Retrieval for Object Detection at Test Time

Edward Ayers[*], **Jonathan Sadeghi**[*], **John Redford, Romain Mueller**[†], **Puneet K. Dokania**[†]

Five AI Ltd., UK
edward.ayers@five.ai, jonathan.sadeghi@five.ai, john.redford@five.ai
romain.mueller@five.ai, puneet.dokania@five.ai

Abstract

There is a longstanding interest in capturing the error behaviour of object detectors by finding images where their performance is likely to be unsatisfactory. In real-world applications such as autonomous driving, it is also crucial to characterise potential failures beyond simple requirements of detection performance. For example, a missed detection of a pedestrian close to an ego vehicle will generally require closer inspection than a missed detection of a car in the distance. The problem of predicting such potential failures *at test time* has largely been overlooked in the literature and conventional approaches based on detection uncertainty fall short in that they are agnostic to such fine-grained characterisation of errors. In this work, we propose to reformulate the problem of finding "hard" images as a query-based hard image retrieval task, where queries are specific definitions of "hardness", and offer a simple and intuitive method that can solve this task for a large family of queries. Our method is entirely post-hoc, does not require ground-truth annotations, is independent of the choice of a detector, and relies on an efficient Monte Carlo estimation that uses a simple stochastic model in place of the ground-truth. We show experimentally that it can be applied successfully to a wide variety of queries for which it can reliably identify hard images for a given detector without any labelled data. We provide results on ranking and classification tasks using the widely used RetinaNet, Faster-RCNN, Mask-RCNN, and Cascade Mask-RCNN object detectors. The code for this project is available at https://github.com/fiveai/hardest.

Introduction

How difficult (or hard) an image can be for a model at inference? A large body of work has already paid significant attention in defining the notion of hard (or useful) images for standard classification and object detection tasks. The main objective of these works was to either perform active learning mainly to reduce the labelling and training cost (Gal, Islam, and Ghahramani 2017; Choi et al. 2021; Roy, Unmesh, and Namboodiri 2018; Aghdam et al. 2019; Haussmann et al. 2020; Yu et al. 2022; Feng et al. 2019), or to improve the training mechanism (Shrivastava, Gupta, and Girshick 2016; Lin et al. 2017; Pang et al. 2019; Chen et al. 2020),
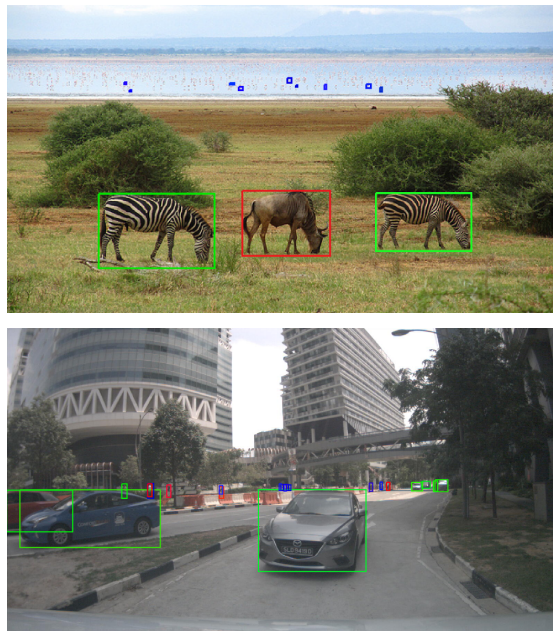

Figure 1: Examples of images showing that "hardness" is task-specific for object detectors. Green, red, and blue boxes denote true positives, false positives, false negatives, respectively. (Top) Output of RetinaNet on an image from the COCO *val* set. Is this image hard for the detector because of the misclassified 'wildebeest' in the foreground or because of the undetected 'birds' in the background? (Bottom) Output of Cascade-RCNN on a frame from the nuImages *val* set. Should this image be considered to be hard for this detector because of the undetected 'pedestrians' even though they are further away compared to the correctly detected 'cars'? Clearly, there is no single correct answer unless a specific requirement of performance is described.

perform metric learning (Suh et al. 2019; Bucher, Herbin, and Jurie 2016), or even to improve the reliability of object detectors towards out-of-distribution samples (Harakeh and Waslander 2021; Du et al. 2022). All these approaches attempt to improve the performance or robustness of object detectors by focusing on their training and generally assume

---

that ground-truth labels are available. In contrast, our objective is to identify hard images for a given *trained* detector, that is images for which the detector is likely to make a mistake, without having access to ground-truth labels.

First note that, in contrast to classification where there is a well-defined definition of error (a misclassification), there are multiple ways to define mistakes in object detection. For example, an image can be considered hard if a detector produces many false-positives on it, but this by far not the only possibility. Another one is to say that an image is hard if the detector makes many mistakes on close objects rather than on distant ones. Therefore, what makes an image hard depends on the situation at hand and one should be able to quantify "hardness" accordingly by defining the associated notion of a mistake. We illustrate this point on Figure 1, where we show two example images that exhibit a variety of possible definitions of errors for object detection. Note that the widely used uncertainty metrics that are often used to quantify "hardness", such as entropy (Shannon 1948) and Dempster-Shafer (Sensoy, Kaplan, and Kandemir 2018), to name a few, are completely agnostic to such requirements. In this work, we revisit these notions and take a step towards defining and quantifying requirement-specific hardness metrics without having access to the ground-truth labels.

