Object Identification from Few Examples by Improving the Invariance of a Deep Convolutional Neural Network

Giulia Pasquale[1,2,3], Carlo Ciliberto[2,4], Lorenzo Rosasco[2,3,4] and Lorenzo Natale[1]

***Abstract*— The development of reliable and robust visual recognition systems is a main challenge towards the deployment of autonomous robotic agents in unconstrained environments. Learning to recognize objects requires image representations that are discriminative to relevant information while being invariant to nuisances, such as scaling, rotations, light and background changes, and so forth. Deep Convolutional Neural Networks can learn such representations from large web-collected image datasets and a natural question is how these systems can be best adapted to the robotics context where little supervision is often available.**

In this work, we investigate different training strategies for deep architectures on a new dataset collected in a real-world robotic setting. In particular we show how deep networks can be tuned to improve invariance and discriminability properties and perform object identification tasks with minimal supervision.

I. INTRODUCTION

Object recognition plays a fundamental role in many robotics tasks, such as reaching for objects, grasping, manipulation, navigation, interaction with humans and many others. Machine learning has driven recent advances in computer vision and lead to remarkable results in a variety of challenging problems [1], [2], [3], [4], [5]. While there are already promising results, robotic vision context comes with its own challenges [6], [7], [8], [9], [10]. State-of-the-art vision systems are based on deep Convolutional Neural Networks (CNNs) trained on large web-data corpora to learn rich discriminative representations invariant to nuisances such as viewpoint changes. In the robotics context the emphasis is not on web-scale image retrieval but rather on the effective recognition of everyday objects in structured context, such as a laboratory or an office. The objects to be learned might not be known a priori and the collection of labeled data is a costly operation. Indeed, learning from limited supervision is a hallmark of human learning which is key for robotics. It is then natural to ask how current deep learning systems can be best adapted to this setting.

A simple yet useful idea is that of using the network parameters obtained by training on large web-data corpora collections to *initialize* the training of deep networks by back-propagation on new potentially smaller datasets. This approach, often referred to as *fine-tuning* [3], [4], [11], seemingly allows to "transfer" representations learned on larger datasets to smaller ones. Fine-tuning however is a tricky procedure and comes in different flavors. On the one hand, it is possible to perform a light tuning, by updating only the last few layers of the networks. On the other hand, a more aggressive tuning can be performed to change the network more "deeply" based on the new data. More generally, fine-tuning requires finding the best architecture setting, a question which is well-known to be a challenge when using deep networks.

In this paper, we compare various strategies to fine-tune and adapt a CNN architecture to perform typical object recognition tasks faced by the iCub [12] humanoid robot in its usual environment. In order to test the discrimination and invariance properties of the obtained architectures, we collect and make available for the community a new image dataset, comprising multiple objects belonging to different categories and undergoing isolated visual transformations.

Our main contributions are: 1) we investigate how different approaches to fine-tuning affect the overall recognition capability of a network, specifically related to the kind of transformations observed during training; 2) we identify a strategy to achieve remarkable accuracies in a robotic environment, for which we report performance on a challenging 50-objects identification task; 3) we release a new object recognition dataset for robotics comprising 150 instances evenly divided into 15 categories, where objects visual transformations have been isolated in order to test the invariance properties of recognition systems.

With respect to recent work [13] considering the exploitation of deep CNNs in a problem setting similar to the one we address here, we operate in a less constrained scenario, for which we also provide a faithful benchmark. Moreover, by isolating the different nuisances typically affecting the performance of recognition systems in this kind of applications, we are able to measure specific network invariances and come up with a working approach for a real-world robotic object identification application.

The rest of this work is organized as follows: Sec. II briefly reviews recent work on deep CNNs and introduces the main concepts related to these architectures. Sec. III-A describes the dataset used for experiments and the acquisition protocol adopted to collect it. Sec. III-B reports the technical details of the models used in this paper. Finally, Sec. IV describes our experimental analysis and empirical observations, while Sec. V offers concluding remarks and direction for future investigation.

