Modeling Robot's World with Minimal Effort

M. Villamizar, A. Garrell, A. Sanfeliu and F. Moreno-Noguer
Institut de Robòtica i Informàtica Industrial, CSIC-UPC
{mvillami,agarrell,sanfeliu,fmoreno}@iri.upc.edu

***Abstract*— We propose an efficient Human Robot Interaction approach to efficiently model the appearance of all relevant objects in robot's environment. Given an input video stream recorded while the robot is navigating, the user just needs to annotate a very small number of frames to build specific classifiers for each of the objects of interest. At the core of the method, there are several random ferns classifiers that share the same features and are updated online. The resulting methodology is fast (runs at 8 fps), versatile (it can be applied to unconstrained scenarios), scalable (real experiments show we can model up to 30 different object classes), and minimizes the amount of human intervention by leveraging the uncertainty measures associated to each classifier. We thoroughly validate the approach on synthetic data and on real sequences acquired with a mobile platform in outdoor and challenging scenarios containing a multitude of different objects. We show that the human can, with minimal effort, provide the robot with a detailed model of the objects in the scene.**

I. Introduction

Over the last decade there has been an enormous progress in the field of visual object detection and classification. Impressive and efficient results are obtained, despite the inherent challenges due to large intra-class dissimilarities and inter-class similarities and other factors as diverse as clutter, occlusion or illumination changes [9], [25]. Robotics is one of the fields that has benefited most of this current progress in visual object detection, with a diversity of applications such as people detection and tracking [5], [17], object recognition and grasping [2], [8], robot navigation and localization [6], [10], [14], [27], among others.

However, there are some particular robotics applications, specially those related to Human Robot Interaction (HRI), where the aforementioned works are not suitable. This is because these existing methods usually train their classifiers offline, using a huge number of annotated data and taking a potentially large amount of time. In contrast, in HRI the interaction with the robot needs to be fast and dynamic, and thus it is fundamental to develop classifiers which can be trained and adapted on the fly, with just very little training data, and as efficiently as possible. In [11], [26] we already proposed online object detectors which met some of these properties, and could be easily trained and adapted to model one single object.

In this paper, we go a step further, and extend previous approaches to multiple classes, i.e., we propose a methodology to model several object appearances on the fly, using the minimal amount of manually annotated data as possible, and still keeping the real time efficiency. The core of our algorithm is based on a randomized tree classifier [7], [19], [20], but which is progressively refined with user annotated data. Despite building a different classifier per object, the whole system still remains very efficient, as features –i.e., ferns– are shared among classifiers, and the only difference between classifier is their particular spatial fern distribution. Additionally, we also gradually minimize the amount or human intervention, while avoiding drifting problems, by proposing an uncertainty-based active learning strategy [18], [23], which for this paper makes it adaptive. Note that this issue is critical in order to maintain long-term interactions with robots, as if the robot keeps asking for annotating images insistently, people tend to quickly give up the interaction [11], [21].

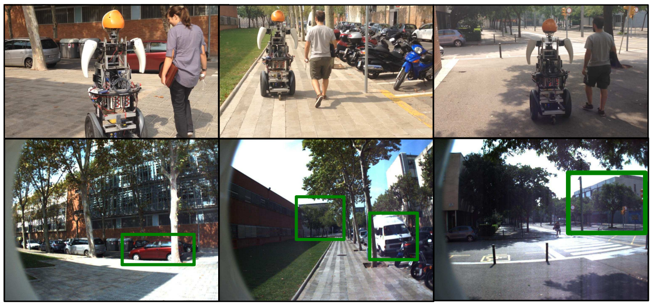

Fig. 1. **Interactive and real-time approach for learning and detecting multiple objects through human-robot interaction.**

Fig. 1 shows an example of how this approach runs in practice when showing the robot the contextual objects that need to be learned. Each time the human user seeks to model a new object of interest, he/she marks a bounding box around the object in the input image, via a mouse, keyboard or touchscreen. The robot initializes a model for this new object and runs a detector on subsequent frames for this, and the rest of objects in the database. When the robot is not confident enough about the detections and class predictions, it requests the human assistance to provide the true class labels, which, in turn, are used to update the classifier.

The remainder of the paper is organized as follows: Sec. II describes the related work and puts in context our contributions. In Sec. III the proposed approach is explained with all its main ingredients. Sec. IV-B describes the experiments conducted to evaluate the proposed learning approach. We report results using both synthetic an real data. The former are used to thoroughly assess the limits of the method in terms of number of classes it can handle or classification rate. Real experiments demonstrate that up to 30 object classes can be efficiently learned in challenging scenarios.

