BAOD: Budget-Aware Object Detection

Alejandro Pardo \(^1\), Mengmeng Xu \(^2\), Ali Thabet\(^2\), Pablo Arbeláez\(^1\), and Bernard Ghanem\(^2\)

\(^1\)Universidad de los Andes, Colombia
\(^2\)King Abdullah University of Science and Technology (KAUST), Saudi Arabia

Abstract

We study the problem of object detection from a novel perspective in which annotation budget constraints are taken into consideration, appropriately coined Budget Aware Object Detection (BAOD). When provided with a fixed budget, we propose a strategy for building a diverse and informative dataset that can be used to optimally train a robust detector. We investigate both optimization and learning-based methods to sample which images to annotate and what type of annotation (strongly or weakly supervised) to annotate them with. We adopt a hybrid supervised learning framework to train the object detector from both these types of annotation. We conduct a comprehensive empirical study showing that a handcrafted optimization method outperforms other selection techniques including random sampling, uncertainty sampling and active learning. By combining an optimal image/annotation selection scheme with hybrid supervised learning to solve the BAOD problem, we show that one can achieve the performance of a strongly supervised detector on PASCAL-VOC 2007 while saving 12.8% of its original annotation budget. Furthermore, when 100% of the budget is used, it surpasses this performance by 2.0 mAP percentage points.

1. Introduction

Object detection in images is a fundamental computer vision problem with applications in many tasks including face/pedestrian detection [44, 1, 21, 41, 12, 56], counting [6, 23], and visual search [10, 39]. Building a successful object detector encompasses three main dimensions: (1) the image dataset to be annotated for training the detector. A larger dataset allows for a more accurate detector, but the number of training images is limited by the annotation budget; (2) the annotation scheme used to label the training images. One could annotate either image-level labels (the categories of the objects are known but their locations are unknown, denoted weakly supervised annotation) or instance-level labels (both categories and locations are known, denoted strongly supervised annotation) [41, 35, 29, 40]; (3) the detection model. Most works on object detection fix the first two dimensions and only explore the third. In fact, they tend to focus on optimizing the detection model based on one or more training datasets that typically provide the same kind of annotations. In this paper, we fix the detection model and investigate solutions in the first two dimensions of the problem.

Figure 1: Budget-Aware performance of detectors with different levels of supervision. The models are trained on PASCAL VOC 2007 trainval (VOC07), PASCAL VOC 2012 trainval (VOC12) as specified in the legend. Our proposed Budget aware object detection (BAOD) (green -∇- curve) has a higher mAP than FSOD (yellow -| - curve) and WSOD (orange - - curve) methods at most budgets. Given a larger unlabeled image pool (Blue -♦- curve, VOC07+VOC12), our BAOD can reach a higher mAP using the same budget needed to annotate VOC07 with instance-level labels. Since the dataset is finite, WSOD cannot increase its performance with more budget.

*denotes equal contribution
A large group of object detectors fall under the umbrella of Fully-Supervised Object Detection (FSOD) [45, 46, 15, 14, 11, 47, 18, 34]. It has been shown in recent years that these techniques can reach high detection performance, especially with the introduction of large datasets with strong annotations [50, 13, 35]. This requirement makes FSOD methods expensive and time consuming. In contrast, Weakly-Supervised Object Detection (WSOD) [4, 9, 24, 5, 61, 16, 32, 20, 48, 54, 22] aims at building object detectors from cheaper but less informative image-level or weak annotations.

In this paper, we propose a trade-off between dataset annotation cost and model precision in order to combine both weak and strong annotations and train an object detector with hybrid supervision. We put all the detectors on the same footing when different annotation schemes are available. Ideally, detectors should only be compared when they are trained using image datasets that offer the same amount of information (not necessarily the same number of images). Since this notion is difficult to define quantitatively, we take the training budget of a detector as a unifying surrogate measure. Here, we define budget as the effort, or cost, to annotate a dataset, thus combining the first two dimensions of the detection problem: dataset scale and annotation scheme. In fact, with a fixed budget and a set of unlabeled images, the number of images we can label depends highly on the annotation cost for each image. This cost varies significantly between image- and instance-level annotations. Typically, annotating a bounding box around an object in an image is significantly more expensive than simply annotating its category [41, 50]. Therefore, an FSOD method with the same budget as a WSOD method contains fewer images in its training set.

We explore strategies to build better object detector models when constrained with a training budget, a novel problem we coin budget-aware object detection (BAOD). As such, we focus on choosing the best images for training and how to annotate them. For this, we survey several selection methods to sequentially choose both the image and type of annotation following an active learning paradigm. Additionally, we propose a novel hybrid training procedure for object detectors that can use both strong (instance-level) and weak (image-level) annotations. Figure 1 shows that actively selecting images and their annotations to sequentially train a hybrid supervised detector outperforms its FSOD and WSOD counterparts, when the budget is larger than 20% of the budget needed to annotate VOC2007 trainval at the instance-level. Moreover, the yellow curve shows that we can reach an even higher mAP if we use the same budget of VOC07 to annotate images from both datasets VOC07 and VOC12.

Contributions. (1) We propose the BAOD problem and present a new evaluation criterion Budget-Average mAP for object detection algorithms. This criterion takes into account both detection performance and budget. (2) We study several strategies (e.g., optimization and learning-based) to select both training images and their annotation scheme. (3) We propose a hybrid supervised learning strategy that combines category and localization information to train a robust detection model that handles both weak and strong annotations. (4) Following a BAOD approach (i.e., combining intelligent image and annotation scheme selection with hybrid supervised learning), we show that the mAP test performance on VOC07 can be improved by 2 percentage points for the same budget used to annotate the training set of VOC07 (by combining it with VOC12). We also show the opposite, i.e., that state-of-the-art test performance on VOC07 can be achieved, while saving 12.8% of the budget used in strongly annotating its training set.

2. Related Work

2.1. Fully Supervised Object Detection (FSOD)

With the development of deep learning, many CNN based methods have been proposed to solve the FSOD problem, such as YOLO [46], SSD [37], RCNN [15], and its variants [47, 14, 11, 18, 34]. Although these methods achieve impressive detection results, providing them with large-scale instance-level (bounding-box) annotations for training is costly.