Our main contributions are as follows. First, we provide a framework that allows to define a large family of requirement-specific hardness metrics that define a notion of error. We call them *hardness queries*. The central idea is to provide fine-grained error behaviours of a given detector at test time (e.g., set of false-negative (FN) and false-positive (FP) bounding boxes) to allow users to compose a large family of complex queries. For example, if provided with the set of *potential* false positive and false negative bounding boxes, a user may compose a query and define hardness as a metric that quantifies the mistake severity on these sets. Let us say that the user's query is to "find images where the number of false-positive boxes strictly dominate the number of false-negative ones". Then, the metric quantifying such query would simply be $\mathbb{1}_{|\text{FN}|>|\text{FP}|}$, where each image for which the indicator turns out to be one would be classified as the hard image under the the given query. In such a way, many queries, much more expressive than this one (discussed in the main paper), can be composed depending on the requirements of the application.

Since, in the absence of ground-truth annotations, it is not possible to define the set of fine-grained errors mentioned above, we propose a simple approach to define a distribution of *pseudo* ground-truth labels (a multivariate Bernoulli distribution) that is then used to define a distribution over these error sets. This distribution is then employed via an efficient Monte Carlo sampling procedure to estimate the expected query-based hardness per image. Through extensive experiments using a suite of widely used object detectors (RetinaNet, Faster-RCNN, Mask-RCNN, and Cascade Mask-RCNN), we show that our method provides state-of-the-art performance (with significant margins in many cases) on ranking and classification tasks that tackle the problem of finding query-based hard images at test time.

Defining Query-Based Hardness

Let us denote a general object detector as a function D_η that maps an input image $\mathbf{x}$ to a list of *detection instances*[1] $\hat{\mathbf{y}} = D_\eta(\mathbf{x})$, where each instance $\hat{y}_i = (b_i, c_i, s_i)$ consists of a regressed bounding box b_i enclosing the i-th detected object, a class label $c_i \in \{1, \cdots, K\}$, and a detection score $s_i \in [0, 1]$. Here η is the threshold such that only instances with $s_i \geq \eta$ are produced by the detector D_η. We denote the ground-truth instances by $\mathbf{y} = \{y_i : i = 1, \ldots\}$, where each instance is of the form $y_i = (b_i, c_i)$. We do not make any further assumptions regarding the architecture or the training procedure of the detector.

Image Hardness the Conventional Way

A widely accepted approach to capture how difficult or hard a certain input is for a given classification model is to compute the entropy of its predictive distribution (Gal, Islam, and Ghahramani 2017). The concept can be extended to object detectors as well (Roy, Unmesh, and Namboodiri 2018). Assuming that we have access to the full probability vector $\mathbf{p}_i \in [0, 1]^K$ over the K classes for each bounding box, we can define the total *entropy* per image as

$$H_\mathbf{x}(\hat{\mathbf{y}}) = -\sum_{i=1}^{|\hat{\mathbf{y}}|} \sum_{k=1}^{K} p_i(k) \log p_i(k), \qquad (1)$$

where we sum over all the detected instances i, and $p_i(k)$ is the probability with which the i-th bounding box belongs to the k-th category. Eq. (1) captures the total uncertainty associated with an image as the sum of the entropy for the predicted bounding boxes. This can be easily adapted to one-vs-all binary setups in which case $p_i(k)$ are independent binary probabilities for each class.

Recently, the Dempster-Shafer framework (Dempster 2008; Sensoy, Kaplan, and Kandemir 2018) received significant attention in image classification tasks and has been shown to outperform entropy in estimating uncertainty for certain out-of-distribution detection and domain shift problems (Liu et al. 2020; Pinto et al. 2022). Contrary to the entropy, this metric is defined in terms of the *evidence* per class (also known as belief) instead of their probabilities, which is defined as $e_i(k) = \exp l_i(k)$, where $\mathbf{l}_i$ is the logit corresponding to $\mathbf{p}_i$. For object detectors, the per-image Dempster-Shafer theory based evidential uncertainty can then be computed as:

$$DS_\mathbf{x}(\hat{\mathbf{y}}) = \sum_{i=1}^{|\hat{\mathbf{y}}|} \frac{K}{K + \sum_k e_i(k)}, \qquad (2)$$

where, similarly to $H_\mathbf{x}(\hat{\mathbf{y}})$, we sum over all the detected instances i in the image.

