Neural Comput & Applic
DOI 10.1007/s00521-016-2223-x

Evolving weighting schemes for the Bag of Visual Words

Hugo Jair Escalante[1] · Víctor Ponce-López[2,3,4] · Sergio Escalera[3,4] · Xavier Baró[2,3,4] · Alicia Morales-Reyes[1] · José Martínez-Carranza[1]

Received: 13 December 2015 / Accepted: 11 February 2016
© The Natural Computing Applications Forum 2016

Abstract The Bag of Visual Words (BoVW) is an established representation in computer vision. Taking inspiration from text mining, this representation has proved to be very effective in many domains. However, in most cases, standard term-weighting schemes are adopted (e.g., term-frequency or TF-IDF). It remains open the question of whether alternative weighting schemes could boost the performance of methods based on BoVW. More importantly, it is unknown whether it is possible to automatically learn and determine effective weighting schemes from scratch. This paper brings some light into both of these unknowns. On the one hand, we report an evaluation of the most common weighting schemes used in text mining, but rarely used in computer vision tasks. Besides, we propose an evolutionary algorithm capable of automatically learning weighting schemes for computer vision problems. We report empirical results of an extensive study in several computer vision problems. Results show the usefulness of the proposed method.

Keywords Bag of Visual Words · Bag of features · Genetic programming · Term-weighting schemes · Computer vision

This paper is an extended and improved version of [12] and it is being submitted to the Special Issue on *Computational Intelligence for Vision and Robotics* of the Neural Computing and Applications Journal.

✉ Hugo Jair Escalante
hugojair@inaoep.mx

Víctor Ponce-López
vponce@cvc.uab.es

Sergio Escalera
sergio@maia.ub.es

Xavier Baró
xbaro@uoc.edu

Alicia Morales-Reyes
a.morales@inaoep.mx

José Martínez-Carranza
carranza@inaoep.mx

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica, 72840 Puebla, Mexico

[2] Universitat Oberta de Catalunya, Barcelona, Spain

[3] University of Barcelona, Barcelona, Spain

[4] Computer Vision Center, Barcelona, Spain

1 Introduction

The Bag of Visual Words (BoVW) is a widely adopted representation for describing the content of images and videos in computer vision problems [42]. This representation is the analogy of the Bag of Words (BoW) representation used in text mining and information retrieval: BoVW accounts for the presence and absence of prototypical patterns (called visual words, and playing the role of words in text processing) that are obtained from training images. This representation has obtained outstanding results in a large number of scenarios [3, 5, 9, 13, 30, 35, 42, 43, 50].

In spite of its effectiveness and popularity, most implementations of BoVW adopt pretty standard weighting schemes, that is, the mechanisms that determine the contribution that visual words have for describing the content of images and videos. For instance, the most common scheme is term-frequency where the BoVW representation is an histogram that accounts for the occurrences of visual words in the image or video. Although competitive

Published online: 01 March 2016

performance has been obtained with this formulation, we think it is worth studying alternative weighting schemes.

This paper explores the suitability of using alternative term-weighting schemes for image and video representation. On the one hand, we report an evaluation of the most common weighting schemes used in text mining, but rarely used for computer vision tasks. Our study comprises unsupervised and supervised weighting schemes. More importantly, we propose an evolutionary algorithm capable of automatically learning weighting schemes for computer vision problems from scratch. The evolutionary algorithm explores the search space of possible weighting schemes that can be generated by combining a set of primitives with the aim of maximizing the classification/recognition performance. We perform experiments in landmark problems in computer vision, namely image categorization (different subsets of the Caltech-101 data set [16]), gesture recognition (the newly introduced Montalbano data set [15]), action recognition (MSRDaily3D Data [47]), places-scene recognition (the well known 15-scenes [30]), insect and bird classification [29, 31] and adult image classification [49]. Experimental results show the effectiveness of the proposed method.

A previous version of this work was published in [12]. Compared to that work, this paper provides a more detailed explanation and motivation for the proposed approach. Furthermore, we extend the experimental evaluation by including additional data sets that correspond to other domains not explored previously. Finally, we also perform a deeper analysis on the resulting weighting schemes.

The remainder of this paper is organized as follows. Next section introduces the BoVW representation and reviews-related work. Section 3 presents common and alternative weighting schemes that have been adopted in text mining and information retrieval but that have not been used in computer vision. Section 4 describes in detail the proposed methodology for evolving weighting schemes. Next, Sect. 5 reports experimental results. Finally, Sect. 6 outlines conclusions and future work directions.

2 BoVW representation

In text mining and information retrieval, the BoW representation is a way to represent documents as numerical vectors, with the aim that such vectorial space captures information about their semantics and content [40]. The idea is to represent a document by a vector of length equal to the number of terms (e.g., words) in the vocabulary associated with the corpus under analysis. Each element in that vector indicates the relevance/importance of the corresponding term for describing the content of the document. Although the BoW makes strong assumptions (e.g., that word order is not important), it is still one of the most used representations nowadays.[1]

Under the BoW, the ith document is represented by a vector $\mathbf{d}_i = \langle x_{i,1}, \ldots, x_{i,|V|} \rangle$, where $x_{i,j}$ is a scalar that indicates the importance of the term t_j for describing the content of the ith document; V is the vocabulary, i.e., the set of different words in the corpus. The way in which $x_{i,j}$ is estimated is given by the so-called term-weighting scheme. There are many ways of defining $x_{i,j}$ in the text mining and information retrieval literature. Usually, $x_{i,j}$ carries information about both: *term-document relevance* (TDR) and *term-relevance* (TR). The former, explicitly measures the relevance of a term for a document, i.e., it captures local information. The most common TDR is the *term-frequency* (TF) weight, which indicates the number of times a term occurs in a document. On the other hand, TR aims to capture relevance of terms for the task at hand, i.e., global information. The most common TR is the *inverse-document-frequency* (IDF), which penalizes terms occurring frequently across the whole corpus. Usually, $x_{i,j}$ combines one TDR and one TR weight.

Perhaps the most common combination is the TF $\times$ IDF weighting scheme [1, 42]. Although this is the standard scheme, for some tasks this may not be the best choice. For instance, in supervised learning tasks, we have information of labels for training samples. However, standard schemes disregard this useful information. This is due to the fact that traditional schemes were originally proposed for information retrieval (an unsupervised problem) [38, 39]. Because of this, recently supervised weighting schemes have been proposed in the text mining community [7].

The success of the bag of words representation in the natural language processing domain has inspired researchers in computer vision as well, and currently the BoVW is among the most used representations for images and videos [3, 5, 9, 20, 28, 30, 35, 42, 43, 50]. In fact, this formulation has trespassed the image and text boundaries, and it has been used for representing audio [34], time series [46], accelerometer [19] signals, etc. In the computer vision analogy, under the BoVW, an image is represented by a vector indicating the importance of visual words for describing the content of the image. In this scenario, a visual word is a prototypical visual pattern that summarizes the information of other visual descriptors extracted from training images. More specifically, the vocabulary of visual words is typically learned by clustering visual descriptors extracted from training images. The centers of the resultant

[1] One should note the text mining community has proposed variants that aim to soften such assumptions, e.g., using n-grams [2], still the BoW is very competitive with such formulations.

clusters are considered as visual words. Commonly, visual descriptors (e.g., SIFT or HOG) are extracted from points or regions of interest, see [20, 50] for comprehensive descriptions of the BoVW representation.

The effectiveness of the BoVW representation depends on a number of factors, including the interest-point-detection phase, the choice of visual descriptor, the clustering step, and the choice of learning algorithm for the modeling task (e.g., classification) [50]. A factor that has not been deeply studied is the role the term-weighting scheme plays. As in text mining, commonly term-frequency or Boolean term-weighting schemes are considered. Despite the fact these schemes have reported acceptable performance in many tasks (including tasks from natural language processing), it is worth asking ourselves whether alternative schemes can result in better performance. To the best of our knowledge, the only work that aims at exploring this issue is the work by Tirilly et al. [43]. The authors compare the performance of different term-weighting schemes for image retrieval. They considered the most common schemes from information retrieval and provide a comprehensive comparative study. In our work, we focus on classification/recognition tasks and consider weighting schemes specifically designed for classification tasks: supervised weighting schemes. In this paper, we aim to answer such question throughout an extensive experimental evaluation. In addition, we propose a genetic programming algorithm to learn weighting schemes by combining a set of primitives. One should note that there are efforts for improving the BoVW in several directions, most notably, great advances have been obtained for incorporating spatio-temporal information [3, 22, 30, 32, 35]. The term-weighting schemes developed in this work can also be applied in those scenarios.

