Weakly-Supervised Object Detection Learning through Human-Robot Interaction

Elisa Maiettini[1] Vadim Tikhanoff[2] Lorenzo Natale[1]

***Abstract*— Reliable perception and efficient adaptation to novel conditions are priority skills for humanoids that function in dynamic environments. The vast advancements in latest computer vision research, brought by deep learning methods, are appealing for the robotics community. However, their adoption in applied domains is not straightforward since adapting them to new tasks is strongly demanding in terms of annotated data and optimization time. Nevertheless, robotic platforms, and especially humanoids, present opportunities (such as additional sensors and the chance to explore the environment) that can be exploited to overcome these issues.**

In this paper, we present a pipeline for efficiently training an object detection system on a humanoid robot. The proposed system allows to iteratively adapt an object detection model to novel scenarios, by exploiting: (i) a teacher-learner pipeline, (ii) weakly supervised learning techniques to reduce the human labeling effort and (iii) an on-line learning approach for fast model re-training. We use the R1 humanoid robot for both testing the proposed pipeline in a real-time application and acquire sequences of images to benchmark the method. We made the code of the application publicly available.

I. INTRODUCTION

Much research on robot vision draws from techniques developed in computer vision. However, robot applications have specific requirements. Particularly, high reliability and fast adaptation are both fundamental requisites of vision systems for robots that need to operate in unconstrained and dynamic environments. Mainstream computer vision solutions typically rely on deep learning based models which have large amount of parameters that need to be tuned at training time. This has well known implications: firstly, the amount of effort required to provide annotations for the, typically, very large training set and, secondly, long training time. The remarkable performance achieved with such techniques are, therefore, appealing for robotics, but their adoption in robotic applications remains limited to those problems for which annotated, large datasets are available and there is no need for on-line adaptation or re-training.

Nevertheless, robotics offers some opportunities that are often unexplored in the literature. For instance, the embodiment of the robot can be exploited to interact with the environment, including humans, to actively acquire training data. Moreover, since robots are frequently equipped with multiple sensory modalities, there is additional information that can be used to aid learning. This is especially true for humanoid robots, such us the iCub [1] and R1 [2], which are equipped with depth, force or tactile sensors.

[1] Humanoid Sensing and Perception, Istituto Italiano di Tecnologia, Genoa, Italy

[2] iCub Tech, Istituto Italiano di Tecnologia, Genoa, Italy

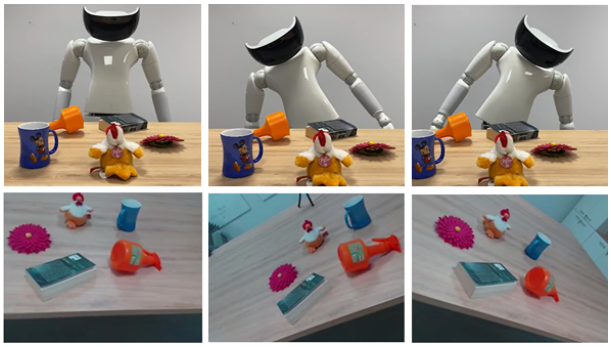

Fig. 1. Example frames of the R1's exploratory behaviors used in the proposed application. Different R1's poses (first row) allows to acquire different views of the objects of interest (second row).

In this paper, we describe the implementation of a complete pipeline for training an object detection system, embedded on a humanoid robot. The system allows to train and iteratively adapt an object detection model, with minimal labeling effort. To do so, the system is designed to conveniently exploit (i) the interaction with a human teacher, (ii) a fixed set of exploratory behaviors and (iii) a weakly-supervised learning algorithm, to reduce the number of frames that are manually annotated while preserving performance. To achieve a smooth interaction between the robot, the environment and the teacher, the system also integrates an architecture for object detection that can be quickly re-trained on-line.

We implemented the system on the R1 humanoid robot [2], to demonstrate its performance in a real-time application. The functioning of the system is demonstrated in the video provided as supplementary material. In addition, we extensively benchmark the system using a series of sequences that were recorded and manually annotated to provide ground-truth, demonstrating the effectiveness of the pipeline in reducing the annotation effort, while retaining performance. Finally, we made the code of the application publicly available[1] for reproducibility.

The remaining of this paper is organized as follows: in Sec. II, we introduce the state-of-the-art of current object detection in robotics. In Sec. III, we present the components of the proposed pipeline, which are analyzed and validated in Sec. IV. Finally, Sec. V concludes the paper.

