Improving Bag of Visual Words Representations with Genetic Programming

Hugo Jair Escalante*, José Martínez-Carraza*, Sergio Escalera† Víctor Ponce-López†,‡ Xavier Baró‡
* Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, 72840, Mexico
Email: {hugojair,carranza}@inaoep.mx
† University of Barcelona Spain, Computer Vision Center, Spain
Email: sergio@maia.ub.es
‡ Universitat Oberta de Catalunya, Spain, Computer Vision Center, Spain
Email: {vponcel,xbaro}@uoc.edu

***Abstract*—The bag of visual words is a well established representation in diverse computer vision problems. Taking inspiration from the fields of text mining and retrieval, this representation has proved to be very effective in a large number of domains. In most cases, a standard term-frequency weighting scheme is considered for representing images and videos in computer vision. This is somewhat surprising, as there are many alternative ways of generating bag of words representations within the text processing community. This paper explores the use of alternative weighting schemes for landmark tasks in computer vision: image categorization and gesture recognition. We study the suitability of using well-known supervised and unsupervised weighting schemes for such tasks. More importantly, we devise a genetic program that learns new ways of representing images and videos under the bag of visual words representation. The proposed method learns to combine term-weighting primitives trying to maximize the classification performance. Experimental results are reported in standard image and video data sets showing the effectiveness of the proposed evolutionary algorithm.**

I. Introduction

The bag of visual words representation is a state-of-the-art methodology for effectively describing the content of images and videos in computer vision tasks [1]. Inherited from the text processing domain, this representation captures information about the occurrence of patterns from a previously learned codebook. Usually, an image is represented by an histogram that accounts for the occurrence of elements of the codebook (i.e., the visual words), see e.g., Section 11.3 in [2]. This way of representing images has proved to be successful in a large number of applications and domains [1], [3], [4], [5], [6], [7], [8], [9]. Although this representation is very popular, it is somewhat surprising that in most studies a term-frequency weighting scheme is considered for representing images (i.e., the representation simply accounts for the frequency of occurrence of visual words). Since the information retrieval and text mining communities have a long tradition with representing documents with bags of words [10], [11], it is worth asking ourselves, whether alternative weighting schemes proposed in the previous communities could be beneficial for computer vision tasks as well. This paper focuses in this problem.

Specifically, we explore in this paper the suitability of common unsupervised and supervised term-weighting schemes, that have been proposed for information retrieval and text categorization, for representing images and videos under the bag of visual words. More importantly, we devise an evolutionary algorithm that learns new weighting schemes for representations based on visual words. The evolutionary algorithm searches the space of possible weighting schemes that can be generated by combining a set of primitives, this is with the aim of maximizing the classification/recognition performance. A working hypothesis of our work is that weighting schemes alternatives to the traditional term-frequency one may lead to better performance. We perform experiments in landmark problems in computer vision, namely: image categorization (different subsets of the Caltech-101 data set [12]), gesture recognition (the newly introduced Montalbano data set [13]), places-scene recognition (the well known 15-scenes [5]), and adult image classfication [14]. Results confirm our hypothesis and motivate further research in this direction.

The remainder of this paper is organized as follows. Next section describes the bag of visual words representation. Section III elaborates on alternative schemes from the information retrieval and text categorization fields. Section IV introduces the evolutionary technique to learn weighting schemes. Section V presents our experimental study. Finally, Section VI outlines conclusions and future work directions.

II. The bag of visual words representation

In text mining and information retrieval, the bag of words representation is a way to map documents into a vector space, with the aim that such space captures information about the semantics and content of documents. 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 to the corpus under analysis. Each element of this vector indicates the relevance/importance of the corresponding term for describing the content of the document. Although the bag of words makes strong assumptions (e.g., that word order is not important), it is still one of the most used representations nowadays.

Formally, the i^{th} 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 relevance of term t_j for describing the content of the i^{th} document; V is the vocabulary, i.e., 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 weight (IDF), which penalizes terms occurring frequently along the corpus. Usually, $x_{i,j}$ combines a TDR and a TR weight; perhaps the most common combination is the $TF \times IDF$ weighting scheme [15], [1].

