Dictionary based Pooling for Object Categorization

Sean Ryan Fanello[1,2], Nicoletta Noceti[2], Giorgio Metta[1] and Francesca Odone[2]
[1]*iCub Facility, Istituto Italiano di Tecnologia, Via Morego 30, 16163, GE, Italia*
[2]*DIBRIS, Università degli Studi di Genova, Via Dodecaneso 35, 16146, GE, Italia*

Keywords: Dictionary based Image Pooling, Sparse Representation, Object Categorization, iCub, iCubWorld Data-Set.

Abstract: It is well known that image representations learned through ad-hoc dictionaries improve the overall results in object categorization problems. Following the widely accepted coding-pooling visual recognition pipeline, these representations are often tightly coupled with a coding stage. In this paper we show how to exploit ad-hoc representations both within the coding and the pooling phases. We learn a dictionary for each object class and then use local descriptors encoded with the learned atoms to guide the pooling operator. We exhaustively evaluate the proposed approach in both single instance object recognition and object categorization problems. From the applications standpoint we consider a classical image retrieval scenario with the Caltech 101, as well as a typical robot vision task with data acquired by the iCub humanoid robot.

1 INTRODUCTION

If, from a methodological point of view, image categorization is considered by many the very essence of computer vision, its applicative aspects are equally important. The possible application domains are countless and include industry, communications, entertainment, robotics, just to name a few. Not only object categorization is one of the hardest tasks of artificial intelligence, but also, in domains such automation and cognitive robotics, visual recognition is a cornerstone of very complex systems that include many other components — pose estimation, grasp, manipulation (Collet et al., 2011; Taylor and Kleeman, 2003; Ekvall et al., 2003; Gordon and Lowe, 2006). For these reasons, in the last decades the problem of designing effective visual representations for classification tasks has been given considerable attention. Since it is nowadays acknowledged recognition algorithms can be more effectively trained from examples than programmed, visual recognition has been tackled by both the computer vision and machine learning communities.

An important result of this joint effort are the so-called *hierarchical representations* which achieve remarkable performances in complex visual recognition tasks once they are used in combination with supervised learning algorithms – see for example (Lazebnik et al., 2006; Wang et al., 2010). Despite the good results obtained on benchmark and challenges, the application of these approaches to real scenarios is still limited. The goal of this paper is to propose an effective image representation pipeline which is able to generalize to different contexts: from common computer vision datasets oriented to image retrieval, e.g. Caltech-101 (Fei-Fei et al., 2004), to real Human-Robot Interaction (HRI) scenarios (Fanello et al., 2013a).

A very influential method for representing the image content is the so-called Bag of Words (BoW) paradigm (Csurka et al., 2004) (also referred to as Bag of Keypoints) based on a vector quantization of local keypoints. This approach has been extended by the work of Lazebnik et al. (Lazebnik et al., 2006), which introduces the *Spatial Pyramid Representation* (SPR) to preserve the spatial configuration in images, and leads to a very popular framework within the visual recognition community.

In classification tasks, it is well known that the sparsity of data representations improves the overall classification accuracy (Fanello et al., 2013c; Viola and Jones, 2004; Huang and Aviyente, 2008; Destrero et al., 2009), therefore Yang et al. (Yang et al., 2009) improve the spatial pyramid pipeline by replacing the vector quantization procedure with a sparse coding step. Different extensions to (Yang et al., 2009) have been proposed in the recent literature (Boureau et al., 2010; Boureau et al., 2011; Feng et al., 2011; Jia et al., 2012; Russakovsky et al., 2012; Chen et al., 2012) — all these methods being based on an unsupervised

Fanello S., Noceti N., Metta G. and Odone F..
Dictionary based Pooling for Object Categorization.
DOI: 10.5220/0004654602690274
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 269-274
ISBN: 978-989-758-004-8