Term-weighting learning with evolutionary algorithms has been studied within information retrieval and text categorization domains [6, 11, 18]. In [6], authors learn information retrieval weighting schemes with genetic programming. They aim to combine a few primitives trying to maximize average precision. In [11, 18], authors use genetic programming for learning weighting schemes for text classification tasks. This work focuses on learning weighting schemes for computer vision tasks.

3 Common and alternative weighting schemes

As explained above, perhaps the most used weighting scheme for information retrieval and text mining tasks is the so-called TF $\times$ IDF [1, 39]. Although good results have been reported in many applications with it, alternative weighting schemes have been proposed aiming to capture additional or problem-specific information with the goal of improving retrieval or classification performance [1, 7, 26, 44]. For instance, for text classification tasks, supervised term-weighting schemes have been proposed [7, 26]. These alternatives aim at incorporating discriminative information into the representation by defining TR weights that account for the discriminative power of terms. For instance, by replacing the IDF term (in the TF $\times$ IDF scheme) by a discriminative term IG (the *information gain* of the term), resulting in a TF $\times$ IG scheme. Common and alternative weighting schemes are described in Table 1.

The first three weighting schemes in Table 1 are common in text mining and information retrieval, and their usage dates back to the 80s [38], being the Boolean scheme the simplest one (only accounting for the occurrence of terms). On the other hand, the last three schemes were proposed in the last decade and still are not well known within text mining. To the best of our knowledge, these alternative weighting schemes have not been evaluated in the context of computer vision (see Sect. refsec:bow). Therefore, a first contribution of this paper is to assess the suitability of such schemes for computer vision problems. The next section introduces our evolutionary algorithm for learning term-weighting schemes for the BoVW.

4 Evolving visual-word weighting schemes

In addition to the evaluation of non-traditional weighting schemes in computer vision, a second contribution of this work is the proposal of an evolutionary algorithm capable of automatically determining new weighting schemes from scratch. Our proposal is motivated by the following observations. First, we observe that traditional weighting schemes were proposed by researchers based on their own expertise, biases, and needs. Also, so far, it has been the norm to use the same weighting scheme for every data set under analysis. In fact, in computer vision tasks, the weighting scheme is rarely considered a factor that can have an impact on the performance of models based on the BoVW formulation.

In this paper, we address the question of whether the weighting scheme design process can be automated by employing evolutionary algorithms. Our proposed method uses genetic programming to learn how to combine a set of TDR/TR primitives with the aim of obtaining a weighting scheme that optimizes classification performance. This term-weighting-scheme learning formulation removes, to some extent, designers biases and does not rely on user expertise.[2] Instead, weighting schemes are sought such that

[2] Please note that traditional weighting schemes have been proposed by researchers based on their own experiences and biases, making strong assumptions and relying on intuition.

Table 1 Weighting schemes used in text mining and information retrieval

Acr.	Name	Formula	Description	References
B	Boolean	$x_{i,j} = \mathbf{1}_{\{\#(t_i,d_j)>0\}}$	Prescense/abscense of terms	[38]
TF	Term-frequency	$x_{i,j} = \#(t_i, d_j)$	Frequency of occurrence of terms	[38]
TF–IDF	TF-inverse doc. freq.	$x_{i,j} = \#(t_i, d_j) \times \log(\frac{N}{df(t_j)})$	TF penalizing corpus-based frequency	[38]
TF–IG	TF-information gain	$x_{i,j} = \#(t_i, d_j) \times IG(t_j)$	TF times term information gain	[7]
TF–CHI	TF-Chi-square	$x_{i,j} = \#(t_i, d_j) \times CHI(t_j)$	TF times χ^2 term-relevance	[7]
TF–RF	TF-relevance freq.	$x_{i,j} = \#(t_i, d_j) \times \log(2 + \frac{TP}{\max(1,TN)})$	TF times RF relevance	[26]

For every scheme, $x_{i,j}$ indicates how relevant the term t_j is for describing the content of the ith document under the corresponding weighting scheme. N is the number of documents in training data set, $\#(d_i, t_j)$ indicates the frequency of term t_j in the ith document, $df(t_j)$ is document frequency of the term t_j, i.e., the number of documents in which term t_j occurs, $\mathrm{IG}(t_j)$ is the information gain of term t_j, $\mathrm{CHI}(t_j)$ is the χ^2 statistic for term t_j, and TP, TN are the true-positive and true-negative rates for term t_j (i.e., number of positive, respectively, negative, documents that contain term t_j)

they maximize the performance in the task under analysis. Hence, our automatic technique allows us to learn tailored schemes for every data set/task being approached.

Figure 1 presents a general diagram of the proposed approach. A set of primitives is extracted from the BoVW representation of training images. These primitives are obtained by counting visual words occurrence statistics. Next, they feed a genetic program that learns how to combine such primitives to generate a term-weighting scheme. The output of the genetic program is a way to represent images that has been learned automatically. Next, both training and test images are represented according to the learned scheme and, finally, a predictive model is learned and their performance evaluated. The remainder of this section describes our proposed method.

4.1 Genetic programming

Our solution to learn term-weighting schemes is based on Genetic Programming (GP) [27]. GP is an evolutionary algorithm, that is an optimization algorithm inspired by biological evolutionary systems. In evolutionary algorithms, solutions to the problem at hand are seen as individuals that interact among them and with the environment (the search space) in such a way that the survival of the population is sought (optimization criterion). The general flow of a typical evolutionary algorithm is shown in Fig. 2: an initial population of solutions/individuals is created (randomly or by a pre-defined criterion), after that, individuals are selected, recombined,[3] mutated and then placed back into the solutions' pool, this process is repeated for a given number of generations and the algorithm returns the best individual found.

[3] Please note that in GP, for each individual, either mutation or crossover is performed each time, but not both. This is different from other variants like genetic algorithms.

The main distinctive feature of GP, when compared to other evolutionary algorithms, is that in GP, nonlinear and complex data structures are used to represent solutions (individuals). For instance, the most common representations for individuals in GP are trees and graphs, whereas for most of evolutionary algorithms, numerical vectors are used. This feature of GP makes it appropriate for facing very complex problems, in most cases related to modeling tasks. This is one of the reasons for which we adopted GP for learning weighting schemes. Nevertheless, the main motivation for using GP for our problem is that we are interested in learning a function that tells us how to combine the different primitives (including the decision of telling which primitives are worth to combine). In this scenario, GP provides a natural solution to the problem, encoding candidate functions as individuals (i.e., trees) and searching for the best one. Clearly, this problem cannot be approached with either traditional optimization or heuristic optimization techniques.

4.2 GP for term-weighting scheme learning

Our approach to generate weighting schemes uses genetic programming to learn how to combine a set of primitives that have been used for building weighting schemes in the past (see Fig. 1). That is, we devise a genetic program that searches for the combination of primitives that maximizes the classification performance of the task under analysis (e.g., image classification). A standard tree representation is adopted in which leafs correspond to primitives and non-terminal nodes correspond to operators by which primitives can be combined; in such a way that the evaluation of a tree leads to a term-weighting scheme (see Fig. 3).

Therefore, under this formulation, we explore the search space of weighting schemes that can be coded by the trees, where, common/alternative weighting schemes are included in the search space. The remainder of the section elaborates on the different components of the proposed genetic program.

