RoboCup@Home-Objects: Benchmarking Object Recognition for Home Robots

Nizar Massouh(✉), Lorenzo Brigato, and Luca Iocchi

Department of Computer Control and Management Engineering,
Sapienza University of Rome, Rome, Italy
{massouh,brigato,iocchi}@diag.uniroma1.it

Abstract. This paper presents a benchmark for object recognition inspired by RoboCup@Home competition and thus focusing on home robots. The benchmark includes a large-scale training set of 196K images labelled with classes derived from RoboCup@Home rulebooks, two medium-scale test sets (one taken with a Pepper robot) with different objects and different backgrounds with respect to the training set, a robot behavior for image acquisition, and several analysis of the results that are useful both for RoboCup@Home Technical Committee to define competition tests and for RoboCup@Home teams to implement effective object recognition components.

Keywords: Object recognition · Benchmarking · Service robots

1 Introduction

RoboCup@Home competition[1] aims at developing and benchmarking home service robots that can help people in everyday tasks. The competition is organized around a set of tasks in which several functionalities must be properly integrated [6,9]. Among these functionalities, *object recognition* is present in many tasks and it is thus very important for the competition as well as for actual deployment of home robots. Active object recognition was also benchmarked as a Technical Challenge in RoboCup@Home 2012. Homer@UniKoblenz achieved the highest score in this challenge by using SURF features and Hough transform clustering[2] applied to high resolution photos acquired by a digital camera. In this challenge, the robot had to move to the table where objects were located and thus active motion actions needed to be carefully designed to reach good view points for image acquisition.

In the last years, we have witnessed a significant effort in improving object recognition performance, specially boosted by the development of Convolutional Neural Networks (CNNs) and large-scale image databases (e.g., ImageNet [3]).

[1] https://athome.robocup.org.
[2] http://wiki.ros.org/obj_rec_surf.

S. Chalup et al. (Eds.): RoboCup 2019, LNAI 11531, pp. 397–407, 2019.
https://doi.org/10.1007/978-3-030-35699-6_31

Consequently, RoboCup@Home teams have shifted to machine learning techniques that promise very good results. However, such results are strongly influenced by the quality of training data and by computational resources available. Thus, many teams have to bring to the competition computational resources suitable to train CNNs and have to spend a lot of effort in acquiring images and train the networks during the setup days. While acquiring images about the specific objects chosen for the competition and training CNNs during the setup days is a suitable way of implementing the object recognition functionality of @Home robots, we believe that there are other processes that can help in implementing an effective object recognition functionality exploiting pre-trained CNNs and without requiring availability of competition objects, image acquisition and training during the setup days.

In this paper we present a benchmark for RoboCup@Home object recognition based on a large-scale training set acquired from the web and pre-trained models. More specifically, we provide: (1) a novel large-scale data set for RoboCup@Home (named **RoboCup@Home-Objects**) with over 196K images acquired from the web and automatically labelled with 8 main categories and 180 classes typically used in RoboCup@Home; (2) pre-trained CNNs on this data set that can be used by RoboCup@Home teams; (3) a test sets containing thousands of images acquired from the web with objects similar to the ones actually used in recent RoboCup@Home competitions; (4) a test set containing thousands of images taken from Pepper robot in a scenario similar to the ones encountered in the competitions; (5) a method based on active robot behaviors to improve quality of image acquisition and take advantages of pre-trained CNNs to improve actual performance of object recognition without any training on the specific competition objects; (6) a performance analysis that allows RoboCup@Home Technical Committee to better define competitions tasks involving object recognition. Although created for the RoboCup@Home community, we believe the benchmark, the models and the results will be interesting for all researchers aiming at integrating object recognition in home robots.

Data, models and results will be fully available in the web site https://sites.google.com/diag.uniroma1.it/robocupathome-objects[3].

2 Related Work

Ever since the exceptional results of Alexnet [7] in the ImageNet Large Scale Visual Recognition Challenge of 2012 (ILSVRC12) [3] the use of Deep Learning and CNNs for robot vision applications increased substantially. Deep Networks need large-scale annotated databases to be successfully trained or they will suffer from over-fitting. This made ImageNet the most used database for Deep network architectures. Another approach to avoid over-fitting and learn a new task is fine-tuning [1]. Fine-tuning is the method of re-training parts of a pre-trained network to fit a new task or new annotated data. Anyway fine-tuning still requires large-scale annotated data sets. As manually annotating large-scale

[3] Currently under development, will be completed before RoboCup@Home 2019.

