# Ask the image: supervised pooling to preserve feature locality

Sean Ryan Fanello[1,2,3] Nicoletta Noceti[2] Carlo Ciliberto[1,2]
Giorgio Metta[1] Francesca Odone[2]

iCub Facility - Istituto Italiano di Tecnologia[1]

DIBRIS - Università degli Studi di Genova[2] Microsoft Research[3]

## Abstract

*In this paper we propose a weighted supervised pooling method for visual recognition systems. We combine a standard Spatial Pyramid Representation which is commonly adopted to encode spatial information, with an appropriate Feature Space Representation favoring semantic information in an appropriate feature space. For the latter, we propose a weighted pooling strategy exploiting data supervision to weigh each local descriptor coherently with its likelihood to belong to a given object class. The two representations are then combined adaptively with Multiple Kernel Learning. Experiments on common benchmarks (Caltech-256 and PASCAL VOC-2007) show that our image representation improves the current visual recognition pipeline and it is competitive with similar state-of-art pooling methods. We also evaluate our method on a real Human-Robot Interaction setting, where the pure Spatial Pyramid Representation does not provide sufficient discriminative power, obtaining a remarkable improvement.*

## 1. Introduction

Most recent visual recognition systems find their roots in the Bag of Words (BoWs) paradigm [7], that significantly evolved over the last 10 years. In its original formulation, images are seen as unordered collection of descriptors quantized into visual words (during the *coding stage*). These quantizations are then mapped into a histogram representation (in the *pooling stage*) used as an input for an image classifier. This basic approach has been extended by the work of Lazebnik et al. [22], which introduces the so called *Spatial Pyramid Representation* (SPR) to preserve the spatial configuration in images.

In classification tasks, it has been shown that the sparsity of the data representations improves the overall classification accuracy – see for instance [13, 31, 19, 8] and references therein. Thus, Yang et al. [34] improves the SPR pipeline by replacing the vector quantization procedure with a sparse coding step. More recently this approach has been extended in different directions, improving the data representation [21], designing mid-level features [2], or increasing the robustness of the pooling stage [3, 15, 20, 26, 4]. Our work falls within the latter group, since we argue that the weaker step of the current pipeline is indeed pooling.

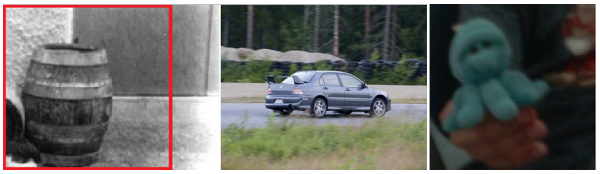

Figure 1. Spatial bias on different datasets. Left: an image from Caltech-101 is wrongly classified because of an unusual position of the object (too far on a side). Center: a standard configuration for the PASCAL VOC dataset. Right: an example of the iCub-World 1.0 that does not present any spatial bias.

Common pooling operations, such as max or average on image regions, may produce an unrecoverable loss of spatial information if the regions are not designed appropriately. Usually, in the SPR framework, pooling is performed on handcrafted subregions of the image which are strongly dependent on the particular dataset used. For instance, in the Caltech-101 objects appear at different scales, but tend to occupy the center of the images, thus a partition in $2^l \times 2^l$ segments is appropriate [22]. In PASCAL VOC [10] the object of interest is usually placed in the upper, center or lower regions of the image, thus a common partition is with $3 \times 1$ segments [24, 36, 35]. Fig. 1 shows different spatial bias on three datasets for image categorization. The image on the left, from Caltech-101, is rather interesting: in such image the object is not positioned in accordance with the majority of images of the same dataset, and it is wrongly classified by a standard spatial pyramid approach. As soon as we crop a part of the image (highlighted with the red square), bringing the object towards the image center, the same algorithm

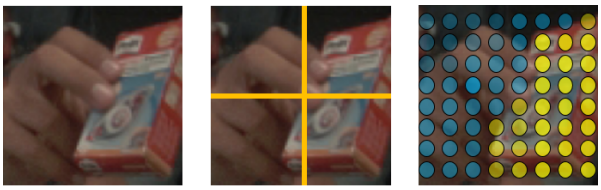


Figure 2. Another image from the iCubWorld 1.0 (A) correctly classified by our pooling method (with a combination of spatial pooling (B) and supervised semantic pooling (C)) and erroneously classified by approaches that rely solely on spatial pyramids.

produces a correct categorization. We can infer that pooling on hand crafted regions is effective if we have a prior on objects position as in close-up photos and portrays. Instead, in natural images in general this information is not available or is not reliable. An example is provided in Fig. 2.