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 bag of visual words is among the most used representations for images and videos [2], [9], [1], [5], [6], [4], [7], [3], [8], [16]. In analogy, under this representation an image is represented by a vector indicating the relevance of visual words for representing the content of the image. Where a visual word is a prototypical visual pattern that summarizes the information of other visual descriptors extracted from training images. More specifically, the visual words vocabulary is typically learnt by clustering visual descriptors extracted from training images. The centers of the resultant clusters are considered as visual words. Commonly, visual descriptors (e.g., SIFT) are extracted from points or regions of interest, see [2], [3] for comprehensive descriptions of the bag of visual words representation.

The success of this representation in computer vision 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 target modeling task (e.g., classification) [3]. A factor that has not been deeply studied is the role that the term-weighting scheme plays. Commonly, term-frequency or Boolean term-weighting schemes are considered in computer vision tasks. 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. [9]. 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. In addition, we propose a genetic program to learn weighting schemes by combining a set of primitives. One should note that there are efforts for improving the bag of visual words in several directions, most notably, great advances have been obtained for incorporating spatial information [6], [4], [5], [17], [18]. The term-weighting schemes developed in this work can also be applied with the previous extensions.

Term-weighting learning with evolutionary algorithms has been studied within information retrieval and text categorization domains [19], [20], [21]. In [19] the authors learn information retrieval weighting schemes with genetic programming, they aim to combine a few primitives trying to maximize average precision. In [20], [21] authors use genetic programming for learning weighting schemes for text classification tasks. This work focuses on learning weighting schemes for computer vision tasks.

III. Alternative weighting schemes

As explained above, the most used weighting scheme for information retrieval and text mining tasks is the so called $TF \times IDF$ [15], [11]. 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 [22], [23], [24], [15]. For instance, for text classification tasks, supervised term-weighting schemes have been proposed [22], [23]. 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 I.

To the best of our knowledge, alternative weighting schemes from Table I have not been evaluated in the context of most computer vision tasks (see Section II). Therefore, a contribution of this paper is to assess the suitability of such schemes for landmark computer vision problems. Next we describe our proposed approach for automatically learning term weighting schemes.

IV. Learning weighting schemes for visual words

So far we have described standard and alternative weighting schemes used in text mining and information retrieval (see Table I), in Section V we evaluate the performance of such schemes in computer vision problems. Although these are among the most popular schemes, one should note that they have been *manually* proposed by researchers based on their own expertise, biases, and needs; where 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 affect the performance of models based on the bag of visual words formulation. In this paper, we propose a novel approach for automatically generating weighting schemes for computer vision tasks. Our proposed method uses genetic programming to combine a set of TDR/TR primitives with the aim of obtaining a weighting scheme that outperforms traditional ones. Our approach removes biases of designers and does not rely on user expertise. Instead, weighting schemes are sought such that 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 (step 1 in Figure 1) is extracted from training images. These primitives are obtained by counting visual word occurrence statistics. Next, they are feed into a genetic program that learns how to combine such primitives to give rise to term-weighting scheme (step 2). The output of the genetic program is a way to represent images that has

TABLE I. COMMON TERM WEIGHTING SCHEMES USED IN TEXT MINING AND INFORMATION RETRIEVAL. IN EVERY SCHEME, $x_{i,j}$ INDICATES HOW RELEVANT TERM t_j IS FOR DESCRIBING THE CONTENT OF THE i^{th} 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 i^{th} DOCUMENT, $df(t_j)$ IS THE NUMBER OF DOCUMENTS IN WHICH TERM t_j OCCURS, $IG(t_j)$ IS THE INFORMATION GAIN OF TERM t_j, $CHI(t_j)$ 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., THE NUMBER OF POSITIVE, RESP. NEGATIVE, DOCUMENTS THAT CONTAIN TERM t_j).