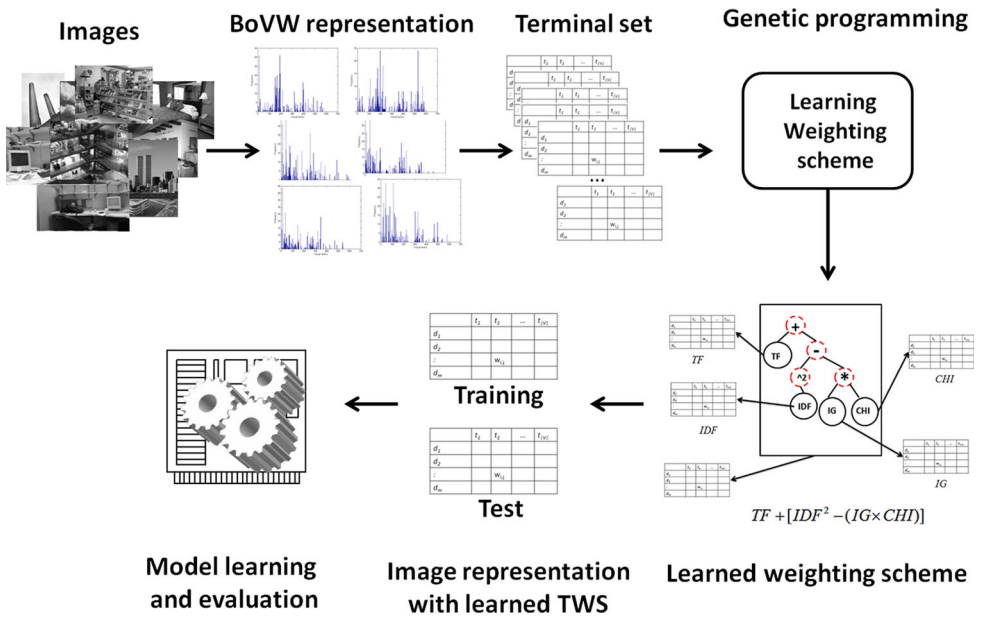

Fig. 1 General diagram of the proposed approach

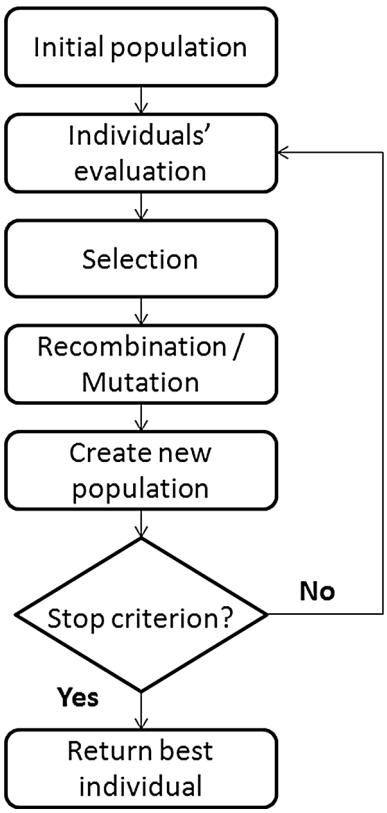

Fig. 2 A generic evolutionary algorithm

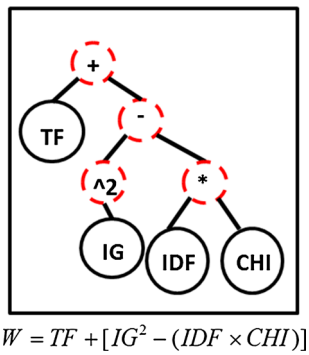

Fig. 3 Adopted representation for individuals. *Dashed nodes* represent operators (taken from the function set) and *solid-line* nodes indicate terminals; below the tree we show the term-weighting scheme derived from it

4.2.1 Representation

As mentioned in Sect. 3, weighting schemes are mainly composed out of two type of factors: TDR an TR weights, which determine the importance of terms into documents and the relevance of terms themselves, respectively. Accordingly, the proposed method uses as terminals TDR and TR primitives (together with useful constants and other weighting schemes), which can be combined by a predefined set of operators. An individual (i.e., solution) in the genetic program is thus a tree formed by these terminals and operators, where the evaluation of the tree leads to a term-weighting scheme. Figure 3 depicts a typical individual and the resultant weighting scheme.

The set of terminals considered in this work is shown in Table 2, whereas for the operators (non-terminals) we considered the function set shown in Table 3.

Each terminal in Table 2 is a matrix of size $N \times |V|$. TDRs are themselves matrices of that dimensions, but TRs are row vectors of length $|V|$ (i.e., they indicate the relevance of each

Table 2 Terminal set

Variable	Meaning
$\mathbf{W}_1$	N, constant matrix, number of training documents
$\mathbf{W}_2$	$\|V\|$, constant matrix, number of terms
$\mathbf{W}_3$	CHI, matrix containing in each row the vector of χ^2 weights for the terms
$\mathbf{W}_4$	IG, matrix containing in each row the vector of information gain weights for the terms
$\mathbf{W}_5$	TF-IDF, matrix with the TF-IDF term-weighting scheme
$\mathbf{W}_6$	TF, matrix containing the TF term-weighting scheme
$\mathbf{W}_7$	FGT, matrix containing in each row the global term-frequency for all terms
$\mathbf{W}_8$	TP, matrix containing in each row the vector of true positives for all terms
$\mathbf{W}_9$	FP, matrix containing in each row the vector of false positives
$\mathbf{W}_{10}$	TN, matrix containing in each row the vector of true negatives
$\mathbf{W}_{11}$	FN, matrix containing in each row the vector of false negatives
$\mathbf{W}_{12}$	Accuracy, matrix where each row contains the accuracy obtained when using the term as classifier
$\mathbf{W}_{13}$	Accuracy_Balance, matrix containing the AC_Balance each (term, class)
$\mathbf{W}_{14}$	Bi-normal separation, BNS, an array that contains the value for each BNS per (term, class)
$\mathbf{W}_{15}$	DFreq, document frequency matrix containing the value for each (term, class)
$\mathbf{W}_{16}$	FMeasure, F-measure matrix containing the value for each (term, class)
$\mathbf{W}_{17}$	OddsRatio, an array containing the OddsRatio term-weighting
$\mathbf{W}_{18}$	Power, matrix containing the Power value for each (term, class)
$\mathbf{W}_{19}$	ProbabilityRatio, matrix containing the probabilityRatio each (term, class)
$\mathbf{W}_{20}$	Max_Term, Matrix containing the vector with the highest repetition for each term
$\mathbf{W}_{21}$	RF, matrix containing the RF vector
$\mathbf{W}_{22}$	TF $\times$ RF, matrix containing TF $\times$ RF

Table 3 Considered function set for the genetic program

Operator	Name	Arity
$+$	Addition	2
$-$	Substraction	2
$*$	Product	2
$/$	Division (protected)	2
$\log_2 x$	Logarithm b-2	1
$\sqrt{x}$	Square root	1
x^2	Square power	1

term). To make all matrices comparable (and henceforth suitable for combination under the function set $\mathcal{F}$), TRs are converted into matrices by repeating the row vector N times. Therefore, all of the operators in the function set act on a scalar basis, that is, they are applied element-by-element. It is worth mentioning that for supervised TR factors, we use information extracted from training images only; i.e., no supervised information is used from the test set.

The initial population is generated with the ramped half-half strategy, which means that half of the population is created with the full method (i.e., all trees have the same deep, *maxdepth*) and the other half is created with the grow method (i.e., trees have deep of at most *maxdepth*), see [27] for details.

4.2.2 Fitness function

The goal of our genetic programming formulation is to obtain a weighting scheme that maximizes classification performance. Therefore, the goodness/fitness of each solution should be tied to the classification performance of a model using the representation induced by the weighting scheme. Specifically, given a solution to the problem, we first evaluate the tree to generate a weighting scheme using the training set, as shown in Fig. 3. Once training documents are represented by the corresponding weighting scheme, we perform a $k-$fold cross-validation procedure, using a given classifier, to assess the effectiveness of the solution. In $k-$fold cross-validation, the training set is split into k disjoint subsets, and k rounds of training and testing are performed; in each round $k-1$ subsets are used as training set and 1 subset is used for testing, the process is repeated k times using a different subset for testing each time. The average classification performance is used as the fitness function.

