Noname manuscript No.
(will be inserted by the editor)

Analysis of Single and Dual Dictionary Strategies in Pedestrian Classification

V. Javier Traver · Carlos Serra-Toro

Received: date / Accepted: date

Abstract Sparse coding has recently been a hot topic in visual tasks in image processing and computer vision. It has applications and brings benefits in reconstruction-like tasks and in classification-like tasks as well. However, regarding binary classification problems, there are several choices to learn and use dictionaries that have not been studied. In particular, how single-dictionary and dual-dictionary approaches compare in terms of classification performance is largely unexplored. We compare three single-dictionary strategies and two dual-dictionary strategies for the problem of pedestrian classification ("pedestrian" vs "background" images). In each of these five cases, images are represented as the sparse coefficients induced from the respective dictionaries, and these coefficients are the input to a regular classifier both for training and subsequent classification of novel unseen instances. Experimental results with the INRIA pedestrian dataset suggest, on the one hand, that dictionaries learned from only one of the classes, even from the background class, are enough for obtaining competitive good classification performance. On the other hand, while better performance is generally obtained when instances of both classes are used for dictionary learning, the representation induced by a single dictionary learned from a set of instances from both classes provides comparable or even superior performance over the representations induced by two dictionaries learned separately from the pedestrian and background classes.

V. J. Traver and (formerly) C. Serra-Toro
Institute of New Imaging Technologies
Jaume-I University
Castellón (Spain)
Tel.: +34 964-728327
E-mail: vtraver@uji.es

Keywords Dictionary learning · Sparse representations · Binary classification · Pedestrian classification

1 Introduction

Many signals of interest can be represented as a linear combination of a (limited) number of words from a dictionary. These words, also known as atoms, can be predefined with known functions such as wavelets, or learned from some training data. Due to its theoretical and practical benefits, these sparse representations and the corresponding methodologies [8] have in the last decade attracted greater attention from the computer vision community, mostly for *reconstruction*-like tasks, including image denoising [9, 24], inpainting [10], and facial images compression [3]. More recently, they have also been applied in *classification* tasks [42] such as face recognition [47, 7], object and pedestrian detection [41, 32, 17, 22] or action recognition [4, 1, 55, 45, 53]. Benefits that sparse representations can provide include removing irrelevant or noisy variables, obtaining more easily interpretable models, and overfitting prevention [28].

An overview of recent work on sparse coding and dictionary learning is provided in the following paragraphs (Sect. 1.1).

1.1 Related work

The goal of this overview is not to be comprehensive, since the literature is vast, but to provide an idea of relevant research problems and progresses being made. The review then focuses on problems closer to the one addressed in this paper.

Dealing with the computational cost. Learning dictionaries from data is generally preferred over analytically computed ones, but this learning comes at a significant computational cost. One approach to reduce the computational complexity is to impose a separable structure on the dictionary so that separable dictionaries can be learned, which allows larger signals (e.g. image patches) and efficient reconstruction tasks [12]. Instead of considering dictionaries of 1D atoms, the so-called 2D dictionaries are learned, with brings significant memory savings [14]. One interesting idea is to build dictionaries that are sparse themselves, which turns out to be a formulation that is both efficient (like analytical dictionaries) and flexible (like learned dictionaries) [34]. Building on this work, more general approaches [40] and improvements [43] have been devised. Since batch algorithms for dictionary learning and sparse coding may consume much computer memory, incremental versions of such algorithms are also designed [16, 25, 43] for when memory is scarce and/or training sets are large.

Including manifold structure. While geometrical information of data can be useful for discrimination, most sparse coding techniques ignore this structure. To address this, a graph-based algorithm was introduced to explicitly capture the manifold of the data [56]. However, since Laplacian regularization is shown to have some drawbacks such as poor generalization ability, a non-linear generalization [21], and Hessian regularization have been proposed as alternatives [57], for multi-view learning [19], including action recognition [20].

Incorporating task awareness. In the context of their use for learning tasks, dictionaries may be learned discriminatively [52] for classification and, more generally, for a variety of other tasks [27]. A unified objective function including reconstruction error, classification error and a label consistency constraint allows to learn the dictionary, the coding parameters and the classifier parameters simultaneously [16]. When a target domain differs from a source domain, as in the case of off-frontal faces (target) and frontal faces (source), learning a common dictionary that represents both domains can be preferable [37]. Similarly, for multi-view action recognition, besides view-specific dictionaries, view-shared features can be modelled by a common dictionary which turns out to be able to represent actions from unseen views [55].

Enhancing images by exploiting multimodality. For some tasks, dictionaries for several images of different characteristics can be combined. For instance, for image deblurring, dictionaries for blurred and for clean image patches are first (jointly) learned. The latter dictionary is then used to get a reconstructed clean patch from the sparse representation of a blurred patch obtained with the former dictionary [23, 39]. Similar ideas have been developed for other problems such as super-resolution [51], or pan-sharpening [59].

Improving the sparsity concept. Sparse-based classification for face verification [47] is among the first and mostly studied applications of sparse representations for image classification. Some authors have challenged the idea that the sparsity concept really applies or bring any advantage to this problem [33, 38]. However, subsequent work has been overcoming the limitations of the sparse-based classification regarding noise in training data and reduced number of instances per class. Essentially, the improvements have come by modeling separately clean prototypes of the target identities and the intra-class variability that can even be shared by faces from different persons [7]. This allows that fewer face images per person are required [6]. Further improvements are possible by modelling linear variations (e.g. wearing glasses or lighting issues) and non-linear ones (e.g. facial expression changes) [11].

Sparse coding in pedestrian classification. Not much research has been carried out regarding pedestrian classification or detection using sparse representations. Within the more general problem of object detection, local histograms of sparse codes are shown to outperform the conventional histogram of oriented gradients (HOG), particularly with large patch sizes [32]. The non-linear extension of sparse-based classification by using the kernel trick [54] has been explored in the context of hierarchical local representations, with better performance than non-sparse methods [44]. For detecting situations such as "a person riding a bike", which involve two classes, the concatenation of the two class-specific dictionaries learned from the corresponding data provides better results than by using these two dictionaries separately [41]. Also sparse coding has been shown to generally outperform PCA in pedestrian classification [49], and can be used for pedestrian detection refinement [18].

Instead of learning a dictionary from raw data, the HOG descriptor computed over this data can directly be used as the atoms of the dictionary [17]. Additionally, in order to handle occlusion and background clutter, this dictionary is complemented with the canonical basis. L_1-norm minimization applied on vHOG [58] (a variable-size block version of HOG) is reported to outperform both HOG+SVM and vHOG+Adaboost [50]

for pedestrian detection. An histogram of sparse coefficients derived from several dictionaries computed with different sparsity constraints is shown to yield improved performance [22].

1.2 Overview and contributions of this work

This work addresses one practical aspect of dictionary usage for pedestrian classification. As a binary problem ("pedestrian" vs "background" classes), it has its own particularities worth studying. For instance, in face verification or action recognition, many possible classes (person identities or action categories) are considered and sparsity can be advantageously related to more (non-zero) sparse coefficients being concentrated on the part of the representation corresponding to a single class (out of many others). Unlike these multi-class scenarios, the pedestrian classification is a two-class problem, which could even be set as one-class problem, where the class of interest (pedestrian) is relatively well-defined, and the out-of-class instances (non-pedestrian, background clutter or texture-poor areas) is very broad and not so well-defined. It is known that the choice of the dictionary has an impact on the semantics of the data that is captured [48], and this may affect the classification performance.

Therefore, it is relevant to study the effect on the classification performances of learning and using different dictionaries. By considering two classes, one choice is to learn two different class-specific dictionaries, but it is also possible to just learn a single dictionary, for the pedestrian (positive) class, for the non-pedestrian (negative) class, or for both classes. In turn, there are several choices when generating the sparse representation of a new image given one or two of these dictionaries. Since it is not straightforward to decide in advance which option is the best one, this work focuses on experimentally evaluating five different approaches: three single-dictionary strategies and two dual-dictionary strategies.

As in most cases in image processing, the atoms of the dictionary are taken or learned in the space of the raw image data. However, in classification tasks one might also consider higher-level image representations. Although of significant importance, the difference between both approaches has generally been overlooked. To address this issue, this work explores whether raw images or the well-known histogram of oriented gradients (HOG) descriptor [5] is more adequate for the problem at hand. At least in image processing task, it is also customary to divide an input image into a grid for computational or discriminatory purposes, with either all cells in the grid contributing to a single dictionary or having one dictionary per cell. In contrast, we use the full image or HOG descriptor as a single atom. Finally, we use a general-purpose classifier instead of heuristic sparse-representation-based scoring or decision functions.

Concretely, the two main contributions of this work are:

- Proposing and comparing different options to learn and use dictionaries for the binary problem of pedestrian classification: three single-dictionary strategies and two dual-dictionary strategies
- Comparing the classification performance between using raw images and high-level signals when learning the dictionaries.