Acronym	Name	Formula	Description	Ref.
B	Boolean	$x_{i,j} = \mathbf{1}_{\{\#(t_i,d_j)>0\}}$	Indicates the prescense/abscense of terms	[10]
TF	Term-Frequency	$x_{i,j} = \#(t_i, d_j)$	Accounts for the frequency of occurrence of terms	[10]
TF-IDF	TF - Inverse Document Frequency	$x_{i,j} = \#(t_i, d_j) \times \log(\frac{N}{df(t_j)})$	An TF scheme that penalizes the frequency of terms across the collection	[10]
TF-IG	TF - Information Gain	$x_{i,j} = \#(t_i, d_j) \times IG(t_j)$	TF scheme that weights term occurrence by its information gain across the corpus	[22]
TF-CHI	TF - Chi-square	$x_{i,j} = \#(t_i, d_j) \times CHI(t_j)$	TF scheme that weights term occurrence by its χ^2 statistic	[22]
TF-RF	TF - Relevance Frequency	$x_{i,j} = \#(t_i, d_j) \times \log(2 + \frac{TP}{\max(1,TN)})$	TF scheme that weights term occurrence by its RF relevance	[23]

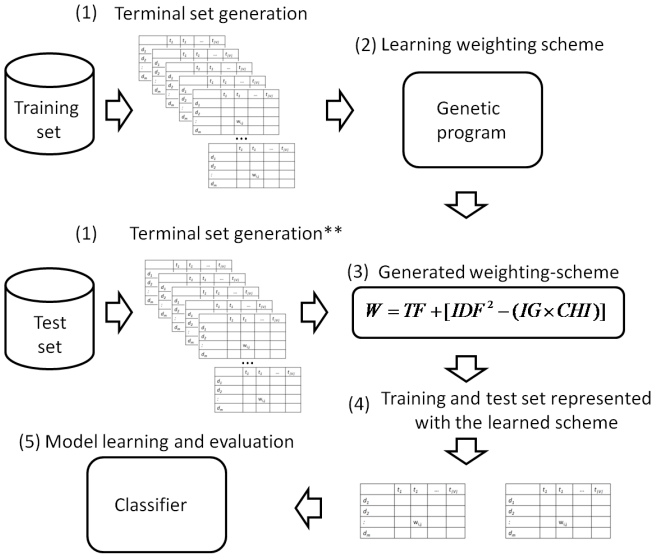

Fig. 1. General diagram of the proposed approach. ** Information from the classes in the test set is not used.

been learnt automatically (step 3). Next, both training and test images are represented according to the learned scheme (step 4) and, finally, a predictive model is learned and their performance evaluated. The rest of this section describes our proposed method.

A. Genetic programming

Our solution to learn term-weighting schemes is based on genetic programming (GP) [25]. GP is an optimization algorithm that was inspired by biological evolutionary systems. 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. The general flow of a typical genetic program is shown in the left plot of Figure 2: an initial population of solutions/individuals is created (randomly or by a pre-defined criterion), after that, individuals are selected, recombined, mutated and then placed back into the solutions pool, this process is repeated a number of times and the algorithm returns the best individual found.

A distinctive feature of GP, when compared to other evolutionary algorithms, is in that complex data structures are used to represent solutions (individuals), for example, trees or graphs. Thus, GP can be used for solving complex learning/modeling problems.

B. GP for term-weighting-scheme learning

Our approach to generate weighting schemes uses genetic programming to learn to combine a set of primitives that have been used for building weighting schemes in the past (see Figure 1). The genetic program searches for the combination of primitives that maximizes the classification performance of the task under analysis (e.g., image classification). A tree representation is adopted in which leafs correspond to primitives and non-terminal nodes correspond to operators by which primitives are combined; the evaluation of a tree leads to a term-weighting scheme, see Figure 2, right.