This problem has been addressed in the recent past, with discriminative approaches that learn the appropriate spatial distribution (or partitioning) of features for a given class[15, 27, 9] or with unsupervised methods that take into account the similarity of the features by pooling in a joint space and feature domain [3]. Our work builds on these observations and proposes a novel procedure that combines a standard max pooling on the image domain with a supervised weighted pooling in the feature domain. Unlike [15, 27, 9] our goal is to learn an appropriate partitioning for a given image, in such a way that we do not incorporate in our learned partitions the possible training set bias. We define a "semantic" feature space where features common to a given object class are close to one another. In this space pooling is performed with a soft assignment, meaning that a given feature may participate to different groups with different weights. Such weights are estimated from the data via a mid-level classification process that determines how related is a given feature code to a specific class. Unlike other methods in the literature [3] the two contributions of pooling in the image and in the feature domain are kept separated as they are conceptually very different. They are combined only at the end of a process, by means of multiple kernel learning [32], and they provide a convincing and more compact final representation.

This paper proposes the following contributions. First, it offers a new perspective on discriminative pooling, taking into account the affinity of each local feature to individual classes. The proposed soft assignment on the feature space allows us to balance the mid-level features locality with their semantical meaning. Second, it proposes a combination of complementary pooling approaches which leads to a more compact image description than previously proposed methods. Finally, this combination adapts to the specific set of data or task, thanks to multiple kernel learning.

We assess our method and compare it with the state of the art on benchmark datasets (PASCAL VOC and Caltech-256). Furthermore, we show the effectiveness of the proposed approach on a dataset acquired in Human-Robot Interaction (HRI) scenario (the iCubWorld 1.0 Dataset[1]) which allows us to test our image descriptor on data not affected by any spatial bias.

The remainder of the paper is organized as follows. Sec. 2 reviews the current visual recognition pipeline, setting the basis to present the contributions of our approach, in Sec. 3. Sec. 4 is devoted to the experimental analysis, while Sec. 5 is left to final discussions.

## 2. Preliminaries

In this section we review the state-of-art classification pipeline based on the coding-pooling scheme. This will set the notation to introduce and discuss our contributions.

### 2.1. General Classification Framework

Fig. 3 depicts the visual recognition pipeline commonly adopted by state-of-the-art methods. It can be divided in four main stages:

**Features Extraction.** A set of local descriptors $\mathbf{x}_1, \ldots, \mathbf{x}_M$ are extracted from an image. Examples are image patches, SIFT [23], or SURF [1]. According to [14], in categorization tasks a dense regular grid is to be preferred, thus we adopt a dense grid of SIFT.

**Coding Stage.** The coding stage maps $\mathbf{x}_1, \ldots, \mathbf{x}_M$ into a new representation $\mathbf{u}_1, \ldots, \mathbf{u}_M$, where $\mathbf{u}_i \in \mathbb{R}^K$ with $K$ the dictionary size. The codes $\mathbf{u}_i$ are obtained by minimizing the reconstruction error

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

where $\mathbf{D}$ is a dictionary (fixed or learned from the data), $\|\cdot\|_F$ is the Frobenius norm, and $\mathbf{C}$ is a (possible) constraint. Coding methods differ in the regularization term $\mathrm{R}(\mathbf{u})$ and the constraints $\mathrm{C}(\mathbf{u})$. Examples are Vector Quantization (VQ) [22], Sparse Coding (SC) [34] and Locality-constrained Linear Coding (LLC) [33]. We rely on LLC as a good compromise between performances and a reduced computational cost.

**Pooling Stage.** The codes $\mathbf{u}_i$ are local by definition, failing to capture higher level image statistics. A pooling operator $g$ is thus required, which gathers the codes located in $S$ overlapping regions $Y_s, s = 1...S$, within a single vector