With respect to the conference paper this work builds on [36], the second contribution is new since previously only the high-level descriptor was tested. As for the first contribution, the strategies are now compared more systematically, including the comparison under the same amount of training signals, which was an issue not considered before, and statistical tests have been applied to find out when and which strategies significantly differ. Therefore, by providing some insights into dictionary usages, the work can guide researchers and practitioners when choosing adequate dictionaries and parameters for particular problem settings.

The rest of the paper is organized as follows. The methodology, including the different dictionary strategies, is described first (Sect. 2). Then, extensive experimentation is reported, covering the difference between raw-images and higher-level representations (Sect. 4.1), the effect of dictionary size and the sparsity constraint (Sect. 4.2), and the impact of the size of the training set for dictionary learning (Sect. 4.3). Conclusions are finally provided (Sect. 5).

2 Methodology

Let us consider a dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$, i.e. with k atoms (code-words) each of dimension n. The "size" of a dictionary is defined as its number of atoms, k. Given a signal $\mathbf{x} \in \mathbb{R}^n$, it can be represented by the sparse representation $\boldsymbol{\alpha} \in \mathbb{R}^k$ found from a given dictionary $\mathbf{D}$. In visual tasks where reconstruction is required, a signal $\mathbf{x}$ can be (approximately) recovered by $\tilde{\mathbf{x}} = \mathbf{D}\boldsymbol{\alpha}$. On the contrary, in a classification task, the sparse representation itself, $\boldsymbol{\alpha}$, can be used as the feature vector, which is the approach taken in this work.

In the following sections, we describe first how the different sparse representations proposed are defined (Sect. 2.1), then the design decisions regarding the formulation used to learn the dictionaries and compute the corresponding sparse representations (Sect. 2.2), and

how the sparse representations are used for classification (Sect. 2.3). Finally, we detail how these different components are combined (Sect. 2.4).

2.1 Sparse representations

Typically, in face recognition and other multi-class problems, several dictionaries are learned, one per class, which are afterwards concatenated into a larger dictionary. However, in a binary problem like pedestrian classification, we wonder whether both dictionaries are required or just any one of them can successfully be used, or which is the best way to combine them. Therefore, the following strategies are explored (Table 1):

Single-dictionary strategies. A single dictionary is learned, but three possibilities (S^+, S^-, and S^*) are considered depending on which data is used for learning, either the positive instances (i.e. those from the positive class), the negative instances, or all of them, respectively.

Dual-dictionary strategies. Here, the two class-specific dictionaries are learned separately and then either concatenated into a larger one $[\mathbf{D}^+, \mathbf{D}^-] \in \mathbb{R}^{n \times 2k}$, or considered separately from the set $\{\mathbf{D}^+, \mathbf{D}^-\}$. The sparse representations are respectively obtained from the concatenated dictionary (strategy $S^\pm$), or by concatenating the representations $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ separately built from $\mathbf{D}^+$ and $\mathbf{D}^-$ (strategy S^{+-}). In other words, in $S^\pm$ it is the dictionaries that are concatenated and yield a single sparse representation $\boldsymbol{\alpha}^\pm$ for a given instance $\mathbf{x}$, whereas in S^{+-} it is the coefficients $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ that are concatenated, resulting in $\boldsymbol{\alpha}^{+-} = [\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-]$, also for each instance.

Therefore, the sparse representations have k components in the single-dictionary strategies, and $2k$ for the dual-dictionary strategies. Similar considerations apply for the sizes of the dictionaries, which is k for the single-dictionary strategies, but $2k$ for the dual-dictionary strategy $S^\pm$. In S^{+-}, two dictionaries are separately involved, each of size k.

Regarding S^*, where instances of both classes are used, it is possible to consider a ratio r of positive instances and the remaining ratio $1-r$ of negative instances. Then, the strategies S^+ and S^- can be seen as particular extreme cases of the r-parameterized $S^*(r)$, namely, $S^+ = S^*(1)$, and $S^- = S^*(0)$. In this work we consider $S^* = S^*(\frac{1}{2})$, to have a balanced situation between the extremes S^+ and S^-. Although S^* is considered here as a single-dictionary strategy, it may actually be regarded as an *hybrid* between the *pure* single-dictionary strategies S^+ and S^- and the dual-dictionary strategies $S^\pm$ and S^{+-}, in the sense that it uses instances of both classes (as in the dual strategies) even though a single dictionary is used (as in the single strategies).

As a key practical consideration, we take care that the number of instances used for dictionary learning is the same for all the different dictionaries compared. This eludes the possible undesirable effect on performance caused by using different number of training instances for learning different dictionaries.

The goal of this work is thus to study the relative merits of these five sparse representations, three ($\boldsymbol{\alpha}^+$, $\boldsymbol{\alpha}^-$, $\boldsymbol{\alpha}^*$) with single dictionaries, and two ($\boldsymbol{\alpha}^\pm$, $\boldsymbol{\alpha}^{+-}$) with dual dictionaries, and how they behave in discriminative terms.

As in common practice [28], instances are subtracted the average of the training instances before dictionary learning, and the learned dictionaries are (atom-wise) L_2-normalized. In S^+, S^- and S^{+-}, it is the average of the instances of corresponding class that is subtracted, while in S^* and $S^\pm$ it is the global average (without distinction of classes). The entire procedure is exactly the same for either both input signals (the gray-level images or the HOG descriptor).

2.2 Optimization model

Several optimization models are possible to learn the dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$ and the sparse representation coefficients $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m] \in \mathbb{R}^{k \times m}$ from m training instances $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{R}^{n \times m}$. One choice is to minimize the L_1 norm of the coefficients $\boldsymbol{\alpha}$ while guaranteeing a reconstruction error lower than an upper bound ϵ,

$$\min_{\mathbf{D}, \boldsymbol{\alpha}_i} ||\boldsymbol{\alpha}_i||_1 \quad \text{s.t. } ||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 \leq \epsilon, \quad \forall\, i \in \{1, \ldots, m\}, \tag{1}$$

as used in [47]. Alternatively, one can seek to minimize the reconstruction error for a given sparsity constraint λ (i.e. the maximum number of non-zero entries allowed),

$$\min_{\mathbf{D}, \boldsymbol{A}} ||\mathbf{X} - \mathbf{D}\boldsymbol{A}||_2^2 \quad \text{s.t. } ||\boldsymbol{\alpha}_i||_0 \leq \lambda, \quad \forall\, i \in \{1, \ldots, m\}, \tag{2}$$

which is the approach taken in [32], and the one used here because setting the sparsity constraint λ can be relatively more intuitive than setting the allowed reconstruction error ϵ, since λ is a natural number with known bounds given $\mathbf{D}$, and it is more user-meaningful. Notice that the higher the value of the constraint λ, the lower the sparsity, i.e. less zero-valued coefficients.

For the sake of clarity when we formally formulate our strategies (Sect. 2.4) in terms of these procedures, let $\mathcal{D}(\mathbf{X}; \lambda)$ be the optimization process corresponding to Eq. (2) returning a dictionary $\mathbf{D}$ and the

Table 1 Dictionary learning and usage strategies studied

Strategy name	Training data for dict. learning	Dictionary/-ies notation	Sparse representation notation
S^+	Positive instances	$\mathbf{D}^+$	$\boldsymbol{\alpha}^+$
S^-	Negative instances	$\mathbf{D}^-$	$\boldsymbol{\alpha}^-$
S^*	All instances together	$\mathbf{D}^*$	$\boldsymbol{\alpha}^*$
$S^\pm$	All instances, per class	$[\mathbf{D}^+, \mathbf{D}^-]$	$\boldsymbol{\alpha}^\pm$
S^{+-}	All instances, per class	$\{\mathbf{D}^+, \mathbf{D}^-\}$	$\boldsymbol{\alpha}^{+-} = [\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-]$

sparse representations $\boldsymbol{A}$ for the given training data $\mathbf{X}$. Let $\mathcal{A}(\mathbf{X}; \mathbf{D})$ be the optimization process returning the sparse representation corresponding to data $\mathbf{X}$ for a given dictionary $\mathbf{D}$. This optimization is equivalent to Eq. 2 but fixing the dictionary $\mathbf{D}$ and only optimizing for $\boldsymbol{A}$. To simplify this notation, some given values for the sparsity constraint λ and dictionary size k are assumed and therefore excluded from the notation.

2.3 Classification

One interesting aspect of sparse representations is that they can very directly and quite efficiently be used for classification. Thus, simple decision functions have been proposed in the literature, such as choosing the class whose corresponding dictionary induces representations with either minimum reconstruction error [47], or the maximum sum of coefficients [4]. Although interesting, these kinds of functions have two limitations: they are heuristic in nature, and are not (directly) applicable to single dictionary cases. Therefore, we used a general-purpose classifier that can be computationally costlier, but it does not have these limitations, and it is therefore more suitable for the purpose of comparing the different strategies.

2.4 Formally defining the strategies

After presenting the strategies, the dictionary and sparse representation learning procedures and the classification choices, we can put all together for a more precise presentation, as follows.