In particular, we evaluate the performance of classification models with the f_1 measure. Let TP, FP and FN to denote the true positives, false-positives and false-negative rates for a particular class, precision (Prec) is defined as $\frac{TP}{TP+FP}$ and recall (Rec) as $\frac{TP}{TP+FN}$. f_1-measure is simply the

harmonic average between precision and recall: $f_1 = \frac{2\times Prec\times Rec}{Prec+Rec}$. The average across classes is reported (also called, macro-average f_1), this way of estimating the f_1-measure is known to be particularly useful when tackling unbalanced data sets.

Because under the fitness function k models have to be trained and tested for the evaluation of a single solution, we need to look for an efficient classification model. We considered Support Vector Machines (SVM) as they can deal naturally with the sparseness and high dimensionality of data. However, training and testing an SVM can be a time-consuming process. Therefore, we opted for efficient implementations of SVMs that have been proposed recently [10, 51]. Those methods are trained online and under the scheme of learning with a budget. We use the predictions of an SVM as the fitness function for learning term-weighting schemes (TWS). Among the methods available in [10], we used the low-rank linearized SVM (LLSMV) [51]. LLSVM is a linearized version of nonlinear SVMs, which can be trained efficiently with the so-called block minimization framework [4]. We selected LLSVM instead of alternative methods because this method has outperformed several other efficient implementations of SVMs (see [10, 51]). Thus, we use this approximated SVM during the fitness function. Once a weighting scheme has been learned, however, we use a deterministic SVM to classify the test set. This is to make results comparable and discard the randomness inherent to the approximate solutions.

4.2.3 Genetic operators

The proposed genetic program follows a standard procedure as depicted in Fig. 2. We use the implementation from [41], which considers standard operators for crossover and mutation. Specifically, subtree crossover is considered where, given two parent trees, an intermediate node is randomly selected within each tree. Then, the subtrees below the selected nodes are interchanged between the parents, giving rise to two offspring. The mutation operator is quite standard as well, it consists of identifying a node within the parent tree and replacing the node with another randomly selected (terminals replaced by terminals and non-terminals replaced by operators in $\mathcal{F}$).

4.3 Final remarks

After the evolutionary process finishes, the genetic program returns a term-weighting scheme. Next, training and test images are represented according to this scheme. A classifier is learned using the training representation and its performance evaluated in the test representation. For this evaluation, we consider a deterministic SVM (from the CLOP toolbox [37]), and hence, results are comparable to each other. The next section reports experimental results on several computer vision tasks obtained with learned weighting schemes.

5 Experiments and results

This section presents experimental results that aim at showing the effectiveness of the proposed methodology for learning term-weighting schemes in a variety of computer vision tasks. First, we describe the experimental settings and then report results of our study.

5.1 Experimental settings

For experimentation, we considered standard data sets associated with landmark computer vision tasks. The considered data sets are described in Table 4. All of these data sets are associated with classification/recognition tasks, hence the same evaluation protocol (with slight variations described below for each data set) was adopted. For all but one data set, we generated training and test partitions[4]; the exception was the MSRDaily3D data set for which we report average performance over fivefold cross-validation, see below.

In every data set, the training partition was used both to obtain the visual vocabulary and to learn the term-weighting schemes with the genetic program, recall the program maximizes the f_1 measure under $k-$fold cross-validation. For evaluating the performance of the different weighting schemes, both, training and test images are represented with the schemes (either learned or predefined). Then, a classification model is learned using training images and the performance of the model is evaluated in test images.

Unless otherwise stated, we used the VLFEAT toolbox for processing images [45]. We considered PHOW[5] (Pyramid Histogram Of Visual Words) features as visual descriptors [3].

Regarding our proposed genetic program for term-weighting learning, the average and standard deviation performance of 5 runs is reported. The method was run in all cases for 50 generations with a population of 500 individuals. This is a very standard choice for GP [27], where it is common to use large number of individuals and

[4] Matlab files with the predefined partitions are publicly available under request.

[5] PHOW is an extension to the raw BoVW formulation that aims at incorporating spatial information by means of a pyramidal structure, see [3] for details.

Table 4 Data sets considered for experimentation

Data set	Classes	|V|	# Train	# Test	Images|terms
Image categorization					
Caltech-tiny	5	12,000	75	75	15|12000
Caltech-102 (15)	101	12,000	1530	1530	165|3000
Caltech-102 (30)	101	12,000	3060	3060	330|3000
Birds	6	400	540	60	540|400
Butterflies	7	400	552	67	552|400
Action recognition					
MSRDaily3D	12	600	192	48	192|600
Gesture recognition					
Montalbano	20	1000	6850	3579	2055|600
Scene recognition					
15 Scenes	101	12,000	1475	3010	1475|2000
Pornographic image filtering					
Adult	101	12,000	6808	1702	6808|2000

Column 6 shows the number of *images|terms* (i.e., size of the visual vocabulary) considered during the search process

a small number of generations. Default values were used for the remainder of GP parameters: generational selection mechanism with elitism, lexictour parent selection [33], crossover probability of 0.9, and mutation probability of 0.1.

Because the optimization process may be too time-consuming for some data sets, we learned the weighting schemes by using subsets of the original training sets:

- Only samples belonging to a subset of classes were used. In some cases, the vocabulary was also reduced, see Table 4 column 6.
- The selection of classes was done randomly; while the vocabulary reduction used a frequency criterion (the most frequent terms were retained).

Despite this reductions, at the end of the search process, all of the data and classes are considered for training the final classifier and evaluation. We emphasize that during the search process we use an approximate SVM for computing the fitness function. When evaluating the performance of weighting schemes in test set, we used a deterministic linear SVM. Specific details and considerations for each data set are reported below.

Finally, for comparing the statistical-significance of differences we used a Wilcoxon signed-rank test (as recommended in [8]), with a 0.05 confidence level.

5.1.1 Caltech-101

Caltech-101 [16] is a mandatory benchmark for image classification. It contains objects that belong to 101 different categories (102 including the background category). Sample images from this data set are provided in Fig. 4.

For experiments we considered three subsets: tiny, 101-15 and 101-30. Tiny considers 5 out 102 classes with 15 images per-class for training and 15 for testing; data set 101-15 considers the 102 classes with 15 training and 15 testing images (per-class); finally, data set 101-30 considers the 102 classes with 30 images for training and 30 for testing. Using 3 subsets of Caltech-101 allows us to evaluate the performance of our method for similar categorization problems but with different complexities in terms of the number of categories and samples. In fact, we use these subsets of Caltech-101 to assess the generality capabilities of the proposed approach, see below. For tiny, we used all of the samples during the optimization process, whereas for the other two data sets we used examples from 10 category classes and the background only, where the top 3000 terms where considered.

5.1.2 Birds and butterflies

We also considered two data sets related to animal recognition: birds and butterflies. Figure 5 shows sample images from these data sets. In both cases, the problem is to distinguish birds/butterflies species. Contrary to Caltech-101, these data sets comprise more fine-grained classification problems. Therefore, these data sets comprise a major challenge because instances of different classes may be very similar. For these data sets, we represented images under the BoW using a Discrete Cosine Transform (DCT) descriptor. This choice is based on previous work in the same data sets [32]. For both data sets, we used 90 percent of images for training and 10 percent of images for testing.