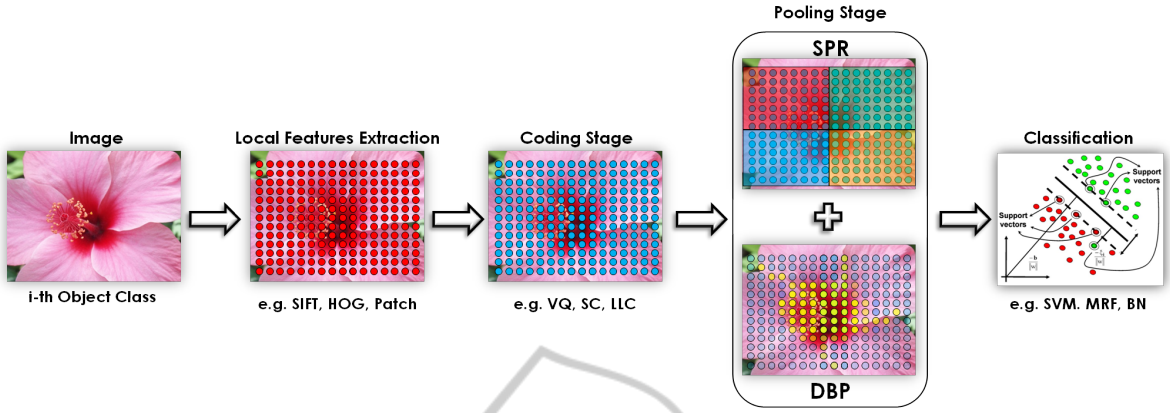

Figure 1: General pipeline for a visual recognition system. We contribute to the Pooling Stage, where we apply a Dictionary Based Pooling (DBP) operator.

learning of an ad hoc dictionary of atoms.
A recent line of research showed how dictionaries learned according to discriminative strategies may produce very effective image representations and should be used if labeled data are available. In (Kong and Wang, 2012; Fanello et al., 2013c) the discriminative strategies involve the coding stage of the pipeline. In this paper we show that discriminative dictionaries can be employed also during the pooling stage, yielding to image representations with an increased discriminative power. We start off from a low-level set of feature descriptors and we learn ad-hoc dictionaries in a discriminative manner. Then, we use these dictionaries to identify the region of the images belonging to a particular class of objects. The regions are then pooled together in order to obtain a compact and meaningful descriptor of the image.
The rest of the paper is organized as follows: in Section 2 we review the background of our work. In Section 3 we describe the method we propose; experiments, results and applications are presented in Section 4, while Section 5 is left to a final discussion.

2 BACKGROUND

In this section we review a classification pipeline commonly used in literature for multi-class image recognition (Lazebnik et al., 2006; Yang et al., 2009; Boureau et al., 2011). This will set the basis to discuss the contributions of our approach.

2.1 Visual Recognition Pipeline

A general visual recognition pipeline based on the use of coding and pooling techniques can be divided in four main stages, as depicted in Fig. 1:

Local Features Extraction. The input image is first described with a set of local features $\{\mathbf{x}_i\}_{i=1}^{M}$. Very popular examples are image patches, SIFT (Lowe, 2004), or SURF (Bay et al., 2008) (either sparse or dense). Taking inspiration from (Fei-fei and Perona, 2005), in this work we compute SIFT descriptors on a regular grid of image locations, thus each image is represented with M descriptors $\mathbf{x}_i \in \mathbb{R}^d$, with $d = 128$.

Feature Coding. It is based on the use of a fixed or data-driven dictionary D of K atoms. The goal is to associate each image feature $\mathbf{x}_i \in \mathbb{R}^d$ with a code $\mathbf{u}_i \in \mathbb{R}^K$ estimated as:

$$\begin{aligned} \mathbf{u}_i = \arg\min_{\mathbf{u}} \|\mathbf{x}_i - \mathbf{D}\mathbf{u}\|_F^2 + \lambda \mathrm{R}(\mathbf{u}) \\ \text{s.t.} \quad \mathrm{C}(\mathbf{u}) \end{aligned} \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm, and $\mathbf{C}$ is a (possible) constraint. Vector Quantization (VQ) (Lazebnik et al., 2006), Sparse Coding (SC) (Yang et al., 2009) and Locality-constrained Linear Coding (LLC) (Wang et al., 2010) are popular examples of coding methods, that mainly differ in the choice of regularization term $\mathrm{R}(\mathbf{u})$ and constraints $\mathrm{C}(\mathbf{u})$. Following (Fanello et al., 2013c), in this work we use Sparse Coding with ad-hoc dictionaries learned from the data (Sec. 2.2).

