2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

On the Impact of Learning Hierarchical Representations for Visual Recognition in Robotics

Carlo Ciliberto†, Sean Ryan Fanello†, Matteo Santoro‡, Lorenzo Natale†,
Giorgio Metta† and Lorenzo Rosasco‡

***Abstract*— Recent developments in learning sophisticated, hierarchical image representations have led to remarkable progress in the context of visual recognition. While these methods are becoming standard in modern computer vision systems, they are rarely adopted in robotics. The question arises of whether solutions, which have been primarily developed for image retrieval, can perform well in more dynamic and unstructured scenarios. In this paper we tackle this question performing an extensive evaluation of state of the art methods for visual recognition on a iCub robot. We consider the problem of classifying 15 different objects shown by a human demonstrator in a challenging Human-Robot Interaction scenario. The classification performance of hierarchical learning approaches are shown to outperform benchmark solutions based on local descriptors and template matching. Our results show that hierarchical learning systems are computationally efficient and can be used for real-time training and recognition of objects.**

I. Introduction

The problem of learning and designing effective visual representations has been recently subject of intense study both in computer vision and machine learning. In these contexts, hierarchical representations coupled with state of art supervised learning algorithms, have achieved remarkable performances in complex visual recognition tasks (see for example [26], [37]). Despite, the good results the application of these approaches in robotics is still limited. The goal of this paper is to assess the impact of learning hierarchical representations for visual recognition in robotics, since we believe that the community could benefit from a thorough evaluation of these methods.

Recently, there has been an increasing interest in robotics toward visual recognition problems as confirmed by the organization of several open challenges [40], [41], [42]. However, these problems have been typically considered as preliminary steps to more articulated tasks, e.g. navigation, manipulation, or other kinds of interaction [18], [3], [10]. As a consequence visual recognition solutions are part of complex systems that include many other components (e.g. pose estimation). Such systems require accurate, hence costly, supervision in the training phase (e.g. uncluttered views of the object [25] or meta-data about its position and orientation with respect to the camera [7]). Indeed, this has led to datasets acquired in highly controlled scenarios (see [40], [41], [42] and also [25]).

† iCub Facility, Istituto Italiano di Tecnologia {carlo.ciliberto,sean.fanello,lorenzo.natale, giorgio.metta} at iit.it. ‡ LCSL, Istituto Italiano di Tecnologia and Massachusetts Institute of Technology matteo.santoro at iit.it, rosasco at mit.edu

Fig. 1. The Human Robot Interaction setting for the data acquisition. Independent motion detection provides a good estimation for the object position in the image.

In this paper we focus on visual recognition in a Human-Robot Interaction scenario. While many data-sets (and benchmarks) are available in computer vision, they are often strongly biased [33] and the good results of hierarchical learning systems are not ensured to carry on to our setting. For these reasons we opted for acquiring a novel dataset reflecting the natural and dynamic setting of human-robot interaction. We considered a scenario where a human supervisor shows 15 different objects in front of the robot. The robot employs motion detection to actively track the objects with its gaze, observe them from multiple viewpoints and extract a subwindow to better localize them in the whole image. Hierarchical representations are then extracted for each image and a multi-class classifier is trained to recognize newly acquired images.

The main contributions of this work are three. First, we provide a rigorous evaluation of hierarchical image representations in a real robotics context, showing that these techniques consistently outperform local descriptors. Second, we observe that advanced supervised learning methods generalize better than template matching: this is a further advantage of compact image representations with respect to unordered sets of local features that cannot be fed directly to a classifier. Finally, we show that all these methods are computationally efficient and can be used for real-time training and recognition of objects. Byproducts of this work are 1) a dataset that we acquired for the experiments and made available for the community (see Sec. IV-A.1), and 2) a practical HRI scheme for low-effort visual data acquisition.

II. State of the Art

A first clear distinction between visual recognition methods can be made based on the application domain: in

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

computer vision the common task is image retrieval from Internet [33], whereas in robotics the variety of applications is incomparably larger and is harder to compare state of the art methods.