[1] https://github.com/robotology/online-detection-demo

II. RELATED WORK

Latest research on object detection for robotics mainly focuses on improving precision in particularly difficult scenarios such as, occlusion and clutter [3], [4], [5], [6], [7]. To this aim, a widely used approach is to rely on deep learning based methods (like e.g. Faster R-CNN [8] and its evolution, Mask R-CNN [9], YOLO [10], Centernet [11] and Cornernet [12]) which demonstrated remarkable performance on general purpose [13], [14], [15] and robotics [16] datasets. A common approach in the state-of-the-art, characterizing the so-called *region-based* approaches, is that of splitting the detection pipeline into three main stages: (i) generation of region candidates that might contain the objects of interest, (ii) region proposals encoding into convolutional features and (iii) region classification and refinement. In the last years, the common trend has been to integrate these three stages into monolithic models, trained end-to-end via back-propagation [8], [9], [17], [18]. A drawback of such architectures is that the training process requires large annotated datasets and long optimization time, and it is, therefore, badly suited for robots operating in dynamic environments. In the following paragraphs, we cover the work that has been done in the literature in order to relieve deep learning based methods requirements in terms of training time and annotated data.

Time-efficient object detection learning. Even though the major trend is to design detection models as monolithic architectures, trained end-to-end, another natural approach is to consider multi-stage architectures, tackling each step of the detection pipeline separately (see e.g., [19], [20] and specifically [21], [22]). As showed in [22], this latter approach is key to obtain fast adaptation capabilities to a novel task. Specifically, the proposed on-line method [22] is composed of a deep learning-based region proposal and feature extractor (based on Faster R-CNN [8]), followed by a Kernel based method for classification and a boostrapping approach to address the well known issue of background-foreground imbalance in object detection [17]. This pipeline is suited for a typical unconstrained robotic setting, as, while keeping the pre-trained feature extractor fixed, the combination of an efficient classifier (FALKON [23], [24]) with an approximated hard negatives bootstrapping (see [22]) provides fast model training. Moreover, lately, the same approach has been adapted to the region proposal generation [25] and object instance segmentation [26]. In this work, we build on [22], proposing a unified pipeline to rapidly train and update a detection model on novel tasks and scenarios, with different human robot interactions for ground-truth acquisition.

Data-efficient object detection learning. In past work [27], the annotation problem has been addressed by exploiting a human robot interaction to collect automatically annotated images. Specifically, within a teacher-learner pipeline [28], motion and depth cues are used to follow with the robot's gaze the object, segment it and automatically assign the correspondent bounding box. While allowing for a natural interaction and accurate object detection [27], this pipeline limits the generalization capabilities to novel, unseen, scenarios [29]. This issue has been addressed with promising results in [30], [29] by exploiting (i) the large amount of un-labeled images acquired by the robot during operation and (ii) the weakly-supervised learning family of techniques (WSL) [31], [32]. In WSL, the most relevant sub-classes of methods, for this work, are *Active Learning* (AL) and *Self-supervised Learning* (SSL). In AL [33], the effort is focused on defining a sample selection policy to choose the most informative samples to be asked for annotation to a human. Recently, AL has been successfully applied to the object detection task [34], [35], [36], [37]. SSL, instead, attempts to exploit the unlabeled instances without querying human experts, by e.g., using the high-confident predictions as pseudo ground-truth [38]. This latter family of approaches does not imply any human annotation, however it may suffer of model drift and degradation due to errors in the self-supervision. Conversely, AL, while being more stable, still requires a human labeling. Recent approaches integrate both AL and SSL into the same detection pipeline, like e.g., the Self-supervised Sample Mining (SSM) [39], [40]. Finally, in [29], the SSM has been integrated with the aforementioned fast object detection learning [22]. While the approach is promising, the required number of manually labeled images (in the order of a few hundreds for a task of 30 objects [29]), prevents from performing the model refinement on-line. In this work, we improve this aspect by integrating a bounding box tracker [41], to further reduce the number of annotations required, by propagating the provided labels between consecutive frames. Moreover, we present a unified system for on-line training and updating an object detection model, which allows to adapt to novel tasks and scenarios with active exploration and human robot interaction.

III. METHODS

In this work, a robot is asked to detect a set of object instances in an unconstrained environment. The aim of the presented application is that of providing a system that allows training and subsequently updating an object detection model. This is done by means of different human robot interaction modalities, while reducing annotation effort by leveraging on autonomous explorative behaviors and a weakly-supervised learning strategy.

A. Overview of the pipeline

The proposed pipeline is initially trained during a brief interaction with a human, in a constrained scenario (hereinafter referred to as *Supervised training phase*). The goal of this initial interaction is that of "bootstrapping" the system, with an initial object detection model. After that, the robot relies on this initial knowledge to adapt to new settings, by actively exploring the environment and asking for limited human intervention (hereinafter referred to as *Refinement*