Feature Pooling. A common approach to overcome the locality of codes $\mathbf{u}_i$ relies on the definition of a pooling operator g that combines the contributions of multiple image locations. Often, this operator takes the codes located at S overlapping regions (e.g. cells of the spatial pyramid), and for each region pools the information in a single vector $\phi_s \in R^K$, $\phi_s = g_{(i \in Y_s)}(\mathbf{u}_i)$, where Y_s denotes the set of locations within the region s. The image is finally represented

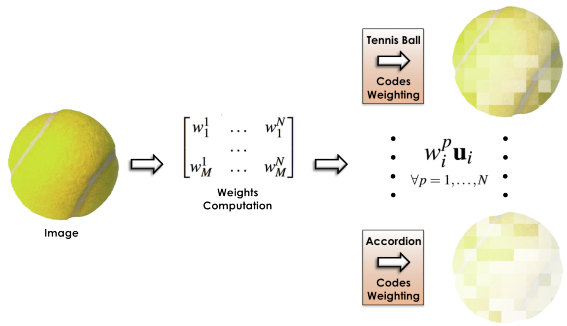

Figure 2: Visual intuition of the Dictionary Based Pooling operator. For each image we compute the weights of the N classes related to each code $\mathbf{u}_i$, $i = 1,\ldots,M$. All the codes are weighted according to the considered class. The max pooling operator will select only relevant features for the considered image.

with a descriptor $\mathbf{z_s} \in \mathbb{R}^{K\times S}$ which is the concatenation of all ϕ_s. Examples of popular pooling operators are *average pooling* and *max pooling*. In this paper, we propose instead the use of a pooling operator which is guided by the discriminative dictionaries (Sec. 3).

Image Classification. The image descriptor is the input of a final classification step. It has been shown that sparse coding is very effective if combined with a linear classifier (Yang et al., 2009), leading to computationally efficient approaches. In what follows, we adopt a linear Support Vector Machines (Vapnik, 1998) following a one-vs-all strategy.

2.2 Discriminative Adaptive Sparse Coding

Our approach to sparse coding relies on learning discriminative dictionaries. We follow the method proposed in (Fanello et al., 2013c). In the remainder of this section we briefly recall the procedure, referring the interested reader to (Fanello et al., 2013c) for further details.
Let us consider a multi-class problem with N classes (objects) and let $\mathbf{X}^p = [\mathbf{x}_1,\ldots,\mathbf{x}_{m^p}]$ be the $d \times m^p$ matrix whose columns are the training vectors of the $p-th$ class. Also, let us define $\overline{\mathbf{X}}^p = [\mathbf{X}^1,\mathbf{X}^2,\ldots,\mathbf{X}^{p-1},\mathbf{X}^{p+1},\ldots,\mathbf{X}^N]$ to be the concatenation of the training matrices of all other classes $q \neq p$. Dictionary learning is based on the minimization of the functional:

$$E = ||\mathbf{X}^p - \mathbf{D}^p\mathbf{U}^p||_F^2 + ||\overline{\mathbf{X}}^p - \mathbf{D}^p\overline{\mathbf{U}}^p||_F^2 + \\ + \lambda||\mathbf{U}^p||_1 + \mu||\overline{\mathbf{U}}^p||_2 \quad (2)$$

with respect to $\mathbf{D}^p, \mathbf{U}^p$ and $\overline{\mathbf{U}}^p$. $\mathbf{D}^p$ is the $d \times K$ dictionary of class p, $\mathbf{U}^p \in \mathbb{R}^{K\times m^p}$ is the codes matrix of class p, while $\overline{\mathbf{U}}^p \in \mathbb{R}^{K\times \overline{m}^p}$, with $\overline{m}^p = \sum_{q=1}^N m^q$, $q \neq p$, are the coefficients related to all other classes. In essence, when learning the dictionary of class p, features belonging to it are constrained to have a sparse representation thanks to the l_1-penalty term, while features of all other classes are forced to be associated with a more dense and smooth code vector. λ and μ are the regularization parameters allowing to control the importance of the two contributions.
As a consequence, features belonging to class p have a higher response if encoded with dictionary $\mathbf{D}^p$ rather than any other dictionary $\mathbf{D}^q$, $q \neq p$, leading to a very discriminative representation.