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

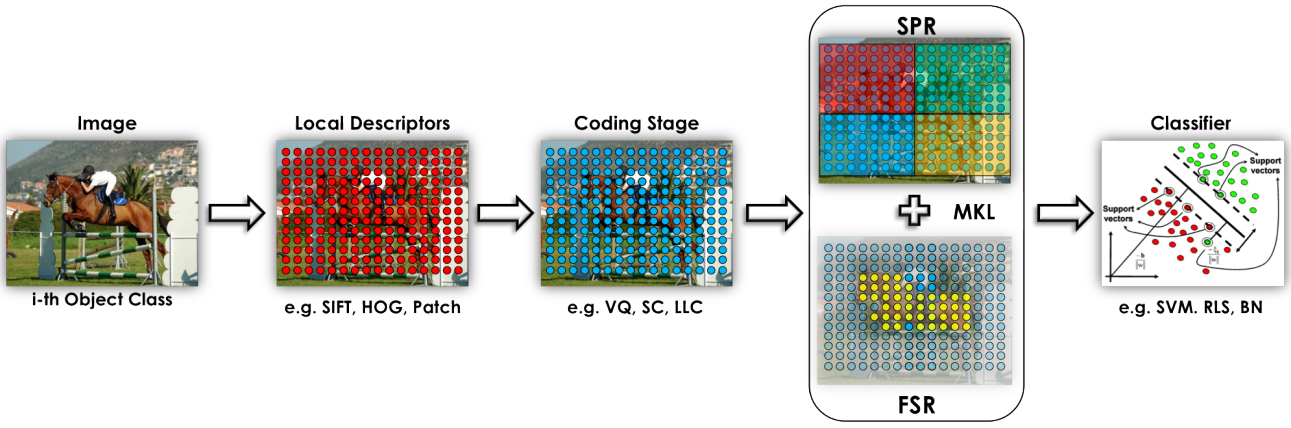


Figure 3. General pipeline for a visual recognition system, where we contribute with a novel supervised pooling.

$\phi_s \in R^K$. Formally

$$\phi_s = g_{(i \in Y_s)}(\mathbf{u}_i). \tag{2}$$

Examples of simple pooling operators are *average pooling* and *max pooling*. It has been shown that the latter obtains the highest performances in classification tasks [2]. The final descriptor of the image $\mathbf{z} \in \mathbb{R}^{KS}$ is the concatenation of all the descriptors $\phi_s$.

Max pooling on the spatial domain can be enriched by taking into account the similarity of local features. In [3] feature vectors from a given region $s$ are clustered with K-means, with a number of clusters $K = P$ common to all the regions. Features codes $\mathbf{u}_i$ are pooled jointly considering each spatial region $s$ and each feature cluster $p$, so that

$$\omega_{s,p} = g_{(i \in Y_s, j \in X_p)}(\mathbf{u}_{i,j}). \tag{3}$$

The final image descriptor $\mathbf{z}$ is the concatenation of all the bin descriptors, thus reaches a considerable size $\mathbf{z} \in \mathbb{R}^{KSP}$.

**Classification.** The final description is fed to a classifier. A standard choice are SVMs [29] because of their computational efficiency in the classification stage. Codes obtained through vector quantization usually require ad-hoc kernels to obtain good performances, whereas sparse coding approaches have shown to be effectively combined with linear classifiers, also ensuring real-time performances [34].

# 3. Combined pooling in image and feature spaces

In this section we present our approach to pooling as a combination of a standard max pooling over a spatial pyramid and of a supervised weighted pooling on the feature space. In the reminder of the section we first describe how we formulate the two different pooling procedures, and then concentrate on the feature space representation, which is the main contribution of the paper. We conclude the section with a description of the way we combined the two obtained representations.

## 3.1. Separating the Image and Feature Domains

The image descriptor proposed in [3] is advantageous in that it accounts for relevant configurations in the feature space. However, it forces all the $S$ spatial regions to be partitioned in an equal number $P$ of states within the feature space, leading to representations of equal length but very different information content. In addition, semantically proximal features (e.g. features lying on a given object) might be separated due to the regular grid of the spatial pyramid. This is undesirable since such an adjacency in feature space would not be reflected by the final description. The image shown in Fig. 2 is an example of such a failure.

Here we slightly change perspective and propose a different combination of pooling in the image and feature space, that leads also to a remarkable reduction in the image descriptor dimension. Indeed notice that, while being true that spatial cells and feature space bins are both designed to capture the geometric properties of the objects, they operate on two very different domains, image and feature space respectively. Therefore it seems more natural to perform pooling separately. We propose to extract two distinct descriptors, the first one $\mathbf{\Phi} \in \mathbb{R}^{KS}$ derived by a standard SPR on the image domain, the second one $\mathbf{\Psi} \in \mathbb{R}^{KP}$ encoding the *Feature Space Representation* (FSR). These two descriptors are obtained by concatenating vectors $\phi_s \in \mathbb{R}^K$ and $\psi_p \in \mathbb{R}^K$ separately:

$$\mathbf{\Phi} = [\phi_1, ..., \phi_S] \tag{4}$$

$$\mathbf{\Psi} = [\psi_1, ..., \psi_P] \tag{5}$$

where

$$\phi_s = g^1_{(i \in Y_s)}(\mathbf{u}_i) \quad \forall s = 1, \dots, S \tag{6}$$

$$\psi_p = g^2_{(i \in X_p)}(\mathbf{u}_i) \quad \forall p = 1, \dots, P. \tag{7}$$

In our method $g^1$ is the usual max pooling operator, while $g^2$ will be described in details later in the section. The final image representation is then obtained by concatenating the two descriptors $\mathbf{z} = [\mathbf{\Phi}, \mathbf{\Psi}] \in \mathbb{R}^{K(S+P)}$.

Notice that if we consider a standard dictionary size $K = 1024$, a spatial pyramid composed of $2^l \times 2^l$ segments with 3 layers ($S = 21$) and $P = 64$ the image representation proposed in [3] has a size $1.3E06$, while our size would be $87040$ (corresponding to a $6\%$ of the descriptor proposed by [3]).