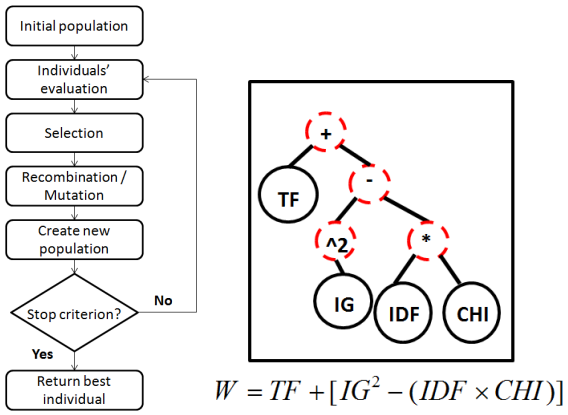

Fig. 2. Left: a generic evolutionary algorithm. Right: adopted representation for individuals. In the latter, dashed nodes represent operators (taken from the function set) and solid-line nodes indicate terminals; below the tree we show the term-weighting scheme derived from it.

1) Representation: As previously mentioned, weighting schemes are mainly composed 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 genetic program uses as terminals TDR and TR primitive (together with useful constants and other weighting schemes), which are 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. The right plot in Figure 2 depicts a typical individual and the resultant weighting scheme.

The set of terminals considered in this work is shown in Table II, as operators (non-terminals) we considered the following function set: $\mathcal{F} = \{+, -, *, /, \log_2 x, \sqrt{x}, x^2\}$. Each terminal in Table II is a matrix of size $N \times |V|$, where N is the number of training documents and V the vocabulary (i.e., the codebook of visual words). TDRs are themselves matrices of that dimensions, but TRs are row vectors of length $|V|$ (i.e., they indicate the relevance of each 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.

TABLE II. 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).
$\mathbf{W}_{14}$	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 $TF \times RF$.

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 [25] for details.

2) Fitness function: Since the goal of the genetic program is to obtain a weighting scheme that maximizes classification performance, 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 in Figure 2, right. 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 directly used as 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 [26], [27]. 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 [27] we used the low-rank linearized SVM (LLSMV) [26]. LLSVM is a linearized version of non-linear SVMs, which can be trained efficiently with the so called block minimization framework [28]. We selected LLSVM instead of alterative methods, because this method has outperformed several other efficient implementations of SVMs, see e.g., [27], [26]. Thus we use this approximated SVM during the fitness function, however, once a weighting scheme has been learnt 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.

3) Genetic operators: The proposed genetic program follows a standard procedure as depicted in Figure 2, left. We use the implementation from [29], which considers standard crossover and mutation operators. 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) 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 learnt using the training representation and its performance evaluated in the test representation. For this evaluation we consider a deterministic SVM (from the CLOP toolbox [30]), hence, results are comparable to each other. The next section reports experimental results on several computer vision tasks obtained with learned weighting schemes.

V. EXPERIMENTS AND RESULTS

This section reports results of an experimental study that aims at evaluating both: the suitability of alternative term-weighting schemes for in computer vision tasks and the performance of learned schemes in the same problems.

A. *Experimental protocol*

For experimentation we considered four standard data sets associated to landmark computer vision tasks. The considered data sets are described in Table III. All of these data sets are associated to classification/recognition tasks, hence the same evaluation protocol (with minor variations described below for each data set) was adopted. For each data set we generated training and test partitions[1]. The training set is used to obtain the visual vocabulary and to maximize the f_1 measure for the genetic program. Unless otherwise stated, we used the VLFEAT toolbox for processing images [31], using PHOW[2] (Pyramid Histogram Of Visual Words) features as visual descriptors [4]. Next, training and test images are represented with the different term-weighting schemes. Then, a classification model is learned using training data and the performance of the model is evaluated in test data.

TABLE III. DATA SETS CONSIDERED FOR EXPERIMENTATION. COLUMN 6 SHOWS THE NUMBER OF $images \mid terms$ CONSIDERED DURING THE SEARCH PROCESS.