3 DICTIONARY BASED POOLING (DBP)

The use of feature dictionary is usually limited to the coding stage of the object classification pipeline. In this work, instead, we propose to extend their use to the pooling stage, exploiting their discriminative power.
Similarly to (Fanello et al., 2013c), we consider a global dictionary $\mathbf{D} = [\mathbf{D}^1,\ldots\mathbf{D}^N]$, of size $d \times K_G$, where $K_G = K \times N$, composed as the concatenation of all discriminative dictionaries previously computed. A feature $\mathbf{x} \in \mathbb{R}^d$ can be decomposed with respect to dictionary and codes as $\mathbf{x} \simeq \mathbf{Du}$, with $\mathbf{u}$ a K_G column vector.
We can interpret each element of code $\mathbf{u}$ as the relevance of each atom in the linear combination. Since we know the correspondence between atoms of the dictionary and classes, $\mathbf{u}$ can be seen as a concatenation of blocks, each one including the responses of a dictionary:

$$\mathbf{u}^{\mathrm{T}} = [(\mathbf{u}^1)^{\mathrm{T}},\ldots,(\mathbf{u}^N)^{\mathrm{T}}]; \quad (3)$$

where $\mathbf{u}^p$ is a K-ary vector representing the response of the p-th dictionary.
We evaluate the strength w^p of code $\mathbf{u}$ with respect to class p as

$$w^p(\mathbf{u}) = \sum_{j=1}^{K} |u(j)^p| \quad (4)$$

where $u(j)^p$ denotes the j-th element of codes block of class p.
As observed in the previous section, highest values in $\mathbf{u}$, and consequently in w, should directly denote a particular affinity with the corresponding class. We thus adopt these measures as weights within a pooling operator working on a partition of the codes space $\{X^p\}_{p=1}^N$, induced by the association of codes with

Figure 3: The iCubWorld 1.0 Dataset. Samples of the 7 classes collected for the *robot* (top strip) and *human* (bottom strip) datasets.

classes. Pooling is performed in each state of the partition according to the following

$$g_{(i\in X^p)}(\mathbf{u}_i) = \max_i (w_i^p(\mathbf{u}_i)\cdot\mathbf{u}_i) \quad \forall p = 1,\ldots,N \quad (5)$$

The weight w_i^p represents a confidence measuring how likely is that the code $\mathbf{u}_i$ has been observed in class p. Roughly speaking, they evaluate how much a given class is able to "see" in a particular image. Fig. 2 shows a visual representation of this principle. On the right, in particular, we report an image depicting a *tennis ball* as "seen" by its true class (above) and by the *accordion* class. Weights associated with the correct class are clearly higher.
For each image, we can finally build a representation $\mathbf{z_n} \in \mathbb{R}^{K_G\times N}$ that is the concatenation of all the weighted responses followed by the max pooling operator.

Combining the Spatial Layout and the Dictionary based Pooling. The spatial pyramid representation leads to an image descriptor $\mathbf{z_s} \in \mathbb{R}^{K_G\times S}$, with S the number of the pyramid cells (see Sec. 2.1), while the proposed DBP generates a descriptor $\mathbf{z_n} \in \mathbb{R}^{K_G\times N}$. The final image representation $\mathbf{z}$ will be the concatenation of the two vectors: $\mathbf{z} = [\mathbf{z_s}, \mathbf{z_n}] \in \mathbb{R}^{K_G\times(S+N)}$. It is common practice to normalize the data before classification, and as a consequence the descriptors become more peaky around zero. It has been experimentally observed the benefit of using a power normalization (Perronnin et al., 2010). Each component of both $\mathbf{z_s}$ and $\mathbf{z_n}$ are exposed to the following power normalization:

$$\begin{aligned}\mathbf{z_s} &= sign(\mathbf{z_s})|\mathbf{z_s}|^\alpha \\ \mathbf{z_n} &= sign(\mathbf{z_n})|\mathbf{z_n}|^\alpha\end{aligned} \quad (6)$$

where $0 \leq \alpha \leq 1$, in our experiments we set $\alpha = 0.5$. This is basically an explicit mapping to another feature space, where the highest code responses have less impact in the descriptor.