Fig. 4 Sample images from the Caltech-101 data set

Fig. 5 Sample images from different categories of the Birds and Butterflies data sets

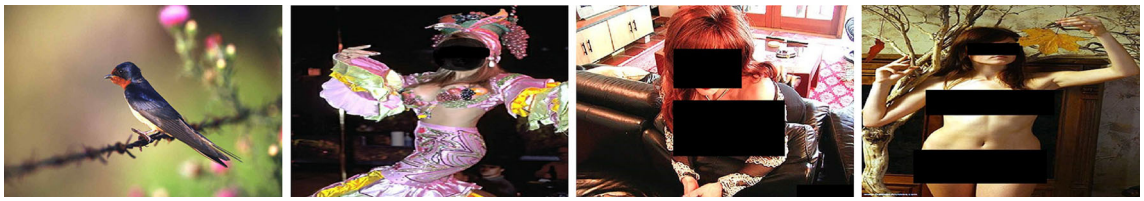

Fig. 6 Sample images from the data set of adult image filtering. The categories are (from *left* to *right*): inoffensive images, lightly dressed persons, partly nude persons, nude persons, and pornographic images (not shown)

5.1.3 Adult image filtering

A data set for adult image filtering was considered as well. The data were made available by [9], and it has been previously used in several publications, see [9, 49]. The data set contains images belonging to five categories, where there is one category for inoffensive images and four categories of increasing level of *adultness*: lightly dressed, partly nude, nude and pornographic, see Fig. 6.

The goal in this task is to associate images with its correct category in such a way that the administrator of a filtering system can decide the level of restriction in the type of images users can have access to (e.g., photographs of lightly dressed persons may be allowed in most sites, even in schools, but nude persons and pornography may be objectionable in most sites). About 80 % of images were used as training set and the remainder as test set, as in [9].

5.1.4 Scene recognition

We consider a benchmark data set for scene recognition [30]. The data set comprises 15 indoor/outdoor categories, where images contain complex scenes. Figure 7 shows sample images from this data set, clearly this is a very challenging task. For this data set, we used the same partitioning proposed in [30]: 100 images per category for training and the rest for testing.

5.1.5 Montalbano

The BoVW has been used to represent videos as well, see e.g., [22, 28, 42]. For this reason, we also decided to include video data sets. Specifically, we considered the Montalbano data set for gesture recognition as provided in [15]. The task consists of recognizing gestures from 20 categories (Italian cultural gestures), see Fig. 8. The available data is depth and RGB video together with skeleton information. For our experiments, we used the features proposed in [36], which combine depth, RGB video and skeleton information by means of convolutional nets and other deep-learning mechanisms. The deep-learning features were clustered and the vocabulary was built. One should note that we approach the gesture recognition problem, that is, given a segmented gesture, to tell the class of the gesture being performed.

Fig. 7 Sample images from the 15-Scenes data set. Categories are from *left* to *right* and from *up* to *bottom*: *bedroom, suburb, industrial, kitchen, living room, coast, forest, highway, inside city, mountain, open country, street, tall building, office and store*

5.1.6 MSRDaily3D

Finally, we considered a benchmark data set for action recognition: MSRDaily3D. This data set comprises 16 actions associated with daily activities, where there are objects in the background and most actions involve human–object interaction. A sample sequence from this data set is shown in Fig. 9. For this data set, we adopted the protocol from [23–25, 48]. Under this setting, we considered 12 out of the 16 actions and performed fivefold cross-validation. We adopted this protocol because it has been adopted in recent work that uses the BoW representation [23–25, 48]; therefore, we can compare the performance of our method with such works. Video sequences were represented with Depth Cuboid Similarity Features (DCSF) and the same parameters for the descriptor as in previous work were used. Descriptors were further processed to represent videos with their bag of features representation.

5.2 Experimental results

Table 5 shows the results obtained by the different weighting schemes (traditional, alternative-supervised and learned) in all of the considered data sets. We report average f_1-measure performance in the test partitions. The * symbol indicates a statistically significant difference between our approach and the method from the corresponding columns.

It can be seen from this table that, in average, the Boolean weighting scheme (column 3) outperforms both, traditional and alternative, term-weighting schemes. This is an interesting result, because, most of the times (normalized) TF or TF-IDF weighting schemes are considered in computer vision tasks. Please note that although the Boolean scheme is the best on average, it is clear from Table 5 that there is no single best weighting scheme for all of the data sets.

Regarding alternative-supervised term-weighting schemes, only TF-RF obtained comparable performance to the TF scheme; however, its performance was lower than the Boolean scheme. The other two supervised schemes performed worse than the baseline. These results are somewhat disappointing, because, intuitively, the incorporation of discriminative information should yield better performance. In spite of these results, our study comparing traditional and alternative weighting schemes is a contribution that brings some light on the performance of such schemes for diverse computer vision tasks. More importantly, we showed the adequacy of the Boolean scheme.

On the other hand, it is clear from Table 5 that the proposed approach for learning visual-word weighting schemes outperforms all the other variants in all of the considered data sets (see column 8, recall for our method we are reporting the average of 5 runs, that is why we report average and standard deviation of performance). For most of the data sets, our GP-based solution improves considerably the performance of all of the other weighting schemes, in fact, the differences in performance between our method and the rest are statistically significant. The average improvement of our genetic program over the Boolean scheme was of around 5 %; we think this improvement makes worth applying our method instead of relying on standard weighting schemes. These results show that, if searched properly, weighting schemes that maximize classification performance may result in improved performance; this is in contrast to using discriminative information by using IG, CHI, etc.

Higher improvements were observed for image categorization and adult image filtering data sets. Whereas marginal improvements were observed for Montalbano and MSRDaily, although results reported for these data sets are quite competitive with the state of the art, see e.g., [15, 25, 48]. The latter behavior can be due to the fact that the descriptors used for these data sets are very discriminative as reported in [15, 36, 48]. In those cases, it may be enough to verify the presence/absence of such discriminative patterns. This is not the case of image categorization data sets for which standard descriptors were used.

In addition to the competitive average performance, it is quite interesting that the standard deviation across runs is