Let $\mathbf{X}_D^+$ and $\mathbf{X}_D^-$ be the data corresponding to the positive (pedestrian) and negative (background) instances used for dictionary learning. Let $\mathbf{X}_C^+$ and $\mathbf{X}_C^-$ be the data used for training the classifier, corresponding to the positive and negative instances. Notice that $\mathbf{X}_D^+ \subset \mathbf{X}_C^+$, and $\mathbf{X}_D^- \subset \mathbf{X}_C^-$, i.e. the data used for dictionary learning are a subset of the complete training dataset. The matrices of sparse coefficients of given data points follow the notation of the vectors of sparse coefficients used in Table 1. For instance, the sparse representation of the m instances, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{R}^{n \times m}$, is denoted as $\boldsymbol{A}^\pm = [\boldsymbol{\alpha}_1^\pm, \ldots, \boldsymbol{\alpha}_m^\pm] \in \mathbb{R}^{k \times m}$ under strategy $S^\pm$. Additionally, a subindex p or n is added to these matrices $\boldsymbol{A}$ to refer to either the positive and negative instances they are computed from.

Let $[\mathbf{Z}_a, \mathbf{Z}_b] \in \mathbb{R}^{m \times (n_a + n_b)}$ be the matrix resulting from horizontally stacking the matrices $\mathbf{Z}_a \in \mathbb{R}^{m \times n_a}$ and $\mathbf{Z}_b \in \mathbb{R}^{m \times n_b}$, (i.e. they have the same number of rows). For our purposes, $\mathbf{Z}$ will either be data instances $\mathbf{X}$, dictionaries $\mathbf{D}$, or sparse coefficients $\boldsymbol{A}$.

Then, the proposed strategies (Sect. 2.1, Table 1) can be defined more precisely as follows:

$$S^+ \equiv \begin{cases} \mathbf{D}^+ = \mathcal{D}(\mathbf{X}_D^+) \\ \boldsymbol{A}_p^+ = \mathcal{A}(\mathbf{X}_C^+; \mathbf{D}^+) \\ \boldsymbol{A}_n^+ = \mathcal{A}(\mathbf{X}_C^-; \mathbf{D}^+) \end{cases} \tag{3}$$

$$S^- \equiv \begin{cases} \mathbf{D}^- = \mathcal{D}(\mathbf{X}_D^-) \\ \boldsymbol{A}_p^- = \mathcal{A}(\mathbf{X}_C^+; \mathbf{D}^-) \\ \boldsymbol{A}_n^- = \mathcal{A}(\mathbf{X}_C^-; \mathbf{D}^-) \end{cases} \tag{4}$$

$$S^* \equiv \begin{cases} \mathbf{D}^* = \mathcal{D}([\mathbf{X}_D^+, \mathbf{X}_D^-]) \\ \boldsymbol{A}_p^* = \mathcal{A}(\mathbf{X}_C^+; \mathbf{D}^*) \\ \boldsymbol{A}_n^* = \mathcal{A}(\mathbf{X}_C^-; \mathbf{D}^*) \end{cases} \tag{5}$$

$$S^\pm \equiv \begin{cases} \mathbf{D}^+ = \mathcal{D}(\mathbf{X}_D^+) \\ \mathbf{D}^- = \mathcal{D}(\mathbf{X}_D^-) \\ \boldsymbol{A}_p^\pm = \mathcal{A}(\mathbf{X}_C^+; [\mathbf{D}^+, \mathbf{D}^-]) \\ \boldsymbol{A}_n^\pm = \mathcal{A}(\mathbf{X}_C^-; [\mathbf{D}^+, \mathbf{D}^-]) \end{cases} \tag{6}$$

$$S^{+-} \equiv \begin{cases} \mathbf{D}^+ = \mathcal{D}(\mathbf{X}_D^+) \\ \mathbf{D}^- = \mathcal{D}(\mathbf{X}_D^-) \\ \boldsymbol{A}_p^{+-} = [\mathcal{A}(\mathbf{X}_C^+; \mathbf{D}^+), \mathcal{A}(\mathbf{X}_C^+; \mathbf{D}^-)] \\ \boldsymbol{A}_n^{+-} = [\mathcal{A}(\mathbf{X}_C^-; \mathbf{D}^+), \mathcal{A}(\mathbf{X}_C^-; \mathbf{D}^-)] \end{cases} \tag{7}$$

Notice that the dictionaries $\mathbf{D}^+$ and $\mathbf{D}^-$, learned for S^+ and S^-, respectively, are the same as those required also for $S^\pm$ and S^{+-}. Thus, when using these strategies together (e.g. during experimentation), these dictionaries can indeed be reused in practice without the need of being recomputed.

Then, for supervised classification, the sparse representations of positive and negative instances, $\boldsymbol{A}_p^s$ and $\boldsymbol{A}_n^s$, with $s \in \{+, -, *, \pm, +-\}$ according to corresponding strategy, are used as input for training the classifier.

The sparse representations for the test data corresponding to the positive and negative classes, $\mathbf{X}^+$ and $\mathbf{X}^-$, are obtained in the same way as for the training data $\mathbf{X}_C^+$ and $\mathbf{X}_C^-$, and hence are not shown here.

3 Experimental setup

3.1 Dataset, classifier and experimental protocol

The INRIA Person Dataset [5] was used in the experiments. Although it is a dataset mostly intended for person detection, it can also be used for classification since its structure contains already cropped positive examples, as well as scenes that are guaranteed to contain no person and thus can be sampled to get a large amount of negative examples. The dataset is also split into training and test subsets.

Therefore, we took the training set of 2,416 already-cropped pedestrian images as positive windows, and the 1,218 human-absent scenes were randomly cropped 5 times thus yielding 6,090 negative windows in total. The test consists of 1,132 positive windows and 453 negative scenes, which we randomly cropped 3 times to obtain an almost balanced test set with 1,359 negative instances. These choices were guided by those made in the paper this one builds on [**?**]. This time, the training and test subsets were joined, and considered as a whole set which is subject to a 5-fold cross validation, thus allowing for a rigorous validation protocol. Therefore, our final dataset consisted of 3,548 positive and 7,449 negative instances.

For each split of the 5-fold, four of the folds, with $m_c = 8,797$ instances, were used for training the classifier, while 2,200 instances were kept for testing. From the set of m_c instances used, only a subset of $m_d < m_c$ instances was considered for dictionary learning for each strategy, and will be indicated in the corresponding section below. All the samplings were stratified, i.e. the proportion between positive and negative instances was maintained in all the folds.

For HOG computation, images were cropped to 128×64 pixels, then divided into blocks of 2×2 square cells, with an overlap between them in each dimension of one cell, each cell having 8 pixels on each side and creating a 9-bin histogram of oriented gradients that were L_2 normalized. The size of the resulting HOG descriptor is $d_H = 3780$. This dimensionality was used to resize the images accordingly, so that the signals used for dictionary learning are approximately of the same length in each case. Images are therefore resized to 86×44, resulting in a vector of size $d_I = 86 \cdot 44 = 3784 \approx d_H$. Notice, however, that the dimensionality of the instances considered afterwards for training and classification is much lower, given by the dictionary size, i.e. k for single-dictionary strategies and $2k$ for the dual-dictionary strategies.

3.2 Hyperparameters

We mostly used a linear SVM, but some tests were performed with a non-linear SVM with a Radial-Basis Function (RBF) kernel for comparison. The values for the regularization parameter C in SVM and the scale parameter γ in the Gaussian function for the RBF SVM were found by a 5-fold cross-validation with grid search, by optimizing the F_1 measure (Section 3.3). The tested ranges were $C \in \{2^i : i \in \{-8, -5, -2, 1, 4, 7, 10\}\}$ and $\gamma \in \{10^i : i \in \{-4, -3, \ldots, 3, 4\}\}$.

Experiments were performed for varying values of the dictionary size k, the sparsity constraint λ, the training set size for dictionary learning m_d, for the five strategies, two classifiers (linear SVM and RBF SVM) and two signals (gray-level images vs HOG). To avoid a combinatorial explosion of tests resulting in unaffordable computational requirements, subsets of these six factors were selected according to the purpose of each experiment, by setting sensible default values for the remaining fixed factors. The values for these factors are specified in the corresponding Sections 4.1–4.3 below.

3.3 Performance assessment

Classification performance is reported using several measures (Table 2). They use the number of true (false) positives, t^+ (f^+), the number of true (false) negatives, t^- (f^-), and the total number of positive (negative) test instances, $n_+ = t^+ + f^-$ ($n_- = t^- + f^+$). As widely known, different measures provide different views of the performance, and some of them, such as the accuracy, do not adequately represent the classifier performance in scenarios of class imbalance. In those cases, the F-measure F_1 or the Matthews Correlation Coefficient (MCC) [30] provide a more unbiased performance summary. The $\mathsf{MCC} \in [-1, +1]$, with $+1$ being perfect classification, 0 random prediction, and -1 complete misclassification, is one good performance metric under class imbalance [2]. Therefore, we use it for summarizing the results of most of the performed tests. All of the measures will be expressed as percentages here, even for MCC whose values can be lower than 0.