### 3.2. Supervised pooling

In our approach the Feature Space Representation (FSR) is built in a supervised way, taking into account the likelihood of a given feature to be observed in an image belonging to a given class. In other words, the representation is aware of the statistically relevant properties of each class individually.

Formally, we consider $P = N$ bins, where $N$ is the number of classes of the problem under exam, and define a weighted version of the max-pooling operator as follows

$$g^2_{(i \in X_p)}(\mathbf{u}_i) = \max_i (w_i^p \mathbf{u}_i) \quad \forall p = 1, \dots, N \tag{8}$$

and thus, according to Eq. 7,

$$\psi_p(j) = \max_i w_i^p u_i(j) \quad \forall j = 1, \dots, K. \tag{9}$$

The weights $w_i^p$ have a natural interpretation as confidence values reflecting how likely it is to observe the code $\mathbf{u}_i$ in an image depicting class $p$. Therefore, in principle, it would be ideal to set for each $\mathbf{u}_i$ the weight $w_i^p = \mathbb{P}(Class = p|\mathbf{u}_i)$. However, since we do not have access to such latent distribution, here we introduce a mid-level classification stage to estimate it.

The underlying idea is to train $N$ classifiers able to recognize subregions of the image and then use their scores as weights $w_i^p$. Indeed, most classification algorithms are somewhat related to the Bayes rule. For instance in binary settings, the predictor provided by Regularized Least Squares (RLS) converges asymptotically to the target function $\mathbb{E}(y|x)$, that is the expectation of the class label, given the input $x$ [11]. In the multi-class case, by adopting a one-vs-all approach to learn the label associated with each $\mathbf{u}_i$, RLS would (asymptotically) provide the $N$ score functions $f_p(\mathbf{u}_i) = \mathbb{E}(y_p|\mathbf{u}_i) = \mathbb{P}(y_p = 1|\mathbf{u}_i)$, where we have associated label $y_p = 1$ or $0$ according to the presence or absence of class $p$ in the image.

From the discussion above, it is clear that RLS is exactly recovering the desired value for the $w_i^p$s. However in this work we used a Support Vector Machine (SVM) [29] algorithm to perform this mid-level classification, after we empirically observed that the two algorithms lead to comparable performances (see [5] for exhaustive comparisons). Indeed, such evaluation needs to be performed several times for each image, causing a demanding computation. One advantage of SVM is that it reduces the computational effort in classification due to the sparse set of support vectors identified during training.

**Mid-Level Classification Weights (MLCW).** We now describe in details how we estimate the weights. Similarly to [2] we consider mid-level features, pooling together coded SIFTs belonging to a small spatial neighborhood into a single descriptor $\phi_s \in \mathbb{R}^K$. This descriptor is more robust to noise than considering single codes independently, and it gives invariance for small changes in the images.

More in details we consider a single level $l$ from the SPR – with small cell size – and decompose the images in the corresponding $2^l \times 2^l$ cells. The codes in each cell $s$ are pooled together with max pooling obtaining a mid-level (or object part) descriptor $\phi_s \in \mathbb{R}^K$ which is then fed to $N$ classifiers (linear SVMs in our case). They produce $N$ scores for each descriptor: $f_p(\phi_\mathbf{s})$ $p = 1, \dots, N$ which we use as weights $w_s^1, \dots, w_s^N$ for all the codes $\mathbf{u}_i$ belonging to the cell $s$. Fig. 4 provides a visual intuition of the method.

**Combining SPR & FSR.** The simplest way to combine the descriptors $\mathbf{\Phi} \in \mathbb{R}^{KS}$ for the SPR and $\mathbf{\Psi} \in \mathbb{R}^{KP}$ for the FSR is to concatenate them. However, depending on the type of data, one part could be more useful than another. A more effective method to combine heterogeneous features is Multiple Kernel Learning (MKL) based on the idea of comparing each feature with an appropriate kernel independently. Then a global kernel $\mathbf{K}_{opt}$ is obtained as a weighted sum of such contributions, where the weights are learnt from data. We use a linear combination of kernels:

$$\mathbf{K}_{opt}(\mathbf{z}_i, \mathbf{z}_j) = d_S \mathbf{K}_S(\mathbf{\Phi}_i, \mathbf{\Phi}_j) + d_P \mathbf{K}_P(\mathbf{\Psi}_i, \mathbf{\Psi}_\mathbf{j}) \tag{10}$$

with $\mathbf{K}_S$ and $\mathbf{K}_P$ linear kernels on SPR and FSR respectively. The weights are learned using the work of Varma and Ray [30]. We refer the reader to [30, 32] for all the details on the method adopted.

## 4. Experiments

In this section we experimentally validate the proposed method on visual recognition tasks. We compare our results with the most recent state-of-art works, with specific reference to approaches using the coding-pooling classification pipeline we also consider.