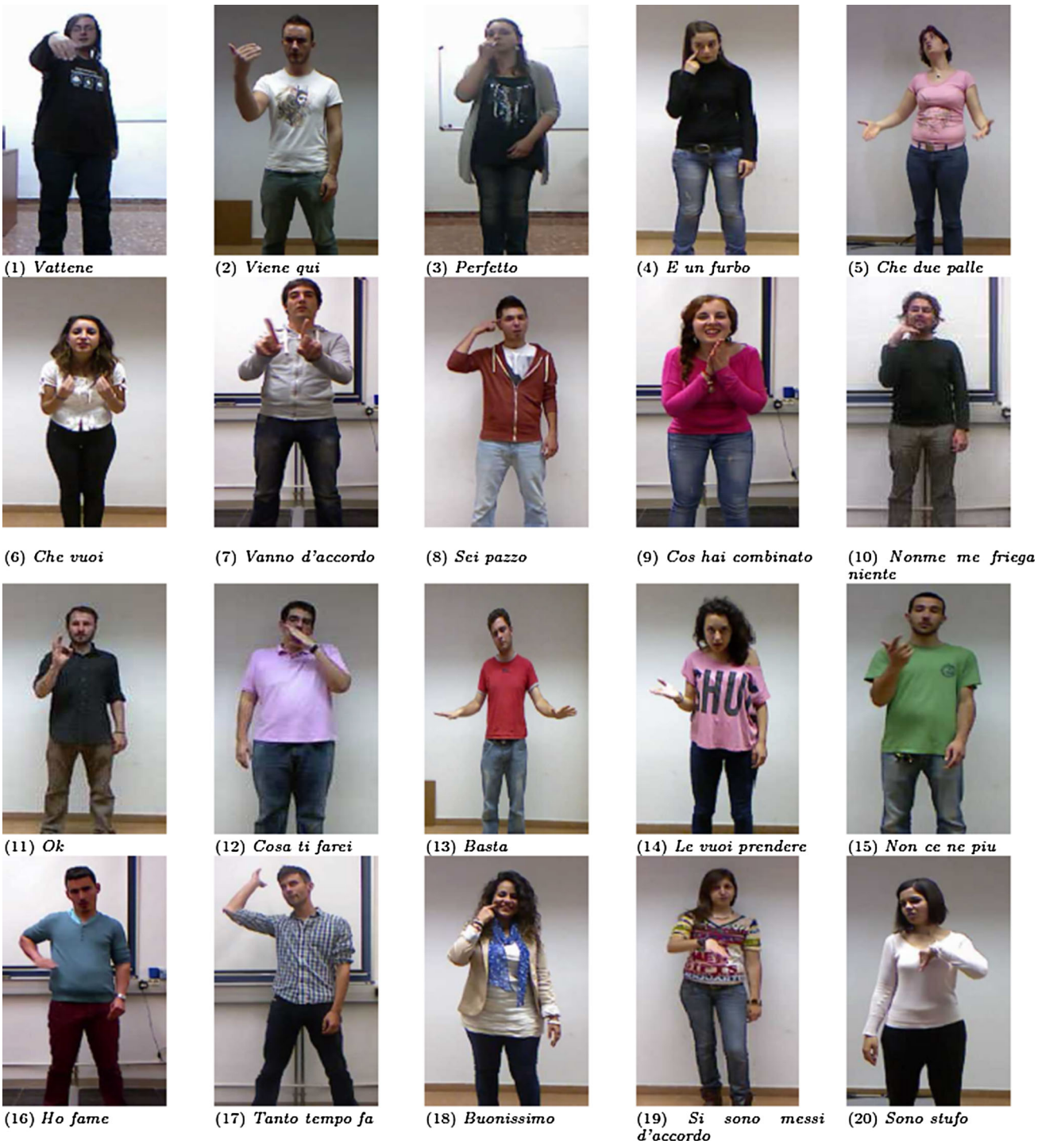

Fig. 8 Sample images from the Montalbano data set. Images from each of the gesture categories are shown [15]

relatively low when compared to the other methods. Thus, evidencing the stability and robustness of the proposed method.

In order to better appreciate the improvements offered by our method, Fig. 10 shows the range of improvement of our method over the best traditional/alternative weighting scheme per data set in terms of absolute and relative differences. That is, we plot the difference in performance between our method (column 8) and the best result among columns 2–7 for each particular data set. This means that our method is not compared with the best scheme in average, but with the best overall for each data set, a somewhat unfair comparison for our approach.

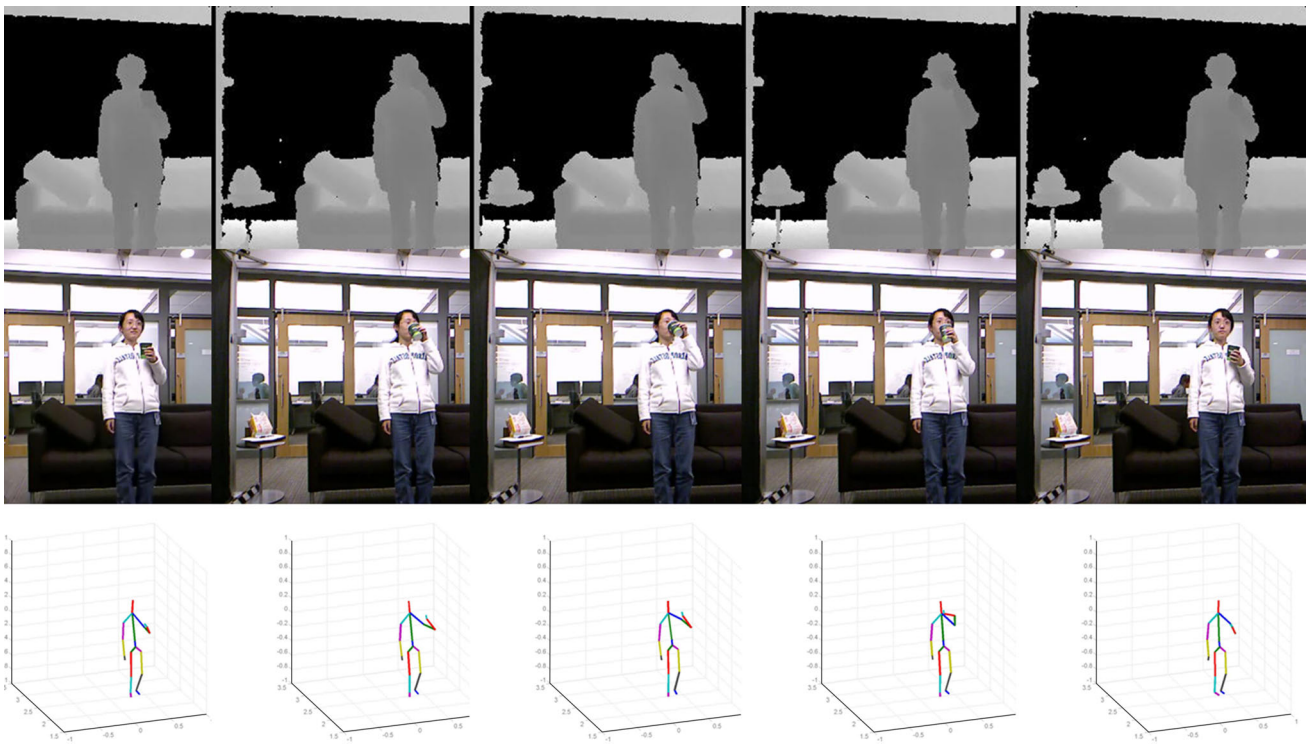

Fig. 9 Sample sequence from the MSRDaily3D data set [47]

Table 5 Classification performance obtained with traditional, alternative and learned weighting schemes

Data set/TWS	Traditional			Alternative-supervised			Learned
	TF (baseline)*	Bol.*	TF-IDF*	TF-RF* [26]	TF-CHI* [7]	TF-IG* [7]	GP (ours)
Tiny	85.65	84.01	76.72	85.65	78.85	80.49	90.75 ± 1.56
101-15	52.26	58.43	48.08	52.30	52.00	51.43	61.05 ± 1.12
101-30	56.61	59.28	49.95	56.68	54.63	52.03	63.04 ± 1.02
Birds	44.68	48.53	30.55	44.68	44.6	43.95	52.95 ± 5.11
Butterflies	26.07	41.44	20.45	26.07	26.08	26.75	42.12 ± 3.07
Adult	52.53	58.35	55.39	52.53	46.39	47.23	62.68 ± 2.08
15 scenes	59.12	61.26	56.51	59.12	55.02	55.07	63.43 ± 0.16
Montalbano	88.55	86.46	88.49	88.55	88.5	88.58	88.79 ± 0.12
MSRDaily3D	75.22 ± 4.2	68.0 ± 6.22	74.72 ± 4.47	75.058 ± 3.9	73.94 ± 5.65	73.77 ± 4.9	76.01 ± 4.01
Average	54.34 ± 22.06	56.91 ± 18.78	50.81 ± 22.38	54.33 ± 22.04	52.46 ± 21.04	52.51 ± 21.11	61.45 ± 18.67

The * symbol indicates a statistically significant difference between our approach and the method from the corresponding columns

From Fig. 10, it can be seen that the GP-based method offers considerable improvements for all but for the Montalbano data set. The difficulty of this task may require running the genetic program using the whole number of classes/samples (for this data set, we used only a third of the total of instances, see column 6 in Table 4). However, as mentioned above, we think this low improvement is due to the very effective visual descriptors over which the BoW representation was generated.

One should note that the proposed method relies on an iterative optimization process that is somewhat computationally expensive. In particular, the adopted representation (tree-based structure), the fact that the terminals are associated with matrices and the estimation of the fitness function[6] (training and testing an SVM classifier under a cross-validation) are the main factors that contribute to

[6] Please note that estimating the fitness function is quite efficient, as it is based on a fast approximation to a linear SVM. So this method can be used for most computer vision applications. Also, we emphasize that the fitness function is only estimated during the learning process, which has to be done a single time and most of the times is performed offline.