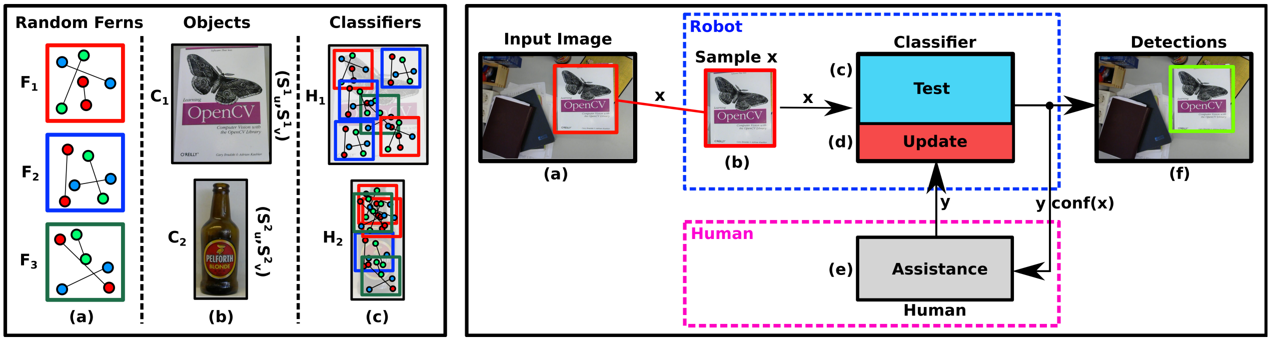

Fig. 2. **General schemes of the proposed interactive learning approach for online object recognition. Left: Two different object classifiers are computed using a shared pool of random ferns, being each one a specific set of pixel-intensity comparisons. Right: online learning using human-robot interactions. The human assists the robot when it is not certain about its sample class prediction.**

II. Related Work and Contributions

We next discuss the related work on the two main topics we address in this paper, the design of online classifiers, and interactive techniques for robot learning:

A. Online Classifiers

Despite showing impressive results, standard methods for object detection and image classification compute the classifiers using intensive and offline learning approaches applied to large datasets [9], [19], [25]. Therefore, most of these offline approaches are not suitable for some particular applications requiring computing the classifier on the fly, either because the training data is obtained continuously, or because the size of the training data is so large that it needs to be loaded progressively. To handle these situations, several online alternatives allowing to sequentially train the classifiers have been proposed [3], [4], [12], [13], [22].

In this paper, the classifier we use is based on an online random ferns formulation [11], [16], [19], [26], which has been showing excellent results, both in terms of classification rates as computational efficiency. In essence, this classifier computes several sets of intensity-based pixel comparisons (Fig. 2 (left-a)) to build the randomized trees which are then used to estimate the posterior class probabilities.

Most previous online versions are focused to single object modeling and tracking [3], [4], [12], [27]. In order to learn multiple models, [26], [11] simply train different classifiers in an independent manner. In contrast, we propose computing simultaneously and in parallel multiple classifiers, one for each object class, and with specific configurations, like the spatial distribution of ferns or the particular object size (Fig. 2 (left-b,c)). This also differs from other state of the art classifiers, that when applied to multiclass problems they require objects with constant aspect ratios of the object [24].

B. Interactive Learning

Active learning techniques have been extensively used in computer vision to reduce the number of training samples that need to be annotated when building a classifier [23]. Approaches such as “query by committee” [1], [15], and “uncertainty-based sampling” [18] close the learning loop using human assistance. In these works, the human user acts as an oracle that annotates/labels those samples that the classifier is not quite confident about their class prediction.

In this work, we propose an interactive learning strategy in which the robot plays a more active role, that is, the discriminative classifiers are built using a combination of the robot predictions with the human assistance (see Fig. 2 (right)). Additionally, we also propose a methodology based on an adaptive uncertainty threshold that progressively reduces the amount of human assistance, making a more ”enjoyable” human-robot interaction. This is also another difference with respect to our own previous works [11], [26]. As it will be shown in the experimental section, using an adaptive threshold we can scale better to several object instances without decrementing the intra-class classification rates.

After having discussed the related work we can summarize the main *contributions* of our approach as follows: (1) Proposing an online approach to learn and detect multiple object instances in images; (2) Designing an interactive learning strategy that progressively improves the discrimination power of the classifiers using human assistance; (3) An adaptive learning scheme to reduce gradually the human interventions; and (4) A real time implementation of the algorithm, which can cope with up to 30 objects at several frames per second.

III. Interactive Learning and Recognition

We next describe each of the main ingredients of our proposed learning strategy. An schematic of how these elements are related is shown in Fig. 2.

A. The Online Classifier

We compute object classifiers using an particular version of the extremely randomized trees [7], [19], [20], which are the so-called online random ferns [11], [19], [26]. We build one such classifier per object from scratch, in a way that the fern features are shared among all classifiers. By doing this, the computation of the ferns features, which is the most computations costly part of the algorithm, is shared by all classifiers. This provides a remarkable speed up compared to when we train each classifier with a different subset of ferns, while classification rates are shown to remain high. Again in Fig. 2 (left), we show two different classifiers, one per each object. Note in Fig. 2 (left-c) that every classifier has the same type of features (ferns), but with a particular spatial distribution.

Let us now describe in detail how posterior class probabilities are computed. Consider a classifier made of J random ferns, in which each Fern F_j is just a set of M binary and random features, $F_j = \{f_1^j, f_2^j, \ldots, f_M^j\}$, representing binary comparisons between pairs of pixel intensities in the image I. Each binary feature can be written as:

$$f(\mathbf{x}) = \mathbb{I}(\mathbf{x}(u_1, v_1, c_1) > \mathbf{x}(u_2, v_2, c_2)) \tag{1}$$

where $\mathbb{I}(e)$ is the indicator function, $\mathbf{x}$ is an image sample, and $\mathbf{x}(u, v, c)$ indicates the intensity-pixel value at coordinates (u, v) with color channel c. These pixel coordinates are defined at random during the learning phase. Fig. 2 (left-a) shows as example three random ferns, each one with three binary features (i.e. colored paired dots). The co-occurrence of these binary features determines the Fern output, $F(\mathbf{x}) = z$, where $z = (f_1, \ldots, f_M)_2 + 1$.