Box-and-whisker plots [35] are provided for visually depicting the average performance and its variability across the 5 folds. To compare the statistical significance of the difference between or within strategies, the Wilcoxon signed-rank test [46] is used. As a paired

Table 2 Measures of classification performance (see the text for the definition of t^+, t^-, f^+, f^-, n_+ and n_-)

Measure	Symbol	Definition
Accuracy	Acc	$\frac{t^+ + t^-}{n_+ + n_-}$
Precision	Pre	$\frac{t^+}{t^+ + f^+}$
Recall	Rec	$\frac{t^+}{n_+}$
F-measure	F_1	$2 \cdot \frac{\mathsf{Pre} \cdot \mathsf{Rec}}{\mathsf{Pre} + \mathsf{Rec}}$
Matthews correlation coefficient	MCC	$\frac{t^+ \cdot t^- - f^+ \cdot f^-}{\sqrt{(t^+ + f^+) \cdot (t^+ + f^-) \cdot (t^- + f^+) \cdot (t^- + f^-)}}$

Table 3 Visual representations of significance degree found by the statistical test given a p-value. The lower the p-value, the higher the significance

⍰	test could not be performed (e.g. not enough data)
●	$p < 0.01$
◕	$p < 0.05$
◔	$p < 0.1$
○	$p > 0.1$ (no significance)

difference test, it is applied to compare results for the same folds. Since a minimum number of samples is required, sometimes it was applied to groups of values (e.g. two close values of λ) to have more samples. The p-values are given in some detailed tabular results, and they are in some cases complemented or replaced by a visual representation (Table 3) of the degree of significance found.

3.4 Software

An efficient implementation of HOG, provided by the library OpenCV [13] and recommended optimal parameters [5] indicated in Section 3.1 were used.

For dictionary learning and sparse coding, the SPArse Modeling Software, v2.6 (SPAMS) [25, 26] was used. The functions used were `spams.trainDL()` (with parameters `mode`=3, and `lambda1`=λ) for dictionary learning, and `spams.omp()` (with parameter `lambda`=λ) for computing the sparse representation of a given signal using a learned dictionary, through the Orthogonal Matching Pursuit (OMP) algorithm [29], both from the Python-interface provided by the library. The number of iterations was set to 150, which was experimentally found to be a safe value for convergence.

For classifier learning, hyperparameter validation, performance computation, and the statistical tests, the functionality of the Python machine learning toolkit, scikit-learn [31] was used. Box-and-whisker plots were drawn with the well-known Python's matplotlib [15].

4 Experimental results

The experimental study includes the following aspects: the comparison of sparse representations derived from raw-image and higher-level representations (Sect. 4.1); the effect of dictionary size and the sparsity constraint (Sect. 4.2); and the impact of the size of the training set for dictionary learning (Sect. 4.3).

4.1 Low- vs high-level signal

We first study the difference of learning the dictionary of either the raw images or the higher-level representations such as the HOG descriptor. To that end, we focus on S^+, and fix the training set size m_d, the sparsity constraint λ and the dictionary size k to sensible default values: $m_d = 1000$, $\lambda = 80$, $k = 400$.

Results (Table 4) are far superior when the HOG descriptor is used for dictionary learning and subsequent sparse coding the images. This is an indication that, while it makes sense to use raw images for sparse coding for reconstruction-like purposes, the use of higher-level representations can be beneficial for classification-like tasks [Conclusion C_1] (Sect. 5).

We also analyze whether a non-linear classifier (an SVM with RBF kernel) may outperform a linear one (SVM without kernel). It can be found (Table 5) that the RBF SVM outperforms the linear SVM, more notably in the case of images than in the case of HOG. However, even with the RBF kernel, results with images are inferior to those with HOG, even with the simpler linear SVM.

The HOG-induced sparse coefficients and the linear SVM are used for the rest of the experiments.

4.2 Effect of dictionary size and sparsity constraint

Both the sparsity constraint λ and the size of the dictionary k may have an impact on the subsequent sparse coefficients and, in turn, the classification performance. We tested $k \in \{60, 100, 200, 400, 800\}$, $\lambda \in \{\lambda' : \lambda' \leq$

Table 4 Performance [Average (std. dev.)] with low-level (raw image) and high-level (HOG) representations.

Signal	Acc	F_1	MCC	Pre	Rec
Image	87.41 (0.65)	79.25 (1.14)	70.55 (1.56)	84.59 (1.07)	74.55 (1.47)
HOG	97.22 (0.29)	95.67 (0.45)	93.62 (0.67)	96.0 (0.63)	95.35 (0.32)

Table 5 Performance with linear (L) and RBF (R) SVM for the first data split

	Acc		F_1		MCC		Pre		Rec	
Signal	L	R	L	R	L	R	L	R	L	R
Image	87.1	91.2	78.7	86.0	69.9	79.7	84.5	88.8	73.7	83.4
HOG	97.2	98.5	95.6	97.7	93.5	96.6	95.9	97.3	95.4	98.0

$k, \lambda' \in \{10, 20, 60, 100, 200, 400, 800\}\}$ under all of the strategies.

4.2.1 General patterns

Results for S^+ (Fig. 1) clearly indicate that for a given dictionary size k, the higher the sparsity constraint λ, the higher the performance [Conclusion C_2]. It can also be observed that, even though good performance can be obtained with relatively small dictionaries, it gets stable when λ approaches k. Although this means that bigger dictionaries are generally advisable for higher performance, it is interesting to note that smaller dictionaries may suffice for a given computational-classification performance trade-off [Conclusion C_2]. For instance, referring to Fig. 1, for a target MCC ≈ 94 (expressed in %) one may choose $k \gtrsim 400$ (with $\lambda > 100$), but the smaller $k = 200$ (with $\lambda \gtrsim 60$) would also meet the requirement, and at half dimensionality.

The need of a big value for the sparse constraint λ may be explained by both the comparatively high dimensionality of the signal and the high intra-class variability of both the positive and the negative classes. Therefore, a given instance can only be well approximated as a linear combination of *many* atoms of the dictionary.

4.2.2 Comparing strategies

These patterns observed for S^+ can also be roughly observed in the other strategies, albeit with some noticeable differences among them. A selection of plots shown side-by-side (Fig. 2) allows an easier comparison among the single-dictionary strategies. Furthermore, and interestingly, using a dictionary learned only from HOG descriptors from background images (S^-) can lead to comparable performance with the case of using the dictionary learned from HOG descriptors of pedestrian images (S^+) [Conclusion C_3]. However, the performance of S^- can be somehow lower, particularly for smaller sparsity constraints. On the other hand, the use of dictionaries learned from HOG descriptors of *both* classes, S^*, leads generally to higher performance, particularly for the lower values of the sparsity constraint λ [Conclusion C_4]. This implies that if very sparse representations are desired, learning a dictionary from instances of both classes can be particularly preferable, even if the total number of training instances is the same, not bigger.

This best single-dictionary strategy (S^*) is compared with the dual-dictionaries. Again, a selection of plots (Fig. 3) indicates that the hybrid strategy S^* offers higher performance than the dual strategies ($S^{\pm}$, S^{+-}), specially for low sparsity constraints λ. One important practical implication of these results is that is is computationally beneficial to learn just *one* dictionary of *mixed* instances over learning two separate class-specific dictionaries, resulting also in representations of lower dimensionality and, optionally, sparser [Conclusion C_4]. Regarding the two dual strategies, not very marked differences exist between them, although S^{+-} seems to behave slightly better than $S^{\pm}$.

4.2.3 Statistical significance

Since the above observations are subjective and qualitative in nature, statistical tests can provide more objective and quantitative insights into whether some interesting differences are actually statistically significant. Wilcoxon pair tests between the set of results corresponding to the 5 data splits for a given strategy, λ, and k were performed. Pairs of relevant strategies were compared for several different subsets of λ and k. Results (Table 6) reveal remarkable differences in most tests. In general, the compared strategies do not differ at larger values of λ and k (Tests 1f,2f,3f), something that could already be suspected by looking at the plots (Figs. 2,3). For instance, S^+ is shown to significantly outperform $S^{\pm}$ in all tested conditions (Tests 3a-3e) except for $k = \lambda \in \{400, 800\}$ (Test 3f). Notice that statistically differences are found between strategies $S^{\pm}$ and S^{+-} (Test 4a-f) even though subjectively their performance look rather similar in the corresponding plots. The comparison of the single-dictionary strategies, S^+ and S^-, with one of the dual-dictionary strategy ($S^{\pm}$)