[1]iCub Facility, Istituto Italiano di Tecnologia (IIT), Genova, Italy

[2]Laboratory for Computational and Statistical Learning, Istituto Italiano di Tecnologia (IIT), Genova, Italy

[3]Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi, Universitá degli Studi di Genova, Genova, Italy

[4]Poggio Lab, Massachusetts Institute of Technology, Cambridge, USA

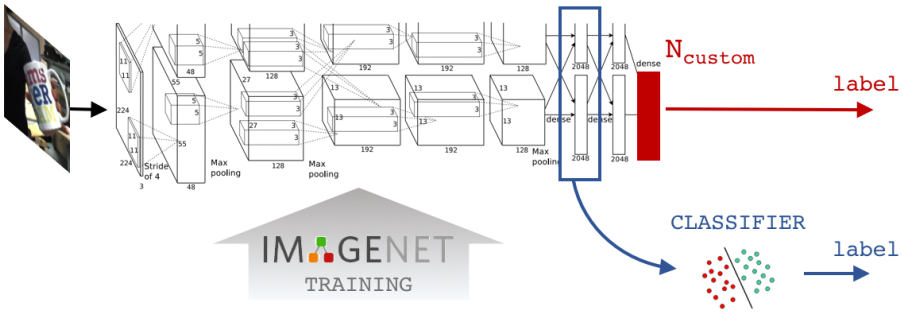

Fig. 1. The two approaches that are commonly adopted to "adapt" deep CNNs trained on large-scale datasets to smaller-scale domains: feature extraction in combination with statistical tools as RLS or SVMs (blue) and fine-tuning (red). In this work in particular we consider the architecture proposed in [1].

II. METHODS AND RELATED WORK

In this Section we introduce deep Convolutional Neural Networks (CNNs) and discuss standard methods to adapt pre-trained models on new tasks.

A. Deep Convolutional Neural Networks (CNNs)

Deep CNNs are hierarchical architectures concatenating signal processing layers to map an input image into a corresponding output representation in a *feature-space* (see Fig. 1). The prototypical structure of a CNN repeats at each layer, usually indicated as *convolution layer*, the following set of operations:

Filtering with respect to a (learned) bank of filters
Spatial Downsampling e.g. via strided convolutions
Non Linearity such as a Sigmoid or ReLU
Spatial Pooling e.g., max-pooling on a local region
Normalization across feature channels

The details of these operations depend to each architecture, but the general principle is that selectivity of the representation is achieved by filtering the signal at each layer with templates tuned to specific patterns (e.g. edges at lower layers, object parts at deeper ones). Such templates can be learned according to supervised or unsupervised strategies. Through this hierarchical processing, the semantic content of the image is progressively "distilled" into a vectorial representation that is ideally robust to visual transformations.

A typical approach in image classification settings is then to concatenate the convolution layers with a certain number of *fully connected layers* (a multi-layer Neural Network), acting as a classifier on top of the representation produced by convolution layers. The output of the last fully-connected layer comprises as many units as the number of classes to be discriminated and, after a *softmax* normalization step can be interpreted as a vector of class probabilities. The modular structure of this model allows to perform end-to-end training of all layers simultaneously by back-propagation [14].

B. Feature Extraction and Statistical Tools

The representation learned by a CNN on a large-scale dataset, such as ImageNet [15], proved to be able to generalize well to other tasks and datasets by employing the network as a "feature extractor": Given a novel image, its representation is obtained by taking the activations of network units at intermediate layers rather than the final classification scores. The intuition behind this strategy is that such representation, learned on a rich variety of examples, encodes the most relevant semantic information of the image, while being robust to most visual nuisances.

This approach has been key to the recent popularity of deep CNNs. Indeed, it has been observed that training a standard classifier such as a *Support Vector Machine (SVM)* or a *Regularized Least Squares Classifier (RLS)* [16] on such features consistently outperforms more sophisticated competitors on a variety of challenging datasets [17], [3], [18], [19], [20], [10], [21]. In Fig. 1 we report a pictorial representation of this strategy for the case of the *CaffeNet* [22] network used in our experiments.

C. Fine-tuning

A previously learned representation can be "adapted" to the new task. This process, known as *fine-tuning*, consists in using the network parameters obtained by training on the original dataset as a *warm restart* when training them by back-propagation on the new one. Fine-tuning has been successfully adopted in a variety of recent works [3], [4], also adapting the representation of networks trained on ImageNet to robotic contexts to predict grasp locations [8] or to jointly process RGB and depth information [6] leading to significant improvements on the state of the art on well-established robotic benchmarks such as RGBD [23].