Limitations   Though both $H_\mathbf{x}(\hat{\mathbf{y}})$ and $DS_\mathbf{x}(\hat{\mathbf{y}})$ can be used at inference to quantify how likely it is that the given image is hard for the detector, using such measures of uncertainty

---

[1]Here $\hat{\mathbf{y}}$ denotes the *final* output of the detector (e.g. after NMS or any other post-processing) and η is typically chosen according to the task-specific requirements for tradeoffs of precision and recall.

to quantify "hardness" is crude as it does not allow one to specify arbitrary requirements of performance. In real-world applications, it is crucial to have a more fined-grained characterisation of detection errors since different types of errors can have vastly different consequences for the downstream task (see Figure 1 for an illustration). Both $H_{\mathbf{x}}(\hat{\mathbf{y}})$ and $DS_{\mathbf{x}}(\hat{\mathbf{y}})$ do not allow specifying such requirements and hence suffer from the fact that they are *agnostic* to the specific performance measure under consideration.

Query-Based Hardness

We now introduce a simple framework that allows to define a large family of hardness metrics, which we denote as *hardness queries* in what follows. Let us assume for now that we have access to the ground-truth bounding boxes $\mathbf{y}$ for each image $\mathbf{x}$. This requirement will obviously not be fulfilled at test time but we present in the next section a method that can approximately resolve this limitation. Given detection instances $\hat{\mathbf{y}} = D_\eta(\mathbf{x})$ and ground-truth bounding boxes $\mathbf{y}$, we can use the Hungarian algorithm on a thresholded intersection-over-union cost matrix (Forsyth and Ponce 2012) to define the error sets $e(\hat{\mathbf{y}}, \mathbf{y}) \in \{fp(\hat{\mathbf{y}}, \mathbf{y}), fn(\hat{\mathbf{y}}, \mathbf{y}), false(\hat{\mathbf{y}}, \mathbf{y})\}$, where fp and fn denote the set of false-positive and false-negative bounding boxes, respectively, and $false = fp \cup fn$. The set of false-negative boxes consists of those ground-truth instances that have not been associated to any detection.

A large family of domain-specific hardness queries capturing the error characteristics of a detector can be introduced using the above fine-grained information about the error categories of the detected boxes. For example, one could consider the relative position of the false-negatives and false-positives to be important when defining hardness, or might want to pay more attention to a particular class (for example, pedestrians) and define hardness accordingly. We introduce below a few query-specific hardness definitions and use them throughout our experiments. However, depending on the task, more expressive queries can easily be constructed.

Examples of hardness queries   In what follows, e denotes one of the error sets fp, fn, or $false$ of the detected instances on an image for a specific detector. For each e, we define the following hardness queries:

Total number of errors An image $\mathbf{x}$ is considered harder than $\mathbf{x}'$ if the detector makes a larger number of errors of a specific type e on that image, that is if $|e(\hat{\mathbf{y}}, \mathbf{y})| > |e(\hat{\mathbf{y}}', \mathbf{y}')|$. The corresponding query is defined by $\mathtt{Total}_{\mathbf{x}}(e(\hat{\mathbf{y}}, \mathbf{y})) = |e(\hat{\mathbf{y}}, \mathbf{y})|$.

Pixel-adjusted errors Here we consider errors on large objects to be more severe than errors on the smaller ones, and introduce a query that captures this requirement. To do so, we simply weight each error by the size of the corresponding bounding box as a fraction of the image area, and define

$$\mathtt{PixelAdj}_{\mathbf{x}}(e(\hat{\mathbf{y}}, \mathbf{y})) = \sum_{b \in e(\hat{\mathbf{y}}, \mathbf{y})} \frac{\mathtt{area}(b)}{\mathtt{area}(\mathbf{x})}, \quad (3)$$

Positive detections $\hat{\mathbf{y}}$ All detections $\hat{\mathbf{y}}_0$ Pseudo ground-truth $\bar{\mathbf{y}}$
$s_i > \eta$ $D_0(\mathbf{x}) = \{(b_i, c_i, s_i)\}_i$ Keep box i with prob. s_i

$\bar{\mathbf{y}} \sim P(\bar{\mathbf{y}})$

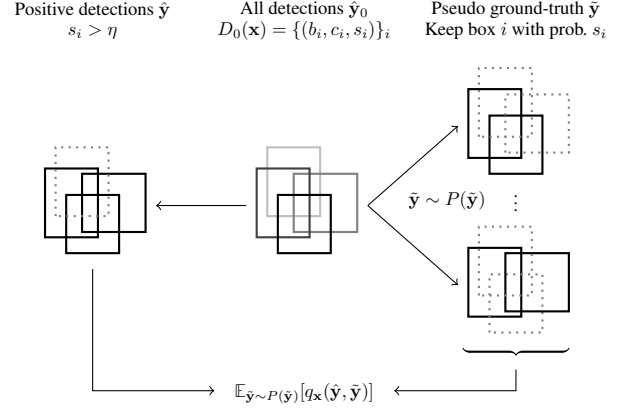$\mathbb{E}_{\bar{\mathbf{y}} \sim P(\bar{\mathbf{y}})}[q_{\mathbf{x}}(\hat{\mathbf{y}}, \bar{\mathbf{y}})]$

Figure 2: Illustration of our method: starting from all detections ($\eta = 0$) we generate pseudo-ground truth annotations $\bar{\mathbf{y}}$ by selecting each box with probability given by their detection score s_i and use them to evaluate positive detections $\hat{\mathbf{y}}$ ($s_i > \eta$) in place of the ground truth using the hardness query $q_{\mathbf{x}}$, see Algorithm 1 for details.

where $\mathtt{area}(.)$ denotes the area of the image or the bounding box.