As mentioned above, classifiers for different objects are computed using shared features. To this end, a small set Θ of R random ferns (a typical value is $R = 10$ ferns) is computed in advance, such that each object classifier can then be computed as a combination of these ferns evaluated at different image locations, Fig. 2 (left-a,c). Since in practice every fern is densely computed at all image locations using a fast convolution, the sharing strategy makes the overall computational cost to be just a function of the number of ferns, and to be independent on the number of classifiers.

The classifier $H_k(\mathbf{x})$ for an object instance k is then built by random sampling with replacement among the J ferns of the shared set Θ, $\mathbf{F_k} = \{F_j^{r,\mathbf{p}}\}_{j=1}^J$, being $r \in \{1, \ldots, R\}$ and $\mathbf{p} \in \mathbb{R}^2$ the image location where the fern j is tested. The response of this classifier $H_k(\mathbf{x})$ over the sample $\mathbf{x}$ is:

$$H_k(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathrm{conf}_k(\mathbf{x}) > \beta \\ -1 & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathrm{conf}_k(x)$ is the confidence of the classifier on predicting that $\mathbf{x}$ belongs to the object k, and β is a confidence threshold whose default value is 0.5. Thus, if the output of the classifier is $H(\mathbf{x}) = +1$, the sample $\mathbf{x}$ is considered as an object or positive sample. Otherwise, this sample is assigned to the background or negative class.

The confidence of the classifier is defined according to the following posterior:

$$\mathrm{conf}_k(\mathbf{x}) = p(y = +1|\mathbf{F_k}(\mathbf{x}), \boldsymbol{\eta}_k), \tag{3}$$

where $\boldsymbol{\eta}_k$ are parameters of the classifier, and $y = \{+1, -1\}$ makes reference to the class label. In turn, this posterior probability is computed by combining the posterior of the J ferns:

$$p(y = +1|\mathbf{F_k}(\mathbf{x}), \boldsymbol{\eta}_k) = \frac{1}{J}\sum_{j=1}^{J} p(y = +1|F_j^{r,\mathbf{p}}(\mathbf{x}) = z, \eta_k^{j,z}), \tag{4}$$

where z is the fern output and $\eta_k^{j,z}$ is the probability that the sample $\mathbf{x}$ belongs to the positive class in the k-th classifier, and output z of fern j. Since the posterior probabilities follow a Bernoulli distribution, $p\left(y|F_j^{r,\mathbf{p}}(\mathbf{x}) = z, \eta_k^{j,z}\right) \sim \mathrm{Ber}(y|\eta_k^{j,z})$, we can write that

$$p\left(y = +1|F_j^{r,\mathbf{p}}(\mathbf{x}) = z, \eta_k^{j,z}\right) = \eta_k^{j,z}. \tag{5}$$

The parameters of these distributions are computed through a Maximum Likelihood Estimate (MLE) over the input samples and their corresponding labels, provided by the human user during the interaction with the robot. That is,

$$\eta_k^{j,z} = \frac{N_{k,+1}^{j,z}}{N_{k,+1}^{j,z} + N_{k,-1}^{j,z}} \tag{6}$$

where $N_{k,+1}^{j,z}$ is the number of positive -object- with output z for fern j . Similarly, $N_{k,-1}^{j,z}$ corresponds to the number of negative samples for the fern j with output z.

B. Interactive Learning

Fig. 2 (right) shows the online learning strategy to train a specific classifier k. Given an input image I, the classifier is tested at every image location and multiple scales using a sliding window approach [28]. At each location (u, v), the image sample $\mathbf{x}$ (local image region defined by the object size (s_u^k, s_v^k)) is evaluated on all J ferns of the classifier to obtain the confidence $\mathrm{conf}_k(\mathbf{x})$ (Eq. 3). Subsequently, the class label for this sample, $y = \{+1, -1\}$, is estimated according to the response of the classifier and the threshold β (refer to Eq. 2).

In order to reduce the number of false positives and avoid drifting problems (produced when updating the classifier with erroneously labeled samples), frequent in non and semi-supervised learning approaches [3], [12], we use an uncertainty-based active learning strategy [18], [23] that in combination with an adaptive uncertainty threshold reduces gradually the amount of human assistance. Active learning minimizes the risk of misclassification by updating the classifier only with samples which have been annotated/labeled by the user.

Therefore, in situations where the classifier is not certain about the class estimate y, because the confidence over the sample $\mathbf{x}$ is ambiguous (near to the threshold β), the system opts for requiring the human help so as annotate the true class of the sample. This request q can be written as:

$$q(\mathbf{x}) = \mathbb{I}(\beta + \theta/2 > \mathrm{conf}_k(\mathbf{x}) > \beta - \theta/2), \tag{7}$$

where θ corresponds to the uncertainty threshold. If $q(\mathbf{x})$ is true the system asks for human assistance. Otherwise, this sample is discarded and not used to update the classifier. Note that by doing this we are just feeding the classifier with labeled samples that are close to the decision boundary, improving thus, its discriminability power.