The structure of the network is typically updated by substituting the output layer to account for the new task (e.g. different classes). Hence, a crucial difference between training a network from scratch and fine-tuning it is that in the latter case the parameters of most layers can be initialized with the result of the first optimization and may not need to be changed dramatically during the second training phase. Their learning rates (namely the step-size used by the stochastic gradient descent during back-propagation) can therefore be set to zero (no update is done) or to very small values, while the learning rate of the output layer is maintained to a higher value. In practice, this allows to train the CNN on much smaller datasets in significantly shorter training times.

Fig. 2. Representation of the 15×10 objects present in the *iCubWorld - Transformations* dataset that we make available to the community. For each instance, we acquired in two separate days different sequences containing specific transformations ($2D$ Rotation, Scale, Background), extracts of which are represented to the left

If, on the one hand, the fine-tuning approach is extremely flexible, in that it allows adapting the network with respect to a novel task, on the other hand it is also prone to overfitting, eventually disrupting the visual representation learned by the pre-trained CNN. In this sense, a relevant issue is whether it can be in general more or less favorable to adapt the intermediate layers of the network with respect to the new training data (i.e. with higher learning rates) or keep a more conservative approach. This question is addressed in Sec. IV-A, where we investigate also the role of the dataset used for fine-tuning with respect to learning invariances and overall recognition performance.

III. EXPERIMENTAL SETUP

In this section we describe the setup that we used for the experimental analysis reported in Sec. IV. We begin by presenting the dataset we collected to investigate our questions regarding invariance and then discuss the technical details of the CNN used for our experiments.

A. *iCubWorld - Transformations*

A major focus of this work is to characterize and quantify the ability of a network to learn representations that are invariant to visual transformations of an object. To investigate these aspects of CNNs, we collected a dataset where different object transformations are isolated and tested separately. Since we are interested in solving object recognition within a real robotic scenario, we performed our data acquisition on an actual robot, the iCub [12].

The procedure employed for data acquisition was proposed in [24] and then adopted to collect the first and second releases of the *iCubWorld* dataset (shortened to *iCW* for simplicity in the following), a benchmark for object recognition methods in robotics [25], [26]. The acquisition protocol is summarized in the following (for more details we refer the reader to the original work): a human "teacher" shows a set of objects to the robot, which uses a tracking routine [21], [27], [28] to follow them with its gaze and extract an approximate bounding box around them. Supervision, in form of the object's label, is provided verbally by the human.

In this work we adopted this strategy to acquire a novel dataset which is remarkably larger than the initial *iCubWorld* (in terms of both number of images and object classes) and is organized in order to isolate specific visual transformations of the observed objects. The dataset, that we call *iCubWorld - Transformations*, comprises 150 objects evenly divided into 15 categories, which are shown in Fig. 2. For each object

TABLE I

SUMMARY OF THE *iCubWorld - Transformations* DATASET

# Cat.	# Obj. per Category	Transformations	# Days	# Frames per Acq. Session
15	10	*Rotation, Scale Background*	2	~ 150
Globally $\sim 140K$ images in 900 acquisitions				

we had the human apply three transformations, starting the acquisition from the same view of the object for each session. We report examples of such process in Fig. 2:

Rotation The human rotated the object in front of the camera, parallel to the image plane while keeping the object at the same scale and position

Scale The human moved the hand holding the object back and forth, thus changing the object's scale with respect to the cameras

Background The human moved in a semi-circle around the iCub, keeping approximately the same distance and pose of the object in the hand with respect to the cameras. Therefore only the background changes dramatically while the object appearance remains approximately the same

For each acquisition session we collected $150 - 200$ images at 8 frames per second. A square 256×256 crop around the object of interest is extracted from each image, originally at 640×480 resolution, using the procedure described in [21]. For each object, we performed data acquisition during 2 separate days in order to consider the effect of possible biases (such as lighting). Table I summarizes the details of the dataset, which was made publicly available[1]). Figure 2 illustrates the 150 objects present in the dataset, organized by category. Three video excerpts show examples of object transformations.

B. Network Setup

Here we review the details of the network models considered for our analysis.

Reference Network. As a reference model we adopt the architecture proposed in [1] and trained on ImageNet (depicted in Fig. 1). Such network has been extensively applied to a variety of tasks [6], [13] and we use its publicly available implementation within *Caffe* deep learning framework [22], namely the *BVLC Reference CaffeNet* model.

