Robust Object Recognition Through Symbiotic Deep Learning In Mobile Robots*

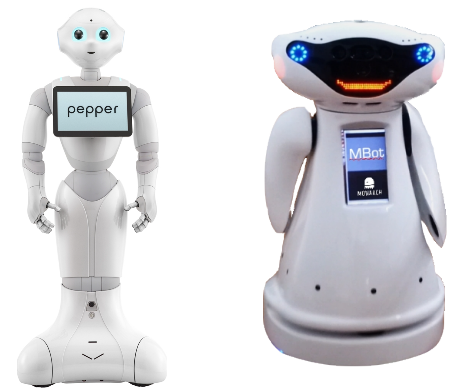
João Cartucho[1], Rodrigo Ventura[2] and Manuela Veloso[3]

***Abstract*— Despite the recent success of state-of-the-art deep learning algorithms in object recognition, when these are deployed as-is on a mobile service robot, we observed that they failed to recognize many objects in real human environments. In this paper, we introduce a learning algorithm in which robots address this flaw by asking humans for help, also known as symbiotic autonomy approach. In particular, we bootstrap YOLOv2, a state-of-the-art deep neural network and create a HUMAN neural net using only the collected data. Using an RGB camera and an on-board tablet, the robot proactively seeks for human input to assist in labeling surrounding objects. Pepper, based in CMU, and Monarch Mbot, based in ISR-Lisbon, are the social robots that we used to validate the proposed approach. We conducted a study in a realistic domestic environment over the course of 20 days with 6 research participants. To improve object recognition, we used the two neural nets, YOLOv2 + HUMAN, in parallel. The robot collects data about where an object is and to whom it belongs by asking. This enabled us to introduce an approach where the robot can search for a specific person's object. We view the contribution of this paper to be relevant for service robots in general, in addition to Pepper and Mbot. Following this methodology, the robot was able to detect twice the number of objects compared to the initial YOLOv2, with an improved average precision.**

I. INTRODUCTION

Human-robot symbiotic learning is an increasingly active area of research. [1], [2], [3], [4]. Anthropomorphic robots are being increasingly deployed in real-world scenarios, such as homes, offices, and hospitals [5], [6], [7]. However, exposure to real environments raises multiple challenges often overlooked in controlled laboratory experiments. For instance, robots equipped with state-of-the-art neural nets trained for object recognition still fail to provide accurate descriptions for the majority of the objects surrounding them outside controlled environments.

This paper tackles the aforementioned problem using a symbiotic interaction approach, in which the robot seeks human assistance in order to improve its object recognition skills. Our primary aim is to improve the accuracy of robot object detection and recognition. In addition, we also tackle the challenge of looking for a specific person's object - a crucial ability for assisting the disabled and elderly.



(a) Pepper the robot (b) Mbot the robot

Fig. 1: Mobile robots used for the evaluation of the proposed method. Photos: SoftBank/Aldebaran and IDMind Robotics

This is achieved by deploying a learning algorithm that empowers the robot to ask humans for help. Over time, it can be measured that the human input increases the robot's effectiveness. The learning process is bootstrapped by an external state-of-the-art neural net — YOLOv2 — for real-time object detection [8].

The robot, using its RGB camera, in conjunction with its on-board tablet, explores its environment whilst labeling objects. The robot then confirms its object recognition by interacting with a human, asking them to respond to simple Yes/No questions and/or by requesting that the human adjusts a rectangle positioned around an object in the tablet.

The other key functionality that was explored was the ability for the robot to determine which objects belong to which humans. Human input is crucial for this task. Providing accurate information will equip the robot with means to actively seek a personal object, on request. This task was made possible due to our realization of the wide applications that can be derived from the data that the robot collected.

The social robots used to test our approach were Pepper (shown in Figure 1a), a service robot developed by Softbank/Aldebaran Robotics and specifically designed for social interaction [9], and Monarch Mbot (shown in Figure 1b). In addition, the two robotic platforms were located in separate working environments: Pepper was based in CMU, USA, and Mbot in ISRobotNet@Home[1], Test Bed of ISR-Lisbon,

*This work was supported by the FCT project [UID/EEA/50009/2013] and Partially funded with grant 6204/BMOB/17, from CMU Portugal.