Occlusion-aware errors Here we consider images with many occluded objects to be harder than the less cluttered ones. Since we do not have 3D information, we make the reasonable assumption that overlaps with true positive boxes is a good proxy for measuring occlusions. Similarly to the pixel-adjusted case, we quantify occlusion-aware hardness as:

$$\mathtt{OccAware}_{\mathbf{x}}(e(\hat{\mathbf{y}}, \mathbf{y})) = \sum_{\substack{b \in e(\hat{\mathbf{y}}, \mathbf{y}) \\ b' \in \mathtt{tp}(\mathbf{x})}} \frac{\mathtt{inter}(b, b')}{\mathtt{area}(b)}, \quad (4)$$

where, $\mathtt{inter}(b, b')$ is the area of the intersection between b and b'.

In the appendix, we show on Figure 10 that the Spearman's rank correlation coefficient between all of the proposed hardness queries varies widely, indicating that they are indeed able to capture different notions of hardness.

Quantifying Query-Based Hardness

In the previous section, we assumed knowledge of the ground-truth bounding boxes in order to define application-specific notions of hardness. In what follows, we show that this limitation can be circumvented by defining a *distribution* over possible ground-truths given detected instances (we call them pseudo ground-truths) and that this distribution can be used to estimate the hardness of an image.

Our main insight is to treat the detection scores of the detected instances as posterior predictive probabilities and use samples from their implied distribution to estimate a given hardness query. More precisely, for a given image $\mathbf{x}$, we first obtain the set of *all detected instances* $\hat{\mathbf{y}}_0 = D_0(\mathbf{x})$.[2] This

---

[2]In practice, we do not set η to be exactly zero but rather choose a very small value.

includes detections with $s_i < \eta$ which would appear as false negatives in the filtered detector output. For each detected instance $\hat{y}_i = (b_i, c_i, s_i) \in \hat{\mathbf{y}}_0$, we then introduce a Bernoulli distributed random variable $X_i \sim \text{Bernoulli}(s_i)$ parameterised by the corresponding detection score s_i. Therefore, if $|\hat{\mathbf{y}}_0| = m$, we construct a multivariate Bernoulli distribution with m independent variables. We finally generate pseudo ground-truth instances $\tilde{\mathbf{y}}$ by selecting each detection $\hat{y}_i \in \hat{\mathbf{y}}_0$ independently according to its Bernoulli distribution, i.e. if $X_i = 1$. This generates a distribution $P(\tilde{\mathbf{y}})$ over pseudo ground-truth instances.

Denoting by $q_{\mathbf{x}}(\hat{\mathbf{y}}, \tilde{\mathbf{y}})$ a generic hardness query (such as Eq. 3), we estimate the hardness of an image by computing the expectation of the corresponding query over the pseudo ground-truth using the following Monte Carlo estimator:

$$SS_{\mathbf{x}}(\hat{\mathbf{y}}; q) = \mathbb{E}_{\tilde{\mathbf{y}} \sim P(\tilde{\mathbf{y}})}[q_{\mathbf{x}}(\hat{\mathbf{y}}, \tilde{\mathbf{y}})] = \int q_{\mathbf{x}}(\hat{\mathbf{y}}, \tilde{\mathbf{y}}) \mathrm{d}P(\tilde{\mathbf{y}})$$

$$\approx \frac{1}{N} \sum_{n=1}^{N} q_{\mathbf{x}}(\hat{\mathbf{y}}, \tilde{\mathbf{y}}^{(n)}), \quad \tilde{\mathbf{y}}^{(n)} \sim P(\tilde{\mathbf{y}}), \quad (5)$$

where N is the number of Monte Carlo samples. We term this methodology *score sampling* and summarize it in Algorithm 1. Sampling $\tilde{\mathbf{y}} \sim P(\tilde{\mathbf{y}})$ is extremely efficient as it typically requires a low number (typically ≈ 10 in our experiments) of parallelizable Bernoulli trials. An illustration of our approach is given in Figure 2.

A key assumption in our approach is that a detector will generally assign a low probability to ambiguous (or incorrectly classified) objects. Therefore, even if the assigned probability of such an object is above the detection threshold, using this probability as the parameter of the Bernoulli distribution will likely lead to multiple unsuccessful trials and that bounding box will be included in the pseudo ground-truth only some of the time. Conversely, bounding boxes with a high detection score will be sampled more frequently and be included in the pseudo ground-truth most of the time. This assumption falls short when a detector is wrong with a very high confidence. Since modern detectors are highly accurate and also show good calibration properties (refer to Figure 5 in the appendix for confidence histograms), we do not expect this assumption to fail frequently. An example would be RetinaNet (Lin et al. 2017) which uses focal loss that has been shown to provide good calibration properties for image classification tasks (Mukhoti et al. 2020). However, as a future work, it would be interesting to explore the impact of varying the calibration of object detectors on our approach.