4 EXPERIMENTS

In this section we validate the proposed dictionary based pooling method. We consider three datasets: iCub World 1.0, iCubWorld Categorization[1] and a subset of the Caltech-101 (Fei-Fei et al., 2004). We compare our approach with state of the art methods (Yang et al., 2009; Fanello et al., 2013c), with the goal of showing that our pooling stage can improve the overall performances. We denote with:

- **SC** the method in (Yang et al., 2009).
- **SC + DASC** the approach proposed in (Fanello et al., 2013c).
- **DBP** (SC + DASC + Dictionary Based Pooling) the method described in Sec. 3.

4.1 Implementation Details

We provide here the details concerning the system parameters. As for the local feature extraction, we extract fixed-scale SIFT on patches of size 16×16 pixels, centered on a fixed grid every 8 pixels.
In the coding stage we set the global dictionary size K_G to 1024, while each class dictionary has $K = \frac{K_G}{N}$ atoms. In this way we ensure a fair comparison with the baseline methods, i.e. all the image representations have the same size. The regularization parameters λ and μ of Eq. 2, and the cost parameter C of SVMs have been selected with a 5-fold cross validation on the training set ($\mu = 0.15$ and $\lambda = 0.1$).

4.2 iCubWorld 1.0

We first evaluate the proposed method in a real Human-Robot Interaction (HRI) setting, where the goal is to recognize single instance of objects. The dataset we refer to has been acquired with the iCub humanoid robot (Metta et al., 2008), and is composed of 7 classes with 500 frames per class, for both the training and the test phase respectively.
Acquisitions have been made with respect to two different modalities, the **Robot Mode** and the **Human Mode** (Fanello et al., 2013a; Fanello et al., 2013b) (see Fig. 3). The Robot Mode dataset contains images acquired by iCub while handling an object of interest. The robot moves the arm in order to observe the object from multiple points of view. The Human Mode dataset contains images depicting a human actor holding one of the seven objects in his hand and showing it to the robot. The robot actively tracks the object, which is presented to the robot from multiple points of view.
The recognition has been performed per frame, temporal information is not used. The results we obtained

[1]The iCubWorld 1.0 and iCubWorld Categorization Datasets can be downloaded from `http://www.iit.it/en/projects/data-sets.html`

Table 1: Accuracy results for the iCubWorld 1.0 Dataset, for both Robot Mode (RM) and Human Mode (HM). We show results when no pyramid is used (No SPM) and with 3-level pyramid (SPM).

	Method	Accuracy RM	Accuracy HM
No SPM	**SC**	70.65%	66.83%
	SC + DASC	76.00%	69.57%
	DBP	**81.82%**	**77.57%**
SPM	**SC**	84.11%	75.44%
	SC + DASC	84.33%	77.73%
	DBP	**86.04%**	**80.97%**

Figure 4: The iCubWorld Categorization dataset. It contains 10 classes acquired with a HRI scheme.

are summarized in Tab. 1 and show how, in this first robotics scenario, dictionary based pooling boosts the performances of the reference methods.

4.3 iCubWorld Categorization

For the second experiment we used a recent object categorization dataset acquired with a HRI setting (Fanello et al., 2013a). The modalities of the acquisition are similar to iCubWorld 1.0, but the focus is on object categorization. It comprehends 10 object categories of different complexity with respect to shape and textures. For each category 3 objects instances of 200 frames each are used for training and 200 frames are used for the testing phase for each new object instance. The particular complexity of this dataset is due to the presence of structured clutter, meaning that the context/background does not improve the

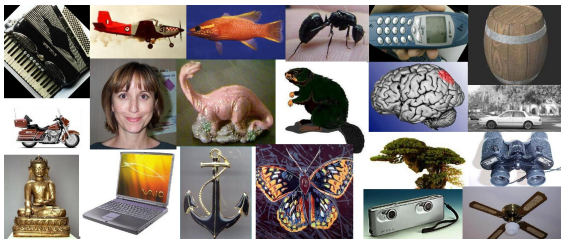

Figure 5: The selection of 20 classes from the popular Caltech-101 dataset, that we considered within the object categorization experiments.

Table 2: Accuracy results for the iCubWorld Categorization Data-Set. We show results when no pyramid is used (No SPM) and with 3-level pyramid (SPM).