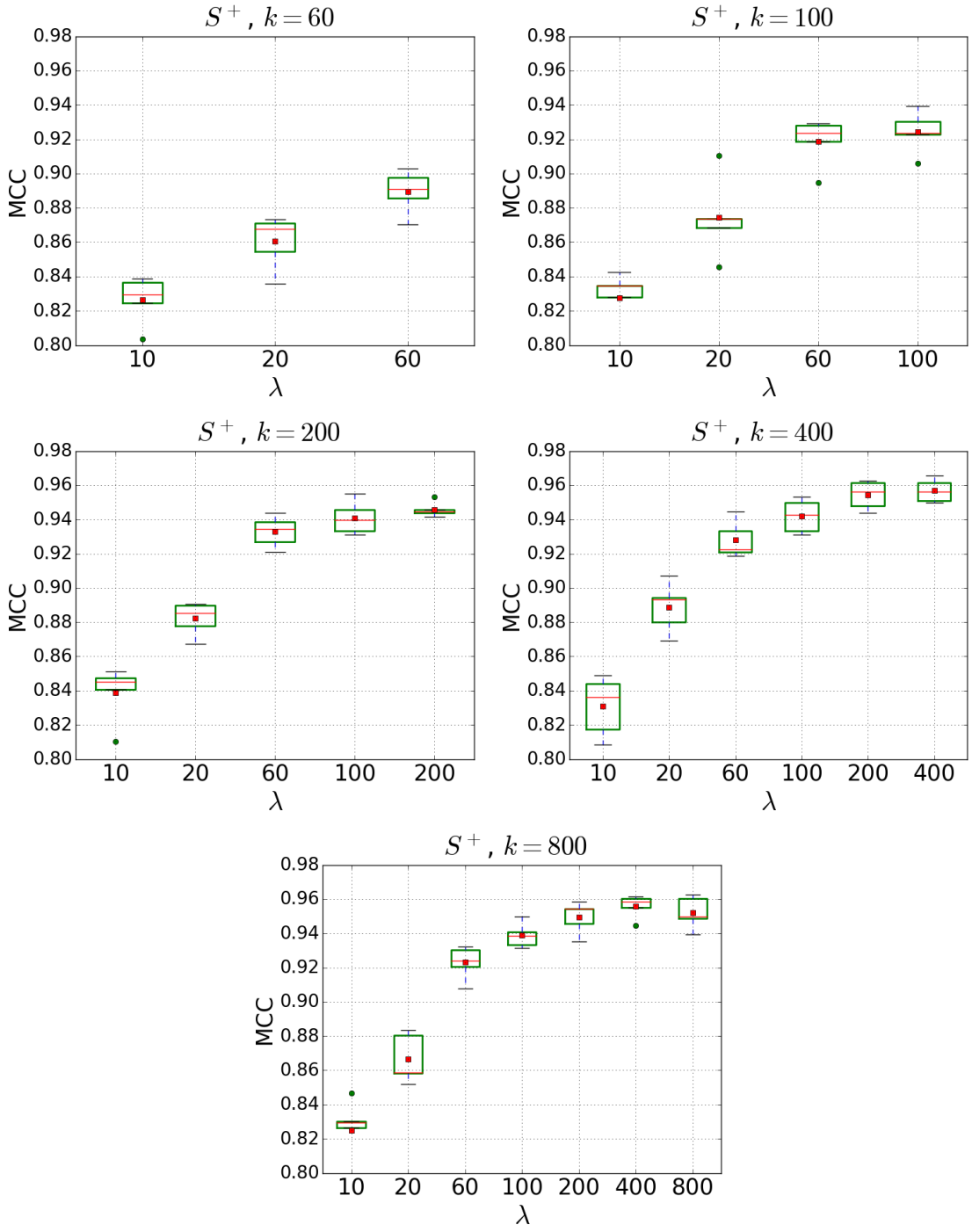

Fig. 1 Performance for different dictionary sizes (k) and varying sparsity constraint (λ) for strategy S^+

discloses some differences, generally at lower confidence level (Tests 5b-f and 6f). When the performance of the two compared strategies are found to differ, it can generally be found which strategy performs better by simple visual comparison of the corresponding plots.

Besides comparing strategies for given k and λ, there are other comparisons of practical interest. For instance, within the same strategy S^*, does performance improve significantly with bigger dictionaries? We can find (Table 7) that this may be true for higher values of the sparsity constraint λ (Test 1d), but not otherwise (Tests 1a-c). The performances between different sparsity constraints are found statistically different at several dictionary sizes (Tests 2a-e). When comparing the performance between dictionaries of different sizes (and lower sparse constraints), some (Test 3b) or no (Tests 3a,c)

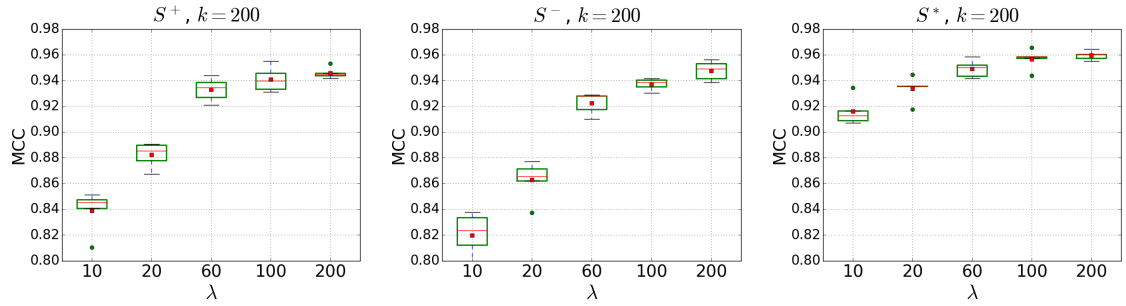

Fig. 2 Comparing strategies S^+, S^- and S^* for the same dictionary size (k) and varying sparsity constraint (λ)

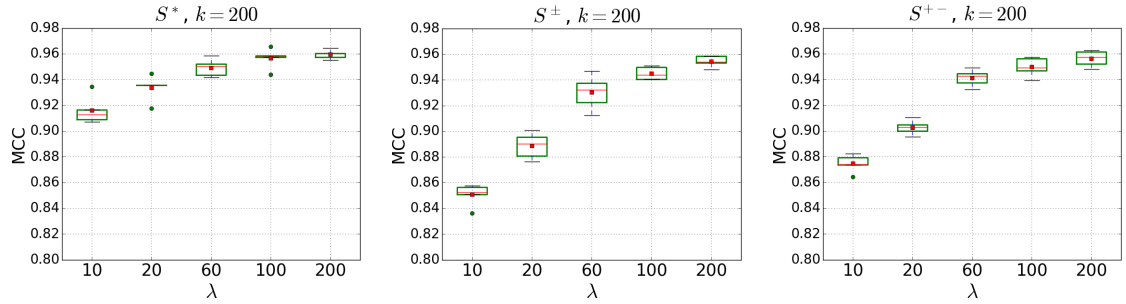

Fig. 3 Comparing strategies S^*, $S^\pm$ and S^{+-} for the same dictionary size (k) and varying sparsity constraint (λ)

statistical differences are found, which emphasizes the idea that lower dimensionality is possible for comparable performance.

4.3 Effect of training size for dictionary learning

The previous tests were performed for a fixed (reference) training set of size $m_d = 1000$ for dictionary learning. However, it is interesting to understand how much this size affects the performance within each strategy, and how the different strategies compare under different training sizes. To that end, we tested three other training sizes $m_d \in \{10, 100, 500\}$, i.e. one hundredth, one tenth and half the reference size for $k = 200$ and $\lambda \in \{60, 100\}$. The classification performance with $m_d = 10$ (i.e. two orders of magnitude lower than the reference size) was very poor and therefore excluded from the results reported here. A selection of illustrative plots (Fig. 4 for single dictionaries and Fig. 5 for dual dictionaries) suggests that when the training size is an order of magnitude smaller, the performance decays noticeably [Conclusion C_5]. In absolute terms, the performance gap seems to be larger in the single-dictionary strategies. Nevertheless, the performance does no degrade when using half the number of instances. The paired Wilcoxon test, performed by joining the results with $\lambda = 60$ and $\lambda = 100$, confirms that the difference between $m_d = 100$ and $m_d = 500$ is statistically significant (p-value $= 0.00506 < 0.01$), but it is not between $m_d = 500$ and $m_d = 1000$. These results suggest that some training instances and learning time can be saved without an impact on classification performance.

When comparing the strategies pair-wise for the three training sizes m_d (Fig. 6), it can be observed that performances have higher variance with smaller training set size, which makes sense. The pattern of when the strategies differ significantly across the size of the training set is not very clear. Tentatively, one might argue that the difference between any of the two pure single-dictionary strategies with any other strategy (either single or dual) tend to slightly decrease with bigger training sets. On the other hand, the difference between the hybrid strategy, S^*, and the dual-dictionary strategies, $S^\pm$ and S^{+-}, tends to increase [Conclusion C_6]. Arguably, the most interesting observation is that with small training sets the hybrid or the dual strategies are preferable, but with bigger training sets the differences shrink somehow. Regarding S^+ and S^-, one may sensibly conclude that if limited instances are available, it is better to learn and use a dictionary of positive instances. Nevertheless, as more instances are available, this choice is less important and S^- can do a good job with only negative instances.