Experiments

We now illustrate the effectiveness of our approach in a variety of settings. The following assumes a hardness query q and a given detector D_η at a given score threshold η but we generally omit references to it for the ease of notation. Given an annotated dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) : i = 1, \ldots, D\}$, we denote by $q_i = q_{\mathbf{x}_i}(\hat{\mathbf{y}}_i, \mathbf{y}_i)$ the *ground-truth* hardness score of the i-th image obtained using the ground-truth bounding boxes $\mathbf{y}_i$, and by $\hat{q}_i$ the hardness score estimated by one

---

Algorithm 1 Score Sampling (SS)

Input: Image $\mathbf{x}$, detector D_η, hardness query $q_{\mathbf{x}}(\hat{\mathbf{y}}, \mathbf{y})$.
Parameter: Number of samples N.
Output: Estimated hardness $\hat{q}$ using score sampling.
1: $\hat{\mathbf{y}}_0 \leftarrow D_0(\mathbf{x})$
2: **for** $n \in \{1, \ldots, N\}$ **do**
3:     **for** $(b_i, c_i, s_i) \in \hat{\mathbf{y}}_0$ **do**
4:         $X_i \sim \text{Bernoulli}(s_i)$
5:     **end for**
6:     $\tilde{\mathbf{y}}^{(n)} \leftarrow \{(b_i, c_i) \text{ for } (b_i, c_i, s_i) \in \hat{\mathbf{y}}_0 | X_i = 1\}$
7: **end for**
8: $\hat{\mathbf{y}} \leftarrow \{(b_i, c_i, s_i) \in \hat{\mathbf{y}}_0 | s_i > \eta\}$
9: $\hat{q} \leftarrow \frac{1}{N} \sum_{n=1}^{N} q_{\mathbf{x}}(\hat{\mathbf{y}}, \tilde{\mathbf{y}}^{(n)})$
10: **return** $\bar{q}$

---

of the methods under consideration. In the case of *score sampling (our)*, it is the approximation to the expectation $\mathbb{E}_{\tilde{\mathbf{y}} \sim P(\tilde{\mathbf{y}})}[q_{\mathbf{x}_i}(\hat{\mathbf{y}}_i, \tilde{\mathbf{y}}_i)]$ as defined in Eq. 5, while for the baselines, it is either computed using the entropy or the DS.

Datasets and detectors To illustrate the generality of our method, we evaluate off-the-shelf models on public-domain datasets for which weights are readily available. We perform our evaluations on the the COCO dataset (Lin et al. 2014) and nuImages (Caesar et al. 2019) and consider the following detectors:

- *coco-retina* and *coco-rcnn* are RetinaNet (Lin et al. 2017) and Faster-RCNN (Ren et al. 2015) with a ResNet-50-FPN backbone trained on the COCO dataset, from `torchvision` (Paszke et al. 2019).
- *mmdet-maskrcnn* and *mmdet-cascade* are Mask-RCNN (He et al. 2017) and Cascade Mask-RCNN (Cai and Vasconcelos 2019) with a ResNet-50-FPN backbone trained on nuImages, from `mmdet` (Chen et al. 2019). We dismiss all instance masks in our experiments and only consider bounding boxes. We remap the nuScenes and MMDetection label schemas to a simplified two-class schema (pedestrian and vehicle) which we describe in the appendix.

We set the pre-NMS score threshold to $\eta = 0.05$ for all detectors. We perform our evaluations on the *val* sets of the COCO and nuImages datasets.

Query-based Hardness Throughout the experiments, we use the queries introduced above but our evaluation protocol can be extended to a large family of single-image measure of performance. More precisely, we will consider all combinations of the counting, pixel-adjusted, and occlusion-aware queries applied to the error categories `fp`, `fn`, and `false = fp ∪ fn`, giving us 9 hardness queries in total.

Baselines We consider entropy (Eq. 1) and Dempster-Shafer (Eq. 2) uncertainty estimates as our baselines. Both methods measure the uncertainty associated with an image using a categorical distribution over all possible classes for each detected box. For two-stage architectures (Faster-RCNN, Mask-RCNN, and Cascade-RCNN), we modify the

standard implementation to get access to the set of K logits for each box which are then normalised to a categorical distribution with probabilities $\mathbf{p}$ using a softmax. In this case, we use use Eq. (1) and (2) directly with K being the total number of classes (including the background class). The situation is different for RetinaNet, in which each box gets instead K independent binary scores $\mathbf{p} \in [0,1]^K$, representing the one-vs-all probability for that class. It is not clear in this case how to define a predictive categorical distribution over all classes and we instead treat this situation as a binary problem with $K = 2$ and compute the uncertainty estimates for the detection score directly.

Implementation details We use `pycocotools` (Lin et al. 2014) for processing bounding boxes, including for computing the association. We use an IOU threshold of 0.5 for the association of bounding boxes in `pycocotools`. We compute score sampling (Algorithm 1) using 10 Monte Carlo samples in all experiments. This provides a favourable balance between accuracy and computational expense, and we provide a sensitivity analysis of the number of samples in the appendix.