[1]João Cartucho is with the Institute for Systems and Robotics, Instituto Superior Técnico, 1049-001 Lisboa, Portugal. Visiting scholar at Carnegie Mellon University, Pittsburgh, PA 15213, USA to.cartucho@gmail.com

[2]Rodrigo Ventura with the Institute for Systems and Robotics, Instituto Superior Técnico, 1049-001 Lisboa, Portugal rodrigo.ventura@isr.tecnico.ulisboa.pt

[3]Manuela Veloso with the Department of Machine Learning, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA mmv@cs.cmu.edu

[1]More info can be found at http://isrobonet_athome.isr.tecnico.ulisboa.pt/

Preprint submitted to 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2018.

Portugal. The experiments were conducted in a realistic domestic scenario. By having many research participants interact and modify the test environment, the unpredictability of object placement and arrangement was ensured.

At the end of our experiment, the robot was able to detect twice the number of objects compared to the initial YOLOv2, with an improved average precision.

This paper is structured as follows, in section: II, we review the state-of-the-art; III, we introduce the learning algorithm; IV, the YOLOv2 + HUMAN; V, we describe the experimental setup; VI, we analyze the results; and finally in VII, the conclusions.

II. RELATED WORK

The complexity of indoor environments grows exponentially due to a various number of factors, increasing the difficulty for robots to complete tasks successfully. The particular task of learning and recognizing useful representations of places (such as a multi-floor building) and manipulating objects has been a subject of active research namely, in a symbiotic interaction with humans [10].

In this paper, we aim to recognize objects coupled with their bounding-box. A few works have explored a human-based active learning method specifically for training object class detectors. Some of them focus on human verification of bounding-boxes [11] or rely on a large amount of data corrected by annotators [12]. Others explore how people teach or are influenced by the robot [13].

It is worth mentioning that symbiotic autonomy has been actively pursued in the past [14], [15]. Some interesting approaches are focused on improving the robot's perception with remote human assistance [16]. Research groups have also worked on developing autonomous service robots such as the CoBots [2], the PR2 [17] and many others.

Considering spatial information analysis and mapping, studies using ultrasonic imaging with neural networks [18], 3D convolutional neural networks with RGB-D [19] and a novel combination of the RANSAC and Mean Shift algorithms [20] have been in development for several decades, which demonstrates a clear evolutionary trajectory that enable the developments of the present.

Other works using scene understanding and image recognition show a strong affinity with this paper. Works such as: using saliency mapping plus neural nets to tackle scene understanding [21]; full pose estimation of relevant objects relying on algorithmic processing and comparison of a dataset of images [20]; feature-matching technique implemented by a Hopfield neural network [22]; and data augmentation [23].

When it comes to showing an object located in the real world to the robot, previous work has investigated alternative ways of intuitively and unambiguously selecting objects, using a green laser pointer [24]. In our approach, we took advantage of the robot's tablet purposely inbuilt for interacting with humans.

Finally, there has been work done in the area of getting robots to navigate in a realistic setting and recognizing objects in order to place them on a map [25]. In our case, using input from human interaction allows the robot to generate this information. This enables the robot to store where each object was seen and how many times it was seen at each location.

Fig. 2: Mbot (top) and Pepper (bottom) learning

III. LEARNING ALGORITHM

The algorithm consists of three parts: First (A), the robot captures images to learn and predict what and where the objects are located in the images. Then (B), the robot asks the humans questions while confirming its previous predictions and collecting additional information. Finally (C), the robot trains using all the collected knowledge. This allows the robot to improve its future predictions.

A. Predicting the objects

The robot starts by navigating to a location it hasn't visited before and captures different images. These are the images that the robot will learn about for that day.

Before requesting help from a human, the robot predicts what the objects are and where they are located in each of the images.

To make these predictions the robot uses YOLOv2 [8], a neural net trained in COCO dataset[26], able to detect up to 80 classes of objects[2]. From "person" to "bed" or "bicycle" this neural net includes generic classes that apply to a vast number of different scenarios.

B. Interacting with Humans

When interacting with humans the robot will be asking questions: (1) Labeling the objects and (2) Identifying to whom an object belongs.

1) Labeling the objects: For labeling the objects the robot will first confirm the previous predictions and then ask the human if all the objects in the image are labeled.

To confirm the predictions the robot asks two Yes/No questions. Given a prediction, for example, the object "cat", the robot asks:

[2]The 80 COCO class labels can be found at http://cocodataset.org

Preprint submitted to 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2018.