With the aim of adapting the human assistance in accordance to the performance of the classifier, we define an adaptive threshold that depends on the incremental classification rate over the requested samples. That is,

$$\theta = 1 - \xi\lambda_k, \tag{8}$$

where ξ is a sensitivity parameter assigned by the user, and λ_k measures the performance of the classsifier k. In turn, this performance rate can be computed by

$$\lambda_k = M_k^c / M_k^q \tag{9}$$

being M_k^q and M_k^c the numbers of requested samples and correctly classified samples, respectively. A sample $\mathbf{x}$ is

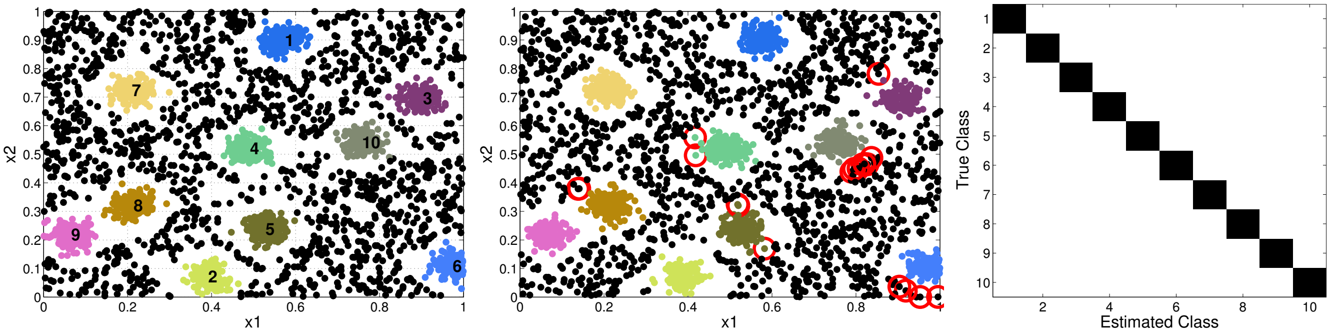

Fig. 3. **Performance of the proposed approach for the 2D classification problem. Left: 1500 positive and negative training samples used to compute the online classifiers. Center: Test samples used to evaluate generalization. Right: Confusion matrix showing the intra-class classification.**

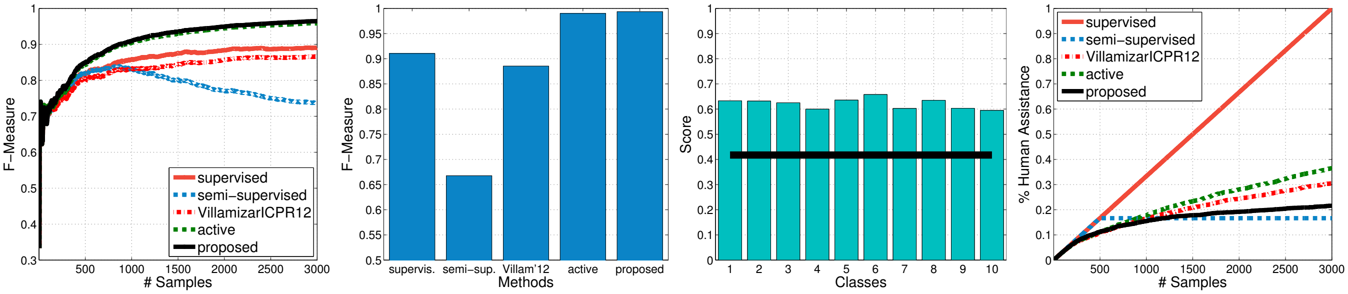

Fig. 4. **Classification results for the 2D problem. Left: Incremental classification rates for different learning approaches. Center-Left: Classification rates on the test samples. Center-Right: Average classification scores across classes. Right: Degree of human assistance.**

correctly classified when the class label y coming from the classifier agrees with the true class label given by the user.

Once the samples have been labeled, they are used to recompute the probabilities $\eta_k^{j,z}$ of Eq. 6, and update the classifier. For instance, let us assume that a sample $\mathbf{x}$ is labeled as $+1$, and that it activates the output z of the fern $F_j^{r,\mathbf{P}}$, i.e, $F_j^{r,\mathbf{P}}(\mathbf{x}) = z$. We will then update the classifier by adding one unit to the i-th bin of the histogram of $N_{k,+1}^{j,z}$. This is repeated for all ferns. With these new distributions, we can recompute the priors $\eta_k^{j,z}$ and update the classifier.

IV. Experimental Results

We now present results of the proposed approach in synthetic and real data. Synthetic experiments will be used to accurately assess the limits of the learning approach in terms like the potential number of classes it can handle or the amount of human assistance required by ours and alternative learning strategies. Real experiments will demonstrate the applicability of the system when used in our robotic platform.