Qualitative Results

We first study qualitatively the efficacy of our method by searching for the 5 hardest and 3 easiest images for a variety of hardness queries for *coco-rcnn* on the COCO *val 2017* set, and then compare these with the actual hardest images obtained using the ground-truth, as well as those found by the baseline methods. These results are presented for the error category `fp` in Figure 3. We display for each image the set of true positive, false positive, and false negative bounding boxes in order to allow for visual inspection of the performance of the detector. We show the corresponding figure for *mmdet-maskrcnn* on nuImages in the appendix.

Contrary to the entropy and DS baselines, our method is able to successfully identify images that have the expected error characteristics for a given query. For example, when the hardness query is `Total(fp)`, which asks for images that are likely to have a large number of false positives, our method finds images featuring a large number of objects (such as books on a shelf, parked cars, etc) many of them being false positives. This is qualitatively similar to the images ranked using the ground-truth. Interestingly, annotations in the COCO dataset often show shortcomings where objects within a large group are not annotated consistently but we nevertheless observe that our method is able to find such images successfully.

The same is true for the `PixelAdj(fp)` and `OccAware(fp)` queries, where our method finds images that have large false-positive detections and many overlapping boxes, respectively. Note that in the case of `OccAware(fp)`, our method guessed the top two images correctly. The images identified by the non query-based baselines are not similar to any specific query and mainly seem to feature a large number of bounding boxes in the image, which is easily explained since these methods measure the total uncertainty over all boxes, see Eq. (1–2).

The images ranked last in terms of hardness (easiest ones)

are somewhat less interesting as they all feature a low number of correctly classified boxes. It is worth noting that in the case of entropy and DS uncertainty, the lowest ranking images have *no* ground-truth bounding boxes and get a score of zero. For our method, we show in the appendix that the variance of the scores is sufficient to generate varied pseudo ground-truth configurations, see Fig. 14.

In the Appendix, we also show a histogram (Figure 13) for the estimated and ground-truth false positive bounding-boxes using *coco-rcnn*, and note that, in fact, many images have a hardness score of zero as they are easy for the detector to perform inference on (e.g., single centred object images). Therefore, it becomes crucial to provide a better ranking as there is a small sub-population of potential hard images. Our qualitative experiments below are precisely to capture how well different approaches do in ranking images based on query-based hardness measures.

Query-Based Hard-Image Ranking

We now evaluate quantitatively the performance of our method at ranking images in order of hardness for a given query. For any given method, we sort all images in the dataset in decreasing order of estimated hardness scores $\hat{q}_i$ and compare the resulting ranking to the ground-truth ranking that is obtained by sorting images in decreasing order of ground-truth hardness q_i. We evaluate ranking quality using the normalised discounted cumulative gain (nDCG) which is a well-known metric for ranking-based tasks, see for example (Järvelin and Kekäläinen 2002). The discounted cumulative gain is defined as follows:

$$\text{DCG} = \sum_{i=1}^{D} \frac{\langle q_j \rangle_{j, \hat{q}_j = \hat{q}_i}}{\log_2(\text{rank}(\mathbf{x}_i; \mathcal{D}) + 1)}, \tag{6}$$

where $\text{rank}(\mathbf{x}_i; \mathcal{D})$ is the rank of image i when the dataset $\mathcal{D}$ is sorted in decreasing order of estimated hardness score $\hat{q}_i$, and $\langle q_j \rangle_{j, \hat{q}_j = \hat{q}_i}$ is the average ground-truth hardness score of all images $\mathbf{x}_j$ that have the same estimated hardness score as $\mathbf{x}_i$, i.e. for which $\hat{q}_j = \hat{q}_i$. This ensures that the DCG is independent under re-orderings of images with the same estimated hardness score. The normalised DCG is then defined by dividing the gain by the idealised gain as $\text{nDCG} = \text{DCG}/\text{DCG}_{gt}$, where DCG_{gt} is the DCG of the ground-truth ranking, see (Järvelin and Kekäläinen 2002).

In Table 1, we observe that in practically all the cases our method (SS) matches or exceeds (significantly) the performance of the entropy and the DS baselines. Since the baseline methods do not allow query-based hardness computation, we do not expect these techniques to perform well on the pixel-aware and occlusion-adjusted hardness queries. This is evident in Table 1 for the pixel-weighted and overlap-adjusted measures of hardness. The gap between methods is much smaller in the case of the `Total` queries which indicates that the entropy and DS are a good proxy for finding the total number of errors in an image. In particular, entropy outperforms our method slightly for the query `Total(fn)` which could be due to the fact that the detectors are not perfectly calibrated.