Fine-tuning Strategies. We consider the two most used approaches for fine-tuning (see Sec. II-C), which differ one from the other with respect to the learning rates used for the layers of the CNN. We refer to them as the *conservative* and *adaptive* fine-tuning strategies: both adopt a learning rate of 10^{-2} for the weights of the output layer (*fc8*) of the network. However, the *adaptive* strategy applies a learning rate of 10^{-3} to the remaining layers from *fc7* and below, while the *conservative* one does not adapt these layers to the new dataset (i.e. the learning rate of the layers below *fc8* is set to zero). These two settings are the two representative extremes between adapting or not the network representation to the new domain, indeed using learning rates above 10^{-3} for internal layers prevents the fine-tuning from converging. Other parameters are left to their original value (e.g. momentum is 0.9, weight decay is 0.0005, dropout ratio is 0.5 for all layers from *fc7* and below).

Preprocessing, Training and Prediction. Regarding image pre-processing, we kept the same protocol adopted for training *CaffeNet* on ImageNet within *Caffe* [22]: incoming images are resized to be 256×256 and the mean image of the training set is removed.

For training, a random 227×227 crop is extracted from each image (randomly mirrored horizontally) before passing it to the learning algorithm. The batch size for back-propagation iterations was set to 256. Note that the training set was shuffled before starting the fine-tuning process since we observed that similarity of images within a batch negatively affects convergence of the back-propagation algorithm. When fine-tuning a network, we evaluated the model's performance on a validation set every epoch and we finally chose the epoch achieving highest validation accuracy. We repeated this process multiple times to account for statistical variability and chose the model providing the highest validation accuracy.

Following [22], at test phase the prediction over one image was taken as the average prediction score over 10 crops extracted from the center and the corners if the image, together with their mirrored versions.

IV. EXPERIMENTAL ANALYSIS

In this section we report our experimental analysis on the problem of learning from few examples using a pre-trained CNN. Due to the close connection between generalization and invariance, we first investigate the effect of fine-tuning on the generalization properties of a network (Sec. IV-A), then, based on our findings, we proceed to assess the ability of tuned architectures to provide invariant representations within the problem of object identification (Sec. IV-B and IV-C). In particular, we approach the problem of generalizing the visual appearance of several objects undergoing different transformations from only a few example images.

Our analysis allows, on the one hand, to better understand how different choices of fine-tuning strategies affect the overall recognition capabilities of a CNN, and, on the other hand, evaluates a protocol to build invariant representations for learning from few examples.

All our experiments are performed on the *iCubWorld* (*iCW*) dataset.

A. Fine-tuning and Invariance

We begin by evaluating how the choice of the dataset used for fine-tuning can impact the overall generalization capabilities of a network.

[1] https://robotology.github.io/iCubWorld/

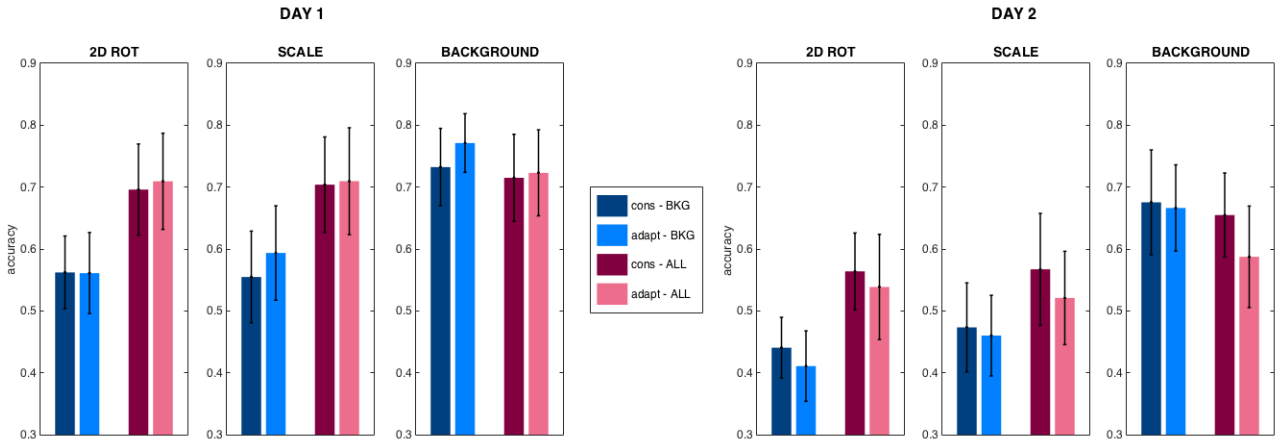

Fig. 3. Classification accuracy of *conservative* and adaptive architectures on *Background* (BKG, blue) and all transformations (ALL, red). Results are reported separately for the *Background*, *Rotation* and *Scale* transformations on *Day 1* and *Day 2*. Bars report average accuracy on 10 trials.