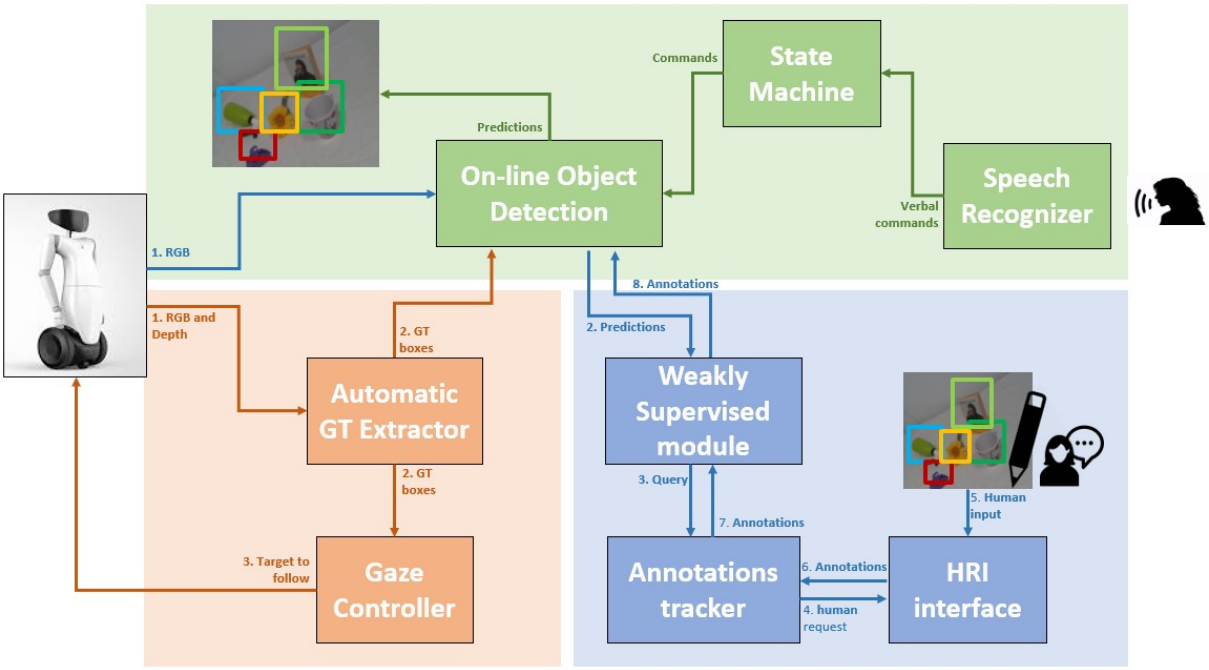

Fig. 2. Architecture of the proposed pipeline. The green block represents the *Object detection system*, the red block is the *Automatic in-hand supervision extraction* and finally, the blue block depicts the *Weak-supervision system*. The flow of red arrows represents the *Supervised training phase*, the blue one represents the *Refinement phase* and the green one is in common between the two phases. Refer to Sec. III for further details.

phase). The *Refinement phase* iteratively builds new training sets. Using the initial detector, the robot detects candidate bounding boxes on the incoming images. These predictions are evaluated to determine their confidence. High confidence labels are added to the training set, with a SSL strategy. Instead, when low confidence bounding boxes are detected, the robot stops the exploration and asks the human teacher to either accept or refine them, with an AL strategy. The teacher provides refined annotations using a graphical interface on a tablet. The new annotations are used to initialize a set of trackers, which propagate bounding boxes as the exploration resumes.

To summarize, the proposed application, is represented in Fig. 2. This can be divided in three main components: (i) the *Object detection system* (green blocks), (ii) the *Automatic in-hand supervision extraction* (red blocks) and (iii) the *Weak-supervision system* (blue blocks). We rely on the YARP[2] middleware for connecting the different modules. In this section, we describe the main components of the presented pipeline.

B. Object detection system

The *Object detection system* is composed of three main modules: (i) the *State Machine*, (ii) the *Speech recognizer* and (iii) the *On-line object detection*. The former one, which is implemented in LUA[3] and rFSM[4], orchestrates the different components of the application. There are three main states: (i) *Inference*, which is the default state, (ii) *Supervised train* and (iii) *Weakly-supervised train*. The different states are triggered by the user's verbal commands or questions which are received by means of the *Speech recognizer* module. The *On-line object detection* is described in details in the following paragraphs.

On-line object detection. This module detects the objects of interest from the stream of images from the robot's cameras. For its implementation, we follow the on-line object detection learning method proposed in [22]. This is a *region-based* approach (see Sec. II) which consists of two stages: (i) region proposals and feature extraction, and (ii) region classification and bounding box refinement. The first stage relies on layers from Faster R-CNN [8] (specifically, the convolutional layers, the Region Proposal Network [8] and the *RoI Pooling layer* [20]). This part is used to extract from an image a set of candidate regions that might contain the objects of interest, so-called Regions of Interest (RoIs), and encode them into a set of convolutional features. The second stage is composed of a set of Kernel based binary classifiers (one for each class of the considered detection task) and Regularized Least Squares (RLS) [42], respectively for the classification and the refinement of the proposed RoIs. We used the recently proposed FALKON [23], [24] for region classification. While the first stage is trained only once and off-line on the available data, the second one can be updated on-line as new data come. Specifically, the classifiers are trained with an approximate bootstrapping approach, called Minibootstrap [22], which addresses the well-known issue in

[2] https://www.yarp.it
[3] https://www.lua.org/
[4] https://github.com/kmarkus/rFSM

object detection of background-foreground class imbalance, while maintaining a short training time.