Currently, the most popular approach in recognition for robotics is to exploit 3D information to obtain invariant models of the observed scene. Most proposed methods build global topological representations of the objects that encode local geometric relations [11] or perform clustering to directly derive tridimensional templates from point clouds [1]. Systems that perform recognition based only on visual cues are typically employed to solve pose estimation problems [8], [32], [10], [21]. They often share the following core strategy: local features (e.g. SIFT [27]) are first extracted from raw images and then matched with a learned object template via robust outliers rejection schemes (RANSAC) [19].

In computer vision, the community has focused on designing or learning descriptive representations for the visual signal. This perspective finds its root in the Bag of Words (BOW) paradigm [20], whose principle is to capture statistically relevant properties of the image content. These methods, combined with the Spatial Pyramid Representation (SPR) [26], achieved good results on standard datasets (e.g. Caltech-101 [16], Pascal VOC [12]) and they were further extended by replacing vector quantization with a sparse coding step [39]. It has been observed that sparsity of the data representation improves the overall classification accuracy – see [22], [9] and references therein. Therefore, in the attempt to extend the successful framework in [39], many recent works have focused on finding novel dictionary learning algorithms [24], [15], designing mid-level features [4] or improving the pooling process [5], [23].

We are aware of very few works in Robotics in which authors fully exploited these newest algorithms and methods for hierarchical image representation techniques [18], [3].

III. Methods for Image Representation and Learning

In this Section we describe the typical pipeline, reporeted in Fig. 2, adopted to obtain hierarchical representations of an image. Most of these methods (Fig. 2(Top)) share a common low-level stage with typical robotics approaches (Fig. 2(Bottom)), namely the extraction of local features, while substantial differences arise in the subsequent phases. Indeed, hierarchical representation methods aim to capture the statistically relevant properties of the scene while template matching techniques organize the local keypoints in a robust geometrical model of the object of interest. In the end however, both approaches aim to obtain a visual representation which is invariant with respect to projective transformations. A training set of these invariant representations is then provided to a learning system that is in charge of identifying the correct object instance within test samples.

A. Image Representation

Hierarchical representations consist in a cascade of so-called coding and pooling stages. Coding performs signal reconstruction of the local image contents to enforce specific structures on the representation (e.g. sparsity [39], [15]); pooling combines neighboring codes into a single vector, acquiring robustness to small spatial deformations. Within this framework we can make a clear separation between matching techniques, which perform only the first local extraction to build the object model, and hierarchical methods, which arrange multiple coding/pooling layers subsequently.

1) Low-Level Descriptors: The low-level feature extraction process performs a local analysis of image patches. The outcome is a sequence of descriptors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^n$ encoding local responses to a predefined (or in some cases learned from data) set of filters. Common filters are image patches [17] of SIFT-like filters [27]. Often, descriptors are extracted from a dense regular grid on the image, following the study in [17]. However, when the spatial position is crucial for subsequent tasks (e.g. pose estimation), low-level features are extracted on geometrically characteristic locations - the so-called keypoints. These keypoints are used to build a robust 2D or 3D object model (e.g. [27], [8]).

2) Hierarchical Image Representation: Higher level representations are build on top of local descriptors. They usually require an initial unsupervised learning step that adapts the representation to the data. An overcomplete basis is learned from training descriptors and is organized in a Dictionary matrix $\mathbf{D} \in \mathbb{R}^{n \times K}$ (n feature size, K dictionary size). Given $\mathbf{D}$, a coding operator $g(\mathbf{D}, \mathbf{x}) = \mathbf{u}$, maps an input feature $\mathbf{x} \in \mathbb{R}^n$ into a new feature space $u \in \mathbb{R}^K$ (with $K > n$). Typically, coding operators share the goal of minimizing the reconstruction error between the input feature $\mathbf{x}$ and the signal reconstruction $\mathbf{Du}$

$$g(\mathbf{D}, \mathbf{x}) = \min_{\mathbf{u}} \|\mathbf{x} - \mathbf{Du}\|_F^2 + \lambda \mathrm{R}(\mathbf{u}) \quad \text{s.t.} \quad \mathrm{C}(\mathbf{u}) = 0 \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm. Coding methods differ in the regularization term $\mathrm{R}(\mathbf{u})$ and the constraints $\mathrm{C}(\mathbf{u})$. Examples are Vector Quantization (VQ) [26], Sparse Coding (SC) [39] and Locality-constrained Linear Coding (LLC) [37]. Popular methods for learning the dictionaries are K-Means [20], or Dictionary Learning techniques [39].