Figure 3: Hardest images (left) and easiest images (right) for the *coco-rcnn* detector on the COCO *val* set for different queries. Entropy and Dempster-Shafer (DS) measures of uncertainty are unable to adapt to different hardness queries and are hence query agnostic. Green, red, and blue boxes denote true positives, false positives, false negatives, respectively. Note that the ground-truth ranking is different for every query. See also Figure 12 in the appendix for examples on the nuImages dataset.
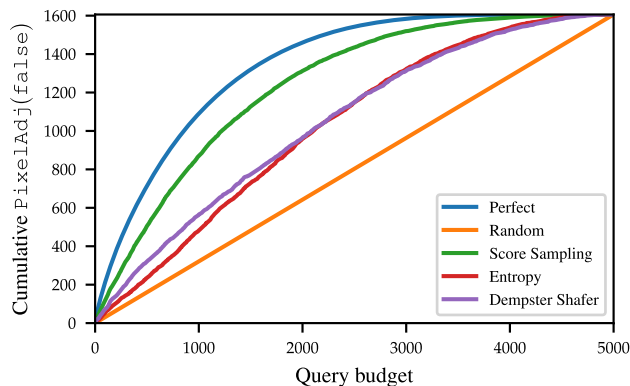
Figure 4: Cumulative ground-truth hardness score of images in the COCO *val* set and *coco-rcnn* detector for images in descending order of estimated hardness scores. `PixelAdj(false)` hardness query is used here.

Ranking performance using query-budget   Another way to visualise the ranking performance would be to consider the cumulative ground-truth hardness of all the images mined within a certain query budget. This allows us to easily visualise *how quickly* hard images can be found within a dataset by each method. We show this for the `Pixel(false)` query and the *coco-rcnn* detector in Figure 4 and observe that our method not just provides better ranking (as shown in Table 1), but also provides the hard images much quicker than the other baselines. We note that the entropy and the DS baselines perform similarly, and are barely better than a random ranking, which corresponds to the diagonal line through the origin. We display similar figures for other detectors, datasets and hardness definitions in the appendix (Figures 6–9).

Query-Based Hard-Image Classification

We finally consider the task of *finding* hard images for a specific detector and hardness query within a dataset. Given a definition of hardness, an image is defined to be "hard" if its ground-truth hardness q_i is above a chosen threshold t_{hard}, and "easy" otherwise. This effectively assigns a binary class label to each image and we evaluate the performance of our method at solving this *hard vs. easy* classification task. To do so, we construct a binary classifier by introducing a threshold t_{score} on the estimated hardness score, and classify an image as "hard" if $\hat{q}_i > t_{score}$ and "easy" otherwise. Performance of this binary classifier can be evaluated using typical metrics for binary classification, such as AUROC, across all possible score thresholds t_{score}. If needed, a score threshold with a given application-specific trade-off of precision and recall can be easily obtained using a held-out dataset.

We evaluate our method and baselines across different hard vs. easy image ratios by choosing the hardness threshold t_{hard} such that this ratio is 5%, 10%, 25%, and 50% for each dataset. We present the mean area under the ROC curve (mAUROC) over these threshold choices in Table 2 in the appendix. In almost all cases, score sampling exceeds or matches the performance of the baseline techniques sim-

Hardness query	Entropy	DS	SS (our)	Entropy	DS	SS (our)
	Normalised DCG			Normalised DCG		
	coco-retina			mmdet-maskrcnn		
`Total(fp)`	0.81	0.81	**0.88**	0.92	0.88	**0.95**
`Total(fn)`	0.91	**0.92**	**0.92**	**0.89**	0.83	**0.89**
`Total(false)`	0.92	0.92	**0.93**	0.95	0.89	**0.96**
`PixelAdj(fp)`	0.66	0.66	**0.86**	0.52	0.55	**0.86**
`PixelAdj(fn)`	0.74	0.74	**0.85**	0.50	0.51	**0.75**
`PixelAdj(false)`	0.76	0.76	**0.87**	0.58	0.61	**0.82**
`OccAware(fp)`	0.75	0.75	**0.91**	0.87	0.83	**0.96**
`OccAware(fn)`	0.81	0.81	**0.89**	0.81	0.74	**0.85**
`OccAware(false)`	0.82	0.83	**0.92**	0.89	0.83	**0.94**
	coco-rcnn			mmdet-cascade		
`Total(fp)`	0.90	0.81	**0.91**	0.92	0.87	**0.94**
`Total(fn)`	**0.90**	0.77	**0.90**	**0.89**	0.83	**0.89**
`Total(false)`	**0.94**	0.83	**0.94**	**0.95**	0.89	**0.95**
`PixelAdj(fp)`	0.79	0.80	**0.93**	0.46	0.49	**0.83**
`PixelAdj(fn)`	0.72	0.77	**0.81**	0.51	0.52	**0.75**
`PixelAdj(false)`	0.81	0.84	**0.90**	0.56	0.58	**0.84**
`OccAware(fp)`	0.85	0.77	**0.94**	0.84	0.80	**0.94**
`OccAware(fn)`	0.83	0.70	**0.86**	0.81	0.75	**0.86**
`OccAware(false)`	0.88	0.78	**0.93**	0.87	0.81	**0.93**

Table 1: Ranking performance of Entropy, Dempster-Shafer, and Score sampling (Our) measured by the Normalised Discounted Cumulative Gain (nDCG) against the ground-truth ranking for different `Total`, `PixelAdj`, and `OccAware` queries. Higher is better.

ilarly as for our previous ranking experiments. Again, we observe score sampling performs much better for hardness definitions which measure a specific property of the detection error.