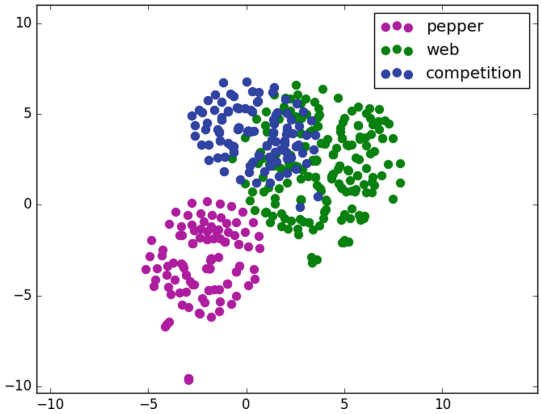


Fig. 1. The t-SNE visualization of the data-sets distributions' extracted features of the fully connected layer (FC7) with our Alexnet@Home180.

data sets requires too much human effort, automatic acquisition can produce suitable large-scale data sets with significantly lower effort. In particular, we can use the Web to collect many images related to a new task in an autonomous matter. In recent years successful attempts have been made to generate large scale databases from the web [2,4]. Work on automatic data collection for robot vision applications with deep networks was proposed for example in [8]. In the benchmark proposed in this paper, we have automatically downloaded images from the Web to build data sets for training and testing CNN-based object recognition for the RoboCup@Home competition.

3 Dataset Creation

In this section we will describe the process of creating 3 datasets: (1) a main dataset of 196K images acquired from the web that is used for training, (2) a benchmark dataset of about 5K images downloaded from the web as images similar to the ones published on RoboCup@Home github; (3) a benchmark dataset acquired from Pepper robot. It is important to notice that these datasets are acquired from different sources and thus come from different distributions, as shown in Fig. 1. Data acquired from different sources allows for a more realistic assessment of the performance that one can expect when using this technology. We thus believe that the experimental performance reported in this paper will provide a reliable estimation of expected performance of the tested networks in actual RoboCup@Home competition.

Below we will briefly describe the data sets acquired, while all the details are provided in the above mentioned web site.

RoboCup@Home-Objects Training Set. To successfully create a dataset that can be used for the RoboCup@Home competition we first need to define a

structure for the categories. Past competitions' used objects and their categories can provide an insight to select ours. We were able to pinpoint 8 main categories: *Cleaning_stuff, Containers, Cutlery, Drinks, Food, Fruits, Snacks and Tableware.* Although some of the categories can be considered subsets of others we will place them at the same hierarchical level and define a list of children for each class. This step will help us increase the variability of our categories. Most of the categories can be considered products and with the popularity of online shopping we will be able to get a specific list of products for each parent. Amazon.com is currently the most used online shopping platform in the world and that allowed it to build a very up-to-date hierarchy of products. We gather 180 children (all mutually exclusive leaves) of our 8 parents. Table 1 shows how the children are distributed among the eight categories. The label of each image will be composed of the parent and the child: "parent/child". We would like to prove that having this hierarchical structure can be used as an advantage by allowing us to switch between a specific label to a more general category (parent category). This should prove useful when encountering never before seen objects.

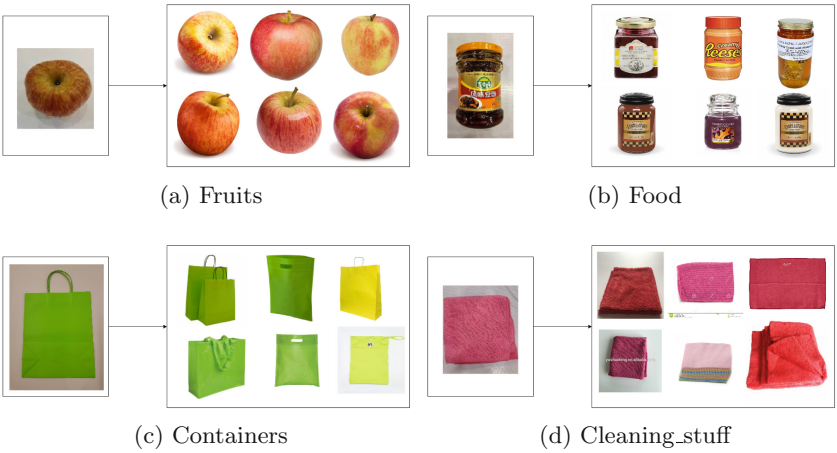
Table 1. Distribution of the 180 classes of the RoboCup@Home-Objects dataset.