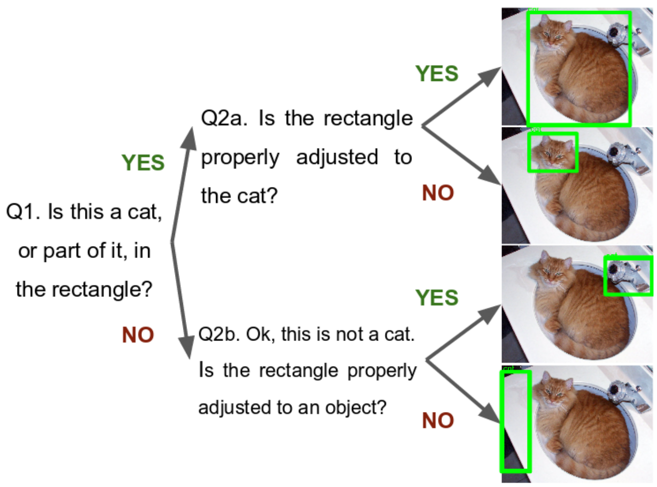


Fig. 3: E.g. Outcomes from Q1 and Q2.

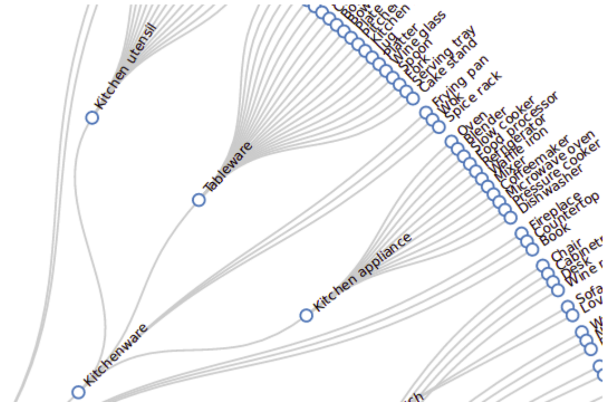


Fig. 4: Some classes that can be added to the robot using the labels from OpenImages dataset.[27]

- Q1: Is this a cat, or part of it, in the rectangle?
- (if Yes to Q1) - Q2a: Is the rectangle properly adjusted to the cat?
- (if No to Q1) - Q2b: Ok, this is not a cat. Is the rectangle properly adjusted to an object?

The robot provides instructions to the human explaining that the properly adjusted rectangle refers to a rectangle that is specifying the extent of the object visible in the image.

These first two questions (Q1 and Q2) return 4 possible outcomes (see Figure 3). If the human says No to Q2a the robot asks for an adjustment of the bounding-box. If the human says Yes to Q2b the robot asks for a relabeling.

After confirming and correcting the predictions the robot then asks:

- Q3: Are all the objects in this image labeled?
- (if Yes to Q3): The image is ready for part (C) - Training.
- (if No to Q3): The robots ask for the human to adjust the rectangle to an object and label it.

When labeling, a list of the previously labeled objects shows up. If the person does not find the object in this list, a new class can be added from a predefined dictionary of classes. This dictionary restrains the lexical redundancies within a language, for example "waste bin" and "trash can" are categorized as only the "waste container" class.

When adding a new class, the robot shows the person the zoomable diagram (Figure 4), organized by categories from the OpenImages dataset [27]. This diagram includes 600 classes, in an organized structure. The robot only accepts new classes from this list.

We tagged the objects in this list as personal or not (e.g. classes like "mobile phone" and "backpack" are tagged as personal objects and classes like "oven" and "table" as not personal). This is used to ask the second type of questions:

2) To whom an object belongs: After labeling all the objects in the image, the robot then asks to whom each of these objects belong (if they are tagged as personal objects). This way the robot also stores information that will be useful when searching for an object belonging to a specific person.

C. Training

The interaction with humans results in a set of labeled images. Every time the robot collects a multiple of 50 images it trains a neural net, HUMAN50, HUMAN100, HUMAN150 and so on, using all the human labeled images. From these images 70% is used for training and 30% is used for testing.