In this paper, we constrain the annotation budget and focus on how to reach the best detection performance within this budget by intelligently selecting between both weak and strong annotations. Su et al. [52] reported that it takes 26 seconds to draw one bounding-box without quality control, and 42 seconds with it. There has been recent progress in developing tools to further reduce annotation time (most recently to 7 seconds) [40, 41, 29]. Our work investigates a complementary aspect of annotation to emphasize that some images do not need to be strongly annotated to achieve excellent performance. While our discussion of budget aware object detection only considers the labor cost for image annotation, we note that sequentially training detection models also consume computational resources. However, given the continual development of hardware acceleration and re-organization, which leads to steady decrease in their cost, manual annotation remains the more expensive component in the detection problem.

2.2. Weakly Supervised Object Detection (WSOD)

If there are no instance-level labels available at training time, a WSOD method can be trained. Most classical approaches cast WSOD as a Multiple Instance Learning (MIL) problem [51, 32, 20, 48, 4, 22]. Bilen et al. [5]...
were the first to utilize a deep CNN (denoted as the weakly supervised deep detection network or WSDDN) to solve this problem. WSDDN selects positive samples by multiplying the score of recognition and detection. Zhang et al. [59] proposed a simple yet effective post-processing step to mine pseudo ground truth bounding boxes used for iterative FSOD training.

We use weak supervision to improve pseudo object labels. These labels are generated from a previously trained detector, and are post-processed through a pseudo label mining process. The image-level labels help remove false predictions that correspond to a wrong object category.

2.3. Hybrid Supervised Learning

In our work, we study hybrid supervised learning for object detection. This type of learning exploits multiple types of supervision during training. Semi-supervised learning is a special case of this family, since it learns a model from a set of labeled and unlabeled data. Several works [33, 49, 7, 30, 43] try to solve the generic semi-supervised learning problem through teacher-student learning. They first train a teacher model from the strongly annotated subset, then the predictions obtained from the teacher model on the unlabeled data are used to regress a second model that is called the student model. Chéron et al. [8] also used several kinds of annotations to train an activity recognition model, and they revealed that strongly annotating every training sample is not necessary to achieve noteworthy localization results in the video domain.

In our case, since images can be labeled either weakly or strongly, a hybrid supervised dataset is always considered. Inspired by Rosenberg’s work [43], we train a teacher-student model to use the hybrid dataset. Our teacher detector is learned from strong annotations only, while the student detector is learned from both ground-truth and/or processed pseudo object labels.

2.4. Active Learning

In general, active learning is a sequential decision making process that iteratively selects the most useful examples an oracle should annotate and add to the labeled training set. It aims at training more accurate models with the minimum data required. This field has been widely studied in the context of image and video classification [42, 2, 28, 26, 60, 17], object and action localization [19, 29, 27, 40, 25], human pose estimation [36] and visual question answering [38]. A commonly used approach to selecting new training images is by means of their entropy score [55]. The intuition is that higher entropy examples attribute to more learning information. More recent research directly predicts the improvements of adding a new sample to the training set, and uses this measurement as a selection criteria [28]. As specified in previous work, active learning aims at maximizing performance while minimizing the human cost in labeling the training samples [42, 36, 19].

The BAOD problem can also be formulated as a sequential decision making process. We study a few well-known selection techniques in our new active learning pipeline, which contains two active processes: (1) select the next batch of training images to annotate and (2) decide the type of annotation for each selected image. Thus, we focus on the annotation sequence that can provide the most useful information to the detector. To the best of our knowledge, we are the first to use active learning and hybrid training to study object detection.

3. Budget Aware Object Detection (BAOD)

In this section, we explore several ways to sequentially create an object detection dataset with a fixed budget, which includes both image-level and instance-level annotations. Then, we introduce a hybrid supervised learning procedure to take advantage of this hybrid labeled training set. A complete overview of the whole active learning procedure is shown in Figure 2.

3.1. Hybrid Dataset using Active Learning

Our hybrid supervised dataset should consider which images to include in training and how to label them. To tackle this problem, we propose an active hybrid learning framework which acts as an active learning agent to simulate the annotation process. At every active step \( t \), we have a budget constraint \( d \) to spend on annotations. Each image in this set belongs to one of three pools: unlabeled \( U_t \), weakly labeled \( W_t \), or strongly labeled \( S_t \). At each active step, the hybrid

![Figure 2: Overview of active learning pipeline to construct a hybrid labeled dataset.](image-url)
supervised dataset $W_t \cup S_t$ is used to train a hybrid detector, which will be detailed in Section 3.2. This detector is used to find a selection function that takes an action on $U_t$ or $W_t$ pools. As shown in Figure 2, we have three possible actions with their associated cost: (1) annotate strongly $x_1$ (i.e. send the image from $U_t$ to $S_t$), (2) annotate weakly $x_2$ (i.e. send the image from $U_t$ to $W_t$), or (3) strongly annotate a weakly annotated image $x_3$ (i.e. send the image from $W_t$ to $S_t$). Once the actions have been made, the image sets $U_t$, $W_t$, and $S_t$ are updated. We proceed iteratively until we either run out of images in $P_t = U_t \cup W_t$ or run out of budget $d$.

To this end, we study three active learning sampling functions (Random, Uncertainty, and Learning Active Learning [28]) within four action selection methods: Random Sampling (RS), Uncertainty Sampling (US), Optimization based on US and Optimization based on Learning Active Learning (LAL).

For RS, we randomly choose an active batch of images at each active step to include into $W_t$ or $S_t$. For US, images are sorted in descending order of uncertainty (measured by entropy) to choose high uncertainty images to train on with full supervision\(^1\), based on an uncertainty score denoted as $s_k$, and the top images are included into $W_t$ or $S_t$. We adhere to the following annotation policy for RS and US when a budget constraint is enforced. For both RS and US, we prioritize weak labels first. In other words, so long as the budget constraint is not exceeded, (1) the image batch that is selected in each active step for these two methods will only contain images from $U_t$ that will be weakly labeled or, (2) if $U_t$ becomes empty, images from $W_t$ will be selected and get a strongly label.

There are several ways to evaluate $s_k$ for an image [55, 41]. We follow convention and model this score in Eq. 1:

$$s_k = \frac{1}{M} \sum_{i=1}^{M} \sum_{p \in p_i} -p i \log(p)$$

This is an entropy measure for each of the $M$ bounding boxes in an image using the classification score predicted by the current detection model. Collecting $M$ predictions from an image $I_k$, each prediction has a probability score vector $p_i \in [0, 1]^N$ for the $c$ object categories.