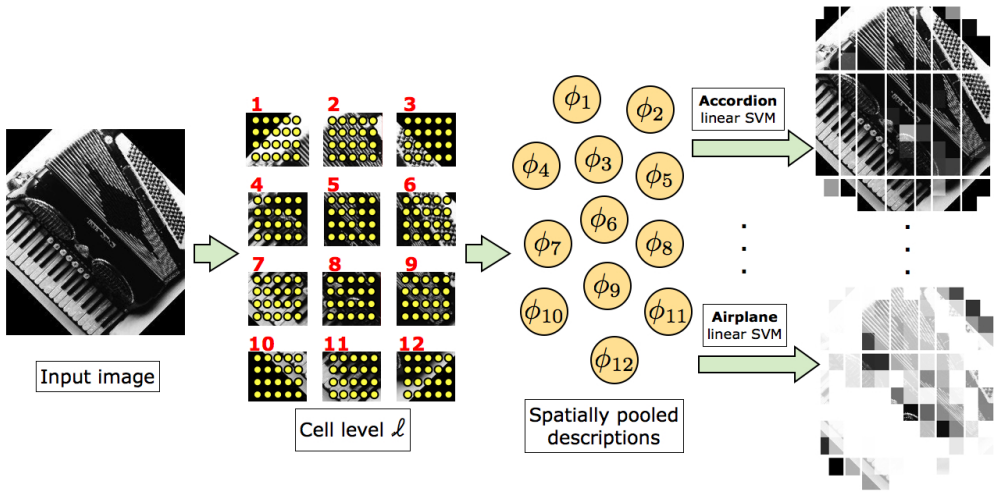


Figure 4. A visual intuition of the mid-level classification stage for the weighted pooling in the feature space (see text for the details). The estimated weights are associated with the various semantic classes and give an impression of what a certain classifier is able to see in the image. In the example, higher weights correspond to a higher alpha channel values: the class *Accordion* "can see" the input image better that the class *Airplane*.

We evaluate our method on three datasets – PASCAL VOC 2007 [10], Caltech-256 [17] and the iCubWorld 1.0 Dataset, that respond to different challenges typical of visual recognition problems. The PASCAL and Caltech-256 are two popular reference benchmarks for image categorization, characterized by a relevant intra-class variability. On the contrary, the iCubWorld refers to the problem of recognizing specific objects instances. iCubWorld images have a strong structured background that does not favor the use of context information within the recognition problem. Moreover, the demonstrator's or robot's hands are always present and act as distractors when building the representation. Finally, the iCubWorld dataset has no spatial bias: objects can occupy different parts of the image of different size and proportion. In the remainder of the section we describe in detail the experimental analysis.

### 4.1. Implementation Details

We provide here all the system parameters to favor the reproducibility of the presented results. We denote with N the number of classes (categories or objects), K is the size of the dictionary, S the number of states on the spatial layout of the image, while P is the number of states of the feature space. As for the local feature extraction, we use a dense grid of SIFTs located every 8 pixels and extracted from $16\times 16$ image patches. In the coding stage we set the dictionary size $K = 4096$ for Caltech-256 and PASCAL VOC 2007, while $K = 1024$ for the iCubWorld 1.0 Dataset.

For the spatial layout we use standard $2^l \times 2^l$ segments with scales $l = 0, 1, 2$ ($S = 21$) for the Caltech-256 and iCubWorld 1.0, whereas in the PASCAL Benchmarks we use the layout suggested by the winner of the VOC 2007 [24] (S=8). Mid-level object classifiers of Sec. 3.2 have been trained on the scale pyramid level $l = 4$. Our partition within the feature space is induced by the membership of codes to the class, thus $P = N$.

### 4.2. PASCAL VOC 2007

PASCAL is a challenging dataset composed by images of 20 object categories gathered from Flickr and characterized by a high variability of viewing angle, illumination, objects size, pose and appearance. Also, occlusions are quite frequent. The classification performance is evaluated using the Average Precision (AP) measure, the standard metric used by PASCAL challenge [10].

We start off by evaluating the benefit of our weighted supervised pooling and compare in Table 1 our approach with two reference methods. The first one is max pooling on a standard SPR [33]. To allow for a fair comparison, we ran the code provided by the authors on SIFT features and set the same parameters we adopted in our method. The next one is the pooling method suggested by [3]. In this case, lacking a publicly available code, we relied on our implementation and set $P = 16$, which seems a good tradeoff between accuracy and descriptor size according to [3]. As for our method, we report the performances achieved by simply concatenating the two contributions of our image description (MLCW), and after having applied multiple kernel learning (MLCW+MKL).