The HUMAN neural net uses the same structure and algorithm as YOLOv2 but is trained using only the images labeled by humans. The reason why we train it from scratch is to avoid catastrophic interference - neural nets forget upon learning new classes. For example, if we added a new class to YOLOv2, for instance "lamp", we would have to make sure that not only the images collected by the robot but also the thousands of images used to train YOLOv2 have the "lamp"s labeled if present in each image. Otherwise, the neural net when training with the default thousands of images would always identify all the present "lamp"s in those pictures as false detections and therefore it wouldn't be able to learn to recognize this object.

After training, the HUMAN neural net is also used in part (A) - Predicting the objects. Using the predictions from both neural nets, the robot is able to predict more objects present in an image.

1) Filtering images used for training: Images with no labeled objects are not used for training. We also remove repeating labels over the same area in the image.

If a person answers Yes to Q1 (correct label), and No to Q2a (bounding-box poorly adjusted), reffering to two different predictions pertaining to the same object, then we get two repeated labels. To avoid this problem, if in an image, an IoU (Intersection over Union) larger than 0.5 is detected between the same labels, we only use the outer bouding-box for training.

The IoU is a measure of how close two bounding-boxes match. To calculate the IoU we divide the intersection area by the union (total area) of the bounding-boxes.

Preprint submitted to 2018 IEEE/RSJ International Conference
on Intelligent Robots and Systems. Received March 1, 2018.

Fig. 5: Domestic scenario where we ran the experiments.

In practice, an IoU score below 50% is probably an incorrect match. This value was set in the PASCAL VOC competition [28], humans tend to be slightly more lenient than the IoU larger than the 50% criterion [11].

IV. YOLOv2 + HUMAN

In a real human scenario robots will need to interact with an unpredictable set of classes.

We used an approach that combines two neural nets (YOLOv2 and HUMAN), running simultaneously. The HUMAN neural net serves as an auxiliary one to the one that was trained with thousands of images in a global effort.

When the two neural nets detect an object in the same area in the picture (IoU larger than 50%), we use the prediction with higher confidence. By default, a confidence from 0.0 to 1.0 is given by the YOLOv2 algorithm, associated to each prediction [8].

V. EXPERIMENTAL SETUP

Using our current computational resources, running the two neural nets as separate processes requires an external computer with a GPU. In our trails we ran the computations and trained the neural nets in this external computer[3]. The average time it took to train the neural nets was 6 hours while YOLOv2 [29] is able to recognize the objects in real-time (our bottleneck was the robot's WiFi connection).

A. Domestic Environment

Using a realistic domestic scenario (Figure 5) composed of one bedroom, one living room, one dining room and a kitchen. The experience was conducted over the course of 20 days, taking a total of 5 hours. Six research participants acted as the human input for the robot, answering questions about objects in order to evaluate if the robot could train the HUMAN neural net when confined to a small-scale environment. We also tested the applicability of using the two neural nets in parallel: YOLOv2 + HUMAN and evaluated the correctness of the robot's predictions.

[3]GPU:NVIDIA's GeForce GTX 1080 Ti; CPU: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz

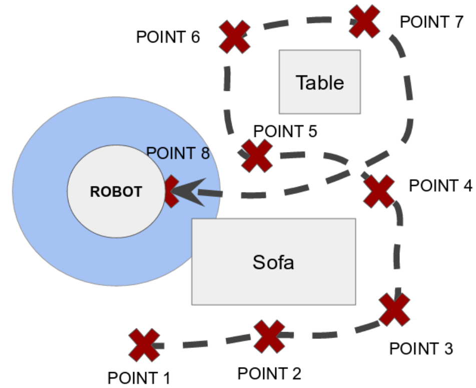


Fig. 6: Points of reference comprising the surrounding area.

1) Generating Points of Reference: The robot should be able to recognize the surrounding objects independently of its current pose (location + orientation), relative to a fixed world frame. There are infinite poses within a confined environment but the robot can't feasibly ask infinite questions to fulfill its objective. To solve this, it starts by defining a set of sparse points. When the distance between the robot and all the existing reference points is greater than a fixed distance it creates a new reference point (Figure 6).

2) Using the Points of Reference: The robot should also be able to recognize the objects in different light conditions (related to various hours of the day), and at different positions, angles and perspectives. We defined that each day the robot will navigate to 1 point of reference per day, at a random time and capture a total of 8 images at different orientations relative to the world frame. This way it captures different exposures and the implicit unpredictability of day-to-day objects.