	Method	Accuracy
No SPM	**SC**	38.07%
	SC + DASC	39.37%
	DBP	**43.51%**
SPM	**SC**	44.01%
	SC + DASC	44.89%
	DBP	**49.28%**

Table 3: Accuracy results for the 20 classes of the Caltech-101. We show results when no pyramid is used (No SPM) and with 3-level pyramid (SPM).

	Method	Accuracy
No SPM	**SC**	64.55%
	SC + DASC	66.81%
	DBP	**73.62%**
SPM	**SC**	76.95%
	SC + DASC	84.43%
	DBP	**86.24%**

recognition performances and it cannot be exploited as in standard image retrieval data-sets (Fanello et al., 2013a). In Tab. 2 we show the results for the categorization test ($T3$ test in (Fanello et al., 2013a)). Even in this challenging data-set the proposed approach outperforms the baseline methods.

4.4 Caltech-101

Finally we show that our method well generalizes also to standard computer vision dataset oriented to image retrieval problems. For this test we used a selection of 20 classes from the very popular *Caltech-101* dataset (Fei-Fei et al., 2004). The classes are the same used in (Fanello et al., 2013c) and are depicted in Fig. 5. We followed the standard evaluation procedure: for each class we used 30 of the available images as training set, while the others have been used for the test phase (max 50 per class). Again even in absence of the spatial pyramid, our method greatly improves the overall accuracy. With a 3-level pyramid combined with the DBP we obtain a substantial gain in the final accuracy. Tab. 3 summarizes the results.

5 DISCUSSION

In this work we dealt with the widely accepted coding-pooling pipeline for visual recognition and proposed a pooling method guided by the use of discriminative dictionaries. We considered a typical

multi-class scenario and learned a dictionary for each object class. Then, we used local descriptors encoded with the learned atoms to guide the pooling stage: we designed a pooling operator making use of weights directly obtained from the coded descriptors.
We performed an extensive evaluations of the method in both single instance object recognition and object categorization problems, and stressed the representation we proposed considering a classical image retrieval scenarios – using the very popular Caltech 101 – as well as on a typical robot vision task – with data acquired by the iCub humanoid robot. Results clearly speak in favor of our approach, showing that the dictionary based pooling strategy we proposed outperforms previous approaches. Our method is also computationally effective thanks to compactness of the description and usability with linear kernels.

ACKNOWLEDGEMENTS

This work was supported by the European FP7 ICT project No. 270490 (EFAA), project No. 270273 (Xperience) and project No. 288382 (Poeticon++).

REFERENCES

Bay, H., Ess, A., Tuytelaars, T., and Vangool, L. (2008). Speeded-up robust features. *CVIU*, 110:346–359.

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

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

Chen, Q., Song, Z., Hua, Z., Y., H., and Yan, S. (2012). Hierarchical matching with side information for image classification. In *CVPR*.

Collet, A., Martinez, M., and Srinivasa, S. S. (2011). The MOPED framework: Object Recognition and Pose Estimation for Manipulation. *The International Journal of Robotics Research*.

Csurka, G., Dance, C., Fan, L., Willamowski, J., and BrayLixin, C. (2004). Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*.

Destrero, A., De Mol, C., Odone, F., and A., V. (2009). A sparsity-enforcing method for learning face features. *IP*, 18:188–201.

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

Fanello, S., Ciliberto, C., Santoro, M., Natale, L., Metta, G., Rosasco, L., and Odone, F. (2013a). icub world: Friendly robots help building good vision data-sets. In *CVPRW*.

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

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

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

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

Feng, J., Ni, B., Tian, Q., and Yan, S. (2011). Geometric lp-norm feature pooling for image classification. In *CVPR*, pages 2609–2704.

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

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

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

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

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

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

Metta, G., Sandini, G., Vernon, D., Natale, L., and Nori, F. (2008). The icub humanoid robot: an open platform for research in embodied cognition. In *8th Work. on Performance Metrics for Intelligent Systems*. Website: http://www.icub.org.

Perronnin, F., Sánchez, J., and Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *ECCV*.

Russakovsky, O., Lin, Y., Yu, K., and Fei-Fei, L. (2012). Object-centric spatial pooling for image classification. In *ECCV*.

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

Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley and Sons, Inc.

Viola, P. and Jones, M. (2004). Robust real-time face detection. *IJCV*, 57:137–154.

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

Yang, J., Yu, K., Gong, Y., and Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*.