Table 6 Paired tests comparing A vs B for diferent pairs of strategies under given conditions

Test	A	B	conditions	p-value	significance
1a	S^+	S^-	$k \in \{60, 100\}$, $\lambda \in \{10, 20\}$	0.00014	●
1b	S^+	S^-	$k = 200$, $\lambda \in \{10, 20\}$	0.00506	●
1c	S^+	S^-	$k \in \{400, 800\}$, $\lambda \in \{10, 20\}$	0.02277	◓
1d	S^+	S^-	$k = 200$, $\lambda \in \{100, 200\}$	0.72128	○
1e	S^+	S^-	$k \in \{400, 800\}$, $\lambda \in \{100, 200\}$	0.00282	●
1f	S^+	S^-	$k \in \{400, 800\}$, $\lambda \in \{400, 800\}$	0.90956	○
2a	S^+	S^*	$k \in \{60, 100\}$, $\lambda \in \{10, 20\}$	$9e-05$	●
2b	S^+	S^*	$k = 200$, $\lambda \in \{10, 20\}$	0.00506	●
2c	S^+	S^*	$k \in \{400, 800\}$, $\lambda \in \{10, 20\}$	$9e-05$	●
2d	S^+	S^*	$k = 200$, $\lambda \in \{100, 200\}$	0.00506	●
2e	S^+	S^*	$k \in \{400, 800\}$, $\lambda \in \{100, 200\}$	0.00022	●
2f	S^+	S^*	$k \in \{400, 800\}$, $\lambda \in \{400, 800\}$	0.57006	○
3a	S^*	$S^\pm$	$k \in \{60, 100\}$, $\lambda \in \{10, 20\}$	$9e-05$	●
3b	S^*	$S^\pm$	$k = 200$, $\lambda \in \{10, 20\}$	0.00506	●
3c	S^*	$S^\pm$	$k \in \{400, 800\}$, $\lambda \in \{10, 20\}$	$9e-05$	●
3d	S^*	$S^\pm$	$k = 200$, $\lambda \in \{100, 200\}$	0.00506	●
3e	S^*	$S^\pm$	$k \in \{400, 800\}$, $\lambda \in \{100, 200\}$	0.00039	●
3f	S^*	$S^\pm$	$k \in \{400, 800\}$, $\lambda \in \{400, 800\}$	0.17285	○
4a	$S^\pm$	S^{+-}	$k \in \{60, 100\}$, $\lambda \in \{10, 20\}$	0.00151	●
4b	$S^\pm$	S^{+-}	$k = 200$, $\lambda \in \{10, 20\}$	0.00506	●
4c	$S^\pm$	S^{+-}	$k \in \{400, 800\}$, $\lambda \in \{10, 20\}$	$9e-05$	●
4d	$S^\pm$	S^{+-}	$k = 200$, $\lambda \in \{100, 200\}$	0.05934	◔
4e	$S^\pm$	S^{+-}	$k \in \{400, 800\}$, $\lambda \in \{100, 200\}$	0.00102	●
4f	$S^\pm$	S^{+-}	$k \in \{400, 800\}$, $\lambda \in \{400, 800\}$	0.00632	●
5a	S^+	$S^\pm$	$k \in \{60, 100\}$, $\lambda \in \{10, 20\}$	$9e-05$	●
5b	S^+	$S^\pm$	$k = 200$, $\lambda \in \{10, 20\}$	0.02182	◓
5c	S^+	$S^\pm$	$k \in \{400, 800\}$, $\lambda \in \{10, 20\}$	0.03334	◓
5d	S^+	$S^\pm$	$k = 200$, $\lambda \in \{100, 200\}$	0.02182	◓
5e	S^+	$S^\pm$	$k \in \{400, 800\}$, $\lambda \in \{100, 200\}$	0.05222	◔
5f	S^+	$S^\pm$	$k \in \{400, 800\}$, $\lambda \in \{400, 800\}$	0.07829	◔
6a	S^-	$S^\pm$	$k \in \{60, 100\}$, $\lambda \in \{10, 20\}$	$9e-05$	●
6b	S^-	$S^\pm$	$k = 200$, $\lambda \in \{10, 20\}$	0.00506	●
6c	S^-	$S^\pm$	$k \in \{400, 800\}$, $\lambda \in \{10, 20\}$	0.52565	○
6d	S^-	$S^\pm$	$k = 200$, $\lambda \in \{100, 200\}$	0.00934	●
6e	S^-	$S^\pm$	$k \in \{400, 800\}$, $\lambda \in \{100, 200\}$	0.57549	○
6f	S^-	$S^\pm$	$k \in \{400, 800\}$, $\lambda \in \{400, 800\}$	0.04799	◓

Table 7 Paired tests comparing A vs B within S^* under given conditions

Test	A	B	conditions	p-value	significance
1a	$k = 60$	$k = 200$	S^*, $\lambda \in \{10, 20\}$	0.87848	○
1b	$k = 60$	$k = 400$	S^*, $\lambda \in \{10, 20\}$	0.05934	◔
1c	$k = 400$	$k = 800$	S^*, $\lambda \in \{10, 20\}$	0.24112	○
1d	$k = 400$	$k = 800$	S^*, $\lambda \in \{100, 200\}$	0.00506	●
2a	$\lambda = 10$	$\lambda = 20$	S^*, $k \in \{60, 100\}$	0.00506	●
2b	$\lambda = 10$	$\lambda = 20$	S^*, $k \in \{100, 200\}$	0.00506	●
2c	$\lambda = 10$	$\lambda = 60$	S^*, $k \in \{60, 100\}$	0.00506	●
2d	$\lambda = 10$	$\lambda = 60$	S^*, $k \in \{100, 200\}$	0.00506	●
2e	$\lambda = 60$	$\lambda = 100$	S^*, $k \in \{100, 200\}$	0.07446	◔
3a	$k = 200$, $\lambda \in \{100, 200\}$	$k = 400$, $\lambda \in \{200, 400\}$	S^*	0.16881	○
3b	$k = 200$, $\lambda \in \{100, 200\}$	$k = 800$, $\lambda \in \{200, 400\}$	S^*	0.03666	◓
3c	$k = 400$, $\lambda \in \{100, 200\}$	$k = 800$, $\lambda \in \{200, 400\}$	S^*	0.24112	○

5 Conclusions

The results of the experiments under the tested conditions led to the following conclusions, regarding the pedestrian classification performance.

C_1: The sparse representations induced from dictionaries learned from higher-level descriptor such as HOG

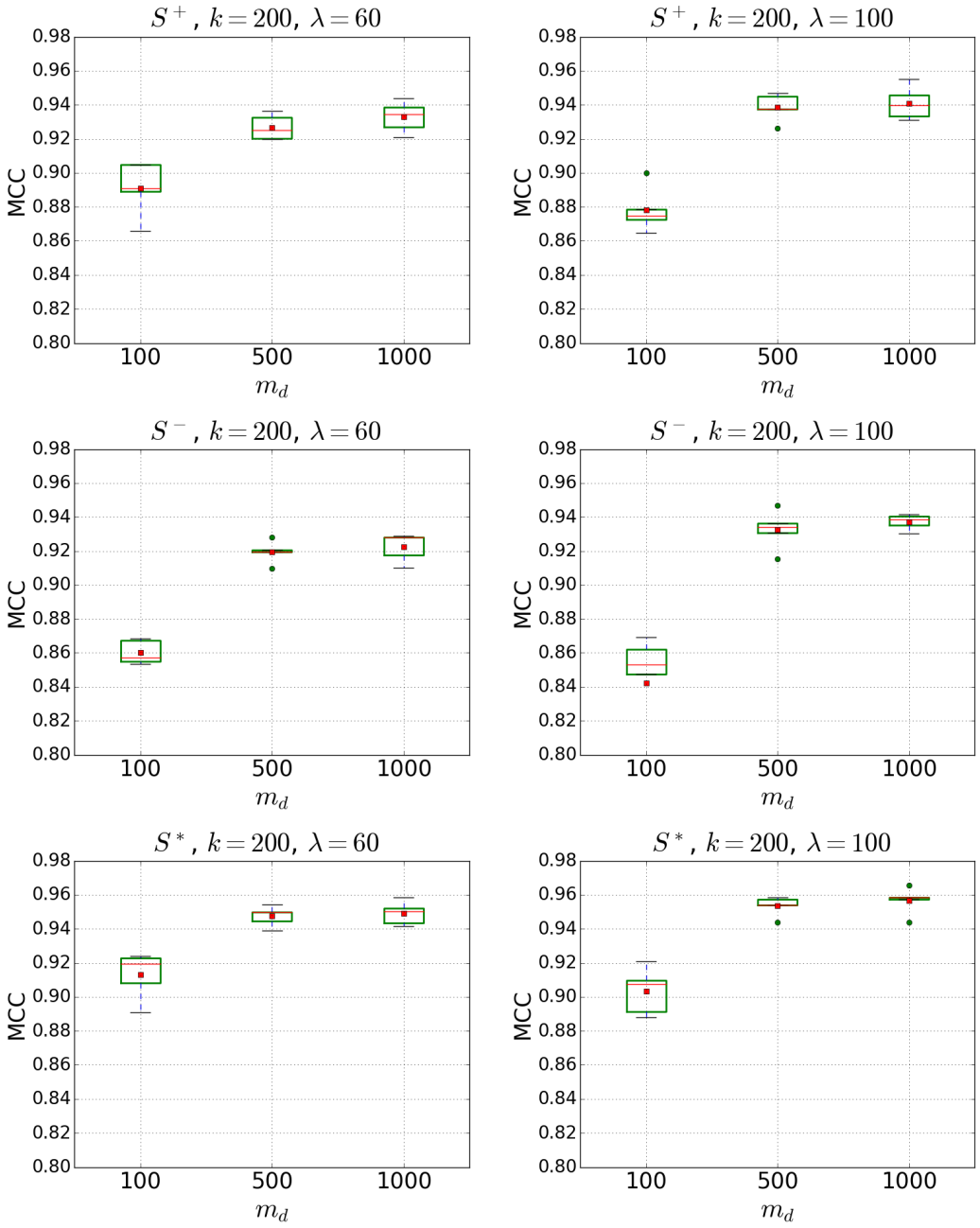

Fig. 4 Performance of single-dictionary strategies by varying the training size (m_d) for dictionary learning

are more discriminative than those from dictionaries learned directly from gray-level images.