Image Categorization					
Data set	**Classes**	$\|\mathbf{V}\|$	**# Train**	**# Test**	$\mathbf{im.}\|\mathbf{terms}$
Caltech-tiny	5	12000	75	75	15|12000
Caltech-102 (15)	101	12000	1530	1530	165|3000
Caltech-102 (30)	101	12000	3060	3060	330|3000
Gesture recognition					
Data set	**Classes**	$\|\mathbf{V}\|$	**# Train**	**# Test**	$\mathbf{im.}\|\mathbf{terms}$
Montalbano	20	1000	6850	3579	2055|600
Scene recognition					
Data set	**Classes**	$\|\mathbf{V}\|$	**# Train**	**# Test**	$\mathbf{im.}\|\mathbf{terms}$
15 Scenes	101	12000	1475	3010	1475|2000
Pornographic image filtering					
Data set	**Classes**	$\|\mathbf{V}\|$	**# Train**	**# Test**	$\mathbf{im.}\|\mathbf{terms}$
Adult	101	12000	6808	1702	6808|2000

Regarding our genetic program for term-weighting learning, the average and standard deviation performance over 5 runs of the genetic program is reported. The method was run in all cases for 50 generations with a population of 500 individuals, default values were used for the remainder of GP parameters.

Because the optimization process may be too time consuming, 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 III 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 [32]).

1) Caltech-101: Caltech-101 [12] is a mandatory benchmark for image classification, it contains objects that belong to 101 different categories, the inclusion of a background category makes it a 102-classes data set. Sample images from this data set are provided in Figure 3.

Fig. 3. Sample images from the Caltech-101 data set.

We performed experiments with 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 problems of comparable, but different, complexity. In fact, we use these subsets of Caltech-101 to assess the generality capabilities of the proposed approach, see below.

2) Adult image filtering: A data set for adult image filtering was considered. The data was made available by [7], and it has been previously used in several publications, see [7], [14]. 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 Figure 4.

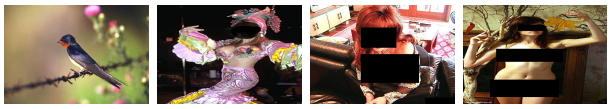

Fig. 4. Sample images from the data set of adult image filtering. The categories are (from left to right): inoffensive images, lightly dressed persons, partly nude persons, nude persons, and pornographic images (not shown).

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., photos 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 [7].

3) Scene recognition: We consider a benchmark data set for scene recognition [5]. The data set comprises 15 indoor/outdoor categories, where images contain complex scenes. Figure 5 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 [5]: 100 per category for training and the rest for testing.

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

[2]PHOW is an extension to the raw bag of visual words formulation that aims at incorporating spatial information by means of a pyramidal structure, see [4] for details.

Fig. 5. Sample images from the 15-Scenes data set. Categories are from left to right and from up to bottom: *bedrom, suburb, industrial, kitchen, living-room, coast, forest, highway, inside-city, mountain, open-country, street, tall-building, office, store.*

4) Montalbano: The bag of visual words has been used to represent videos as well, see e.g., [1], [16], [18]. For this reason we decided to include a video data set too. Specifically, we considered the Montalbano data set for gesture recognition as provided in [13]. The task consist of recognizing gestures from 20 categories (Italian cultural gestures), see Figure 6. The available data is depth and RGB video together with skeleton information. For our experiments we used the features proposed in [33], 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.

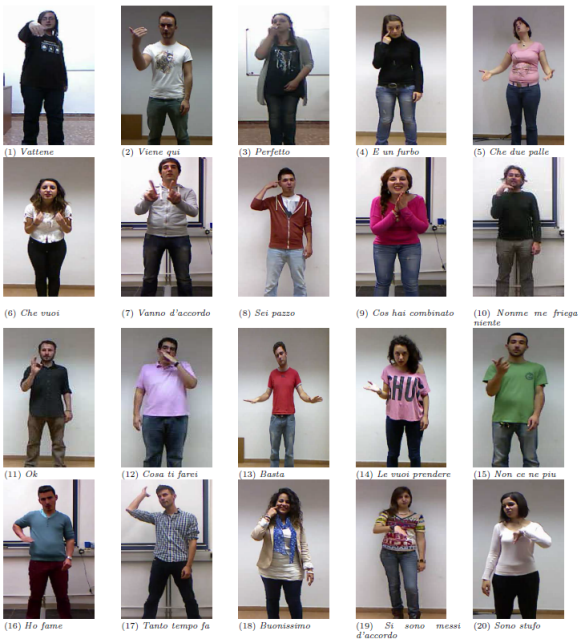

Fig. 6. Sample images from the Montalbano data set. Images from each of the gesture categories are shown [13].