A. Synthetic Data - 2D Classification Problem

We initially analyze the performance of the proposed online method on a synthetic 2D classification problem, that will reveal the influence of certain parameters or different learning strategies on the classification results and on the number of samples that need to be manually annotated.

Fig. 3 (left) shows an example of a 2D classification problem where 10 positive classes (colored points) and one negative class (black points spread out over the feature space) are randomly and sequentially fed to the online learning system in order to compute the classifiers. In this particular scenario, the classifiers are built using individual 2D decision stumps as binary features (Eq. 1). That is, $f(\mathbf{x}) = \mathbb{I}(\mathbf{x}_i > \phi)$, where i and $\phi \in (0, 1)$ are the feature axis and threshold defining the space partition.

The classification performance of the proposed approach is shown in Fig. 3 (center) where the classifiers computed using the training samples (left side) are evaluated on a set of test samples in order to measure the generalization capability. We see that the most samples are correctly classified and only a small fraction of them are misclassified, indicated in the figure through the red circles. Quantitatively speaking, the method obtains a F-measure rate of 0.994 to distinguish positive samples from negative ones (two-class separability). Fig. 3 (right) shows the confusion matrix in specifically recognizing each of the 10 positive classes (using ground-truth sample labels). We see that the method achieves high classification rates both to separate the positive and negative classes and to correctly classify the positive subclasses.

Fig. 4 (left) shows the incremental classification performance of the method as the training samples are given sequentially to the online classifiers. Here, we evaluate five different learning alternatives:

Supervised: The classifiers are trained with all samples. Each sample is labeled by the human user (human label).

Semi-supervised: The first n samples are labeled by the human (human label), whereas the rest ones are labeled using the classifier confidence (machine label).

Active: The classifiers are only trained/updated in cases of high uncertainty in the predictions. The human resolves the ambiguity by providing the sample label.

Villamizar ICPR2012 [26]: This approach combines active and semi-supervised learning. Active learning for uncertain samples and self-learning for certain samples.

Proposed: The classifier uses active learning in combination with the adaptive uncertainty threshold.

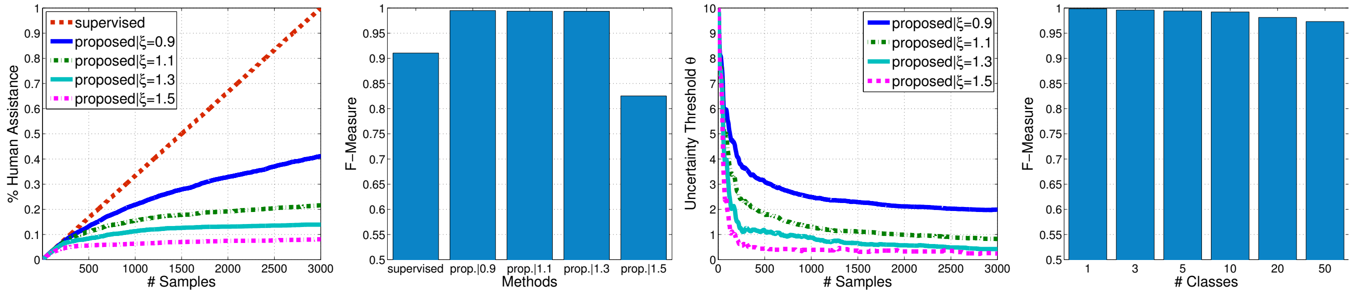

Fig. 5. **Classification results for different degrees of human intervention. Left: Percentages of human labels. Center-Left: Classification rates on the test samples. Center-Right: Adaptive uncertainty threholds. Right: Classification rates in terms of the number of classes.**

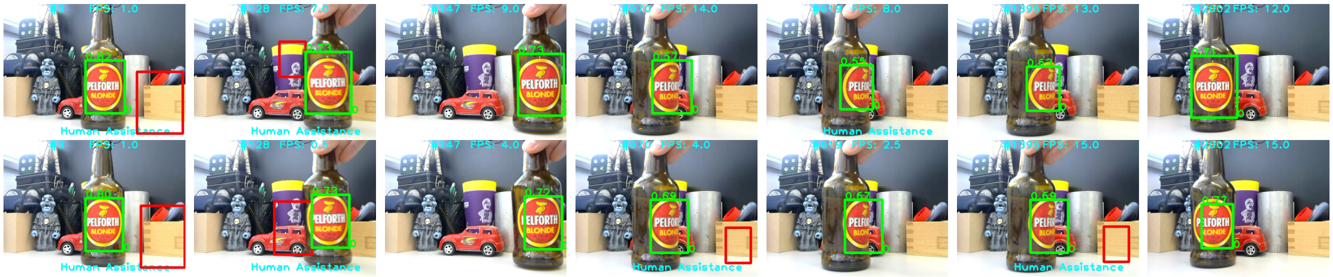

Fig. 6. **Detection results for a single object. Top Row: Detection outputs of the proposed method. Bottom Row: Outputs of the approach proposed in [26]. Green rectangles indicate object hypotheses whereas red ones are background hypotheses labeled by the user during the assistance.**

Observe that all learning approaches start with low classification scores, but then they begin to improve progressively as more samples are provided to the classifiers. However, note that at some point, the semi-supervised learning deteriorates the classifier performance. This is because the self-learning suffers from drifting problems, making the classifier to be constantly updated with erroneously labeled samples. By contrast, our proposed method and the active learning obtain the best performance since the classifiers are computed with highly informative samples (uncertain samples) and human labels. This focuses the classifier mainly on the decision boundaries and makes it more discriminative that using all training samples (supervised method).