B. Looking for a specific's person object

When the robot asks a human a question about an object associated with a certain Point of Reference, it gathers the spacial information of that object.

If the robot registers, for example a "doll", as being 1 time in the living room and 10 times in the bedroom, if asked to search for that same object it will hierarchically search by divisions with greater number of occurrences of that object. This somewhat emulates the thought process of humans when they are looking for a misplaced object, wherein they will look for the most common places where they see or use said object.

Furthermore when a user asks where his or hers object is, the robot, using the neural nets (YOLOv2 + HUMAN150), starts by detecting the objects. When the searched personal object class is detected, e.g. "backpack", the robot compares the part of the image inside the bounding-box with all the previous instances of "backpack's it has recorded and associates it to the one with the highest amount of features in common (using OpenCV's SIFT code[4]).

[4]The code can be found at https://opencv.org/

VI. RESULTS

To evaluate the proposed system, we compared the neural nets using an external ground-truth composed of 100 pictures. These images were captured by the robot in different places of the experience scenario without the constraint of being in a Point of Reference. They also included different lightning conditions and 10 blurred pictures (robot moving).

A. Average Precision

The neural nets predictions were judged by the precision/recall (PR) curve. The quantitative measure used was the average precision (AP) [30], [31], [32], [28].

It was computed as follows:

- First, we computed a version of the measured precision/recall curve with precision monotonically decreasing, by setting the precision for recall r to the maximum precision obtained for any recall $r' > r$.
- Then, we computed the AP as the area under this curve by numerical integration. No approximation was involved since the curve is piecewise constant.

First, we map each detection to a ground-truth object instance. There is a match if the labels are the same and the IoU (Intersection over Union) is larger than 50% (value established in the PASCAL VOC competitions [31]), by the formula:

$$IoU = \frac{Area(B_p \cap B_g)}{Area(B_p \cup B_g)}, \quad (1)$$

where $B_p \cap B_g$ and $B_p \cup B_g$ respectively denotes the intersection and union of the predicted and ground-truth bounding-boxes.

In the case of multiple detections of the same object only 1 (one) is set as a correct detection and the repeated ones are set as false detections [28].

B. Correctness of Predictions

During the 20 day experiment the robot trained 3 different neural nets: HUMAN50, HUMAN100 and HUMAN150. In the first week (less than 50 images), the robot was using only YOLOv2 to generate the predictions. In the second week the robot used YOLOv2 and HUMAN50. And finally, during the last week it used YOLOv2 and HUMAN100.

Looking at Table I, II and III we can see the results from the Yes or No questions (Q1 and Q2).

We observed that the HUMAN neural net, although generating a smaller amount of predictions, it was also able to produce true positives (✓Label and ✓Bounding-box), which indicate that the robot is learning.

The number of true positives increased by 25% from Human50 to HUMAN100, suggesting the evolution of the HUMAN neural net with more images retrieved from the learning algorithm.

TABLE I: Predictions - Image 0 to 50.

YOLOv2 - total number of predictions: 161

values in %	Bounding Box ✓	Bounding Box ×
Label ✓	52.2	18.6
Label ×	13.7	15.5

TABLE II: Predictions - Image 50 to 100.

YOLOv2 - total number of predictions: 139

values in %	Bounding Box ✓	Bounding Box ×
Label ✓	64.7	12.9
Label ×	5.0	17.3

HUMAN50 - total number of predictions: 38

values in %	Bounding Box ✓	Bounding Box ×
Label ✓	28.9	50
Label ×	0.0	21.1

TABLE III: Predictions - Image 100 to 150.

YOLOv2 - total number of predictions: 181

values in %	Bounding Box ✓	Bounding Box ×
Label ✓	59.1	15.5
Label ×	4.4	21.0

HUMAN100 - total number of predictions: 31

values in %	Bounding Box ✓	Bounding Box ×
Label ✓	54.8	22.6
Label ×	6.4	16.13

C. Evaluation of the neural nets

Table IV shows the AP for each of the classes and finally the mAP (mean Average Precision). Since this value is a mean, when comparing the results we need to take into account the number of classes.

Relatively to the 15 classes of YOLOv2 present in the ground-truth the mAP of YOLOv2 was 41.29% and of YOLOv2 + HUMAN150 was 45.8%, higher than the original value. Additionally, YOLOv2 was only able to detect 14 of the total number of classes while YOLOv2 + HUMAN150, 27.