Below, we explain the remaining active selection methods: Optimization based on US and LAL.

### 3.1.1 Optimization Based Active Selection using US

At the $t$-th active step, let $P_t = \{I_k\}_{k=1}^N$ be the $N$ images that can be further annotated, and $s \in \mathbb{R}^N$ be their corresponding uncertainty scores (as defined above). We have three possible action vectors $x_1, x_2, x_3 \in \{0, 1\}^N$, as defined earlier. If the $k$-th element of $x_1$ is 1, we annotate the image $I_k$ using option $i$. Assume the next active batch of annotations has a linear impact on the model performance increment $\delta_t$.

$$\delta_t = f_1(P_t) x_1 + f_2(P_t) x_2 + f_3(P_t) x_3$$

To quantitatively show the contribution of new images, we also assume that the uncertainty score is a complete statistic of an unlabeled or weakly labeled set of images. Then, we can simply approximate $(f_1, f_2, f_3)$ as linear functions. We observe that many active learning studies [31, 19] have experimentally shown that incorporating images with a higher uncertainty score in training may further improve the detector, but at the risk of having more difficulty in producing true positive predictions. Therefore, we model $(f_1, f_2, f_3)$ as linear functions that tend to favor actions $x_1, x_3$ over action $x_2$ for images with high enough uncertainty score (i.e. higher than the median). As such, and by combining the intuitions above, the expected increment in performance is modeled as:

$$\delta_t \approx s^\top (x_1 + x_3) + (\mu 1 - s)^\top x_2$$

For the active selection method based on optimization using US, we seek to maximize $\delta_t$, while staying within the budget constraint. To model this latter constraint, we define $a$ as the cost of strongly annotating an unlabeled image, $b$ as the cost of weakly annotating an unlabeled image, and $c$ as the cost of strongly annotating an already weakly labeled image. Therefore, this selection method seeks to solve the following binary optimization problem for $(x_1, x_2, x_3)$:

$$\max_{x_1, x_2, x_3 \in \{0, 1\}^N} s^\top (x_1 + x_3) + (\mu 1 - s)^\top x_2$$

s.t. $x_3 \leq \psi$

$$x_1 + x_2 \leq 1 - \psi$$

$$1^\top (a x_1 + b x_2 + c x_3) \leq d$$

$$1^\top (a x_1 + b x_2 + c x_3) \geq d - a$$

Here, we define a vector $\psi$ as the indicator vector for images that have already been weakly annotated, i.e. the $k$-th component of $\psi$ is 1 if $I_k \in W_t$ and 0 otherwise. Using this indicator, the first two constraints in Eq (4) enforce that only one action is performed on each image, i.e. among all the $k$-th components of $(x_1, x_2, x_3)$, only one can be 1. The third and fourth constraints enforce that the budget be used as much as possible in each active step.

The linear binary problem in Eq (4) is NP-hard in general, so exact solvers (e.g. the Branch and Bound Algorithm) tend to have long run-times especially when $N$ is large. As a tradeoff between optimization accuracy and per active step run-time, we employ a conventional linear relaxation of the original problem to form an approximate linear
program (LP), which can be efficiently solved at large-scale with off-the-shelf LP solvers. More details about this problem are shown and proven in the supplementary material.

### 3.1.2 Optimization Based Active Selection using LAL

The uncertainty score \( s \) is not the only selection measure that has been studied in the active learning literature. In the Learning Active Learning (LAL) method proposed by Konyushkova et al. [28], the increment in performance function \( \delta_t \) is learned to be a function of the current model state as well. More concretely, given images \( I_k \in P_t \) and the current detection model, we build a feature vector \( v = [O_t, s] \) that concatenates both the current model state \( O_t \), represented as the average precision curves under five different Intersection over Union (IoU) thresholds, and the uncertainty scores \( s \). Following the LAL method in [28], we train a Support Vector Regression (SVR) model to regress the actual increment in mAP performance from \( v \) at each active step for both weak and strong annotation actions. Obviously, these SVRs are trained on a separate detection dataset than the one used to evaluate BAOD. At each active step and by denoting the output predictions of these SVRs as \( h_w \) and \( h_s \), we formulate the same constrained optimization problem in Eq (4) but with the learned objective:

\[
\delta_t = h_w^T x_2 + h_s^T (x_1 + x_3)
\]

### 3.2. Hybrid Supervision for Object Detection

At each active step, we train a hybrid supervised detector using both strong and weak supervision, i.e., using images from \( W_t \) and \( S_t \) (refer to Figure 3 for an illustration).

**Teacher-Student Model.** In this framework, we train an initial detector with the initially strongly annotated image set \( S_0 \). In each active step afterwards and to overcome undesirable local minima, we learn this detector using pre-training on Imagenet [50]. However, the model can also be fine-tuned from previous active steps to reduce computation time. Training on \( S_t \) at each active step can only learn a decent detector, but it can also transfer knowledge to weakly annotated images. This detector works as a teacher that predicts objects in every image in the weakly labeled image set \( W_t \). If these predictions are post-processed properly, they can be viewed as pseudo labels, which we merge with the ground truth instance-level labels in \( S_t \) to train a fully supervised student detector. The student detector is also pretrained from Imagenet. Since it has both strong and weak annotations from training samples in \( W_t \cup S_t \), we expect the student detector to perform better than its teacher.

**Post-processing.** The predictions from the teacher model at each step have many redundant or erroneous pseudo labels. We use minimal knowledge to post-process them so as to focus our study on the active selection methods and the benefit of hybrid training. More concretely, given an image \( I_k \) with a weak annotation \( \omega \in \{0, 1\}^c \), where \( c \) is the number of categories, we assume that the teacher model gives \( M \) positive predictions with localization \( P \in \mathbb{R}^{M \times 4} \) (4 components defining a bounding box), classification \( A \in \{0, 1\}^{M \times c} \), and the confidence score for each of the \( M \) positive predictions \( q \in [0, 1]^M \). \( P, A \) and \( q \) are obtained from the teacher model. Among these \( M \) predictions, we seek to mine the pseudo labels in the form of a sparse \( M \)-dimensional binary vector \( y \) that solves the following constrained optimization:

\[
\begin{align*}
\max_{y \in \{0, 1\}^M} & \quad y^\top q \\
\text{s.t.} & \quad y^\top (1 - w) = 0 \\
& \quad \text{IoU}(P_i, P_j) \leq \alpha \quad \forall y_i, y_j = 1 \\
& \quad 1 \leq \|y\|_0 \leq \beta
\end{align*}
\]

The intuition behind solving this problem is that we seek to maximize the confidence score across all pseudo labels indexed by \( y \). The first constraint enforces image-level consistency of the pseudo labels with the ground truth weak annotations. The second constraint removes predictions that are highly overlapping (similar to an NMS post-processing step). The third constraint enforces a sparsity condition on \( y \), thus limiting the number of possible pseudo labeled bounding boxes (potential objects) to be between 1 and \( \beta \).
In our experiments, we take $(\alpha = 0.3, \beta = 3)$. Details on this optimization are in the supplementary material.