The output of the coding stage is a set of local coded descriptors $\mathbf{u}_1, \ldots, \mathbf{u}_t$. A pooling map allows to combine these separate information and take into account higher-level statistics of the image. This operator takes the codes located at S overlapping regions (e.g. cells of a spatial pyramid), and obtains a single succinct representation. The final descriptor $\mathbf{z} \in \mathbb{R}^{KN_c}$ of the image, consists in the concatenation of the descriptors obtained from each of these regions. The N_c regions are usually obtained by partitioning images in $2^l \times 2^l$ patches with scales $l = 0, 1, 2$. The most known hierarchical learning method, Bag of Words (BOW) [20], consists in the combination of VQ and average pooling, on top of a first layer of dense SIFT descriptors. On the other hand, it has been empirically observed that Sparse Coding (SC) favors max pooling over average pooling [4].

A different perspective is given by the HMAX framework, which is an algorithmic model of the recognition process

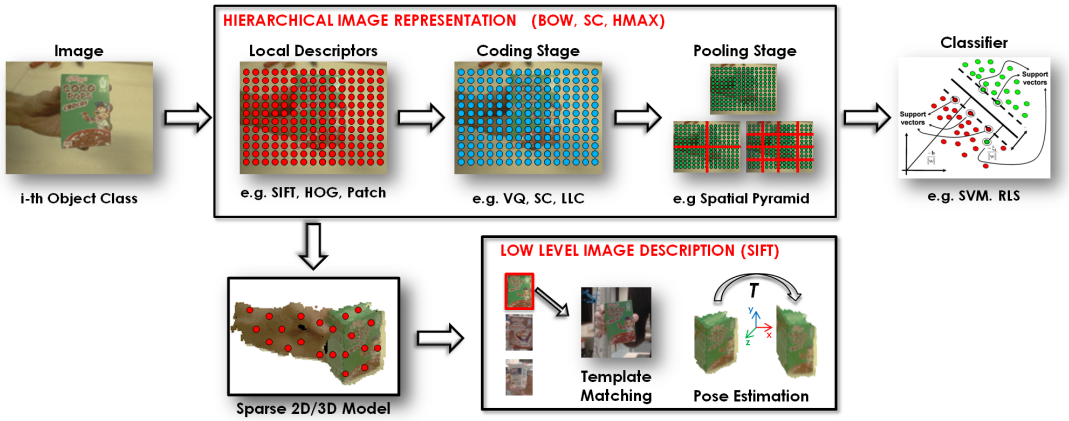

Fig. 2. General outline of object recognition approaches.(Top) Standard computer vision pipeline for hierarchical image representations extraction, (bottom) template matching and pose estimation.

in humans. HMAX retraces the human's ventral stream structure of simple and complex cells forming a hierarchy of alternating layers (see [30] for more details) that can be interpreted as a sequence of coding and pooling stages. First, a set of muti-scale Gabor filters with different orientations and bandwidths are convolved with the image,followed by a max-pooling step on multiple scales.

The second layer computes the similarity measure between responses of the first pooling layer and a dictionary of filter prototypes previously learned from data. Finally, the output of a spatial pooling operator over the entire image and over all scales is returned. In the end, the number of components of the feature vector is equal to the number of previously learned (or predefined) prototypes.

In this work we evaluated the performance of the Bag of Words, Sparse Coding (both using Spatial Pyramid Representations) and HMAX since we believe that they are representative of the current principal trends in computer vision.