Similarly, Fig. 4 (center-left) shows the classification rates (F-measure) on the set of test samples. We see again that the proposed method, together with the active learning, achieves the best classification performance and generalization capability. The average classification scores across the different classes are shown in Fig. 4 (center-right). All scores are above the classification threshold β (Eq. 2). The black thick line corresponds to the average score for the negative class.

As regards to the amount of human intervention, Fig. 4 (right) displays the percentages of human labels as a function of the number of incoming samples. The semi-supervised learning just uses labels for the first 500 samples. Note also that the supervised learning uses all human labels (represented by a diagonal line) to compute the classifiers, whereas our method reduces considerably the number of human assistance. The classifiers are computed by only using 22% of the training samples. By contrast, the active learning continues requiring human assistance until reaching about 38% of the samples. This shows that the human intervention is progressively reduced thanks to the adaptive uncertainty threshold without deteriorating the classification rates.

This behavior is observed in Fig. 5 where the proposed method is evaluated in terms of the adaptive uncertainty threshold. Fig. 5 (left) shows the human assistance percentages for four different values of the sensitivity parameter ξ. We observe how the number of required human annotations decreases as the sensitivity parameter gets larger until obtaining less than 10% of the training samples. However, this at the expense of an important reduction in the classification rates, see Fig. 5 (center-left). In Fig. 5 (center-right) we can see the adaptive threshold values through the incoming samples. As a general trend, the threshold decreases rapidly as the classifiers get more confident in their predictions.

Finally, Fig. 5 (right) shows the classification rates on the test samples for varying numbers of classes. Note that increasing the number of classes in the learning phase, produces a small drop in the classification rates. This is because we are consideing a large number of classes for such a small feature space (2D).

B. Experiments with Real Data

In this section, the proposed method is evaluated over three different object recognition scenarios. The first experiment consists in learning and detecting one single object in a cluttered scene; the second one is focused on face detection and identification for two persons; and finally, the method is tested for learning and recognizing multiple objects (up to 30) while the robot navigates through an urban scenario.

Fig. 6 shows some examples of the first experiment. Particularly, this experiment has over 2000 images containing the object (beer bottle) at diverse locations and viewpoints. The user initializes the classifier with the first frame. The top row of Fig. 6 corresponds to the output of the proposed method (for $\xi = 1.1$). This is compared in the bottom row with the detection results obtained by our previous

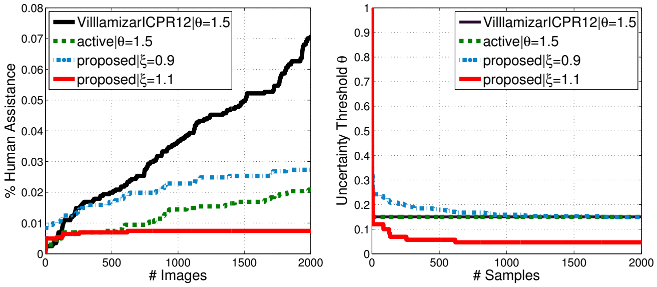

Fig. 7. **Performance comparison of different learning approaches in terms of the amount of human assistance (left) and the uncertainty threshold (right).**

approach [26]. Results show that both methods are able to learn and detect the object through the whole sequence. This is indicated via the green rectangles around the object. Red rectangles are detection hypotheses that the user has manually labeled as incorrect during the interaction. Yet, despite the good performance of both methods, the proposed approach has the benefit that the amount of human assistance is significantly reduced. This is shown in Fig. 7 (left) where the percentages of human assistance are displayed together with the active learning approach. Our method obtains the lowest rate of human assistance while correctly detecting the object in all frames.

Additionally, Fig. 7 (right) shows the uncertainty thresholds for the aforementioned learning approaches. Observe that our method gradually reduces the uncertainty threshold during the learning phase whereas the active learning and [26] keep a constant value ($\theta = 0.15$).

Fig. 8 shows the results when the proposed approach is evaluated for face detection and recognition. Like in the previous experiment, the classifiers are interactively trained using human assistance. In this case, the method learns and detects two people simultaneously while they interact with the robot. This contrasts to [26] where the classifiers are independently computed and one at a time. The displayed sequence snapshots show that our method can effectively learn multiple faces and detect them in the subsequent frames. Furthermore, the method runs in real-time, except for the assistance periods, and the retrieved identity of people is correct. This issue is shown by the small images beside the detection boxes.

Finally, the method is evaluated for learning and detecting multiple objects in urban scenarios. Fig. 9 depicts some sample images showing the performance of the classifiers and the ability of the proposed method to learn several objects in real-time and interactively when the robot navigates within the environment. We can see that objects like cars, doors and buildings are easily learned and recognized (green boxes) by the system. It is important to emphasize that most red rectangles, wrong hypotheses, are because the classifier has been recently initialized and it yields a small number of false positives. Nevertheless, these false predictions are shortly removed by updating the classifiers with human-labeled samples.