4. Experiments and Analysis

4.1. Experimental Setup

Datasets. As in most WSOD methods, we use the PASCAL VOC 2007 (VOC07) or 2012 (VOC12) datasets [13] to perform most of the experiments. Given the active learning pipeline of our method, we emulate the active learning procedure using VOC07-12 annotations to selectively annotate 20 categories in 5011 images in VOC07 or 16551 images in the union of VOC07 and VOC12 (VOC0712). All detection models are evaluated on the VOC07 test dataset. For each annotation type (weak or strong), we assume that each image has a fixed annotating cost/time, which is not necessarily true in practice but it simplifies the analysis. In most of the experiments, we set $(a = 34.5, b = 1.6, c = a - b)$, in unit seconds, according to the annotation procedure of [50]. In Section 4.4, we vary these cost value to $(a = 7, b = 1.6, c = a - b)$ following the more efficient annotation procedure of [41]2. The cost of fully annotating VOC07 trainval is denoted as 100% (or total) budget for every experiment.

Evaluation Metrics. To evaluate the active selection methods with the hybrid detector, we compute a budget-performance curve at various budget limits. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP. In doing so, we propose a new budget-aware metric denoted as Budget-Average mAP, measured as the normalized area under the budget-mAP curve for a certain budget range. The budget axis varies the percentage of the total budget, and the model detection performance is taken to be mAP.

Implementation Details. In every active step, we choose Faster-RCNN as the object detection model (teacher and student), which is trained in the hybrid supervised way described in Section 3.2. Both teacher and student models use VGG16 as the backbone network pre-trained on ImageNet [50]. We follow the default setup in [58] to train Faster-RCNN. During training, the total number of epochs is set to 10, the learning rate is 0.01 for the first 8 epochs and 0.001 for the remaining. The batchsize is set to 16 on a four-GPU cluster nodes equipped with Titan Xp. The student model is cloned from the ImageNet pretrained VGG16 in every active step and has the same training schedule as the teacher model1. For LAL experiment, we collect 10 categories from the MS-COCO dataset [35] to train the SVRs. Since the LAL method is dataset agnostic [28], we take the SVR training categories to be different from the 20 categories in PASCAL VOC.

4.2. Advantages of Uncertainty Sampling and Hybrid Training

Uncertainty Sampling. In order to explore the influence of the uncertainty score on model performance, we emulate the annotation process of the oracle using VOC07 and incorporate it into the active learning pipeline with only strong annotations and use a FSOD method to train our model. We use two selection methods (RS and US) to build the dataset for FSOD. From the first two columns in Table 5 that compare these two sampling methods, we see that collecting images with large uncertainty score is more effective and preferable than random sampling in the three budget ranges.

Hybrid Training. Table 5 also shows the influence of the active selection scheme on model performance but when the hybrid supervised learning process is used to combine weak and strong annotations at each active step. We again use RS and US to select the images in each step for hybrid training. It is clear that both hybrid methods outperform their corresponding FSOD counterpart at every budget range, especially when the budget range is low (under 50%).

Table 1: Budget-Average mAP using fully and hybrid training pipelines with random and uncertainty selection. Uncertainty sampling is always better than random sampling selection, and hybrid training is always better than FSOD.

<table>
<thead>
<tr>
<th>Selection Method</th>
<th>FSOD</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sampling Function</strong></td>
<td><strong>RS</strong></td>
<td><strong>US</strong></td>
</tr>
<tr>
<td><strong>Low Budget Range</strong></td>
<td>52.5</td>
<td>53.1</td>
</tr>
<tr>
<td><strong>Mid Budget Range</strong></td>
<td>62.5</td>
<td>63.6</td>
</tr>
<tr>
<td><strong>High Budget Range</strong></td>
<td>67.9</td>
<td>68.7</td>
</tr>
</tbody>
</table>

4.3. Optimization-Based Active Selection

Here, we evaluate the optimization-based selection methods (using US and LAL) described in Section 3.1, when they are combined with hybrid supervised training. Table 2 compares these methods by measuring their average mAP at each budget range and compares them to the RS and US selection methods for references. These results indicate that the optimization methods are the best ways to combine and use the hybrid annotations at every budget, where the optimization-based US method is slightly better 3

2We left more discussion on the annotating time assumption in the supplementary material.

3The source code of the framework will be made publicly available.
than its LAL counterpart. As such, we denote the former method as the BAOD approach, which was mentioned and highlighted in Figure 1. Interestingly, we observe that the RS method requires 62% of the total VOC07 budget (all images are strongly annotated) to achieve 95% of the detection performance at that budget (i.e., 67.4% mAP). In comparison, the BAOD method requires only 48.5% of the total budget to reach the same performance. This performance gap attests to the effectiveness of this method. We include more results in the supplementary material.

Table 2: Budget-Average mAP using simple hybrid training and optimization methods. US based optimization is slightly better than LAL one. The optimization methods perform better than the simple hybrid random selection and uncertainty selection methods in the three budget ranges.