Parent name	Number of children
Cleaning_stuff	37
Containers	17
Cutlery	15
Drinks	17
Food	22
Fruits	23
Snacks	26
Tableware	23

The list of children is then used as a query list to search and download images on the web. With Google, Yahoo and Bing as our search engines we download images for each child's category name. These search engines are known to have a bias toward photos of objects with a clean background. After the data collection we use Perceptual Hashing on the images to identify and eliminate duplicates. After cleaning the database, we end up with a total of 196K images that we call RoboCup@Home-Objects.

RoboCup@Home github-seeded Web Benchmark. The RoboCup@Home github repository[4] contains the list and photos of objects actually used in several competitions. We have used these photos as seeds to create a benchmark from visually similar images collected from the web. We took advantage of the reverse

[4] https://github.com/RoboCupAtHome.



(a) Fruits (b) Food

(c) Containers (d) Cleaning_stuff

Fig. 2. Examples of different categories from the RoboCup@Home github-seeded Web images.

image search provided by Google to produce this dataset. Google's reverse image search takes an image with an optional label and provides images that are visually and semantically similar. After collecting the competitions' images of objects we end up with 160 photos divided in 8 categories (our parent categories). We then proceed to use each of these photos as seeds providing their category as a label and we downloaded the first 50 returned images. After cleaning the duplicated images we end up with a total of 5,750 images labelled with the 8 parent categories defined above. As we can observe in Fig. 2 the downloaded images have the same visual features of the seeds used.

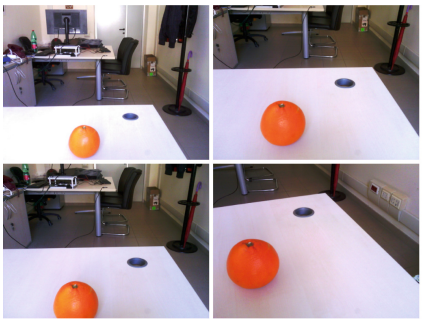
RoboCup@Home Pepper Objects Benchmark. RoboCup@Home Pepper Objects Benchmark has been acquired by using a Pepper robot, which is one of the standard platforms for RoboCup@Home. We selected a set of objects and placed each of them on a table (in different positions and different orientations with respect to a window to increase variability with respect to lighting conditions, including back-light situations).

Image acquisition was performed with an autonomous behavior of the robot reproducing operations during RoboCup@Home tasks. More specifically, the robot is *imprecisely* placed about 30 cm away from the table oriented towards it and executes the behavior illustrated in Fig. 3. For each run, 32 distinct images are thus automatically collected and organized in four groups: A (1 image), B (1 image), C (10 images), D (30 images), with $C \subset D$. Currently, we have acquired images of 14 objects (several runs in different locations), for a total of 2,624 images[5]. Some examples of acquired images are shown in Fig. 4.

[5] This data set will be further increased in the future, possibly as a community effort.

function TAKEIMAGES() $\rightarrow \langle A, B, C, D \rangle$
  Stand posture
  Tilt head down
  Take image A
  Lean forward
  Take image B
  for $i = 1, \ldots, 10$ **do**
    Random head motion
    Take image C_i
    Copy image C_i in D_i
  end for
  Move left
  for $i = 11, \ldots, 20$ **do**
    Random head motion
    Take image D_i
  end for
  Move right
  for $i = 21, \ldots, 30$ **do**
    Random head motion
    Take image D_i
  end for
  return $\langle A, B, C, D \rangle$
end function

Fig. 3. Procedure to acquire images of an object for Pepper benchmark



(a) Fruits

(b) Cleaning_stuff

Fig. 4. Pepper objects examples of different configurations: A - top left, B - top right, C - bottom left, D - bottom right.

4 Models Training

We proceeded to split our RoboCup@Home-Objects training set into 80% of images for training and 20% for validation. We fine-tuned an AlexNet [7] and a GoogleNet [10] pretrained on Imagenet's ILSVRC12 on our data, using the Caffe framework on NVIDIA Deep Learning GPU Training System (DIGITS).

We froze all the layers of the networks and learned the last fully connected layer for Alexnet and the last pool layer for Googlenet while boosting their learning multiplier by +1. We set our initial learning rate to 0.001 which we step down by 10% every 7.5 epochs. We trained both networks for 30 epochs with a stochastic gradient decent (SGD) solver.