To this end, we compare the classification accuracy of *CaffeNet* fine-tuned over two different datasets: one comprising only the *Background* (BKG) transformation and another one comprising *Background*, *Rotation* and *Scale* (ALL) transformations. In order to fairly compare the two, we randomly sub-sampled this latter dataset to maintain the same size as the first one. Moreover, we fine-tuned the network on each dataset according to the *adaptive* or *conservative* strategies described in Sec. III-B.

For this preliminary analysis we focused on the classical application scenario of deep CNNs, object categorization, postponing our study on object identification to the following sections. Specifically, we consider the problem of discriminating between the 15 categories contained in the dataset introduced in Sec. III-A. Regarding model-selection: from the set of 10 object instances available per category, we used 7 objects per category for training, 2 for validation, and tested classification prediction on the remaining object. To account for possible biases across days (e.g. similar lighting, background, etc.) we performed fine-tuning and validation on *Day* 1 while tested on both *Day* 1 and 2. We repeated this experiment 10 times to allow for every object instance per category to be tested on at least once.

Fig. 3 reports the classification accuracy of the four networks fine-tuned according to the *adaptive* or *conservative* strategy on the BKG or ALL transformations.

A first general observation is that, regardless of the invariance learned on ImageNet, both fine-tuning strategies on both datasets exhibit a remarkable performance drop on *Day* 2. More precisely, the *adaptive* strategy (light blue/red) provides the largest gap between the two days, achieving comparable performance to the *conservative* strategy (dark blue/red) on *Day* 1, but being constantly worse on *Day* 2.

A second observation is that both fine-tuning strategies seem to clearly benefit from having access to a richer set of visual transformations. Indeed, networks fine-tuned on the mixed dataset (ALL, red) outperform by more than $\sim 10\%$ of accuracy, on *Scale* and *Rotation*, networks trained only on the BKG dataset (blue). Moreover, they exhibit comparable performance to those trained on BKG also when tested on this transformation itself, suggesting that reducing the number of examples undergoing this specific transformation (as it happens in the mixed dataset, since the two have same size) does not have a disruptive effect on performance.

We repeated the experiment by training only on *Rotation* or *Scale* and observed similar patterns. These results overall warn about fine-tuning deep networks in robotics settings on specific conditions, considering the risk of observing degraded performance when these slightly change. Since this issue is particularly relevant to the robotics context, in the following we further investigate the effect of adapting the network internal representation to the visual transformations characterizing the new domain. To this end, we compare network representations, tuned or not on the *iCW* dataset, for solving object identification tasks.

B. Invariance of the Network Internal Representation

To better understand the effect of fine-tuning on the internal layers of a CNN, we start visualizing them by applying dimensionality reduction techniques. We focus on the models trained on all transformations in Sec. IV-A, since we observed that tuning on a specific transformation strongly degrades performance on the others. In particular, we want to investigate to which extent fine-tuning can be beneficial to improve the network representation invariance to the experienced visual transformations.

We used the t-SNE approach [29], to qualitatively and quantitatively compare representations extracted from a network fine-tuned according to the *adaptive* strategy on the categorization task of Sec. IV-A and from *CaffeNet* original model. We consider the *fc7* layer (see Sec. III-B), i.e., the last one before the output layer (*fc8*). Indeed, we remark that the *conservative* tuning strategy does not change layers up to *fc7*. For the fine-tuned model we chose one random trial among the 10 performed in the previous experiment.

With t-SNE we computed the 2-dimensional embedding of extracted representations of the 15 objects belonging to