B. Classification

After the representation stage, images are described by either a single hierarchical vector $\mathbf{z} \in \mathbb{R}^{KN_c}$ or a set of local features $\mathbf{x}_1, \ldots, \mathbf{x}_t$ ($\mathbf{x}_i \in \mathbb{R}^n$). Two possible strategies are commonly used in order to classify images: template matching or learning approaches. Machine learning methods tend to be more robust to intra-class variations, since they obtain the model from different instances of the same object; matching methods are more versatile since they do not require a batch training procedure. From the point of view of data representation, the first class of methods usually uses a single feature vector per image, whereas matching-based can work with both representations, although they are usually combined with local features.

1) Template Matching: Matching methods treat all the local descriptors extracted from different views of the object as belonging to one single object model [27], [8]. When a new test image is provided the best match among all the object databases and the current image represents the classification response. When additional information is provided (e.g object pose during the training phase, 3D model etc.) robust outlier rejection schemes (e.g. RANSAC) improve the recognition rate. More specifically in this work we build a SIFT database for each object, then all descriptors in a new image were tested with the database. Following the indications in [27], [8] the matches computation is carried out with an approximation of K-NN algorithm, called Best-Bin-First (BBF) [2] checking the first 200 nearest-neighbor candidates. We discard matches when the second closest neighbor is coming from a different object then the first and the ratio of the two distances is greater than 80%.

2) Learning Methods: hierarchical representations are usually combined with more sophisticated learning approaches: the single descriptor is fed to a supervised classifier. Codes obtained through vector quantization usually require ad-hoc kernels to obtain good performances, instead, sparse coding approaches have shown to be effective if combined with linear classifiers, also ensuring real-time performances [39]. In our experiments we used only linear kernels since non-linear approaches are slower and less suited for real-time applications . In our analysis we tested both SVM [35] and RLS. The first one is nowadays a consolidate approach in both robotics and computer vision communities and often it is used as baseline for new learning methods. RLS-based techniques, instead, are among the simplest algorithms for many learning tasks and yet they have been reported to consistently achieve state of the art performances in many contexts[29].

IV. Experiments

A. Setting

We designed a challenging scenario for image acquisition and training. The result is a dynamical Human Robot Interaction (HRI) schema that imposes strong limits on human supervision, similarly to [14]. In our setting, the robot has to learn the visual appearance of different objects presented by a human demonstrator (see Fig. 1) for a very short time interval. At the beginning of each session, the human stands

Fig. 3. Sample images of the 15 objects collected following the procedure described in Sec. IV-A

approximately 1.5 meters from the iCub humanoid robot [34], and pronounces the name of the object in his own hand, showing it to the robot from multiple points of view.

After this initial training phase, which lasts approximately 10 seconds per class, the demonstrator shows again one of the presented objects asking the system to identify it. In order to actively track the object during demonstration, the system relies on an independent motion detector [6] that provides also a rough bounding box around the object of interest. For speech recognition, we used Microsoft Speech Recognition libraries.

1) Dataset: following the procedure described above, we acquired a dataset comprising the 15 objects depicted in Fig. 3. For each instance, a training and test sets were acquired directly from the iCub cameras during respectively 10 and 15 seconds of demonstration. We chose a sampling frequency of 15Hz for images of 640×480 pixels, obtaining a dataset of 2250 images (150 per object) for training and 3000 (200 per object) for test. The crop effected by the motion detector reduced the original images to windows of 160×160 pixels. The dataset was made publicly available for the community at the link `http://eris.liralab.it/download/iCub/datasets/iCubWorld_SingleInstance.zip`.

2) Implementation Details: we employed an open GPU version of SIFT extraction (see [38]) to compute low-level descriptors and implemented the further steps of Bag of Words (BOW) and Sparse Coding (SC) in C++ (publicly available on th iCub repository [34]). The MATLAB (+ GPU) implementation of the system described in [28] was used to obtain the HMAX codes. To perform k-Nearest Neighbors we chose the kd-Trees implementation in the VLFeat library [36], for SVM we used LIBLINEAR [13] while for RLS we relied on the GURLS library [31].