In total there were only four classes where the YOLOv2 had a higher score than YOLOv2 + HUMAN150: "book", "chair", "diningtable" and "pottedplant". In all of these cases, the difference in the score was always smaller than 5%.

In this experiment we also verified the improvement of the HUMAN neural net with an increasing mAP, from 9.22% to 13.46% and finally 16.26% with the number of objects also increasing from 27 to 29 and finally 30.

D. Human mistakes

We identified two primary categories of human error in labelling images: (a) unlabeled objects (happens when objects are difficult to label, e.g. small objects, or when people

TABLE IV: Average Precision using an external ground-truth composed of 100 images in different poses. Values are in percentage (%) and the largest one per row marked in **bold**.

values in %	YOLOv2	H50	H100	H150	YOLOv2 + H150
backpack	0	**45.5**	36.4	22.7	22.7
bed	62.5	0	62.5	67.5	**85.9**
book	**18.3**	5.1	6.1	6.1	17.5
bookcase	-	**33.3**	16.7	0	0
bottle	**23.5**	-	9.1	0	**23.5**
bowl	**34.4**	0	0	0	**34.4**
cabinetry	-	11.9	**14.4**	9.8	9.8
chair	**58.1**	20.3	25.6	32.1	53.8
coffeetable	-	4.8	**9.5**	4.8	4.8
countertop	-	0	0	**23.8**	**23.8**
cup	24.0	0	2.8	25.0	**42.5**
diningtable	**43.1**	21.1	37.8	13.5	39.7
doll	-	-	-	0	0
door	-	17.2	**24.1**	20.7	20.7
heater	-	**15.4**	0	7.7	7.7
nightstand	-	28.6	28.6	**71.4**	**71.4**
person	**42.9**	0	0	0	**42.9**
pictureframe	-	0	0	**17.7**	**17.7**
pillow	-	0	5.0	**13.0**	**13.0**
pottedplant	**65.5**	17.2	17.2	17.4	62.3
remote	**73.2**	0	0	0	**73.2**
shelf	-	0	0	**16.7**	**16.7**
sink	**16.3**	-	0	7.1	**16.3**
sofa	**90.5**	0	0	4.8	**90.5**
tap	-	0	0	**1.4**	**1.4**
tincan	-	0	**1.8**	0	0
tvmonitor	48.3	5.0	20.0	33.8	**63.2**
vase	**18.8**	8.3	0	4.2	**18.8**
wastecontainer	-	15.2	**72.7**	43.9	43.9
windowblind	-	0	0	**23.5**	**23.5**
total of classes	15	27	29	**30**	**30**
mAP (all classes)	**41.29**	9.22	13.46	16.26	31.39
mAP (YOLOv2 classes)	41.29	8.16	14.62	15.6	**45.8**

couldn't find the object in the list, e.g. "fire extinguisher" was not in included in the OpenImages labels); (b) wrong label (e.g. human clicks Yes when should have clicked No).

Despite the human error, we observed that the mean average precision and the total of correct predictions increased, suggesting the improvement of the neural net.

E. Looking for a specific's person object

In Figure7, we can see the results of this experiment, where we simulated the misplacement of a backpack and remotley requested the robot to search for it (using Telegram bot API[5]). As the figure suggests, the robot searched the surrounding environment and after approximately 2 minutes he had a positive match on the subject's backpack. Pertinently there were two more backpacks present at the scene and Pepper was able to identify the desired one.

VII. CONCLUSIONS

This paper presents an approach to address the object recognition limitations of service robots. In particular, we bootstrap YOLOv2, a state-of-the-art neural, based on the teaching provided by the humans in close proximity. The robot then trains a neural net with the collected knowledge and uses two neural nets in parallel: YOLOv2 + HUMAN. By using the two neural nets, the robot gets the ability to adapt to a new environment without losing its previous knowledge. We implemented our learning algorithm in two different robots, Pepper and Mbot, and conducted experiments to test the performance of these neural nets in a domestic scenario. Potential future work includes further enhancing the object recognition capabilities with a focus on sharing information between the robots (located in different places/countries). We view the contribution of this paper to be relevant for service robots in general, in addition to Pepper and Mbot.