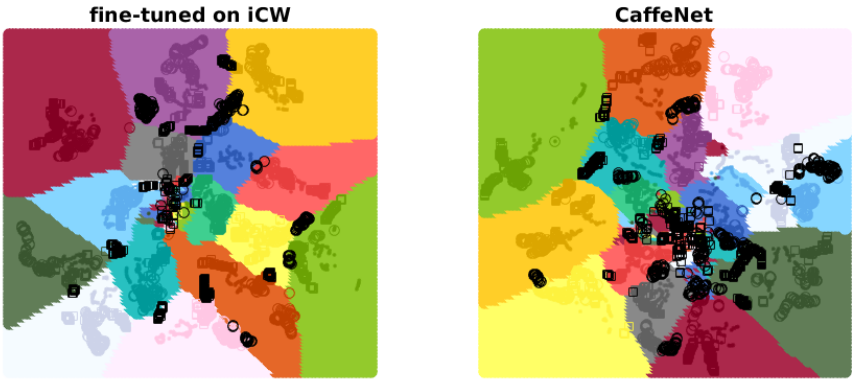

Fig. 4. 2-dimensional embedding of $fc7$ representations of 15 objects, extracted from one network tuned on *iCW* (Sec. IV-B) and from original *CaffeNet*. We report regions predicted by a kNN classifier trained only on the dot markers (*Background* transformation) to discriminate the objects when undergoing *Scale* (square markers) or *Rotation* (circles) transformations. Black markers denote misclassifications.

the test set of the previous experiment. We then performed a *k-Nearest Neighbor (kNN)* object identification experiment on the $2D$ projections of these 15 objects, considering only points from the *Background* set of *Day* 1 for training and testing on *Scale* and *Rotation* sets of *Day* 1 and 2. Fig. 4 represents the t-SNE embedding for the representations coming from the tuned network and *CaffeNet* original model. In particular, we report the regions learned by the kNN classifier from the training points (dot markers) and the test points of *Day* 1 (square markers for *Scale* and circles for *Rotation*). Black markers denote the misclassified points.

The classification accuracy associated to this proof-of-concept experiment confirms what is visually evident, i.e., adapted features outperform by more than 10% off-the-shelf features, achieving a recognition accuracy of 80% vs 68%. We conclude that fine-tuning is indeed able to produce a more invariant representation with respect to the experienced transformations. This wasn't evident from Fig. 3.

C. Object Identification from Few Example Images

The results observed in Sec. IV-B seem to indicate that the internal representation of the tuned *CaffeNet* is more robust to the nuisances affecting the *iCW* dataset. In order to quantify such invariance, in this section we perform extensive tests, considering challenging object identification scenarios where very few example images are provided during training.

We compare again the representations from the original *CaffeNet* model and from models fine-tuned according to the *adaptive* strategy on all transformations of the *iCW* dataset. Analogously to Sec. IV-B, we extract image representations from the *fc7* feature layer. We then train an RLS classifier (as described in Sec. II-B) to the task of object identification on the *Background* dataset of *Day* 1, testing on the remaining two transformations on both days. For the RLS classifier we employed the GURLS library [30], which we used to train a *kernel regularized least squares* classifier with Gaussian kernel. In the following, we refer to *ImageNet+RLS* and *iCW+RLS* to denote classifiers trained respectively on features produced by a *CaffeNet* trained on ImageNet or fine-tuned on *iCW*. We also considered replacing the RLS classifier with another stage of fine-tuning. We tried either the *conservative* or *adaptive* strategy, starting either from the original *CaffeNet* or the model previously tuned on a subset of *iCW*.

Small-scale Object Identification. In this first experiment we carry out a small-scale instance recognition problem, where, similarly to Sec. IV-B, the system is asked to discriminate among 15 objects each belonging to a different category of the *iCW* dataset.

Fig. 5 reports the classification accuracy of the considered methods compared on this task. To evaluate the dependency of different architectures with respect to the dataset size, we used for training respectively 10, 50 or all available frames per object from the *Background* transformation of *Day* 1 (around 150 frames per object). We report the classification accuracy averaged over the *Scale* and *Rotation* transformations separately for the two days; we averaged performance over 10 trials, each one considering the 15 objects that are not used for fine-tuning (see Sec. IV-A).

A first observation is that *iCW + RLS* consistently outperforms all competitors when the number of examples is low. This can be explained noting that, overall, approaches using tuned representations achieve higher accuracies than the ones using off-the-shelf representations, suggesting that the representation produced *CaffeNet* fine-tuned on *iCW* is actually able to adapt to the transformations occurring in the dataset. A second observation is that, while fine-tuning achieves comparable or slightly superior performance to kernel methods when more frames are provided, kernel methods are more stable when the training set is small. This suggest that kernel methods are generally more suited to operate in settings where new objects must be learned on-the-fly from few example images.