We trained the same models on the training set using only the 8 parent categories. We ended up with 4 models: *Alexnet@home180* and *Googlenet@home180*, trained on all the children categories, *Alexnet@home8* and *Googlenet@home8*, trained on the parents categories only. For our models trained on the 180 children categories, we then execute a category mapper to map the result onto the 8 parents category. Table 2 shows the scored Top 1 accuracy percentage of the 4 models on the validation set and we can see that having less categories to learn made the task easier on the models.

Table 2. Validation accuracy percentage of our 4 models.

Model	Accuracy
Alexnet@home8	77.89%
Alexnet@home180	47.85%
Googlenet@home8	81.91%
Googlenet@home180	53.55%

5 Analysis of Results

In this section we present a brief summary of the results obtained using data and models described above. More details are provided in the web site.

Table 3. Top-1 and Top-5 parent majority accuracy of our 4 models trained with RoboCup@Home-Objects and tested on github objects (gh-o) and github-seeded dataset (gh-s).

Accuracy percentage	Top-1		Top-5 parent	
	gh-o (160)	gh-s (5.7K)	gh-o (160)	gh-s (5.7K)
Alexnet@Home180	70.44	65.86	**73.58**	70.84
Alexnet@Home8	67.29	66.74	67.29	66.74
Googlenet@Home180	64.78	67.33	70.44	**71.68**
Googlenet@Home8	**72.95**	**67.86**	72.95	67.86

5.1 Test Accuracy on Github-Seeded Web Benchmark

We are interested in the comparison of the results obtained in the two data sets collected from the web and of the models trained on the parents vs. the children.

As we can observe in Table 3, the results by all 4 models are very close to each other, with a little decrease of performance in the github-seeded benchmark (which contains 5.7 K images) in comparison with the 160 original github images. This means that our method for the creation of the github-seeded benchmark from web images is a successful data augmentation method that mimics the distribution of the source as we can see also in Fig. 1. We notice as well that Googlenet@Home8 slightly outperformed its child model and the other models. In the Top-5 parent section of Table 3 we used the top 5 predicted labels and returned the majority parent, i.e. if the top 5 predicted categories are: *"tableware/bowl", "fruits/orange", "fruits/tangerine", "tableware/coffee_mug", "fruits/melon"* the returned category is *"fruits"*.

Finally, we observe an increase in accuracy when using the models trained on the 180 categories. Googlenet@Home180 was able to outperform our previous best by 4% for the github-seeded benchmark and for the github objects our new best is by Alexnet@Home180. This shows the advantages of mapping the result to the parent label (more general category) which adds flexibility in case of indecision.

5.2 Analysis on Pepper Benchmark

When using the Pepper dataset that, as already mentioned, has a very different distribution with respect to training, and without fine-tuning on the specific objects, there is a general decrease of performance in accuracy, but more importantly a very large variance of the results depending on the object.

By analyzing the performance of our models on this dataset without fine-tuning, we can assess the difficulty of recognizing each particular object. As discussed in the next section, this information can be very useful to competition organizers to choose proper objects for the competition as well as assigning a more suitable score to each kind of object. As an example, we show in Table 4 the result of the application of Googlenet@Home-180 on a set of objects of the Pepper benchmark averaged over all acquisition procedures. We can notice how different the result can change from one object to another. This wide range of results can be contributed to either the quality of the image taken or by how well the object represents its category. In the case of the Cookies our model kept predicting "Food" instead of "Snacks" which can be confusing since snacks can be considered a child category of Food. A detailed analysis has been done (available on the web site) on each object and a rank denoting the difficulty of recognizing every object without fine-tuning has been produced.

Active Image Acquisition. Finally, we have evaluated the active behavior of the Pepper robot in acquiring images from different view-points. To this end, we considered a subset of objects in the Pepper benchmark and two different

Table 4. Result of the Googlenet@Home-180 model on 7 different Pepper objects. The result is reported in Accuracy percentage over all acquisition procedures.

Category	Object	Accuracy
cleaning_stuff	cleaners	93.28%
tableware	cup	70.00%
fruits	orange	60.00%
drinks	water_bottle	43.33%
snacks	snack	36.67%
fruits	banana	9.68%
snacks	cookies	0.00%

Table 5. Accuracy on 5 selected objects varying robot acquisition behavior.