5) Results: This section presents results of traditional, alternative and learned weighting schemes in the data sets described above. As previously mentioned, the most common approach for image representation under the bag of visual words is the TF scheme, hence, we consider it our baseline.

Table IV shows the results in test-sets in terms of f_1 measure. It can be seen from this table that, in average, the Boolean weighting scheme outperforms both, traditional and alternative, term-weighting schemes. This is an interesting result, because, most of the times a (normalized) TF weighting scheme is considered in computer vision tasks.

Regarding supervised term weighting schemes, only $TF-RF$ outperformed the usual TF scheme, but its performance was lower than the Boolean scheme. This is a somewhat disappointing result, because, intuitively, the incorporation of discriminative information should yield better performance. Anyway, we are reporting an experimental assessment of these schemes in a number of tasks, and showing the adequacy of the Boolean scheme.

On the other hand, it is clear from Table IV 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). The improvement is consistent and by a considerable margin in most cases. Besides, one should note that the standard deviation across runs is relatively low, evidencing the stability and robustness of the method.

To better appreciate the improvements offered by our method, in Figure 7 we show 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.

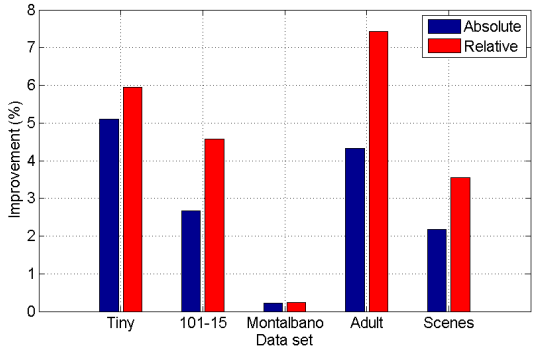

Fig. 7. Absolute (blue-first bar) and relative (red-right bar) improvement for the different data sets, taking as reference the best traditional/alternative weighting scheme for each data set.

From Figure 7 it can be seen that our method offers considerable improvements for all but for the Montalbano data set. The difficulty of this task may require running the genetic programm 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 III). This could be due to the fact that we are modeling videos. On the other hand, the largest improvement was observed for the Adult image data set, this could be due to the fact that specific term-weighting schemes are required for this type of tasks.

TABLE IV. CLASSIFICATION PERFORMANCE OBTAINED WITH TRADITIONAL, ALTERNATIVE AND LEARNED WEIGHTING SCHEMES. THE ⋆ SYMBOL INDICATES A STATISTICALLY SIGNIFICANT DIFFERENCE BETWEEN OUR APPROACH AND THE METHOD FROM THE CORRESPONDING COLUMNS.

	Traditional			**Alternative**			**Learned**
Data set / TWS	**TF (baseline)⋆**	**Bol.⋆**	**TF-IDF⋆**	**TF-RF⋆** [23]	**TF-CHI⋆** [22]	**TF-IG⋆** [22]	**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**
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**
Average	65.78±16.73	67.96±13.44	62.52±16.03	66.61±16.44	62.57±16.93	62.47±17.46	**71.62±14.09**

We now evaluate the generality of the weighting schemes. Figure 8 shows boxplots reporting the average performance obtained by the different term-weighting schemes when applied to all of the data sets. That is, each of the considered methods (columns 2-8 in Table IV) was evaluated in all of the data sets (including data sets for which the weighting scheme was not learned).

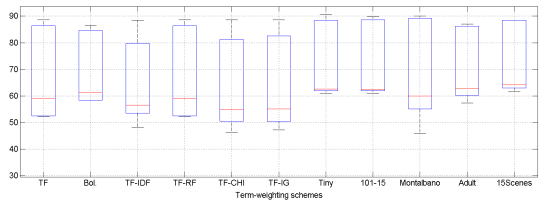

Fig. 8. Performance obtained with the different methods when used to classify all of the considered data sets.