Medium-scale Object identification. We extend our analysis to a 50-classes identification scenario, over all (10) object instances from 5 categories in the *iCW* dataset (namely the

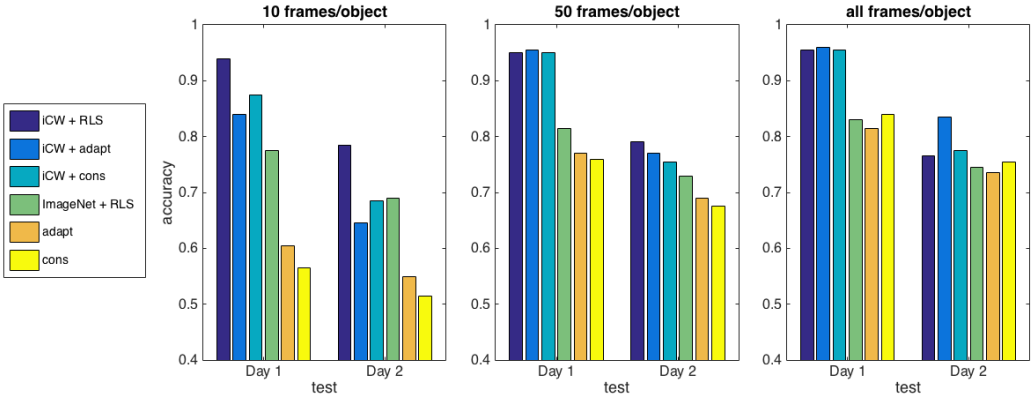

Fig. 5. Comparison of different approaches to object identification. We trained on 10, 50, or all available frames per object, using only the *Background* transformation in *Day* 1, and tested on the remaining two transformations on both days. Bar plots report average accuracy averaged over 10 trials.

book, flower, glass, hairbrush and hair clip). We first fine-tuned CAFFENET for categorization on the remaining 10 categories, according to the same protocol as in Sec. IV-A (7 objects per category for training, 2 for validation, repeating for 10 trials). Then, we used the tuned networks to provide $fc7$ features for the 50-classes identification task. In this last test we are focused on extending our approach to identify objects from few examples, therefore we provide only 10 frames per object at training time (from the *Background* set) and consider only kernel methods over network representations. We note that in this case, the considered 5 categories are neither part of *ImageNet*, nor part of the *iCW* subset used for fine-tuning, hence the networks never experienced them before the identification task.

Table II reports the average classification accuracy over *Scale* and *Rotation* for the two days separately. The first and second rows report respectively performance of RLS applied to off-the-shelf or tuned features according to the protocol described above. The other two rows report two alternatives that we tried when fine-tuning *CaffeNet* on the 10 categories. With *iCW day12 + RLS* we included both days in the fine-tuning training set. With *iCW id day12 + RLS* we changed the task according to which perform fine-tuning: instead of categorization, we fine-tuned to perform object identification over all 100 object instances (considering all transformations, both days, and leaving out a random 20% for validation).

We note that also in this setting fine-tuned features outperform *ImageNet + RLS* by a large margin. Moreover, fine-tuning on both days increases performance on *Day 2*, and fine-tuning to the identification task (*iCW id day12 + RLS*) seems to be the best approach. These results supports the previous observation that, by performing fine-tuning over some objects undergoing visual transformations, the network is indeed adapting previously learned invariances to the novel nuisances and that RLS is a sufficiently robust method to learn to exploit such invariances even from few examples.

The classification accuracy achieved by our approach is remarkable considering that *Background*, *Scale* and *Rotation* sets across days depict the object in very different conditions,as can be observed in Fig. 2, and that only 10 example images per object were provided.

TABLE II

50-OBJECTS IDENTIFICATION ACCURACY WHEN TRAINING RLS ON 10 IMAGES/OBJECT, WHOSE REPRESENTATIONS ARE PROVIDED BY MODELS TUNED ON A SUBSET OF *iCW* ACCORDING TO DIFFERENT STRATEGIES.

Architecture	Day 1	Day 2
ImageNet + RLS	0.68	0.61
iCW + RLS	0.82	0.72
iCW day12 + RLS	0.82	0.77
iCW id day12 + RLS	0.86	0.81

V. CONCLUSIONS AND ONGOING WORK

We addressed the problem of learning to recognize a wide range of objects from minimal amounts of data. Given the recent success of deep CNNs, we assessed the efficacy of different approaches to fine-tune and adapt pre-trained networks to novel tasks.

We considered a challenging scenario where a human supervisor provides only few glimpses of the objects of interest to the robot. In this context we collected and made available a dataset comprising images of 150 objects divided according to changes in $2D$ orientation, scale and background.