As shown in Table 1, the pooling method we propose systematically outperforms the other approaches. This is consistent with the performances reported in the literature. An interesting contribution is proposed in [26] where the authors report a very similar performance (57.2%, obtained

| | plane | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | motor | person | plant | sheep | sofa | train | tv | AP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| max pooling [33] | 68.2 | 57.7 | 39.9 | 61.6 | 24.0 | 57.4 | 73.4 | 53.5 | 49.7 | 36.9 | 42.3 | 39.6 | 73.4 | 62.2 | 79.4 | 23.8 | 42.7 | 48.4 | 68.0 | 47.7 | 52.5 |
| pooling in [3] | 68.4 | 56.9 | 41.1 | 62.9 | 23.8 | 58.8 | 73.9 | 53.4 | 50.1 | 37.2 | 41.7 | 40.4 | 74.3 | 62.1 | 79.5 | 24.1 | 42.4 | 49.3 | 68.8 | 48.8 | 52.8 |
| OCP [26] | **74.2** | **63.1** | 45.1 | **65.9** | 29.5 | **64.7** | **79.2** | **61.4** | 51.0 | **45.0** | **54.8** | 45.4 | 76.3 | **67.1** | 84.4 | 21.8 | 44.3 | 48.8 | 70.7 | **51.7** | 57.2 |
| **MLCW** | 70.5 | 58.6 | 42.9 | 61.6 | 28.3 | 59.4 | 74.8 | 54.8 | 51.4 | 39.4 | 44.3 | 41.4 | 74.9 | 65.5 | 81.8 | 27.6 | 43.9 | 48.9 | 69.9 | 49.8 | 54.5 |
| **MLCW + MKL** | **74.2** | 62.5 | **49.8** | 59.3 | **32.6** | 62.7 | 78.8 | 56.1 | **52.1** | 43.2 | 48.1 | **45.6** | **78.7** | 63.3 | **88.7** | **31.5** | **47.3** | **52.3** | **73.2** | 51.1 | **57.5** |

Table 1. Classification results (AP in %) on PASCAL VOC 2007 with different pooling strategies. Coding is always performed with LLC.

Figure 5. The iCubWorld 1.0 Dataset. Samples of the 7 classes collected for the *robot* (top strip) and *human* (bottom strip) datasets.

with DHOG features and a dictionary of $K = 8192$ atoms) adopting a pooling strategy that assumes a prior on the location of the object of interest. This speaks in favor of the capability of our method of implicitly dealing with objects in general unknown locations within the image. We finally observe that the results reported in [33] (59.3%) are higher than the ones we obtained running their code. This was also observed in [26] and might be due to the different low-level features and the possible presence of post-processing of the resulting image features.

The current winners of VOC Challenge 2012 [4, 28] obtain high accuracy results, with an AP of 64.7% [4], however their approach is based on a combination of multiple features (SIFT,HOG,LBP) and object detectors coupled with a visual saliency map. Moreover, these remarkable results have been obtained specifically on the PASCAL VOC Datasets, whereas our goal is to build a general image representation able to generalize in different contexts.

| Method | Training Class Size | | | |
|---|---|---|---|---|
| | 15 | 30 | 45 | 60 |
| max pooling [33] | 32.0 | 38.4 | 42.2 | 44.3 |
| pooling in [3] | 32.8 | 39.0 | 42.8 | 45.4 |
| **MLCW** | 34.1 | 39.9 | 42.4 | 45.6 |
| **MLCW + MKL** | **35.2** | **40.1** | **44.9** | **47.9** |

Table 2. Classification results (Average accuracy (%)) on Caltech-256 as the size of the training set increases. Coding is always performed with LLC.

### 4.3. Caltech-256

This Caltech-256 dataset consists of 256 object categories. Similarly to Sec. 4.2, we first specifically evaluated the pooling strategy we are proposing. According to previous works, we report in Table 2 the average accuracy for different sizes of the training set. Once again, our method achieves the best performances.

For a more complete comparison with related works, we summarize in Table 3 the accuracies published for a selection of recent methods from state-of-art. Since the results have been obtained under different conditions, we also report some details on the setting they refer to. Although slightly higher, our method is comparable with the works in [3, 15]. As for the first, the same considerations we did referring to the PASCAL dataset hold here, which may suggest some possible improvements of our method (for instance with different features). A comparison with [15] is more difficult since no public code is available, and we failed implementing the method since the alternate optimization procedure proposed by the authors has no clear theoretical grounds.

Table 4 reports a further analysis of the benefit of our Feature Space Representation. Here the comparison is carried out without the boost of the spatial pyramid. In our case we use the FSR only, discarding the spatial representation entirely, while for other methods we only consider the initial level of the pyramid ($l = 0$). This is the fairest comparison between our approach and [3]: in both cases we represent the image by partitioning the feature space on the overall image (and in both cases we obtain a representation of size KP even if P has a different meaning in the two approaches). We report results on different training set sizes and on two different dictionary sizes. In both setting our approach improves previous results of about 5%, suggesting for us the possibility of exploiting only a features space learnt from (labeled) data.