C_2: In general, the larger the dictionaries and the lower the sparsity (corresponding to larger sparsity constraint), the better. However, in some cases it is possible to get some similar performance with smaller dictionaries and/or sparser solutions, with the corresponding computational advantage.

C_3: It is possible to get competitive performance with a variety of dictionary choices, even with a single dictionary learned from only negative instances. However, if instances of only one class have to be used for dictionary learning, the use of instances of the positive class is advisable.

C_4: It seems preferable to use dictionaries learned from a mixture of positive and negative classes over dictionaries of only one of the classes. Nevertheless, for a fixed dictionary size, a dictionary learned from both classes without distinction among them offers equal or better performance than the dual strategies involving learning two separate class-specific dictionaries. This winner strategy in discriminative

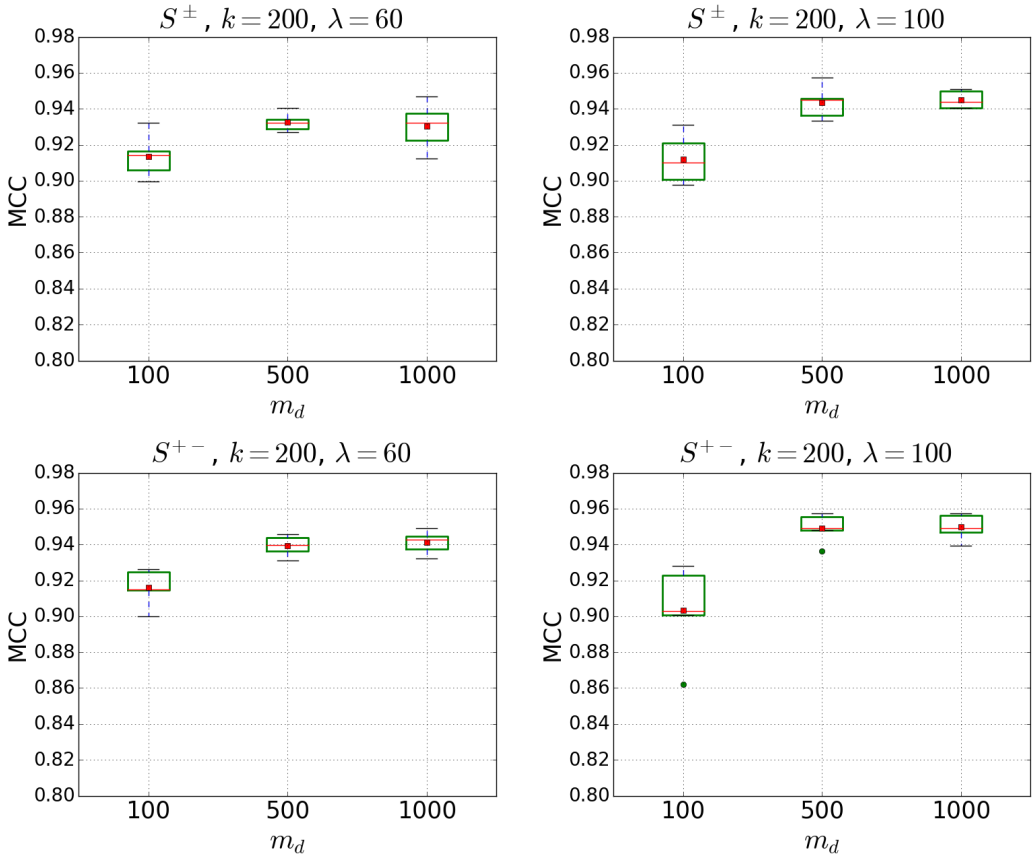

Fig. 5 Performance of dual-dictionary strategies by varying the training size (m_d) for dictionary learning

terms is also computationally advantageous since it requires learning one dictionary, not two, and the sparse representation has half dimensionality with respect the dual-dictionary strategies.

C_5: Although larger training sizes for dictionary learning produce better results, only changes in size which are larger in at least one order of magnitude result in noticeable better performance.

C_6: While the evidence is not strong, the performance gap between pure single- and dual-dictionary strategies decreases when dictionaries are learned with more training instances, whereas the difference between the hybrid strategy and the dual strategies becomes statistically significant with larger training sets.

Some possibilities for further work include: studying the effect of using other optimization models for dictionary learning and sparse coding; exploring whether simple and efficient ad hoc decision functions are possible, and finding out how they perform with respect to general-purpose classifiers; and whether the current representation in terms of sparse coefficients can be rethought so that the sparsity can actually be exploited in order to achieve lower dimensionality without any decay of the classification performance.

Acknowledgments. The collaboration of Á. Hernández-Górriz in an earlier stage of this work is acknowledged. This work is partly funded by the Spanish *Ministerio de Economía, Industria y Competitividad* (TIN2013-46522-P), and *Generalitat Valenciana* (PROMETEOII /2014/062).

References

1. Alfaro A, Mery D, Soto A (2016) Action recognition in video using sparse coding and relative features. In: Computer Vision and Pattern Recognition (CVPR), pp 2688–2697
2. Boughorbel S, Jarray F, El-Anbari M (2017) Optimal classifier for imbalanced data using Matthews correlation coefficient metric. PLoS ONE 12(6)
3. Bryt O, Elad M (2008) Compression of facial images using the K-SVD algorithm. Journal of Visual Communications and Image Representation 19(4):270–282

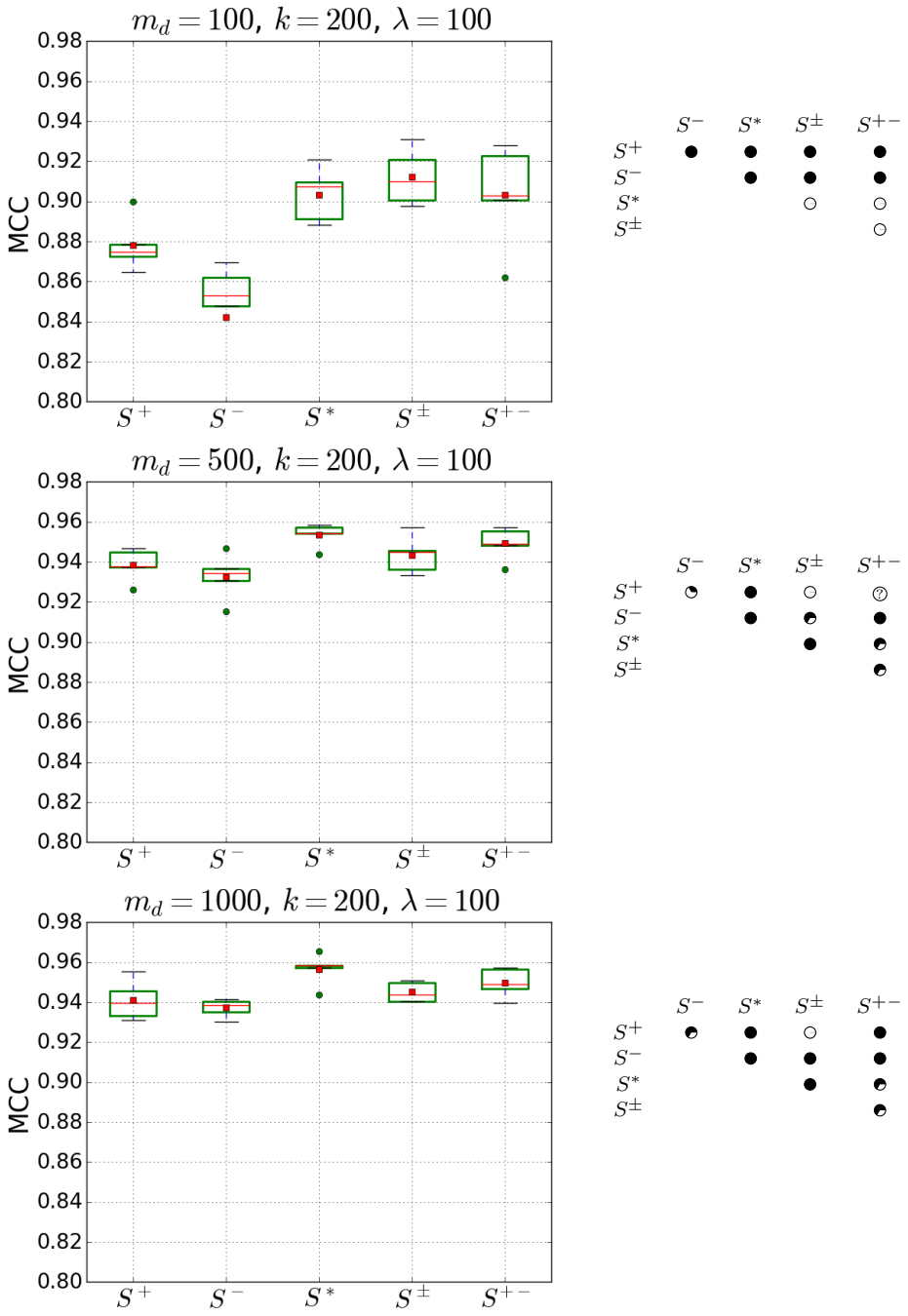

Fig. 6 Performance for different training sizes for dictionary learning (m_d)