Several findings can be drawn from Figure 8. First, it is clear the traditional and alternative weighting schemes do not generalize well (boxplots 1-6, from left to right). The Boolean weighting scheme being the one with better generalization capabilities. Secondly, regarding the performance of learned weighting schemes (boxplots 7 to 11), the schemes learned for Tiny, 101-15 and 15-Scenes data sets tend to generalize better, this can be due to the fact that this are generic image-classification tasks. On the other hand, the scheme learned for Montalbano and Adult data sets did not generalize well, this result confirms the fact that for this specialized tasks (gesture recognition and pornographic image filtering), the genetic program learned very specific term weighting schemes that do not work well for the rest of tasks. Hence proving the importance of adopting ad-hoc weighting schemes for different tasks.

Figure 9 shows the frequency of use of each of the terminals from Table II in the solutions returned by the genetic program for all of the data sets (i.e., a bar in Figure 9 corresponds to a row in Table II). 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 proposal: compare columns 2,4 and 5 to column 8 in Table IV.

Only 6 out of the 22 terminals did not appeared in solutions returned by the genetic program, all of these are terminals ($W_{9,10,12,14,15,20}$) correspond to TR weights, mainly used for feature selection in text classification [34]. Although they have

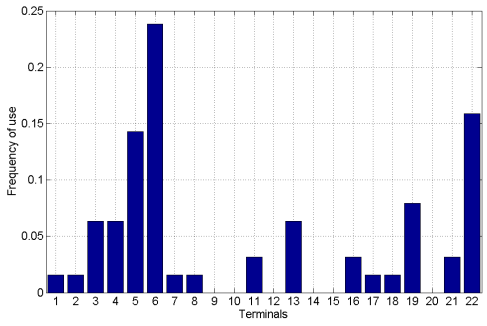

Fig. 9. Frequency of appearance of terminals into the solutions found by the genetic program.

proved to be very effective in [34] (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.

Summarizing experimental results, we have show evidence that the proposed genetic program outperforms significantly existing term-weighting schemes in a number of computer vision tasks. We also showed that the Boolean weighting scheme is a better option than the standard TF. Finally, we showed that while some of the learned schemes generalize well, it is better to use an ad-hoc weighting scheme, learned for each particular data set.

VI. CONCLUSIONS

We have presented a study on the use of traditional and alternative term-weighting schemes for computer vision tasks using the bag of visual words formulation. Our study assess the performance of weighting schemes that have not been used for the approached tasks. More importantly, we propose an evolutionary algorithm for learning term-weighting schemes. To the best of our knowledge, our work is the first that assesses alternative weighting schemes, and it is the first in proposing to learn weighting schemes. After an extensive experimental study, comprising 6 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. Besides this method showed better generalization capabilities.
- 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 extending/modifying our method to improve its performance in video-based tasks, like gesture recognition or video retrieval. Also we would like to evaluate the performance of our method when using the entire data for optimization, and using other evolutionary algorithms to evolve weighting schemes.

ACKNOWLEDGMENT

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] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *International Conference on Computer Vision*, vol. 2, 2003, pp. 1470–1477.

[2] K. Grauman and B. Leibe, *Visual Object Recognition*. Morgan and Claypool, 2010.

[3] J. Zhang, M. Marszablek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.

[4] A. Bosch, A. Zisserman, and X. Munoz, "Image classifcation using random forests and ferns," in *Proc. of ICCV*, 2007.

[5] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the Computer Vision and Image Processing Conference*. IEEE, 2006, pp. 2169–2178.

[6] M. Mirza-Mohammadi, S.Escalera, and P. Radeva, "Contextual-guided bag-of-visual-words model for multi-class object categorization," in *Proc. of CAIP*. Springer, 2009, pp. 748–756.

[7] T. Deselaers, L. Pimenidis, and H. Ney, "Bag of visual words for adult image classification and filtering," in *Proceedings of the International Conference on Pattern Recognition*. IEEE, 2008.