<table>
<thead>
<tr>
<th>Selection Method</th>
<th>Hybrid</th>
<th>Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling Function</td>
<td>RS</td>
<td>US</td>
</tr>
<tr>
<td>Low Budget Range</td>
<td>56.4</td>
<td>55.1</td>
</tr>
<tr>
<td>Mid Budget Range</td>
<td>64.5</td>
<td>65.8</td>
</tr>
<tr>
<td>High Budget Range</td>
<td>68.7</td>
<td>69.3</td>
</tr>
</tbody>
</table>

4.4. Effect of Per-Image Annotation Cost

The cost of strong annotations can vary due the annotation strategy that is used. For instance, Papadopoulos et al. [41] created a method that reduces the time needed to draw bounding boxes to 7 seconds per image on average. Here, we use this cost ($\alpha = 7$) to study the behavior of our BAOD method given a smaller gap between strong and weak annotation. We report these results in Table 3, which depicts the best performing FSOD method and the RS hybrid baseline for reference. We observe the same relative behavior between the methods as in the case when $\alpha$ was several times higher. However, the performance gap between these methods decreases in this case because the number of images that can be weakly annotated for the cost of one strong annotation is much smaller.

Table 3: Budget-Average mAP using a lower cost for strong annotations. If we assume the weak and strong annotation costs are more close (7 seconds and 1.5 seconds), US based optimization (BAOD) still performs better than the simple hybrid random selection and uncertainty selection methods in the three budget ranges.

<table>
<thead>
<tr>
<th>Selection Method</th>
<th>FSOD</th>
<th>Hybrid</th>
<th>Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling Function</td>
<td>RS</td>
<td>US</td>
<td>US (BAOD)</td>
</tr>
<tr>
<td>Low Budget Range</td>
<td>53.1</td>
<td>52.3</td>
<td>54.2</td>
</tr>
<tr>
<td>Mid Budget Range</td>
<td>62.8</td>
<td>63.1</td>
<td>63.6</td>
</tr>
<tr>
<td>High Budget Range</td>
<td>68.3</td>
<td>68.6</td>
<td>69.3</td>
</tr>
</tbody>
</table>

4.5. Improving Detection using Fixed Budget

This experiment simulates a real-world application of our method. We combine VOC07 and VOC12 data to simulate a larger pool of unlabeled images (called VOC0712) to choose from. As reported in Table 4, when the budget is set at 87.2% of the total budget (annotation budget for VOC07), we learn an object detector whose performance is the same as an FSOD detector trained on the entire strongly labeled VOC07 training set. This is a budget saving of 12.8%. Now, if we choose to use this total budget on VOC0712, our method achieves a 73.0% mAP, which is 2% improvement over the aforementioned FSOD detector on the same VOC07 test set. If all the 16551 images in the union of VOC07 and VOC12 set are strongly labeled, we use an extra 230% of budget and only improve 3.4% mAP.
**Figure 5:** Visualization of the selected images in each step. *Left:* Two examples in the warm-up set which is fully annotated by 10% budget. *Up-Right:* Strongly annotated images per step. They are hard examples including occlusion, multiple instance or tiny scale. *Bottom-Right:* Weakly annotated images per step. They are simple in the beginning but the difficulty increases when the detector is mature.

Table 4: Simulation of a larger unlabeled image pool. With 87.2% budget, BAOD achieves the same performance as a detector trained on fully annotated VOC07. If the budget equals to the total budget of VOC07, BAOD achieves 2% mAP improvement over FSOD with the same budget. Further annotating all the images can only improve 3.4% mAP.

<table>
<thead>
<tr>
<th>Train Set Pool</th>
<th>VOC07</th>
<th>VOC0712</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Budget*</td>
<td>100%</td>
<td>87.2%</td>
</tr>
<tr>
<td>Final mAP</td>
<td>0.710</td>
<td>0.710</td>
</tr>
</tbody>
</table>

* The values are normalized by VOC07 trainval set cost.

**4.6. Easy Images and Weak Annotation First**

We analyze the cost and number of images selected at every active selection step to investigate which type of training examples are more helpful in the sequential training process. Based on the final mAP of each category in VOC07 [14], we divide the twenty categories into three groups: Easy, Medium, and Hard. The left plot of Figure 4 illustrates that BAOD spends more budget to annotate Easy categories (shown in the green area) in the first several steps, while the cost of Hard categories (red area) increases when the detector becomes more accurate. These results align with concepts from curriculum learning, in which a larger number of easy samples can be trained on first to bootstrap the model and then hard samples are introduced progressively.

The right plot of Figure 4 measures the percentage of selected images per active step that belong to various subsets of the data (combinations of annotation type chosen and difficulty). Interestingly, the BAOD model tends to select more weak annotations at the beginning from all three groups of objects, since this kind of label is cheap and informative. Given a larger budget, the agent increases the proportion of strong annotations to further improve model performance. Note that there are no new weakly labeled images after active step 9 because all the images are already annotated.

**4.7. Qualitative Results of the Active Selection**

Figure 5 shows some selected strongly and weakly labeled images based on the BAOD experiment in section 4.5. Each row of the images is from the dog category in the VOC07 or VOC12 trainval set, and each column of images is selected at the same active step.

We show that in the first five active steps, strongly annotated images contain dog instances that are difficult to detect due to occlusion, multiple close instances or small-scale. In contrast, the selected weakly annotated images during the same steps are relatively easier to locate. As the model gets mature the difficulty for both levels of annotations increases. For example, the image chosen in step 4 contains three small dogs, and the dogs appeared in step 5’s images are small and black which makes them barely visible.

**5. Conclusion**

In summary, we introduce a novel budget-aware perspective to study the unexplored dimensions of the object detection problem. With a fixed budget, we compare both optimization and learn based sample methods to build diverse hybrid supervised object detection datasets which consist of both image level supervision and instance level supervision. The evaluation of detectors that learned from these budget fixed datasets shows that the handcrafted optimization method on uncertainty score outperforms other general active learning methods including random sampling, active learning, and reinforcement learning (shown in supplementary material). With the optimal set-up, our proposed budget-aware approach can achieve the performance of a strongly supervised detector on PASCAL-VOC 2007 while saving 12.8% of its original annotation budget. Furthermore, when 100% of the budget is used, our approach surpasses this performance by 2 percentage points of mAP.
References


Appendices

A. Solutions to Optimization Functions

A.1. Image Sampling

In section 3.1.1 and 3.1.2 of the submission, we propose two linear functions to approximate the mAP increment. Then we maximize it with several constraints. However, the proposed solution is an integer programming problem, and it cannot be solved in polynomial time. For example, the default solver uses Branch and Bound Algorithm (B&B) and it takes more than 24 hours to find a global maximum. To solve this problem more efficiently, we take the relaxation of the original integer program shown as Eq.7. Noted that the solution space of \( x_1, x_2, x_3 \) becomes an interval \([0, 1]\), but not a discrete set \( \{0, 1\} \), the relaxation Eq.7 can be solved in linear time.