During the *Supervised training phase* (represented by the flow of red arrows in Fig. 2), the application is in the *Supervised train* state and the detection system processes the incoming images and labels, training a new FALKON classifier and a new RLS refiner for each novel class. During the *Refinement phase* (represented by the flow of blue arrows in Fig. 2), instead, the application's state is *Weakly-supervised train* and the detection model is refined on the new scenario. Firstly, the current detection model predicts the incoming unlabeled images. Then the *Weakly-supervised module* evaluates these predictions (as described in Sec. III-D), adding the confident ones to the dataset as self-supervised pseudo ground-truth (with a SSL strategy) and asking for manual annotation for the uncertain ones (with an AL strategy). When the acquisition is completed, the *On-line object detection* updates the FALKON classifiers and the RLS models on the novel data. Finally, during the *Inference* state, the detection system receives the stream of images as input and provides a list of predicted detections as output.

C. Automatic in-hand supervision extraction

This component (red block in Fig. 2) implements the procedure for the automatic annotation acquisition for handheld objects which is used during the *Supervised training phase*. It is composed of the *Automatic ground-truth extractor*[5] and the *Gaze controller*[5] (see Fig. 2). For the former we rely on [27]. This pipeline exploits the depth information and human-robot interaction in order to segment the blob of pixels belonging to the object of interest. Specifically, the human shows the object in front of the camera of the robot. A tracking routine [28] selects the pixels from the depth map that are closer to the robot, segmenting them from the background. It computes a bounding box surrounding this particular blob and sends it to the *Object detection system* and to the *Gaze controller*. The former uses the bounding box as ground-truth for training a new detection model during the *Supervised training phase*, while the *Gaze controller* uses this information to generate a target point to follow in order to make the robot track the object of interest.

D. Weak-supervision system

This component (blue block in Fig. 2) implements the weakly-supervised learning strategy that allows for the detection model's refinement during the *Refinement phase*. This is composed by four main modules: (i) the *Exploration module*, (ii) the *Weakly-supervised module*, (iii) the *Annotations tracker* and (iv) the *Human-Robot Interaction (HRI) interface*.

Exploration module. This module makes the robot follow a specified trajectory, with the aim of exploring the surrounding environment, acquiring different views of the objects. This exploration needs to be paused when the human intervention is required to provide manual annotation and resumed when the labeling process is accomplished. For this work, we considered a fixed set of exploratory movements for the upper body of the robot to acquire new views of the objects. Refer to Fig. 1 for examples frames of the R1's exploratory movements used for the proposed pipeline. More sophisticated actions may be performed, for example, to actively manipulate objects.

Weakly-supervised module. During the exploration, this module receives the predictions of the stream of images from the *On-line object detection*, it evaluates them and decides whether to use them as self-supervision (with a SSL strategy) or to ask for an external labeling (with an AL strategy). For this module, we relied on the weakly-supervised strategy proposed in [30] and developed in [29]. Specifically, first a *Scoring function* assigns a confidence score to each unlabeled image, based on the received predicted detections. This confidence score is, then, used by a *Selection policy* to decide whether an image needs to be queried for annotation or the predicted detections are confident enough to be used for self-supervision. Note that, we simplify the *Scoring function* with respect to the one used in [29] to reduce the per-image processing time. Specifically, instead of using the so-called *Cross-image validation* (see [39] and [29]), the confidence score of the image is obtained by averaging the confidence scores of the single predictions. Then, similarly to previous work, the *Selection policy* compares the per-image confidence scores with two thresholds, namely th_l and th_h. If the confidence score is lower than the th_l, the image is considered doubtful and thus asked for annotation (AL). If instead the score is higher than th_h the predictions in the image are considered confident and thus can be used as pseudo ground-truth for refinement (SSL). All the other predicted images are not used. Using an average score for the whole image in some cases may lead the system to use bounding boxes whose individual score is low. To avoid this situation, we also introduce a minimum threshold (namely, th_m) and verify that all bounding boxes within an image have a score that is at least above this minimum threshold, if this is not the case, the whole image is marked for annotation regardless the average confidence score.