[5]More info can be found at https://core.telegram.org/

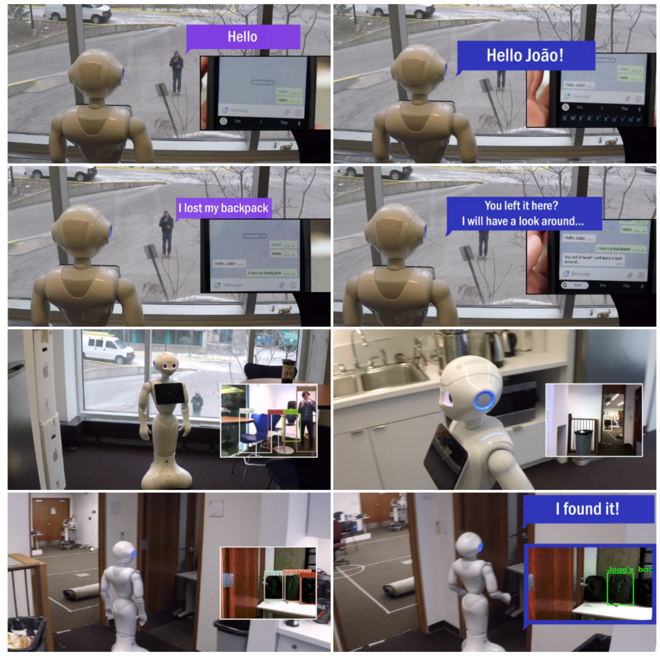


Fig. 7: Pepper looking for João's backpack demo.

ACKNOWLEDGMENT

This work was partially funded by FCT [PEst-OE/EEI/LA0009/2013] and Partially funded with grant 6204/BMOB/17, from CMU Portugal. We thank the members of the CORAL research lab at Carnegie Mellon University, led by Professor Manuela, for their help with the Pepper robot, in particular Robin Schmucker and Kevin Wang.

REFERENCES

[1] Thomaz, Andrea Lockerd, and Cynthia Breazeal. "Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance." In Aaai, vol. 6, pp. 1000-1005. 2006.

[2] Veloso, Manuela M., Joydeep Biswas, Brian Coltin, and Stephanie Rosenthal. "CoBots: Robust Symbiotic Autonomous Mobile Service Robots." In IJCAI, p. 4423. 2015.

[3] Argall, Brenna D., Sonia Chernova, Manuela Veloso, and Brett Browning. "A survey of robot learning from demonstration." Robotics and autonomous systems 57, no. 5 (2009): 469-483.

[4] Thrun, Sebastian, and Tom M. Mitchell. "Lifelong robot learning." Robotics and autonomous systems 15, no. 1-2 (1995): 25-46.

[5] SoftBank Robotics, 2018. Who is Pepper? (2018). Retrieved February 1, 2018 from https://www.ald.softbankrobotics.com/en/robots/pepper

Preprint submitted to 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2018.

[6] Hawes, Nick, Christopher Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrov, Jay Young et al. "The STRANDS project: Long-term autonomy in everyday environments." IEEE Robotics & Automation Magazine 24, no. 3 (2017): 146-156.

[7] Agence France Presse. 2016. Robot receptionists introduced at hospitals in Belgium. (2016). Retrieved February 2018 from https://www.theguardian.com/technology/2016/jun/14/robot-receptionists-hospitals-belgium-pepper-humanoid

[8] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." arXiv preprint 1612 (2016).

[9] Sam Byford (The Verge). 2014.SoftBank announces emotional robots to staff its stores and watch your baby. (2014). Retrieved February 1, 2018 from https://www.theverge.com/2014/6/5/5781628/softbank-announces-pepper-robot

[10] Veloso, Manuela M., Joydeep Biswas, Brian Coltin, Stephanie Rosenthal, Susana Brandao, Tekin Mericli, and Rodrigo Ventura. "Symbiotic-autonomous service robots for user-requested tasks in a multi-floor building." (2012).

[11] Papadopoulos, Dim P., Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. "We don't need no bounding-boxes: Training object class detectors using only human verification." arXiv preprint arXiv:1602.08405 (2016).

[12] Vijayanarasimhan, Sudheendra, and Kristen Grauman. "Large-scale live active learning: Training object detectors with crawled data and crowds." International Journal of Computer Vision 108, no. 1-2 (2014): 97-114.