\[
x_1, x_2, x_3 = \arg\max_{x_1, x_2, x_3 \in [0, 1]^N} s^\top(x_1 + x_3) + (\mu - s)^\top x_2
\]

s.t.
\[
\begin{align*}
x_3 & \leq \psi \\
x_1 + x_2 & \leq 1 - \psi \\
1^\top(ax_1 + bx_2 + cx_3) & \leq d \\
1^\top(ax_1 + bx_2 + cx_3) & \geq d - a
\end{align*}
\]

(7)

To project \( x_1, x_2, x_3 \) back to the discrete set \( \{0, 1\} \), we take three float numbers \( \epsilon_1, \epsilon_2, \epsilon_3 \) as the thresholds of \( x_1, x_2, x_3 \), respectively. In another word, every element in \( x_k \) larger than \( \epsilon_k \) is set as 1, otherwise it is 0.

\[
\begin{align*}
x_1^* & = 1\{x_1 > \epsilon_1\} \\
x_2^* & = 1\{x_2 > \epsilon_2\} \\
x_3^* & = 1\{x_3 > \epsilon_3\}
\end{align*}
\]

(8)

The solution needs to be feasible after thresholding. More concretely, the first two constraints must be satisfied because we cannot give an invalid action (e.g. It’s impossible to annotate an image both weakly and strongly, but it might appear from the solution).

We set \( \epsilon_1 = \epsilon, \epsilon_2 = 1 - \epsilon, \epsilon_3 = \epsilon \), where \( \epsilon \) exhaustively goes from 0 to 1 to satisfy the last two budget constraints. In this case, it can be proven that the solution of Eq.8 is also feasible from the original constraints.

**Proof.** (1) If \( \psi(k) = 0 \), where \( k \) can be the index of any element of vector \( \vec{x}_1 \). Since \( \vec{x}_1, \vec{x}_2, \vec{x}_3 \) is feasible, the first inequality in Eq.7 shows

\[
\hat{x}_3(k) \leq 0 \implies \hat{x}_3(k) = 0.
\]

(9)

The second inequality in Eq.7 gives the inequality for \( \hat{x}_1 \) and \( \hat{x}_2 \).

\[
\hat{x}_1(k) + \hat{x}_2(k) \leq 1 \implies 1 - \hat{x}_2(k) \geq \hat{x}_1(k) \implies 1\{1 - \hat{x}_2(k) < \epsilon\} \leq 1\{\hat{x}_1(k) < \epsilon\}
\]

(10)

Here \( 1\{\} \) is the indicator function. When we apply Eq.8 to check the two restrictions, the discrete solutions \( x_1^*, x_2^*, x_3^* \) are still feasible.

\[
\begin{align*}
x_3^*(k) & = 1\{\hat{x}_3(k) > \epsilon\} = 1\{0 > \epsilon_3\} = 0 = \psi(k) \\
x_1^*(k) + x_2^*(k) & = 1\{\hat{x}_1(k) > \epsilon\} + 1\{\hat{x}_2(k) > 1 - \epsilon\} \\
& = 1\{\hat{x}_1(k) > \epsilon\} + 1\{1 - \hat{x}_2(k) < \hat{x}_2(k)\} \\
& \leq 1\{\hat{x}_1(k) > \epsilon\} + 1\{\hat{x}_1(k) < \hat{x}_2(k)\} \leq 1 = 1 - \psi(k)
\end{align*}
\]

(11)

Here \( 1\{\} \) is the indicator function. When we apply Eq.8 to check the two restrictions, the discrete solutions \( x_1^*, x_2^*, x_3^* \) are still feasible.

\[
\begin{align*}
x_3^*(k) & = 1\{\hat{x}_3(k) > \epsilon\} = 1\{0 > \epsilon_3\} = 0 = \psi(k) \\
x_1^*(k) + x_2^*(k) & = 1\{\hat{x}_1(k) > \epsilon\} + 1\{\hat{x}_2(k) > 1 - \epsilon\} \\
& = 1\{\hat{x}_1(k) > \epsilon\} + 1\{1 - \hat{x}_2(k) < \hat{x}_2(k)\} \\
& \leq 1\{\hat{x}_1(k) > \epsilon\} + 1\{\hat{x}_1(k) < \hat{x}_2(k)\} \leq 1 = 1 - \psi(k)
\end{align*}
\]

(11)

Here \( 1\{\} \) is the indicator function. When we apply Eq.8 to check the two restrictions, the discrete solutions \( x_1^*, x_2^*, x_3^* \) are still feasible.

(2) If \( \psi(k) = 1 \), we have

\[
\begin{align*}
x_3^*(k) & \leq 1 \implies \hat{x}_3(k) = 1 \\
x_1^*(k) + x_2^*(k) & \leq 0 \implies \hat{x}_1(k), \hat{x}_2(k) = 0
\end{align*}
\]

(12)

Similar to previous discussion, we can have \( x_3^*(k) \leq 1 \) and \( x_1^*(k) = 0, x_2^*(k) = 0 \) to fit the two constraints.
A.2. Post Processing

In the article, we give another optimization function to do post processing. This procedure checks the consistency between the image and instance-level annotations, and it removes abundant pseudo labels.

Given an image with the weakly annotation $\omega \in \{0, 1\}^K$, where $K$ is the number of categories. We assume the first detection model (teacher model) gives $M$ positive predictions for the localization $P \in \mathbb{R}^{M \times 4}$, classification $A \in \{0, 1\}^{M \times K}$, and the positive class confidence vector $q \in [0, 1]^M$. Eventually, the post processing returns a sparse $M$-dimensional binary vector $y$ as Eq.13, where $\alpha = 0.3$, $\beta = 3$.

$$\begin{align*}
\min_{y \in \{0, 1\}^M} & -y^T q \\
\text{s.t.} & \quad y^T A(1 - w) = 0 \\
& \forall_{y_i, y_j = 1} IoU(P_i, P_j) \leq \alpha \\
& |y|_0 \in [1, \beta]
\end{align*}$$

In this problem, our objective function takes high confidence predictions by choosing $y \in \{0, 1\}^M$. In the first constraint, $y^T A$ accumulates the selected predictions by their categories. The following inner product with $(1 - w)$ returns zero only when all the predicted classes are in the weak annotation. On the other hand, the second constraint removes the heavily overlapped predictions while the third one makes the binary vector $y$ sparse. We develop Alg.1 to solve the above optimization.