HRI interface and Annotations tracker. Finally, one of the main contributions of this work relies on the *HRI interface* and *Annotations tracker* modules. The former simplifies the annotation process, while the latter propagates the labels provided by the human across consecutive frames. Specifically, when the *Weakly-supervised module* requires annotations for an image, the *HRI interface* is activated, so that the teacher can annotate the frame. This is done with a smooth labeling procedure: the *HRI interface* receives verbal commands and it opens an interactive window where the user can draw the correct bounding boxes or refine the ones predicted by the detection model. Finally, provided annotations are used to initialize the *Annotations tracker* which propagate them in future frames, further reducing

[5]Module taken from `https://github.com/robotology`

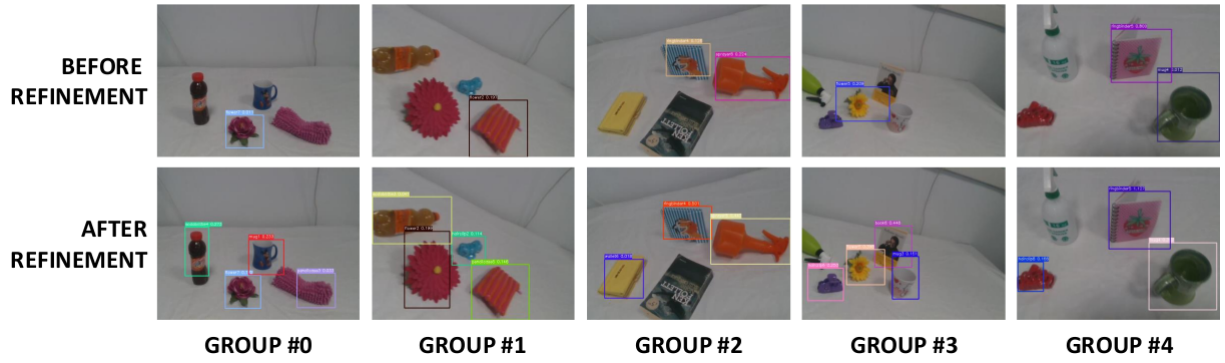

Fig. 3. Examples of detected images, from the 5 objects groups, obtained by using the BEFORE REFINEMENT MODEL and the AFTER REFINEMENT MODEL on new unseen sequences. See Sec. IV-C for details.

the human labeling effort. For this module, we rely on the multi-object tracker *Re3* [41], which we integrated in our application. To avoid the tracker to diverge in case of extreme clutter or occlusions, causing a deterioration of the bounding boxes, we evaluate their quality by computing the overlap between different objects. When the overlap is too high, the quality is considered low. In this case, the robot asks the human teacher to provide a new set of annotations and the tracker is re-initialized. In Sec. IV-B, we quantify the precision of the bounding boxes provided by the tracker and demonstrate that this allows to remarkably reduce the manual annotation effort.

IV. EXPERIMENTS

While the video submitted as supplementary material shows the functioning of the proposed application, in this section, we demonstrate its effectiveness reporting on the performed empirical evaluation. The experimental validation focuses on the refinement pipeline, because the validation of the initial *Supervised training phase* has been explored in previous work [27], [22]. We consider, therefore, the refinement following the initial supervised training of the detector by means of the *Automatic in-hand supervision extraction*, presented in Sec. III-C. This model needs to be refined in order to generalize to a different setting (i.e., a table top) by exploiting unlabeled data collected by the robot during exploration (by means of the *Weak-supervision system*, presented in Sec. III-D).

Given the focus of the experimental validation, for the *Supervised training phase*, we use previously acquired data, i.e. the iCubWorld-Transformations dataset [43] (iCWT). iCWT contains images of 200 objects, demonstrated by a human teacher to the robot, within the conditions described in Sec. III-C. To benchmark the performance of the system after the *Refinement phase* and compare the detection performance, we selected 21 objects, that are part of the 200 objects in iCWT, and divided them in 5 groups. For each group, the objects were placed on a table in two different arrangements. Then, each set was presented to the robot for the *Refinement phase*. During the exploration, the sequence of images were recorded and, subsequently, manually annotated. Manual annotation of the sequences was required to obtain ground-truth for benchmarking purposes, i.e. the ground-truth acquired in this way was not used for training (we call these sequences TABLE-TOP in the remainder of the paper). In the following sections, we report results for 5 different experiments, one for each objects group in the TABLE-TOP. During each experiment, the robot performed a refinment following the *Refinement phase* (see Sec. III). Note that, during this latter phase a human manually provides the annotations, by means of the *HRI interface*.

TABLE I

THIS TABLE REPORTS ON THE REFINEMENT CAPABILITIES OF THE PROPOSED APPROACH, COMPARING BEFORE REFINEMENT MODEL WITH THE AFTER REFINEMENT MODEL. REFER TO SEC. IV-A FOR DETAILS.