[13] Thomaz, Andrea L., and Maya Cakmak. "Learning about objects with human teachers." In Proceedings of the 4th ACM/IEEE international conference on Human robot interaction, pp. 15-22. ACM, 2009.

[14] Rosenthal, Stephanie, Joydeep Biswas, and Manuela Veloso. "An effective personal mobile robot agent through symbiotic human-robot interaction." In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, pp. 915-922. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[15] Rosenthal, Stephanie, and Manuela Veloso. "Using symbiotic relationships with humans to help robots overcome limitations." In Workshop for Collaborative Human/AI Control for Interactive Experiences. 2010.

[16] Ventura, Rodrigo, Brian Coltin, and Manuela Veloso. "Web-based remote assistance to overcome robot perceptual limitations." In AAAI Conference on Artificial Intelligence (AAAI-13), Workshop on Intelligent Robot Systems. AAAI, Bellevue, WA. 2013.

[17] Bohren, Jonathan, Radu Bogdan Rusu, E. Gil Jones, Eitan Marder-Eppstein, Caroline Pantofaru, Melonee Wise, Lorenz Msenlechner, Wim Meeussen, and Stefan Holzer. "Towards autonomous robotic butlers: Lessons learned with the PR2." In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp. 5568-5575. IEEE, 2011.

[18] Watanabe, Sumio, and Masahide Yoneyama. "An ultrasonic visual sensor for three-dimensional object recognition using neural networks." IEEE transactions on Robotics and Automation 8, no. 2 (1992): 240-249.

[19] Maturana, Daniel, and Sebastian Scherer. "Voxnet: A 3d convolutional neural network for real-time object recognition." In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pp. 922-928. IEEE, 2015.

[20] Collet, Alvaro, Dmitry Berenson, Siddhartha S. Srinivasa, and Dave Ferguson. "Object recognition and full pose registration from a single image for robotic manipulation." In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pp. 48-55. IEEE, 2009.

[21] Itti, Laurent, Christof Koch, and Ernst Niebur. "A model of saliency-based visual attention for rapid scene analysis." IEEE Transactions on pattern analysis and machine intelligence 20, no. 11 (1998): 1254-1259.

[22] Nasrabadi, Nasser M., and Wei Li. "Object recognition by a Hopfield neural network." IEEE Transactions on Systems, Man, and Cybernetics 21, no. 6 (1991): 1523-1535.

[23] DInnocente, Antonio, Fabio Maria Carlucci, Mirco Colosi, and Barbara Caputo. "Bridging between computer and robot vision through data augmentation: a case study on object recognition." In International Conference on Computer Vision Systems, pp. 384-393. Springer, Cham, 2017.

[24] Kemp, Charles C., Cressel D. Anderson, Hai Nguyen, Alexander J. Trevor, and Zhe Xu. "A point-and-click interface for the real world:

laser designation of objects for mobile manipulation." In Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on, pp. 241-248. IEEE, 2008.

[25] Ekvall, Staffan, Patric Jensfelt, and Danica Kragic. "Integrating active mobile robot object recognition and slam in natural environments." In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pp. 5792-5797. IEEE, 2006.

[26] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In European conference on computer vision, pp. 740-755. Springer, Cham, 2014.

[27] Krasin I., Duerig T., Alldrin N., Ferrari V., Abu-El-Haija S., Kuznetsova A., Rom H., Uijlings J., Popov S., Veit A., Belongie S., Gomes V., Gupta A., Sun C., Chechik G., Cai D., Feng Z., Narayanan D., Murphy K. OpenImages: A public dataset for large-scale multi-label and multi-class image classification, 2017. Available from https://github.com/openimages

[28] Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes (voc) challenge." International journal of computer vision 88, no. 2 (2010): 303-338.

[29] Marko Bjelonic. 2017. YOLO ROS: Real-Time Object Detection for ROS. (2017). Retrieved January 21, 2018 from https://github.com/leggedrobotics/darknet_ros

[30] Everingham, M. and Van Gool, L. and Williams, C. K. I. and Winn, J. and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

[31] Everingham, Mark, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes challenge: A retrospective." International journal of computer vision 111, no. 1 (2015): 98-136.

[32] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang et al. "Imagenet large scale visual recognition challenge." International Journal of Computer Vision 115, no. 3 (2015): 211-252.