With regards to the computational time, Fig. 10 plots the running times (in frames per second) of the proposed method as a function of the number of object classifiers. Note that the computational cost increases as the number of objects gets larger. However, learning and detecting 20 objects at four frames per second is a remarkable and promising result, especially for current robotic tasks involving online learning and real-time performance.

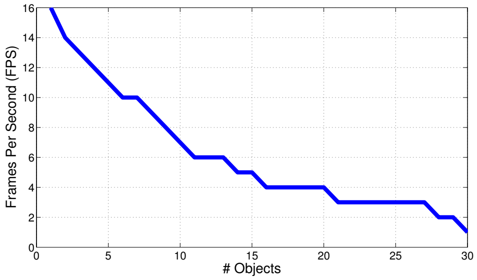

Fig. 10. **Computational efficiency of the proposed approach according to the number of objects.**

V. Conclusions

In this work, we have presented a novel approach for interactively learning the appearance model of multiple objects in real-time. The proposed method uses efficient and reliable random trees classifiers to compute object detectors on the fly and which are progressively refined with the human assistance. The proposed method also includes an uncertainty-based active learning strategy that reduces the amount of human intervention while it maintains high recognition rates. The method has been evaluated extensively in different scenarios such as 2D classification, face recognition, and the learning and detection of contextual objects in urban settings using an autonomous mobile robot.

Acknowledgments. Work partially supported by the Spanish Ministry of Science and Innovation under project DPI2013-42458-P, ERA-Net Chistera project ViSen PCIN-2013-047, and by the EU project ARCAS FP7-ICT-2011-28761.

References

[1] N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *ICML*, 1998.

[2] G. Alenyà, S. Foix, and C. Torras. Using tof and rgbd cameras for 3d robot perception and manipulation in human environments. *Intelligent Service Robotics*, pages 1–10, 2014.

[3] S. Avidan. Ensemble tracking. *PAMI*, 29(2):261–271, 2007.

[4] B. Babenko, M.H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 33(8):1619–1632, 2011.

[5] N. Bellotto and H. Hu. Multisensor-based human detection and tracking for mobile service robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1):167–181, 2009.

[6] A. Corominas, J.M. Mirats-Tur, and A. Sanfeliu. Efficient active global localization for mobile robots operating in large and cooperative environments. In *ICRA*, pages 2758–2763, 2008.

[7] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, (7):81–227, 2011.

[8] A. Edsinger and C.C. Kemp. Human-robot interaction for cooperative manipulation: Handing objects to one another. In *RO-MAN*, 2007.

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[10] G. Ferrer, A. Garrell, and A. Sanfeliu. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In *IROS*, pages 1688–1694, 2013.

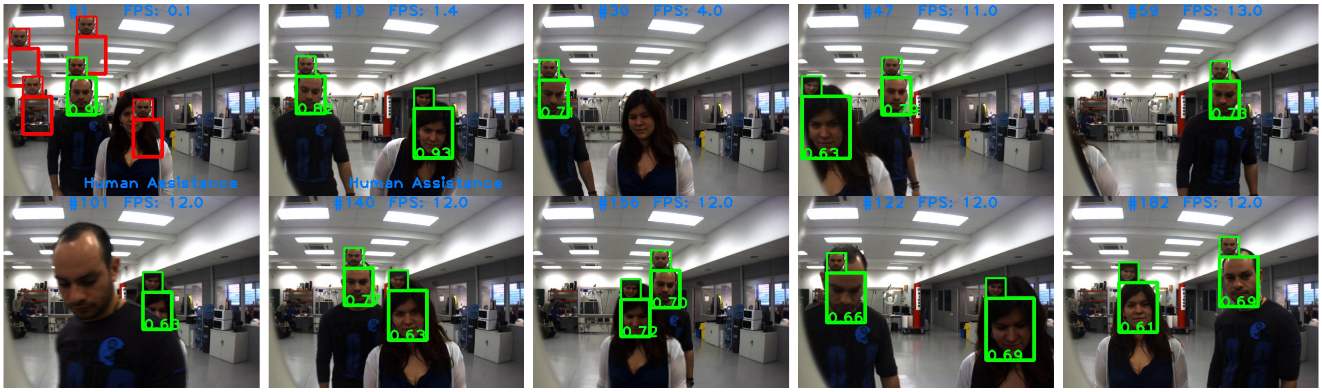

Fig. 8. **Online face recognition. The presented method performs online learning and detection of faces via human-robot interaction.**