### 4.4. iCubWorld 1.0 Dataset

Finally, we evaluate the proposed supervised pooling method in a real Human-Robot Interaction (HRI) setting. This analysis is conducted over two sets obtained by collecting images from the cameras of the iCub robot [25] in different settings, the *Robot Mode* and the *Human Mode* (see Fig. 5). We considered 7 object classes, and acquired image sequences of 500 frames per class (per modality), for both the training and the test phase respectively. The recognition has been performed per frame, temporal information

| Method | #Train. | Feature | Pooling | K | Descr. length | Avg. acc. (%) |
|---|---|---|---|---|---|---|
| Wang et al.[33] | 60 | HOG | max pooling | 4096 | SK | 47.7 |
| Yang et al.[34] | 60 | SIFT | max pooling | 1024 | SK | 40.1 |
| Gemert et al. (from [33]) | 30(*) | SIFT | bag-of-words | 128 | K | 27.2 |
| Boureau et al.[3] | 30(*) | SIFT | max pooling | 1024 | KPS | 41.7 |
| Gao et al. [16] | 60 | SIFT | max pooling | 1024 | SK | 40.4 |
| Harata et al. [18] | 15(*) | SIFT | max pooling | 1024 | SK | 30.2 |
| Feng et al. [15] | 45(**) | SIFT | geom. pooling | 4096 | SK | 47.3 |
| **proposed** | 60 | SIFT | superv. weight. max pool. | 4096 | K(N+S) | **47.9** |

Table 3. Comparison with state-of-art methods on Caltech-256. All methods are based on a 3-level standard spatial pyramid. Values denoted with (*) refer to the only training set size reported in the corresponding paper. In all the other cases, we report the size of the best-performing training set among a selection of possibilities, typically in {15,30,45,60}. In (**) 60 is not reported.

| K | Method | Training Class Size | | |
|---|---|---|---|---|
| | | 15 | 30 | 60 |
| 4096 | max pooling [33] (l=0) | 23.7 | 29.3 | 33.9 |
| | pooling in [3] (l=0) | 24.2 | 29.9 | 34.8 |
| | MLCW (no spatial info.) | 29.1 | 34.7 | 40.8 |
| 256 | max pooling [33] (l=0) | 7.9 | 9.5 | 12.1 |
| | pooling in [3] (l=0) | 9.9 | 11.8 | 13.8 |
| | MLCW (no spatial info.) | 12.0 | 17.0 | 20.0 |

Table 4. Accuracy (%) of our FSR based on MLCW and comparison with other pooling methods without the boost of a spatial pyramid (see text).

| Method | Acc. RM (%) | Acc. HM (%) |
|---|---|---|
| max pooling [33], l=0 | 70.11 | 65.91 |
| pooling as [3], l=0 | 78.00 | 68.37 |
| MLCW (no spat. info.) | **81.97** | **73.34** |
| max pooling [33], l={0,1,2} | 83.37 | 75.37 |
| pooling as [3], l={0,1,2} | 84.20 | 76.73 |
| MLCW, l={0,1,2} | 86.28 | 77.28 |
| MLCW + MKL, l={0,1,2} | **87.14** | **78.34** |

Table 5. Accuracy (%) on the iCubWorld 1.0, for both Robot Mode (RM) and Human Mode (HM). Coding is performed with LLC.

is not used. The two datasets have been recorded in natural and realistic way, more information can be found at http://www.iit.it/en/projects/data-sets.html. The Robot Mode dataset contains images acquired while the objects of interest were held in the iCub hand and the robot was moving its own arm in order to observe them from multiple points of view. By exploiting the known forward kinematics of the system it was possible to extract from each image a bounding a box around the robot manipulator, and therefore around the hand-held object.
The Human Mode dataset contains images depicting a human actor holding one of the seven objects of interest in his hand and showing it to the robot. The robot was actively tracking the object, granting a certain degree of background variability to the images. During the acquisition the object of interests are presented to the robot from multiple points of view. It was possible to estimate a bounding box around the moving object by exploiting an independent motion detection algorithm [6, 12]. Since in both cases the bounding boxes have been computed automatically, we can safely assume there is no bias on the position of the object of interest in the image. Classification results are summarized in Tab. 5: in this robotics scenario our supervised image representation boosts the performances of current state of the art methods, with and without the contribution of a spatial pyramid. Notably, even when no spatial information is employed, the accuracy is high. This suggests that, in Human-Robot Interaction (HRI) settings and more in general when we can not rely on a prior on the object position in the image, hand-crafted image regions may not be a suitable choice for pooling.