Objects group	Before ref. (mAP(%))	After ref. (mAP(%))	Manual annotations
#0	14.4%	90.6%	4 images (12 bbox)
#1	58.2%	92.9%	3 images (10 bbox)
#2	44.1%	95.1%	4 images (12 bbox)
#3	36.2%	78.4%	4 images (12 bbox)
#4	48.5%	87.9%	5 images (13 bbox)

In the reported experimental analysis, for the *On-line object detection*, we adopt the CNN model proposed in [44] as convolutional backbone for the Faster R-CNN based feature extractor. This has been trained on a 100 objects identification task from the iCWT (as explained in [22]) and we used the learned weights for the subsequent on-line learning steps. Note that, the 100 objects have been chosen excluding the 21 in the TABLE-TOP. We rely on [29] to set the hyper-parameters of FALKON and the Minibootstrap. Then, for the *Weakly-supervised module*, we empirically set the $th_l = 0.3$, $th_h = 0.4$ and $th_m = 0.1$.

We report performance in terms of mAP (mean Average Precision) with the IoU (Intersection over Union) threshold set to 0.5, as defined for Pascal VOC 2007 (see [13] for further details).

A. Detection refinement evaluation

For each objects group previously described, we define the two following detection models:

TABLE II

THIS TABLE ANALYZES THE IMPACT OF THE DIFFERENT COMPONENTS OF THE WEAK-SUPERVISION SYSTEM. REFER TO SEC. IV-B FOR DETAILS.

Objects group	Manual annotations	Total AL queries	Total SSL	Tracker (mAP(%))	Pseudo labels (mAP(%))
#0	4 images (12 bbox)	200 images (800 bbox)	0	88.0%	88.0%
#1	3 images (10 bbox)	173 images (692 bbox)	5	97.4%	92.9%
#2	4 images (12 bbox)	197 images (788 bbox)	0	97.0%	97.0%
#3	4 images (12 bbox)	134 images (670 bbox)	24	84.4%	76.0%
#4	5 images (13 bbox)	163 images (652 bbox)	1	95.3%	95.4%

- BEFORE REFINEMENT MODEL, which is obtained after the *Supervised training phase* as described previously.
- AFTER REFINEMENT MODEL, which, instead, is obtained after the *Refinement phase*, updating the detection model, following each TABLE-TOP exploration sequences.

The obtained results are reported in Tab. I. For each of the 5 objects group, we report the accuracy on one of the TABLE-TOP sequences, obtained by (i) the BEFORE REFINEMENT MODEL (**Before ref.** column in Tab. I) and (ii) the AFTER REFINEMENT MODEL which has been refined during that exploration sequence (**After ref.** column in Tab. I). Finally, we report the number of manual annotations required to the human, both in terms of images and manually drawn bounding boxes (**Manual annotation** column in Tab. I).

As it can be observed, for all the 5 objects groups, as expected, the BEFORE REFINEMENT MODEL performs poorly on the new table top scenario. The reason for this is twofold. Firstly, there is a considerable domain shift between the *Supervised training phase* (handheld objects) and the *Refinement phase* (table top). This shift affects the detection accuracy. Secondly, the number of images per objects (150) used during the *Supervised training phase* is small. Note that, this is roughly half of the images used for the same purposes in previous works [30], [29]. This is done in order to show performance of the model under limited annotated data budgets. Both these aspects negatively affect the detection accuracy of the BEFORE REFINEMENT MODEL in the new conditions. However, after the refinement process, the accuracy increases remarkably, for all the 5 objects groups (**After ref.** column in Tab. I), showing the effectiveness of the proposed approach. Notice that this process is performed with quite limited human intervention. The user, indeed, annotated not more than 5 images per group, drawing a total of 13 or less bounding boxes per group.

B. Weak-supervision system analysis

In this section, we analyze the impact of the different components of the *Weak-supervision system* during the *Refinement phase*. To this aim, for each of the 5 objects groups used for the experiment in the previous section, we report in Tab. II: (i) the comparison between the number of manual annotations required by the *Weakly-supervised module* and the ones actually drawn by the human, both in terms of images and drawn bounding boxes (respectively, **Total AL queries** and **Manual annotation** columns in Tab. II), (ii) the number of images used as self-supervision (**Total SSL** column in Tab. II), (iii) the accuracy of the tracker (**Tracker** column in Tab. II) and (iv) the accuracy of the bounding boxes used as ground-truth during the *Refinement phase* (**Pseudo labels** column in Tab. II).

Firstly, as it can be noted, the actual number of AL queries is greatly higher than the number of annotations provided by the human, for all objects groups. This shows the effectiveness of the *Weak-supervision system* in reducing the labeling effort with respect to previous work [29]. This is achieved mainly by the contributions of the self-supervision and the *Annotations tracker*. Regarding the former one, it can be noted in Tab. II that the total number of annotation queries is significantly higher than the number of images chosen for self-supervision during the refinement process. This is mainly caused by the poor performance obtained by the BEFORE REFINEMENT MODEL on the table top scenario due to domain shift. Nonetheless, in 3 out of 5 cases, self-supervision helps in reducing the amount of annotations required. However, as it can be observed in Tab. II, the main contribution in the reduction of the required manual annotations is given by the introduction of the *Annotations tracker*. Indeed, even if the total number of queries is between 134 and 200 (ranging between 652 and 800 of requested bounding boxes), the number of images that were actually annotated by the human is significantly smaller (5 or less), with a correspondent number of requested bounding boxes ranging from 10 to 13. This large reduction allows performing the refinement process on-line, during robot exploration, interactively providing the labels when requested.