3) System Parameters: for template matching tests, a set of sparse SIFT was extracted at a 16×16 scale with the keypoints detector described in [27]. On the other hand, for BOW and SC we used an initial layer of SIFT features extracted from a dense grid on the image. Points were located every 8 pixels, SIFT descriptor scales were set to 16×16 pixels. In both cases (dense and sparse) we tried multiple scales, but we did not find any benefit during recognition. The dictionary size was fixed to 1024 and the pyramid levels is set to 3 for BOW and SC while for HMAX that does not employ pyramidal pooling a dictionary of 4096 features was employed. Larger dictionary sizes did not led to higher results.

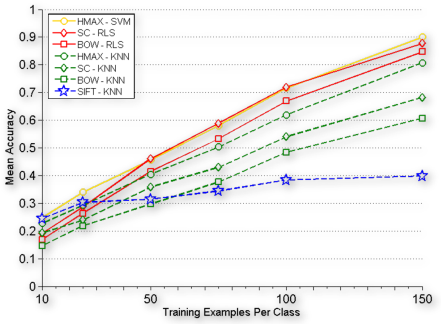

Fig. 4. Recognition performance of the methods considered with respect to increasing number of training examples. Hierarchical descriptors need only few examples per class to outperform local approaches. Statistical learning methods (solid lines) exhibit consistently higher results compared to template matching techniques (dashed lines).

B. Benchmark

The first set of experiments we performed were aimed at comparing the classification performance of both local and hierarchical image representation approaches. The results confirm that object recognition pipelines described in Section III may improve the visual recognition performance of robots. In Tab. I we report the results obtained with the methods described in Section III using all the 2250 (150 per class) training images.

The use of the same learning method, namely the Nearest Neighbor, on the top of a hierarchical representation leads to better classification performance with respect to raw SIFT matching. This evidence clearly supports the use of hierarchical representations, which encode more global image information than local descriptors.

We can also observe that machine learning methods present remarkable benefits over template matching. Although not surprising, it has to be pointed out that in the case of local features, such methods cannot be employed directly to a classifier. This represent a further advantage of having a compact representations of the whole image in a single vectors.

	k-NN (%)	RLS (%)	SVM (%)
☆ SIFT	39.9	-	-
□ BOW	60.6	84.7	83.6
◇ SC	68.2	87.7	86.6
○ HMAX	80.7	86.5	89.1

TABLE I

CLASSIFICATION ACCURACY AVERAGED OVER 15 CLASSES. 150 TRAINING EXAMPLES PER CLASS.

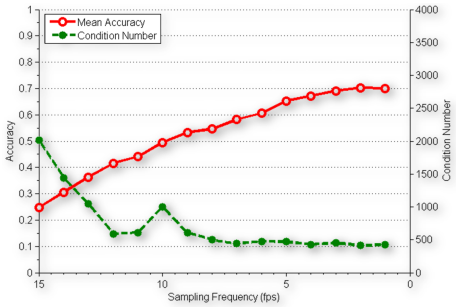

Fig. 5. Classification accuracy (red solid line) with respect to training sets of identical size (10 examples per class) collected with sampling frequencies varying between 15 Hz and 1 Hz. The generalization capabilities of the system increase dramatically when the redundancy (green dashed line) of the training data decreases. Feature: HMAX, Training method: SVM

One crucial aspect of supervised settings is the number of training examples necessary to build a robust visual model of the objects. In Fig. 4 we report the accuracy of the methods showed in Table I trained with increasing number of examples (from 10 to 150 images per class). For each representation we plotted the learning method with the highest classification performance.

Local descriptors better generalize when few examples are provided, however, hierarchical representations and learning methods outperform the competitors after $25-30$ examples per class (i.e. just 2 seconds of training). This suggests that high level features are able to capture large statistics of the visual world, but less efficient when only few observations are provided.

C. Accuracy-Redundancy Tradoff Analysis

In an ideal setting, learning methods require training and test data to be sampled i.i.d. from the same distribution. This assumption is clearly broken by the inherent heteroscedasticity of physical processes that often cause these two sets to result different. This fact is crucial in the setting described in Section IV-A, since training data are collected in a short interval (10 seconds) at a relatively high frequency (15 images per second). As a consequence the scene captured in training images does not change much from frame to frame leading to a redundant training set.