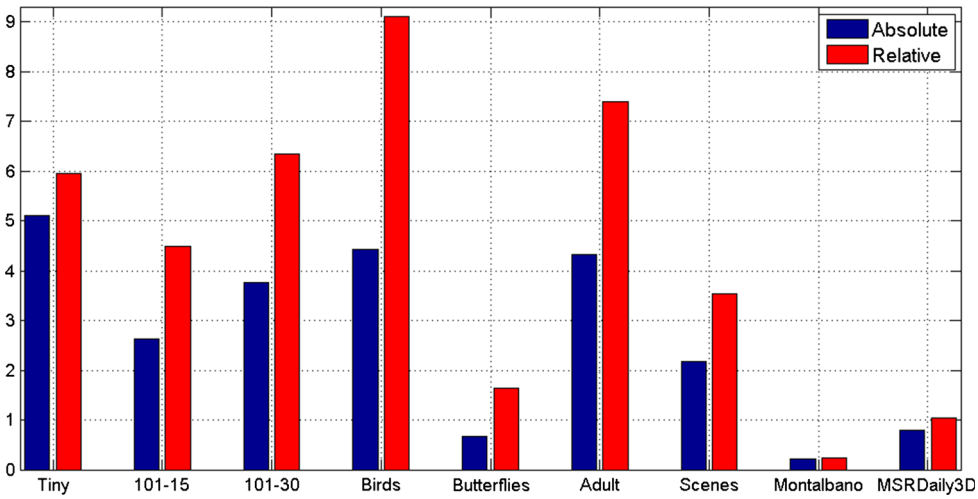

Fig. 10 Absolute (*blue-first bar*) and relative (*red-right bar*) improvement for the different data sets, taking as reference the best traditional/alternative weighting scheme for each data set (color figure online)

Table 6 Sample weighting schemes learned with the proposed approach for selected data sets

ID	Data set	Learned TWS	Formula
1	Caltech101-15	sqrt{[sqrt(RF × TF) + log 2(RF × TF)]}	$\sqrt{\sqrt{W_{22}} + \log 2(W_{22})}$
2	Birds	log 2{[FMeas × (CHI × log 2(TF × RF))]}	$\log 2(W_{16} \times (W_3 \times \log 2(W_{22}))$
3	MSRDaily3D	[(TF × FN) × sqrt(T)]	$((W_6 \times W_{11}) \times \log 2(\sqrt{W_{22}}))$
4	Adult	(sqrt(IDF) × D)	$(\sqrt{W_5} \times D)$
5	Montalbano	log 2[log 2(CHI)] × sqrt(IDF)	$(\log 2(\log 2(W_3)) \times \sqrt{W_5})$
6	15-Scenes	log 2(ProbR + TF × RF)	$\log 2(W_{19} + W_{22})$

In column 2 each weighting is shown as a prefix expression. The names of the variables are self-explanatory. Column 3 shows the mathematical expression of each TWS using the terminal set from Table 2

the computational expensiveness of our model. Nevertheless, in practice, the average running time of the proposed method takes of the order of a few hours. Thus, although the proposed method is somewhat computationally expensive, the average running time is acceptable for most computer vision applications. Please note that the process of learning weighting schemes is a procedure that is performed offline and has to be done a single time. Therefore, we think it is worthwhile spending a few hours using our method, given the potential improvement in performance that can be obtained. On the other hand, one may argue that alternative weighting schemes are less complex (and henceforth require of less processing time to generate the representation). We think this time is negligible, because it involves only a few additional arithmetic operations over more matrices (which are also computed a single time).

5.3 Qualitative analysis

This section presents a qualitative study on the proposed method for learning term-weighting schemes. Table 6 shows sample schemes learned for selected data sets. It can be seen that all of the learned schemes included primitives that capture from supervised information, thus showing the importance of such supervised components. Therefore, we can say that the proposed method effectively learns to combine supervised building blocks that result in competitive weighting schemes. This is in contrast with alternative-supervised schemes that showed limited performance (see Table 5).

From Table 6, it can be seen that the learned weighting schemes are indeed simple expressions (opposed to standard GP solutions that include very complex trees). This is a desirable property that suggests overfitting is not an issue for the proposed method.

Finally, it is interesting to note that very different weighting schemes were obtained for the different data sets, thus giving evidence that a tailored weighting scheme is required for each task.

Figure 11 shows the frequency of use of each of the terminals from Table 2 in the solutions returned by the genetic program for all of the data sets (i.e., a bar in Fig. 11 corresponds to a row in Table 2). It can be seen that three most used terminals are W_6, W_{22} and W_5, which correspond to TF, TF × RF and TF-IDF weighting schemes. This is interesting because, even when these were the most chosen terminals by solutions returned with the genetic program, such terminals were significantly outperformed by our

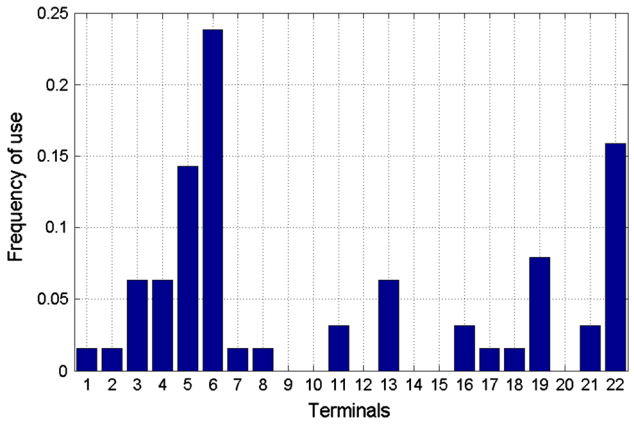

Fig. 11 Frequency of appearance of terminals into the solutions found by the genetic program, see Table 2 for terminals description

proposal: compare columns 2, 4 and 5 to column 8 in Table 5.

Only 6 out of the 22 terminals did not appear in solutions returned by the genetic program. All of these terminals, ($W_{9,10,12,14,15,20}$) corresponding to TR weights, are mainly used for feature selection in text classification [17]. Although they have proved to be very effective in [17] (terminal W_{14} was the best criterion for feature selection in that study), they were not very helpful for building term-weighting schemes for computer vision tasks.

6 Conclusions

The BoVW is one of the most used representations in computer vision tasks. Despite being very effective, it is somewhat surprising that little research has been performed on term-weighting schemes for computer vision. In this direction, this paper introduced a novel methodology for learning weighting schemes to boost the performance of classification models relying on the BoVW. The proposed methodology resulted very effective in a wide variety of computer vision tasks. Additionally, we report an in-depth study on the performance of standard and alternative weighting schemes commonly used in text mining. To the best of our knowledge, our work is the first that assesses alternative weighting schemes, and it is the first in proposing methods to learn weighting schemes for computer vision tasks. From our extensive experimental study, comprising 9 data sets of common computer vision task we can conclude the following:

- Among traditional and alternative weighting schemes, the Boolean one obtained the highest performance.
- Weighting schemes learned with our proposed approach outperformed consistently all other weighting schemes in all of the data sets.
- For different tasks, learning a term-weighting scheme with the proposed approach is much better than applying other schemes (either traditional/alternative or learned for another data set).
- Computer vision tasks that are not too generic (e.g., gesture recognition or adult image filtering) require of tailored weighting schemes, accordingly, schemes learned for this data sets do not generalize well in other data sets.
- Among all of the considered terminals, three weighting schemes were used most often by solutions returned by the genetic program (TF, TF-IDF and TF-RF), however, the way in which the genetic program combined such primitives resulted in much better performance.

Future work includes studying alternative methodologies for learning term-weighting schemes. Specifically, we plan to pose the problem as one of learning/optimizing the representation matrix, where other evolutionary algorithms could be used. Also, we are interested in learning term-weighting schemes for other domains, like audio [34], time series [46] or accelerometer data [19], and other scenarios as one-shot recognition [21] and early classification [14].

Acknowledgments This work was supported by CONACyT under Project Grant No. CB-2014-241306 (*Clasificación y recuperación de imágenes mediante técnicas de minería de textos*) and Spanish Ministry of Economy and Competitiveness TIN2013-43478-P. Víctor Ponce-López is supported by Fellowship No. 2013FI-B01037 and Project TIN2012-38187-C03-02.