Tab. II column **Pseudo labels** shows that the accuracy of the labels considered for the refinement is not perfect (i.e. $< 100\%$). One may argue that noise in the annotations provided by the self-supervision and the *Annotations tracker* may have a negative impact on the refined model. However, except for one case (object group 3), the quality of the used annotations is overall quite high (i.e. $> 88\%$). Moreover, the accuracy achieved in Sec. IV-A demonstrates that the annotation noise does not affect the refinement process, allowing to successfully recover the drop of performance obtained by the BEFORE REFINEMENT MODEL with a small price in terms of additional manual annotations.

C. Generalization capabilities evaluation

Finally, in this section we evaluate the generalization capabilities of the proposed pipeline. To this aim, for each of the 5 objects groups, we evaluate the same models obtained in

TABLE III

THIS TABLE REPORTS THE ACCURACY OF THE BEFORE REFINEMENT MODEL AND THE AFTER REFINEMENT MODEL ON UNSEEN SEQUENCES. REFER TO SEC. IV-C FOR FURTHER DETAILS.

Objects group	Before ref. (mAP(%))	After ref. (mAP(%))
#0	36.0%	75.8%
#1	22.3%	72.8%
#2	42.1%	81.0%
#3	27.3%	53.7%
#4	47.8%	66.0%

Sec. IV-A, namely, BEFORE REFINEMENT MODEL and AFTER REFINEMENT MODEL, on a new exploration sequence. This latter differs from the one used for the refinement in both the objects view poses and the arrangements. This is done in order to evaluate the accuracy of the refined detection models when presented the objects under different view poses and conditions. The results are shown in Tab. III.

As it can be observed, in all cases, the *Refinement phase* performed in Sec. IV-A on one sequence, allows to improve the accuracy on the new unseen sequence as well. This demonstrates that the refinement process does not over-fit the data acquired during the *Refinement phase*, but improves the generalization capabilities of the model. Moreover, the refinement process can be iteratively repeated by performing new exploration sequences, progressively improving the detection model. Examples of detected images from these new sequences, obtained by using both the BEFORE REFINEMENT MODEL and the AFTER REFINEMENT MODEL, are reported in Fig. 3.

V. CONCLUSIONS

In this paper, we propose a unified application for efficiently training and updating an object detection on a humanoid robot. Specifically, the proposed pipeline exploits different interaction modalities based on the interaction with a human teacher and a fixed set of robot exploratory behaviors, to quickly adapt an object detection system effectively reducing human labeling effort while retaining performance. The experimental evaluation on the real robot demonstrated the effectiveness of the proposed approach.

A limitation of this work is that the robot does not interact with the objects physically. We believe that the results can be further improved with the integration of more sophisticated active exploratory actions e.g. pushing, picking up and rotating objects to acquire new, richer views. Exploratory movements could also be linked to the learning process to determine optimal exploratory actions to obtain most informative views (i.e. *best view* selection). The work presented in this paper allow the robot to iteratively adapt its vision system to novel tasks and scenarios, another direction of research would be to integrate continuous learning strategies, like e.g. [45].

REFERENCES

[1] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, "The icub humanoid robot: an open-systems platform for research in cognitive development." *Neural networks : the official journal of the International Neural Network Society*, vol. 23, no. 8-9, pp. 1125–34, 1 2010.

[2] A. Parmiggiani, L. Fiorio, A. Scalzo, A. V. Sureshbabu, M. Randazzo, M. Maggiali, U. Pattacini, H. Lehmann, V. Tikhanoff, D. Domenichelli, A. Cardellino, P. Congiu, A. Pagnin, R. Cingolani, L. Natale, and G. Metta, "The design and validation of the r1 personal humanoid," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 674–680.

[3] A. Zeng, S. Song, K. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morena, P. Q. Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1–8.

[4] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, "Synthesizing training data for object detection in indoor scenes," *CoRR*, vol. abs/1702.07836, 2017.

[5] M. Schwarz, A. Milan, A. S. Periyasamy, and S. Behnke, "Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 437–451, 2018. [Online]. Available: https://doi.org/10.1177/0278364917713117

[6] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 23–30.

[7] X. Chen, Y. Chen, B. You, J. Xie, and H. Najjaran, "Detecting 6d poses of target objects from cluttered scenes by learning to align the point cloud patches with the cad models," *IEEE Access*, vol. 8, pp. 210 640–210 650, 2020.