Conclusive Remarks

In this work, we proposed a framework to enable finding query-based hard images for object detectors without having access to ground-truth labels. We showed that it is possible to construct a distribution over pseudo ground-truth labels and quantify a given query-based hardness measure via an efficient Monte Carlo approximation. We provided extensive analysis to show such approximations rank and find hard images more effectively than general purpose image hardness estimation techniques such as entropy and Dempster-Shafer.

We hope that our work opens possibilities to explore situations where query-based hardness measures are used in practice to improve downstream tasks and also to understand their impact on data-shift scenarios. One crucial assumption in our approach is that the object detectors are considered to be well enough calibrated. Therefore, another future work entails exploring differently calibrated models and understanding their impact on the hardness measures as well as studying the efficacy of this technique on out of domain data. Finally, a more complex and accurate stochastic model could also be devised to provide improved ranking performance.

Acknowledgements

References

Aghdam, H. H.; Gonzalez-Garcia, A.; Weijer, J. v. d.; and López, A. M. 2019. Active learning for deep detection neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3672–3680. openaccess.thecvf.com.

Bucher, M.; Herbin, S.; and Jurie, F. 2016. Hard Negative Mining for Metric Learning Based Zero-Shot Classification. In *Computer Vision – ECCV 2016 Workshops*, 524–531. Springer International Publishing.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2019. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*.

Cai, Z.; and Vasconcelos, N. 2019. Cascade R-CNN: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5): 1483–1498.

Chen, J.; Luo, B.; Wu, Q.; Chen, J.; and Peng, X. 2020. Overlap Sampler for Region-Based Object Detection. In *WACV*, 756–764.

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; Zhang, Z.; Cheng, D.; Zhu, C.; Cheng, T.; Zhao, Q.; Li, B.; Lu, X.; Zhu, R.; Wu, Y.; Dai, J.; Wang, J.; Shi, J.; Ouyang, W.; Loy, C. C.; and Lin, D. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*.

Choi, J.; Elezi, I.; Lee, H.-J.; Farabet, C.; and Alvarez, J. M. 2021. Active learning for deep object detection via probabilistic modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10264–10273.

Dempster, A. P. 2008. A generalization of Bayesian inference. In *Classic works of the Dempster-Shafer theory of belief functions*.

Du, X.; Wang, Z.; Cai, M.; and Li, S. 2022. Towards Unknown-aware Learning with Virtual Outlier Synthesis. In *International Conference on Learning Representations*.

Feng, D.; Wei, X.; Rosenbaum, L.; Maki, A.; and Dietmayer, K. 2019. Deep active learning for efficient training of a lidar 3d object detector. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 667–674. IEEE.

Forsyth, D. A.; and Ponce, J. 2012. *Computer vision: a modern approach*. Pearson.

Gal, Y.; Islam, R.; and Ghahramani, Z. 2017. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, 1183–1192. proceedings.mlr.press.

Harakeh, A.; and Waslander, S. L. 2021. Estimating and Evaluating Regression Predictive Uncertainty in Deep Object Detectors. In *International Conference on Learning Representations (ICLR)*.

Haussmann, E.; Fenzi, M.; Chitta, K.; Ivanecky, J.; Xu, H.; Roy, D.; Mittel, A.; Koumchatzky, N.; Farabet, C.; and Alvarez, J. M. 2020. Scalable active learning for object detection. In *2020 IEEE intelligent vehicles symposium (iv)*, 1430–1435. IEEE.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

Järvelin, K.; and Kekäläinen, J. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4): 422–446.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988. openaccess.thecvf.com.

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

Liu, J. Z.; Lin, Z.; Padhy, S.; Tran, D.; Bedrax-Weiss, T.; and Lakshminarayanan, B. 2020. Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness. In *NeurIPS*.

Mukhoti, J.; Kulharia, V.; Sanyal, A.; Golodetz, S.; Torr, P. H.; and Dokania, P. K. 2020. Calibrating Deep Neural Networks using Focal Loss. In *NeurIPS*.

Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; and Lin, D. 2019. Libra R-CNN: Towards Balanced Learning for Object Detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pinto, F.; Yang, H.; Lim, S.-N.; Torr, P. H.; and Dokania, P. K. 2022. RegMixup: Mixup as a Regularizer Can Surprisingly Improve Accuracy and Out Distribution Robustness. In *arXiv: 2206.14502*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Roy, S.; Unmesh, A.; and Namboodiri, V. P. 2018. Deep active learning for object detection. In *BMVC*, volume 362, 91. bmva.org.

Sensoy, M.; Kaplan, L.; and Kandemir, M. 2018. Evidential deep learning to quantify classification uncertainty. *Advances in Neural Information Processing Systems*, 31.

Shannon, C. E. 1948. A mathematical theory of communication. In *Bell System Technical Journal*.

Shrivastava, A.; Gupta, A.; and Girshick, R. 2016. Training Region-based Object Detectors with Online Hard Example Mining. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Suh, Y.; Han, B.; Kim, W.; and Lee, K. M. 2019. Stochastic class-based hard example mining for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7251–7259. openaccess.thecvf.com.

Yu, W.; Zhu, S.; Yang, T.; and Chen, C. 2022. Consistency-based active learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3951–3960.