Algorithm 1: Noise Cleaning

Input : Weak annotation $\omega \in \{0, 1\}^K$, localization $P \in \mathbb{R}^{M \times 4}$, classification $A \in \{0, 1\}^{M \times K}$, the positive class confidence for $M$ predictions $q \in [0, 1]^M$, $\alpha$, $\beta$;
Output: a binary vector $y \in \{0, 1\}^M$;
1 $y = 1$;
2 for $i = 1 : M$ do
3 $\quad$ if not $A(i, :) < \omega$, $y(i) = 0$;
4 $\quad$ end
5 $y' = $ NMS index for $[P,A,q]$, do it by class with threshold $\alpha$
6 $y = y \ast y'$
7 if $\text{sum}(y) > \beta$ then
8 $\quad$ assign the top $\beta$ remained predictions to $y$
9 end

B. More Budget-Aware Methods and Experiments

B.1. Sample from Small Uncertainty Images

In the Section 3.1 of the submission, to apply Uncertainty Sampling (US) method, we sort images in descending order of uncertainty and choose high uncertainty images. Moreover, we also conducted an experiment in which images were sorted by the uncertainty score in an ascending order, and we pick low uncertainty images in every step. The other settings are the same to US’s. Table 5 compares these two methods.

Table 5: Budget-Average mAP using Largest and Smallest Uncertainty Scores. When the budget is low (10%-30%), collecting images with small uncertainty score (shown as SUS) is more effective. However, when the dataset size is large enough, choosing large uncertainty scores (shown as LUS) is preferable.
When the budget is in low range (10% – 30%), collecting images with small uncertainty score is more effective. However, when the dataset size is larger, choosing large uncertainty scores is preferable. This behaviour follows the conclusions of Bengio et al. [3], who suggested that it is better to train models using curriculum learning, i.e. giving easy images to the model in early training and then giving hard images once the model has learned better representations. This experiment also shows that there are better solutions to the BAOD problem. For example, we can find a better sampling function which also considers the current state of the detection model. Assembling the sampling function into the optimization selection method will produce a better budget-mAP performance.

B.2. Reinforcement Learning

B.2.1 Method

The problem of finding the next active batch of training examples can be naturally formulated as a reinforcement learning problem. This approach allows us to learn a strategy directly from the trial-and-error experience and to avoid the specific mathematical model of the previous sections. The reinforcement learning methods do not need any prior knowledge about the environment, but they can still construct an optimal strategy. Thus, it is easily extensible to stochastic environments with variable set-ups such as strongly-weakly annotation costs ratio and a total amount of budget.

Suppose that we already have a training set, and we regress a detector with performance $m_0$. In the next active step, we evaluate the uncertainties of a subset of unlabeled images. The active learning agent interacts with the environment in the form of the difficulty preference and annotation type. We use $O_t$ as defined in section 3.1.2 of the submission as the observed state, which considers both the classification and localization performance. The agent returns two actions from current state: (1) asking for low/medium/high uncertainty images and (2) giving a strongly/weakly annotation. In conclusion, the RL agent gives a choice $a$ from three by two possible actions matrix.

In the RL training process, we take the detection model increment performance $\Delta_t$ as a reward. If there is still enough budget, the agent chooses the next active batch from the new observed state $O_{t+1}$. If we run out of budget after $T$ active steps, the total reward is $R = \sum_{t=1}^{T} \Delta_t = m_T - m_0$. Here $m_0$ is the performance of a randomized warm-up model with a fixed seed. By maximizing the return $R$, the agent learns a policy $\pi$ that maximizes the final model performance $m_T$.

To train the agent, we learn a policy with $Q$-learning[57] which learns to approximate $Q$-function $Q^\pi(a, O_t)$ with a three-layer neural network. Given an action $a$ and the current state $O_t$, it returns the expected value from policy $\pi$.

B.2.2 Experiment

At every active step, the $Q$ agent gives insights on which images to label and which type of annotations is required. Different from the previous pipeline, the RL method does not evaluate all the unlabeled images, but renders a preference for easy, middle or hard images. This significantly expedites the training process. We firstly compute the number of required images from the budget and annotation types, then we stop evaluating unlabeled images’ uncertainties, once we get enough images. Table 5 shows that RL performs better than FSOD, since it learns a policy to weakly or strongly annotate images and takes advantage of hybrid training. Nevertheless, it performs worse than US and LAL optimization. The reinforcement learning environment is built by VOC12 training set and validation set. We simulate the active learning process on four Titan Xp GPUs for five days to train the $Q$-learning agent, and the agent is used to solve the budget aware object detection problem on VOC07 trainval set and test set.

C. Definition of Easy, Medium and Hard Classes

We analyze the cost and number of images selected at every active selection step to investigate which type of training examples is more helpful in the sequential training process. Based on the final mAP of each category in VOC07, we divide the twenty categories into three groups: Easy, Medium and Hard. The mapping is motivated and shown in Table 2.

D. Discussion on Annotating Time

In the submission, we assume $a = 34.5$ or $a = 7$, and $b = 1.6$. We explain these values in this section.

First of all, annotating an image does not take a fixed time. From the technique used in ImageNet[50], the Median of annotating a bounding box is 25.5s, but an extra of 9.0s and 7.8s is required to control its quality. However, using the state-of-the-art method extreme clicking for efficient object annotation [41] can reduce the strongly annotating time to 7s. To consider most bounding box annotation methods, we assume this value to be 7s, 21s or 34.5s. On the other hand, answering a binary question takes around 1.6 seconds[40], but answering twenty binary question does not take $1.6 \times 20 = 32s$. [40]
Table 6: The mapping from twenty categories in PASCAL VOC to Easy, Medium and Hard classes. This mapping is based on the final mAP of each category.