To better appreciate the impact of training set redundancy we compared the classification performance the system trained using sets of same size (10 image per class = 150 samples), but sampled at different frequencies from the

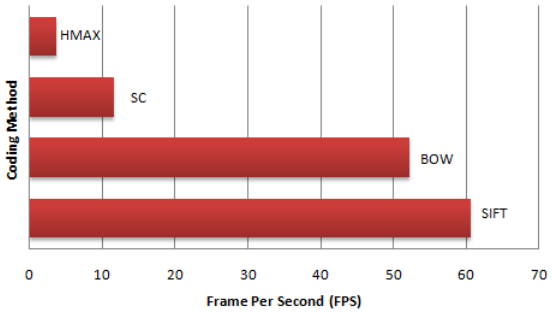

Fig. 6. Frames per second (FPS) of the evaluated coding methods.

original training set of 150 images per class. Following elementary statistical learning principles, we measured the redundancy of each training set with the condition number of the similarity matrix of the training points (which corresponds to the linear kernel matrix used by SVM and RLS). In Fig. 5 is reported (solid red) the classification accuracy with respect to this rough measure of training set redundancy (dashed green). Remarkably, when redundancy decreases enough the system achieves performances comparable to those obtained using the whole 150 examples. This analysis highlights strong connections with regularization theory: slower sampling frequencies improve the system generalization performance when training samples are lacking.

This experiment can be interpreted as a tradeoff analysis between the accuracy of a system trained on a given set of examples and the effort in collecting such data points. Clearly, sets acquired in less controlled settings are likely to be redundant. Therefore what is saved in acquisition effort is payed in accuracy or time required to achieve same classification performance. High accuracy can be achieved with few training samples, at the cost of much effort from the supervisor side (e.g. manually selecting the training set). The graph in Fig. 5 suggests that the training set acquisition strategy adopted should depend on the specific application considered.

D. Computational Efficiency

We compared the computational performances of the evaluated methods on a 2.4Ghz Core 2 Duo Processor and reported them in Fig. 6. The figure shows the clear tradeoff between accuracy and speed: the combination of SIFT and matching algorithm is, as expected, the fastest one even when the training set is large, but as observed it leads to limited learning performance. BOW is probably the best choice when computational efficiency is a priority while SC and HMAX, being relatively slower, are suited for applications that have strong accuracy requirements but are less restrictive on system efficiency.

V. DISCUSSION

Our study was motivated by the observation that: 1) despite the progress in machine learning and computer vision, visual recognition solutions are often limited to simple local

descriptors together with template matching classifiers; 2) since visual recognition problems are often considered in the context of more complex tasks, there is a lack of benchmark results for object recognition in plausible robotics scenarios. Given the above premises in this paper we have extensively evaluated hierarchical learning approaches to object recognition within a challenging human-robot interaction setting. In particular,

- We shown an exhaustive evaluation of state-of-the-art methods for visual recognition tasks, showing their benefits and accuracy.
- We proved that these methods are also suitable for real-time tasks and they do not need expensive training phases.
- We selected a dynamic HRI scenario, where we conducted all the experiments. This scenario can be used as general scheme do acquire data for visual recognition purposes. A byproduct of this study is a dataset which is publicly available for the community.

The presented analysis offers empirical evidence of the benefits that hierarchical learning representations can provide. We restricted our evaluation to the purely visual tasks, the next steps will require to investigate possible uses of these approaches in more articulated task such as grasp or manipulation.

VI. Acknowledgments

This work was supported by the European FP7 ICT projects N. 270490 (EFAA), N. 270273 (Xperience), N. 288382 (Poeticon++) and FIRB project RBFR12M3AC.

References

[1] J. Aleotti, D. Lodi Rizzini, and S. Caselli. Object categorization and grasping by parts from range scan data. In *ICRA*, 2012.

[2] J.S. Beis and D.G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *CVPR*, 1997.