Our analysis shows that fine-tuning the internal layers of a network, adapts the invariance of the representation to the nuisances of the new dataset. By leveraging upon this effect, we tuned a deep *CNN* over multiple visual transformations to obtain a robust representation for object identification. Moreover, we observed that when only few training examples are provided per class, it is beneficial to combine such adapted representation with more robust learning approaches such as Regularized Least Squares classification or Support Vector Machines, using the fine-tuned network as a feature extractor.

Future work will focus on studying invariance to more challenging transformations such as $3D$ rotations for identification or intra-class variability for categorization, potentially exploiting more recent CNNs such as GoogleNet [2] or ResNets [31]. Moreover, we plan to investigate the role of

fine-tuning in a lifelong learning scenario, in order to determine whether such incremental process could be leveraged upon in order to further improve the representation learned by the network.

VI. Acknowledgments

The work described in this paper is supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216; and by FIRB project RBFR12M3AC, funded by the Italian Ministry of Education, University and Research. We gratefully acknowledge NVIDIA Corporation for the donation of the Tesla k40 GPU used for this research.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[6] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 681–687.

[7] J. Sung, S. H. Jin, I. Lenz, and A. Saxena, "Robobarista: Learning to Manipulate Novel Objects via Deep Multimodal Embedding," *ArXiv e-prints*, Jan. 2016.

[8] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3406–3413.

[9] N. Snderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford, "Place categorization and semantic mapping on a mobile robot," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5729–5736.

[10] N. Snderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 4297–4304.

[11] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," *ArXiv e-prints*, Dec. 2013.

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

[13] D. Held, S. Thrun, and S. Savarese, "Robust single-view instance recognition," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2152–2159.

[14] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0816-y

[16] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[17] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 1329–1335.

[18] M. D. Zeiler and R. Fergus, *Visualizing and Understanding Convolutional Networks*. Cham: Springer International Publishing, 2014, pp. 818–833. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10590-1_53

[19] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[20] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, T. Jebara and E. P. Xing, Eds. JMLR Workshop and Conference Proceedings, 2014, pp. 647–655. [Online]. Available: http://jmlr.org/proceedings/papers/v32/donahue14.pdf

[21] G. Pasquale, T. Mar, C. Ciliberto, L. A. Rosasco, and L. Natale, "Enabling depth-driven visual attention on the icub humanoid robot: instructions for use and new perspectives." *Frontiers in Robotics and AI*, vol. 3, no. 35, 2016. [Online]. Available: http://www.frontiersin.org/humanoid_robotics/10.3389/frobt.2016.00035/abstract

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia - MM '14*. ACM Press, 11 2014, pp. 675–678.

[23] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 5 2011, pp. 1817–1824. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5980382

[24] S. R. Fanello, C. Ciliberto, L. Natale, and G. Metta, "Weakly supervised strategies for natural object recognition in robotics," *IEEE International Conference on Robotics and Automation*, pp. 4223–4229, 5 2013.

[25] C. Ciliberto, S. R. Fanello, M. Santoro, L. Natale, G. Metta, and L. Rosasco, "On the impact of learning hierarchical representations for visual recognition in robotics," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 3759–3764.

[26] G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, and L. Natale, "Teaching iCub to recognize objects using deep Convolutional Neural Networks," vol. 43, 2015, pp. 21–25. [Online]. Available: http://www.jmlr.org/proceedings/papers/v43/pasquale15

[27] C. Ciliberto, U. Pattacini, L. Natale, F. Nori, and G. Metta, "Reexamining lucas-kanade method for real-time independent motion detection: Application to the icub humanoid robot," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 4154–4160.

[28] C. Ciliberto, S. R. Fanello, L. Natale, and G. Metta, "A heteroscedastic approach to independent motion detection for actuated visual sensors," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3907–3913.

[29] L. van der Maaten, "Accelerating t-sne using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, pp. 3221–3245, 2014. [Online]. Available: http://jmlr.org/papers/v15/vandermaaten14a.html

[30] A. Tacchetti, P. K. Mallapragada, M. Santoro, and L. Rosasco, "GURLS: A Least Squares Library for Supervised Learning," *Journal of Machine Learning Research*, vol. 14, pp. 3201–3205, 2013. [Online]. Available: http://jmlr.org/papers/v14/tacchetti13a.html

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.