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

[9] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask r-cnn," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[10] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: real-time instance segmentation," *CoRR*, vol. abs/1904.02689, 2019. [Online]. Available: http://arxiv.org/abs/1904.02689

[11] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6569–6578.

[12] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*, Zrich, 2014, oral. [Online]. Available: /se3/wp-content/uploads/2014/09/coco_eccv.pdf,http://mscoco.org

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[16] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *IEEE Robotics and Automation Magazine*, 2015.

[17] A. Shrivastava, A. Gupta, and R. B. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*. IEEE Computer Society, 2016, pp. 761–769.

[18] j. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V.

Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 379–387.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[20] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[21] E. Maiettini, G. Pasquale, L. Rosasco, and L. Natale, "Speeding-up object detection training for robotics with falkon," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018.

[22] ——, "On-line object detection: a robotics challenge," *Autonomous Robots*, vol. 44, no. 5, pp. 739–757, 2020.

[23] A. Rudi, L. Carratino, and L. Rosasco, "Falkon: An optimal large scale kernel method," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3888–3898.

[24] G. Meanti, L. Carratino, L. Rosasco, and A. Rudi, "Kernel methods through the roof: handling billions of points efficiently," *arXiv preprint arXiv:2006.10350*, 2020.

[25] F. Ceola, E. Maiettini, G. Pasquale, L. Rosasco, and L. Natale, "Fast region proposal learning for object detection for robotics," *arXiv preprint arXiv:2011.12790*, 2020.

[26] ——, "Fast object segmentation learning with kernel-based methods for robotics," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[27] E. Maiettini, G. Pasquale, L. Rosasco, and L. Natale, "Interactive data collection for deep learning object detectors on humanoid robots," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, Nov 2017, pp. 862–868.

[28] G. Pasquale, T. Mar, C. Ciliberto, L. Rosasco, and L. Natale, "Enabling depth-driven visual attention on the icub humanoid robot: Instructions for use and new perspectives," *Frontiers in Robotics and AI*, vol. 3, p. 35, 2016.

[29] E. Maiettini, R. Camoriano, G. Pasquale, V. Tikhanoff, L. Rosasco, and L. Natale, "Data-efficient weakly-supervised learning for on-line object detection under domain shift in robotics," *arXiv preprint arXiv:2012.14345*, 2020.

[30] E. Maiettini, G. Pasquale, V. Tikhanoff, L. Rosasco, and L. Natale, "A weakly supervised strategy for learning object detection on a humanoid robot," in *2019 IEEE-RAS 19th International Conference on Humanoid Robotics (Humanoids)*, Nov 2019.

[31] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, 2017.

[32] J. Hernández-González, I. Inza, and J. A. Lozano, "Weak supervision and other non-standard classification problems: a taxonomy," *Pattern Recognition Letters*, vol. 69, pp. 49–55, 2016.

[33] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2012.

[34] H. H. Aghdam, A. Gonzalez-Garcia, J. v. d. Weijer, and A. M. Lopez, "Active learning for deep detection neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[35] E. Haussmann, M. Fenzi, K. Chitta, J. Ivanecky, H. Xu, D. Roy, A. Mittel, N. Koumchatzky, C. Farabet, and J. M. Alvarez, "Scalable Active Learning for Object Detection," *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1430–1435, 2020.

[36] S. V. Desai, A. L. Chandra, W. Guo, S. Ninomiya, and V. N. Balasubramanian, "An adaptive supervision framework for active learning in object detection," *arXiv preprint arXiv:1908.02454*, 2019.

[37] C.-C. Kao, T.-Y. Lee, P. Sen, and M.-Y. Liu, "Localization-aware active learning for object detection," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 506–522.

[38] Y. Li, D. Huang, D. Qin, L. Wang, and B. Gong, "Improving Object Detection with Selective Self-supervised Self-training," in *European Conference on Computer Vision*. Springer, 2020, pp. 589–607.

[39] K. Wang, X. Yan, D. Zhang, L. Zhang, and L. Lin, "Towards human-machine cooperation: Self-supervised sample mining for object detection," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 1605–1613. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Wang_Towards_Human-Machine_Cooperation_CVPR_2018_paper.html

[40] K. Wang, L. Lin, X. Yan, Z. Chen, D. Zhang, and L. Zhang, "Cost-effective object detection: Active sample mining with switchable selection criteria," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 834–850, March 2019.

[41] D. G. A. Farhadi and D. Fox, "Re 3: Real-time recurrent regression networks for visual tracking of generic objects," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 788–795, 2018.

[42] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[43] G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, and L. Natale, "Are we done with object recognition? the icub robots perspective," *Robotics and Autonomous Systems*, vol. 112, pp. 260 – 281, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889018300332

[44] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013. [Online]. Available: http://arxiv.org/abs/1311.2901

[45] T. Schmidt and D. Fox, "Self-directed lifelong learning for robot vision," in *Robotics Research*. Springer, 2020, pp. 109–114.