[3] T. Botterill, S. Mills, and R. Green. Speeded-up bag-of-words algorithm for robot localisation through scene recognition. In *IVCNZ*, 2008.

[4] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, 2010.

[5] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun. Ask the locals: multi-way local pooling for image recognition. In *ICCV*, 2011.

[6] 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 *IROS*, 2011.

[7] A. Collet, D. Berenson, S. Srinivasa, and D. Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA '09)*, May 2009.

[8] A. Collet, M. Manuel, and S. Srinivasa. The MOPED framework: Object Recognition and Pose Estimation for Manipulation. *The International Journal of Robotics Research*, 2011.

[9] A. Destrero, C. De Mol, F. Odone, and Verri A. A sparsity-enforcing method for learning face features. *IP*, 18:188–201, 2009.

[10] S. Ekvall, D. Kragic, and F. Hoffmann. Object recognition and pose estimation using color cooccurrence histograms and geometric modeling. In *Image Vision Computing*, 2003.

[11] K. Eunyoung and G. Medioni. 3d object recognition in range images using visibility context. In *IROS*, 2011.

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

[13] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin. Liblinear: A library for large linear classification. *JMRL*, 2008.

[14] S.R. Fanello, C. Ciliberto, L. Natale, and G. Metta. Weakly supervised strategies for natural object recognition in robotics. *ICRA*, 2013.

[15] S.R. Fanello, N. Noceti, G. Metta, and F. Odone. Multi-class image classification: Sparsity does it better. *VISAPP*, 2013.

[16] L Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVPRW*, 2004.

[17] L. Fei-fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, pages 524–531, 2005.

[18] D. Filliat. A visual bag of words method for interactive qualitative localization and mapping. In *ICRA*, 2007.

[19] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981.

[20] C. Gabriella, R.D. Christopher, F. Lixin, W. Jutta, and B. Cdric. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[21] I. Gordon and D.G. Lowe. What and where: 3d object recognition with accurate pose. In *Lecture Notes in Computer Science*, 2006.

[22] K. Huang and S. Aviyente. Wavelet feature selection for image classification. *IP*, 17:1709–1720, 2008.

[23] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *CVPR*, pages 3370–3377, 2012.

[24] S. Kong and D. Wang. A dictionary learning approach for classification: separating the particularity and the commonality. In *ECCV*, 2012.

[25] K. Lai, L. Bo, X. Ren, and D. Fox. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In *IEEE International Conference on on Robotics and Automation*, 2011.

[26] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, volume 2, pages 2169–2178, 2006.

[27] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[28] Jim Mutch, Ulf Knoblich, and Tomaso Poggio. CNS: a GPU-based framework for simulating cortically-organized networks. Technical Report MIT-CSAIL-TR-2010-013 / CBCL-286, Massachusetts Institute of Technology, 2010.

[29] R. Rifkin. Everything old is new again: a fresh look at historical approaches in machine learning. In *Ph.D. dissertation, MIT*, 2002.

[30] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *PAMI*, 2007.

[31] A. Tacchetti, P. Mallapragada, M. Santoro, and L. Rosasco. Gurls: a toolbox for large scale multiclass learning. In *NIPS workshop on parallel and large-scale machine learning*, 2011.

[32] G. Taylor and L. Kleeman. Fusion of multimodal visual cues for model-based object tracking. In *ACRA*, 2003.

[33] A. Torralba and A.A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[34] icub repository. `http://eris.liralab.it/iCub/main/dox/html/index.html`.

[35] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc., 1998.

[36] A. Vedaldi and B. Fulkerson. Vlfeat: an open and portable library of computer vision algorithms. In *Iinternational Conference on Multimedia*, 2010.

[37] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[38] C. Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). `http://cs.unc.edu/~ccwu/siftgpu`, 2007.

[39] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[40] Solutions in perception challenge. `http://solutionsinperception.org/index.html`.

[41] Mobile manipulation challenge. `http://mobilemanipulationchallenge.org`.

[42] Robotcup@home. `www.robocupathome.org`.