[8] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bra, "Visual categorization with bags of keypoints," in *International workshop on statistical learning in computer vision*, 2004.

[9] A. review of weigthing schemes for bag of vsual words image retrieval, "P. tirilly and v. claveau and p. gros," IRISA, Tech. Rep., 2009.

[10] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inform. Process. Manag.*, pp. 513–523, 1988.

[11] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computer Surveys*, vol. 34, no. 1, pp. 1–47, 2008.

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

[13] S. Escalera, X. Baro, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce, H. J. Escalante, J. Shotton, and I. Guyon, "ChaLearn looking at people challenge 2014: Dataset and results," in *Proc. of ECCV-Chalearn workshop*, 2014.

[14] S. J. Yoo, "Intelligent multimedia information retrieval for identifying and rating adult images," in *Proceedings of the International Conference KES*, ser. LNAI, vol. 3213. Springer, 2004, pp. 164–170.

[15] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley, 1999.

[16] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2–3, pp. 107–123, 2005.

[17] A. P. López-Monroy, M. M. y Gómez, H. J. Escalante, A. Cruz-Roa, and F. A. González, "Bag-of-visual-ngrams for histopathology image classification," in *Proc. of SPIE 8922*, 2013, p. 89220P.

[18] A. Hernández-Vela, M. A. Bautista, X. Perez-Sala, V. Ponce-López, S. Escalera, X. Baró, O. Pujol, and C. Angulo, "Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in rgb-d," *Pattern Recognition Letters*, vol. 50, no. 1, pp. 112–121, 2014.

[19] R. Cummins and C. O'Riordan, "Evolving local and global weighting schemes in information retrieval," *Information Retrieval*, vol. 9, pp. 311–330, 2006.

[20] M. García-Limón, H. J. Escalante, M. M. y Gómez, A. Morales, and E. Morales, "Towards the automated generation of term-weighting schemes for text categorization," in *Proc. of GECCO Comp'14, (Late-breaking abstract)*, 2014, pp. 1459–1460.

[21] H. J. Escalante, M. Garcia, A. Morales, M. Graff, M. Montes, E. F. Morales, and J. Martinez, "Term-weighting learning via genetic programming for text classification," *Knowledge-based Systems*, vol. Online, 2015.

[22] F. Debole and F. Sebastiani, "Supervised term weighting for automated text categorization," in *Proceedings of the 2003 ACM Symposium on Applied Computing*, ser. SAC '03. New York, NY, USA: ACM, 2003, pp. 784–788. [Online]. Available: http://doi.acm.org/10.1145/952532.952688

[23] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *Trans. PAMI*, vol. 31, no. 4, pp. 721–735, 2009.

[24] P. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of Artificial Intelligence Research*, vol. 37, pp. 141–188, 2010.

[25] W. B. Langdon and R. Poli, *Foundations of Genetic Programming*. Springer, 2001.

[26] K. Zhang, L. Lan, Z. Wang, and F. Moerchen, "Scaling up kernel svm on limited resources: A low-rank linearization approach," in *Proc. of AISTATS 2012*, 2012.

[27] N. Djuric, L. Lan, S. Vucetic, and Z. Wang, "Budgetedsvm: A toolbox for scalable svm approximations," *Journal of Machine Learning Research*, vol. 14, pp. 3813–3817, 2013.

[28] K. W. Chang and D. Roth, "Selective block minimization for faster convergence of limited memory large-scale linear models," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2011.

[29] S. Silva and J. Almeida, "Gplab-a genetic programming toolbox for matlab," in *Proc. Nordic MATLAB conf.*, 2003, pp. 273–278.

[30] A. Saffari and I. Guyon, "Quick start guide for clop," TU Graz - CLOPINET, Tech. Rep., 2006.

[31] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008.

[32] J. Demsar, "Statistical comparisons of classifiersover multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[33] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Proc. of ECCV ChaLearn Workshop on Looking at People*, 2014.

[34] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *J. of Mach. Learn. Res.*, vol. 3, pp. 1289–1305, 2003.