<table>
<thead>
<tr>
<th>Category</th>
<th>Easy</th>
<th>Medium</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cat</td>
<td>88.9</td>
<td>82.3</td>
<td>75.8</td>
</tr>
<tr>
<td>car</td>
<td>88.4</td>
<td>82.0</td>
<td>65.7</td>
</tr>
<tr>
<td>bus</td>
<td>88.1</td>
<td>80.4</td>
<td>65.7</td>
</tr>
<tr>
<td>horse</td>
<td>87.6</td>
<td>80.1</td>
<td>68.9</td>
</tr>
<tr>
<td>train</td>
<td>86.6</td>
<td>78.9</td>
<td>70.8</td>
</tr>
<tr>
<td>cow</td>
<td>86.3</td>
<td>77.7</td>
<td>63.6</td>
</tr>
<tr>
<td>dog</td>
<td>85.9</td>
<td>78.9</td>
<td>53.6</td>
</tr>
<tr>
<td>caro</td>
<td>84.3</td>
<td>84.3</td>
<td>77.7</td>
</tr>
<tr>
<td>person</td>
<td>82.3</td>
<td>78.9</td>
<td>75.8</td>
</tr>
<tr>
<td>bike</td>
<td>88.4</td>
<td>84.3</td>
<td>75.8</td>
</tr>
<tr>
<td>sheep</td>
<td>86.3</td>
<td>82.3</td>
<td>80.1</td>
</tr>
<tr>
<td>mbike</td>
<td>88.1</td>
<td>80.1</td>
<td>75.8</td>
</tr>
<tr>
<td>tv</td>
<td>87.6</td>
<td>82.3</td>
<td>75.8</td>
</tr>
<tr>
<td>sofa</td>
<td>88.1</td>
<td>82.3</td>
<td>75.8</td>
</tr>
<tr>
<td>table</td>
<td>88.4</td>
<td>82.3</td>
<td>75.8</td>
</tr>
<tr>
<td>boat</td>
<td>87.6</td>
<td>82.3</td>
<td>75.8</td>
</tr>
<tr>
<td>bottle</td>
<td>88.1</td>
<td>82.3</td>
<td>75.8</td>
</tr>
<tr>
<td>chair</td>
<td>86.3</td>
<td>82.3</td>
<td>75.8</td>
</tr>
<tr>
<td>plant</td>
<td>86.6</td>
<td>82.3</td>
<td>75.8</td>
</tr>
</tbody>
</table>

* data is from the last setup of Table 6 in the report [58].

finds that clicking on one of five on-screen buttons takes around 2.4 second. In the original ImageNet paper, people select a sequence of queries for humans such that we achieve the same labeling results with only a fraction of the cost of the naive approach. It is more efficient to organize the semantic concepts into hierarchies [53]. More importantly images are collected from the internet through keyword search, which implies that many image level labels are free.

In our Budget-Aware Object Detection problem, we only care about budget ratios between strong annotations and weak annotations, although there are multiple instances in one image. For this reason, we have experiments with different ratios to compensate the cost assumptions and to show that at different ratios our method still works. In the last chapter of the supplementary material, we have a table to show the experiments with different cost ratios (CR) in \{34.5 : 1.0, 21 : 1.6, 7 : 1.6\} of strong and weak annotations.

E. Budget-mAP Curves

We present the original Budget vs mAP figures for Table 1,2 and 3 in the submission, respectively. The curves shown here are mAP scores before integration, which are more detailed but noisy. Please check section F for the table representation.

Figure 6: left: Budget-mAP curves using fully and hybrid training pipelines. Orange bars compare the two training pipelines using random sample while Blue bars show both pipelines using US sampling. Middle: Budget-mAP curves using FSOD, hybrid training, and optimization methods. Blue bars show US using FSOD. Orange bars show RS hybrid baseline. Gold bars show US Optimization (BAOD). Dark Grey Bars show LAL Optimization. right: Budget-mAP curves using a lower cost for strong annotations. Blue Bars represent FSOD with US sampling. Orange Bars represent Hybrid training with RS sampling. Gold Bars represent US Optimization. All the methods use a smaller gap between the weak and strong annotation costs.
### Table 7: Table Representation for Raw Experiment Data

<table>
<thead>
<tr>
<th></th>
<th>Budget Percentages</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10.8</td>
<td>15.7</td>
</tr>
<tr>
<td>RS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPT-LS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPT-LAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOC0712</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Budget</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPT-LS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPT-LAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOC0712</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* H for Hybrid Learning  
** CR for cost ratio of strong and weak annotations
G. More Visualizations

Below we show some visualization figures of strongly labelled and weakly images for each of the classes chosen by US based Optimization. Each column indicates one step of the active learning pipeline and each of the rows represents a class. The warm up set is represented on the first column showing only strongly labeled images chosen randomly. For the next 5 steps we can observe on the top sub-row strongly labeled images and on the bottom sub-row weakly labeled images. The bounding boxes shown in this figure are not the original annotations from PASCAL, they were drawn manually only for this visualization purposes.
Class 5

Class 6

Class 7

Class 8

Class 9
<table>
<thead>
<tr>
<th>Class 10</th>
<th>Strongly Annotate</th>
<th>Weakly Annotate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 11</td>
<td>Strongly Annotate</td>
<td>Weakly Annotate</td>
</tr>
<tr>
<td>Class 12</td>
<td>Strongly Annotate</td>
<td>Weakly Annotate</td>
</tr>
<tr>
<td>Class 13</td>
<td>Strongly Annotate</td>
<td>Weakly Annotate</td>
</tr>
<tr>
<td>Class 14</td>
<td>Strongly Annotate</td>
<td>Weakly Annotate</td>
</tr>
<tr>
<td>Class 15</td>
<td>Strongly Annotate</td>
<td>Weakly Annotate</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class 16</th>
<th>Strongly Annotate</th>
<th>Weakly Annotate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class 17</th>
<th>Strongly Annotate</th>
<th>Weakly Annotate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class 18</th>
<th>Strongly Annotate</th>
<th>Weakly Annotate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class 19</th>
<th>Strongly Annotate</th>
<th>Weakly Annotate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 8: **Visualization of the results for each of the 20 classes of PASCAL VOC.** Each column indicates one step of the active learning pipeline and each of the rows represents a class. Step 0 is represented on the first column showing only strongly labeled images chosen randomly. For the next 5 steps we can observe on the top sub-row strongly labeled images and on the bottom sub-row weakly labeled images. The bounding boxes shown in this figure are not the original annotations from PASCAL, they were drawn manually only for this visualization purposes.
References


[40] D. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari. We don’t need no bounding-boxes: Training object class detectors using only human verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 854–863, 2016. 1, 2, 3, 14