## 5. Discussion

We proposed a novel supervised pooling method for visual recognition systems. We designed an image description that combines spatial arrangement of objects (with the classical Spatial Pyramid Representation) and their semantics (with a novel Feature Space Representation) by means of Multiple Kernel Learning. Data supervision was used to devise a weighted pooling strategy, where the weights are related to the coherence of a given feature code with respect to a particular class of objects. Results on standard computer vision benchmarks as well as HRI scenarios showed that the proposed approach effectively boosts current visual recognition system performances and helps attenuating the effect of the spatial bias of many image datasets. Indeed, all the examples shown in Figure 1 were correctly classified by our method and they were not by a standard SPR. See also the image in Figure 2: an object lies on a side of the image. A SPR (as in (B)) fails to correctly classify the object because it is misaligned with the center of the image. The same happens with [3]: if the spatial bin is wrong

also the configuration space it is going to be poorly represented. In our case, the combination of a semantic pooling (B+C) allows us to obtain a correct classification: part of the objects are weighted properly and the related features will have higher impact on the final descriptor.

## References


[1] H. Bay, A. Ess, T. Tuytelaars, and L. Vangool. Speeded-up robust features. *CVIU*, 110, 2008. 2

[2] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, 2010. 1, 3, 4

[3] 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. 1, 2, 3, 4, 5, 6, 7

[4] Q. Chen, Z. Song, Z. Hua, H. Y., and S. Yan. Hierarchical matching with side information for image classification. In *CVPR*, 2012. 1, 6

[5] C. Ciliberto, S. Fanello, M. Santoro, L. Natale, G. Metta, and L. Rosasco. On the impact of learning hierarchical representations for visual recognition in robotics. In *IROS*, 2013. 4

[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

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. BrayLixin. Visual categorization with bags of keypoints. In *W. on Statistical Learning in Computer Vision, ECCV*, 2004. 1

[8] A. Destrero, C. De Mol, F. Odone, and V. A. A sparsity-enforcing method for learning face features. *Trans. on IP*, 18, 2009. 1

[9] N. M. Elfiky, J. Gonzlez, and F. X. Roca. Compact and adaptive spatial pyramids for scene recognition. *Image Vision Comput.*, 30(8):492–500, 2012. 2

[10] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 5

[11] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 2000. 4

[12] S. R. Fanello, C. Ciliberto, L. Natale, and G. Metta. Weakly supervised strategies for natural object recognition in robotics. *ICRA*, 2013. 7

[13] S. R. Fanello, N. Noceti, G. Metta, and F. Odone. Multi-class image classification: Sparsity does it better. *VISAPP*, 2013. 1

[14] L. Fei-fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 2

[15] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric lp-norm feature pooling for image classification. In *CVPR*, 2011. 1, 2, 6, 7

[16] S. Gao, I. W.-H. Tsang, L.-T. Chia, and P. Zhao. Local features are not lonely - laplacian sparse coding for image classification. In *CVPR*, 2010. 7

[17] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. *Technical report, California Institute of Technology*, 2007. 5

[18] T. Harada, Y. Ushiku, Y. Yamashita, and Y. Kuniyoshi. Discriminative spatial pyramid. In *CVPR*, 2011. 7

[19] K. Huang and S. Aviyente. Wavelet feature selection for image classification. *Trans on IP*, 17, 2008. 1

[20] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *CVPR*, 2012. 1

[21] S. Kong and D. Wang. A dictionary learning approach for classification: separating the particularity and the commonality. In *ECCV*, 2012. 1

[22] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 2

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60, 2004. 2

[24] M. Marszalek, C. Schmid, H. Harzallah, and J. Van De Weijer. Learning Object Representations for Visual Object Class Recognition. Visual Recognition Challange Workshop, ICCV, 2007. 1, 5

[25] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori. The icub humanoid robot: an open platform for research in embodied cognition. In *8th Work. on Performance Metrics for Intelligent Systems*, 2008. 6

[26] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei. Object-centric spatial pooling for image classification. In *ECCV*, 2012. 1, 5, 6

[27] G. Sharma and F. Jurie. Learning discriminative spatial representation for image classification. In *BMVC*, 2011. 2

[28] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan. Contextualizing object detection and classification. In *CVPR*, 2011. 6

[29] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc., 1998. 3, 4

[30] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. *ICCV*, 2007. 4

[31] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57, 2004. 1

[32] S. V. N. Vishwanathan, Z. Sun, N. Theera-Ampornpunt, and M. Varma. Multiple kernel learning and the SMO algorithm. *Adv. in Neural Information Processing Systems*, 2010. 2, 4

[33] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010. 2, 5, 6, 7

[34] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009. 1, 2, 3, 7

[35] J. Yang, K. Yu, and T. Huang. Efficient highly over-complete sparse coding using a mixture model. In *ECCV*, 2010. 1

[36] X. Zhou, K. Yu, T. Zhang, and T. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV*, 2010. 1