Model	A	B	C	D
Googlenet@Home-180	13.3%	43.3%	50.0%	50.0%
Mobilenet-Imagenet	26.7%	70.0%	73.3%	73.3%

models: Googlenet@Home-180 (as described above) and MobileNet model pre-trained on ImageNet [5]. These objects are: *banana*, *orange*, *cup*, *water_bottle*, and *plastic_bag*, whose labels are present within the 1,000 output categories of Imagenet trained models.

The results summarized in Table 5 show accuracy over 6 tests per each object and for each robot acquisition behavior A, B, C, D. For evaluating types with multiple images (i.e., C and D) a majority vote scheme was performed and the most voted class in all the images is compared with the true label. As shown, the behavior of leaning towards the object (B) gives significantly better results with respect to nominal behavior (A), while moving the head to view the objects from different view points (C, D) gives only a little additional advantage with respect to B. This observation should help researchers to properly balance acquisition time (C and D behavior are much longer) with recognition accuracy.

Finally, when we can assume that exactly one of these objects is in the image (as it is often the case during the competition), we can consider the highest confidence among only these 5 labels. In this way we obtained 100% accuracy in most cases.

6 Discussion and Conclusions

The results on the RoboCup@Home-Objects data can be useful for both @Home Technical Committee and teams.

From the perspective of the Technical Committee, an analysis of the difficulty in recognizing specific objects can drive choices about definition and scoring of the tests. We describe here three examples. (1) *Easy configuration*, choose a few

objects (e.g., 5 to 10) that are within the RoboCup@Home-Objects/ImageNet labels and place exactly one of them in the environment: expected accuracy is almost 100% with a pre-trained model[6]. (2) *Medium configuration*, choose one object for each of the 8 parent categories of RoboCup@Home-Objects: expected accuracy is around 70% without fine tuning at the competition site, that can be enough in some cases, for example with a proper reasoning system or human-robot interactions to disambiguate or ask for confirmation. (3) *Difficult configuration*, choose objects with non-ImageNet labels among the ones that give worst accuracy results on our benchmark: expected accuracy is too low and on-site fine-tuning is necessary to perform well in the competition.

Another possibility for the Technical Committee is to define categories of objects granting more score for objects that are more difficult to recognize, possibly allowing the teams to choose. This would allow teams not willing to focus on object recognition to choose easy objects, use pre-trained models and focus their development on other functionalities (e.g., object manipulation or human-robot interaction), still having very good performance in object recognition. Evaluation of object recognition difficulty can be easily done by just use our pre-trained models on candidate objects selected for the competitions.

On-going and future work include extension of the data sets and involvement of the community (RoboCup@Home teams and other researchers) to improve the quality of the benchmark and of object recognition functionalities in home robots.

References

1. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: delving deep into convolutional nets. In: Proceedings BMVC (2014)
2. Cheng, D.S., Setti, F., Zeni, N., Ferrario, R., Cristani, M.: Semantically-driven automatic creation of training sets for object recognition. Comput. Vis. Image Underst. **131**, 56–71 (2015)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: Proceedings CVPR, pp. 248–255 (2009)
4. Divvala, S.K., Farhadi, A., Guestrin, C.: Learning everything about anything: webly-supervised visual concept learning. In: Proceedings CVPR, pp. 3270–3277 (2014)
5. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861 (2017). http://arxiv.org/abs/1704.04861
6. Iocchi, L., Holz, D., Ruiz-del-Solar, J., Sugiura, K., van der Zant, T.: RoboCup@Home: analysis and results of evolving competitions for domestic and service robots. Artif. Intell. **229**, 258–281 (2015)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings NIPS (2012)

[6] This configuration was successfully tested at European RoboCup@Home Education Challenge 2019, where inexperienced high-school teams were able to use an almost perfect object recognition module.

8. Massouh, N., Babiloni, F., Tommasi, T., Young, J., Hawes, N., Caputo, B.: Learning deep visual object models from noisy web data: how to make it work. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, 24–28 September 2017, pp. 5564–5571 (2017). https://doi.org/10.1109/IROS.2017.8206444
9. Matamoros, M., Seib, V., Memmesheimer, R., Paulus, D.: RoboCup@Home; summarizing achievements in over eleven years of competition. CoRR abs/1902.00758 (2019). http://arxiv.org/abs/1902.00758
10. Szegedy, C., et al.: Going deeper with convolutions. CoRR abs/1409.4842 (2014)