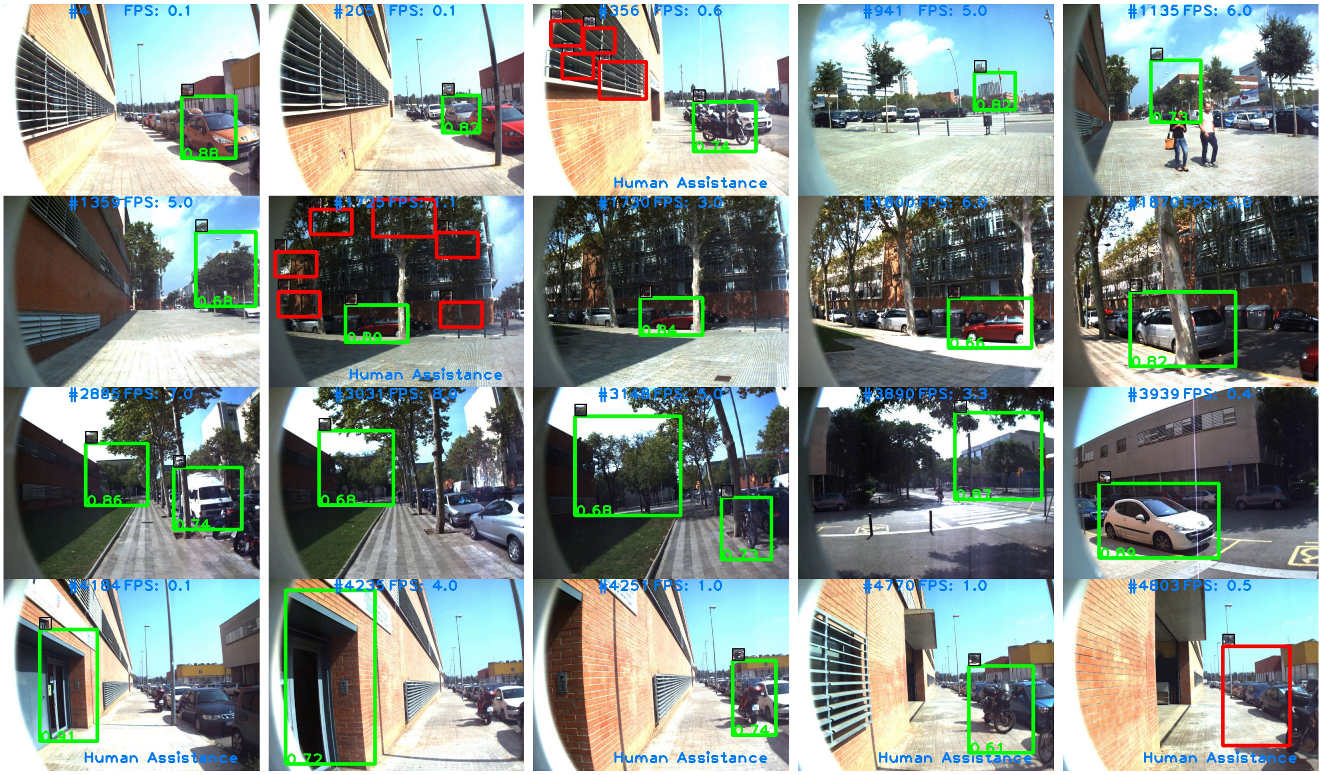

Fig. 9. **Learning and detection results of the proposed method in urban scenarios. The method is capable of computing efficiently multiple classifiers, each one specialized in an object. The classifiers are improved progressively using the human assistance.**

[11] A. Garrell, M. Villamizar, F. Moreno-Noguer, and A. Sanfeliu. Proactive behavior of an autonomous mobile robot for human-assisted learning. In *RO-MAN*, pages 107–113, 2013.

[12] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, 2006.

[13] D. Hall and P. Perona. Online, real-time tracking using a category-to-individual detector. In *ECCV*, 2014.

[14] A. Hornung, K. Wurm, and M. Bennewitz. Humanoid robot localization in complex indoor environments. In *IROS*, 2010.

[15] M. Opper H.S. Seung and H. Sompolinsky. Query by committee. In *Proceedings of the ACM Workshop on Computational Learning Theory*, 1992.

[16] Z. Kalal, J. Matas, and K. Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, 2010.

[17] M. Kobilarov, G. Sukhatme, J. Hyams, and P. Batavia. People tracking and following with mobile robot using an omnidirectional camera and a laser. In *ICRA*, pages 557–562, 2006.

[18] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[19] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *PAMI*, 2010.

[20] D. Ernst P. Geurts and L. Wehenkel. Extremely randomized trees. *Machine learning, 63.1*, pages 3–42, 2006.

[21] P. Rani, C. Liu, N. Sarkar, and E. Vanman. An empirical study of machine learning techniques for affect recognition in human–robot interaction. *Pattern Analysis and Applications*, 9(1):58–69, 2006.

[22] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. Prost: Parallel robust online simple tracking. In *CVPR*, 2010.

[23] B. Settles. Active learning literature survey. *University of Wisconsin, Madison 52*, pages 55–66, 2010.

[24] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 19(5):854–869, 2007.

[25] M. Villamizar, J. Andrade-Cetto, A. Sanfeliu, and F. Moreno-Noguer. Bootstrapping boosted random ferns for discriminative and efficient object classification. *Pattern Recognition*, 45(9):3141–3153, 2012.

[26] M. Villamizar, A. Garrell, A. Sanfeliu, and F. Moreno-Noguer. Online human-assisted learning using random ferns. In *ICPR*, 2012.

[27] M. Villamizar, A. Sanfeliu, and F. Moreno-Noguer. Fast online learning and detection of natural landmarks for autonomous aerial robots. In *ICRA*, 2014.

[28] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.