4. Castrodad A, Sapiro G (2012) Sparse modeling of human actions from motion imagery. International Journal of Computer Vision (IJCV) 100(1)
5. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition (CVPR)
6. Deng W, Hu J, Guo J (2012) Extended SRC: Undersampled face recognition via intraclass variant dictionary. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 34(9):1864–1870
7. Deng W, Hu J, Guo J (2013) In defense of sparsity based face recognition. In: Computer Vision and Pattern Recognition (CVPR)
8. Elad M (2010) Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer
9. Elad M, Aharon M (2006) Image denoising via learned dictionaries and sparse representation. In: Computer Vision and Pattern Recognition (CVPR)

10. Fadili MJ, Starck JL, Murtagh F (2009) Inpainting and zooming using sparse representations. The Computer Journal 52:64–79
11. Gao Y, Ma J, Yuille AL (2017) Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. IEEE Transactions on Image Processing 26(5):2545–2560
12. Hawe S, Seibert M, Kleinsteuber M (2013) Separable dictionary learning. In: Computer Vision and Pattern Recognition (CVPR), pp 438–445
13. Howse J, Joshi P, Beyeler M (2016) OpenCV: Computer Vision Projects with Python. Packt
14. Hsieh SH, Lu CS, Pei SC (2014) 2D sparse dictionary learning via tensor decomposition. In: IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp 492–496
15. Hunter JD (2007) Matplotlib: A 2D graphics environment. Computing in Science and Engineering 9(3):90–95
16. Jiang Z, Lin Z, Davis LS (2013) Label consistent K-SVD: Learning a discriminative dictionary for recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 35(11):2651–2664
17. Krishna Vinay G, Haque SM, Venkatesh Babu R, Ramakrishnan K (2012) Human detection using sparse representation. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)
18. Liang F, Tang S, Zhang Y, Xu Z, Li J (2014) Pedestrian detection based on sparse coding and transfer learning. Machine Vision and Applications (MVA) 25(7):1697–1709
19. Liu W, Tao D, Cheng J, Tang Y (2014) Multiview Hessian discriminative sparse coding for image annotation. Computer Vision and Image Understanding (CVIU) 118(Supplement C):50–60
20. Liu W, Liu H, Tao D, Wang Y, Lu K (2015) Multiview Hessian regularized logistic regression for action recognition. Signal Processing 110:101–107
21. Liu W, Zha ZJ, Wang Y, Lu K, Tao D (2016) p-Laplacian regularized sparse coding for human activity recognition. IEEE Transactions on Industrial Electronics 63(8):5120–5129
22. Liu Y, Lasang P, Siegel M, Sun Q (2016) Multi-sparse descriptor: A scale invariant feature for pedestrian detection. Neurocomputing 184:55–65
23. Lou Y, Bertozzi AL, Soatto S (2011) Direct sparse deblurring. Journal of Mathematical Imaging and Vision 39(1):1–12
24. Mairal J, Elad M, Sapiro G (2008) Sparse representation for color image restoration. IEEE Transactions on Image Processing 17(1):53–69
25. Mairal J, Bach F, Ponce J, Sapiro G (2009) Online dictionary learning for sparse coding. In: International Conference on Machine Learning (ICML)
26. Mairal J, Bach F, Ponce J, Sapiro G (2010) Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research 11:19–60
27. Mairal J, Bach F, Ponce J (2012) Task-driven dictionary learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 34(4):791–804
28. Mairal J, Bach F, Ponce J (2014) Sparse modeling for image and vision processing. Foundations and Trends in Computer Graphics and Vision 8(2–3):85–283
29. Mallat S, Zhang Z (1993) Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing 41(12):3397–3415
30. Matthews BW (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA) - Protein Structure 405(2):442–451
31. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12:2825–2830
32. Ren X, Ramanan D (2013) Histograms of sparse codes for object detection. In: Computer Vision and Pattern Recognition (CVPR)
33. Rigamonti R, Brown M, Lepetit V (2011) Are sparse representations really relevant for image classification? In: Computer Vision and Pattern Recognition (CVPR)
34. Rubinstein R, Zibulevsky M, Elad M (2010) Double sparsity: Learning sparse dictionaries for sparse signal approximation. IEEE Transactions on Signal Processing 58(3):1553–1564
35. Sahay A (2016) Data Visualization, Volume I. Business Expert Press
36. Serra-Toro C, Hernández-Górriz Á, Traver VJ (2017) Strategies of dictionary usages for sparse representations for pedestrian classification. In: Pattern Recognition and Image Analysis, IbPRIA 2017, pp 96–103
37. Shekhar S, Patel VM, Nguyen HV, Chellappa R (2015) Coupled projections for adaptation of dictionaries. IEEE Transactions on Image Processing 24(10):2941–2954
38. Shi Q, Eriksson A, van den Hengel A, Shen C (2011) Is face recognition really a compressive sensing problem? In: Computer Vision and Pattern

Recognition (CVPR)
39. Singh K, Vishwakarma DK, Walia GS (2017) Blind image deblurring via gradient orientation-based clustered coupled sparse dictionaries. Pattern Analysis and Application (PAA)
40. Sironi A, Tekin B, Rigamonti R, Lepetit V, Fua P (2015) Learning separable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 37(1):94–106
41. Sivalingam R, Somasundaram G, Morellas V, Papanikolopoulos N, Lotfallah OA, Park Y (2010) Dictionary learning based object detection and counting in traffic scenes. In: International Conference on Distributed Smart Cameras
42. Spratling MW (2014) Classification using sparse representations: a biologically plausible approach. Biological Cybernetics 108(1):61–73
43. Sulam J, Ophir B, Zibulevsky M, Elad M (2016) Trainlets: Dictionary learning in high dimensions. IEEE Transactions on Signal Processing 64(12):3180–3193
44. Sun R, Zhang G, Yan X, Gao J (2016) Robust pedestrian classification based on hierarchical kernel sparse representation. Sensors 16(8)
45. Wang W, Yan Y, Zhang L, Hong R, Sebe N (2016) Collaborative sparse coding for multiview action recognition. IEEE MultiMedia 23(4):80–87
46. Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics Bulletin 1(6):80–83
47. Wright J, et al (2009) Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 31(2)
48. Wright J, et al (2010) Sparse representation for computer vision and pattern recognition. Proceedings of the IEEE 98(6):1031–1044
49. Xie YF, Su SZ, Li SZ (2010) A pedestrian classification method based on transfer learning. In: 2010 International Conference on Image Analysis and Signal Processing, pp 420–425
50. Xu R, Jiao J, Zhang B, Ye Q (2012) Pedestrian detection in images via cascaded L_1-norm minimization learning method. Pattern Recognition 45(7):2573–2583
51. Yang J, Wright J, Huang TS, Ma Y (2010) Image super-resolution via sparse representation. IEEE Transactions on Image Processing 19(11):2861–2873
52. Yang M, Zhang L, Feng X, Zhang D (2011) Fisher discrimination dictionary learning for sparse representation. In: International Conference on Computer Vision (ICCV), pp 543–550
53. Yao T, Wang Z, Xie Z, Gao J, Feng DD (2017) Learning universal multiview dictionary for human action recognition. Pattern Recognition 64:236–244
54. Zhang L, Zhou WD, Chang PC, Liu J, Yan Z, Wang T, Li FZ (2012) Kernel sparse representation-based classifier. IEEE Transactions on Signal Processing 60(4):1684–1695
55. Zheng J, Jiang Z, Chellappa R (2016) Cross-view action recognition via transferable dictionary learning. IEEE Transactions on Image Processing 25(6):2542–2556
56. Zheng M, Bu J, Chen C, Wang C, Zhang L, Qiu G, Cai D (2011) Graph regularized sparse coding for image representation. IEEE Transactions on Image Processing 20(5):1327–1336
57. Zheng M, Bu J, Chen C (2014) Hessian sparse coding. Neurocomputing 123:247–254
58. Zhu Q, Yeh M, Cheng K, Avidan S (2006) Fast human detection using a cascade of histograms of oriented gradients. In: Computer Vision and Pattern Recognition (CVPR), pp 1491–1498
59. Zhu XX, Bamler R (2013) A sparse image fusion algorithm with application to pan-sharpening. IEEE Transactions on Geoscience and Remote Sensing 51(5):2827–2836