References

1. Baeza-Yates R, Ribeiro-Neto B (1999) Modern information retrieval. Addison-Wesley, Boston
2. Bekkerman R, Allan J (2004) Using bigrams in text categorization. Technical Report, Department of Computer Science. University of Massachusetts, Amherst, vol 1003, pp 1–2
3. Bosch A, Zisserman A, Munoz X (2007) Image classification using random forests and ferns. In: Proceedings of the ICCV
4. Chang KW, Roth D (2011) Selective block minimization for faster convergence of limited memory large-scale linear models. In: SIGKDD conference on knowledge discovery and data mining. ACM
5. Csurka G, Dance CR, Fan L, Willamowski J, Bra C (2004) Visual categorization with bags of keypoints. In: International workshop on statistical learning in computer vision
6. Cummins R, O'Riordan C (2006) Evolving local and global weighting schemes in information retrieval. Inf Retr 9:311–330
7. Debole F, Sebastiani F (2003) Supervised term-weighting for automated text categorization. In: Proceedings of the 2003 ACM symposium on applied computing, SAC '03. ACM, New York, pp 784–788
8. Demsar J (2006) Statistical comparisons of classifiersover multiple data sets. J Mach Learn Res 7:1–30
9. Deselaers T, Pimenidis L, Ney H (2008) Bag of visual words for adult image classification and filtering. In: Proceedings of the international conference on pattern recognition. IEEE

10. Djuric N, Lan L, Vucetic S, Wang Z (2013) Budgetedsvm: a toolbox for scalable svm approximations. J Mach Learn Res 14:3813–3817
11. Escalante HJ, Garcia M, Morales A, Graff M, Montes M, Morales EF, Martinez J (2015) Term-weighting learning via genetic programming for text classification. Knowl Based Syst 83:176–189
12. Escalante HJ, Martinez-Carranza J, Escalera S, Ponce-López V, Baró X (2015) Improving bag of visual words representations with genetic programming. In: Proceedings of the 2015 international joint conference on neural networks. IEEE, pp 3674–3681
13. Escalante HJ, Montes M, Sucar E (2012) Semantic cohesion for image annotation and retrieval. Comput Sist 10(1):121–126
14. Escalante HJ, Sucar E, Morales E (2016) A naive bayes baseline for early gesture recognition. Pattern Recogn Lett 73:91–99
15. Escalera S, Baro X, Gonzalez J, Bautista MA, Madadi M, Reyes M, Ponce V, Escalante HJ, Shotton J, Guyon I (2014) ChaLearn looking at people challenge 2014: dataset and results. In: Proceedings of ECCV—chalearn workshop
16. Fei-Fei L, Fergus R, Perona P (2004) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: Proceedings of the IEEE, CVPRW
17. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3:1289–1305
18. García-Limón M, Escalante HJ, Montes y Gómez M, Morales A, Morales E (2014) Towards the automated generation of term-weighting schemes for text categorization. In: Procddings of GECCO Comp'14, (Late-breaking abstract), pp 1459–1460
19. Gonzalez-Gurrola LC, Moreno R, Escalante HJ, Martnez F, Carlos R (2015) Learning roadway surface disruption patterns using the bag of words representation. IEEE transactions on intelligent transportation systems (under review)
20. Grauman K, Leibe B (2010) Visual object recognition. Morgan and Claypool, San Rafael
21. Guyon I, Athitsos V, Jangyodsuk P, Escalante HJ (2014) The Chalearn gesture dataset (CGD 2011). Mach Vis Appl 25(8):1929–1951
22. Hernández-Vela A, Bautista MA, Perez-Sala X, Ponce-López V, Escalera S, Baró X, Pujol O, Angulo C (2014) Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in rgb-d. Pattern Recognit Lett 50(1):112–121
23. Hoai M, De la Torre F (2012) Max-margin early event detectors. In: IEEE conference on computer vision and pattern recognition. IEEE, Providence, RI, pp 2863–2870
24. Hoai M, Lan Z, De la Torre F (2011) Joint segmentation and classification of human actions in video. In: IEEE conference on computer vision and pattern recognition. IEEE, Providence, RI, pp 3265–3272
25. Huang D, Yao S, Wang Y, De La Torre F (2014) Sequential max-margin event detectors. In: European conference on computer vision
26. Lan M, Tan CL, Su J, Lu Y (2009) Supervised and traditional term-weighting methods for automatic text categorization. Trans PAMI 31(4):721–735
27. Langdon WB, Poli R (2001) Foundations of genetic programming. Springer, Berlin
28. Laptev I (2005) On space-time interest points. Int J Comput Vis 64(2–3):107–123
29. Lazebnik S, Schmid C, Ponce J (2004) Semi-local affine parts for object recognition. In: British machine vision conference, pp 779–788
30. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the computer vision and image processing conference, IEEE, pp 2169–2178
31. Lazebnik S, Schmid C, Ponce JA (2015) Maximum entropy framework for part-based texture and object recognition. In: IEEE international conference on computer vision, pp 832–838
32. Lopez-Monroy AP, Montes y Gomez M, Escalante HJ, Cruz-Roa A, Gonzalez FA (2015) Improving the bovw with discriminative n-grams and mkl. Neurocomputing 175:768–781
33. Luke S, Panait L (2002) Lexicographic parsimony pressure. In: Proceedings of the 2002 genetic and evolutionary computation conference, pp 829–836
34. Manchala S, Prasad VK, Janaki V (2014) Gmm based language identification system using robust features. Int J Speech Technol 17:99–105
35. Mirza-Mohammadi M, Escalera S, Radeva P(2009) Contextual-guided bag-of-visual-words model for multi-class object categorization. In: Proceedings of the CAIP. Springer, pp 748–756
36. Neverova N, Wolf C, Taylor GW, Nebout F (2014) Multi-scale deep learning for gesture detection and localization. In: Proceedings of the ECCV chalearn workshop on looking at people
37. Saffari A, Guyon I (2006) Quick start guide for clop. Technical report, TU Graz—CLOPINET
38. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manag 24:513–523
39. Sebastiani F (2008) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47
40. Sidorov G, Gelbukh A, Gomez-Adorno H, Pinto D (2014) Soft similarity and soft cosine measure: similarity of features in vector space model. Comput Sist 18(3):491–504
41. Silva S, Almeida J (2003) Gplab-a genetic programming toolbox for matlab. In: Proceedings of the Nordic MATLAB conference, pp 273–278
42. Sivic J, Zisserman A (2003) Video google: a text retrieval approach to object matching in videos. Int Conf Comput Vis 2:1470–1477
43. Tirilly P, Claveau V, Gros P (2009) A review of weighting schemes for bag of visual words image retrieval. Technical report, IRISA
44. Turney P, Pantel P (2010) From frequency to meaning: vector space models of semantics. J Artif Intell Res 37:141–188
45. Vedaldi A, Fulkerson B (2010) VLFeat: an open and portable library of computer vision algorithms. In: Proceedings of the 18th ACM international conference on multimedia. ACM, pp 1469–1472
46. Wang J, Liu P, She FH, Nahavandi M, Kouzani A (2013) Bag-of-words representation for biomedical time series classification. Biomed Signal Process Control 8(6):634–644
47. Wang J, Liu Z, Wu Y, Yuan J (2012) Mining actionlet ensemble for action recognition with depth cameras. In: IEEE conference on computer vision and pattern recognition. IEEE, Providence, RI, pp 1290–1297
48. Xia L, Aggarwal JK (2013) Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: IEEE conference on computer vision and pattern recognition. IEEE, Portland, OR, pp 2834–2841
49. Yoo SJ (2004) Intelligent multimedia information retrieval for identifying and rating adult images. In: Proceedings of the international conference KES, vol 3213 of LNAI, pp 164–170. Springer
50. Zhang J, Marszablek M, Lazebnik S, Schmid C (2007) Local features and kernels for classification of texture and object categories: a comprehensive study. Int J Comput Vis 73(2):213–238
51. Zhang K, Lan L, Wang Z, Moerchen F (2012) Scaling up kernel svm on limited resources: A low-rank linearization approach. In: Proceedings of th AISTATS 2012