Improving Image Classification Through Descriptor Combination

A. Mansano, J. A. Matsuoka, L. C. S. Afonso, J. P. Papa
Department of Computing
São Paulo State University
Bauru, Brazil

F. Faria and R. da S. Torres
Institute of Computing
University of Campinas
Campinas, SP, Brazil

***Abstract*—The efficiency in image classification tasks can be improved using combined information provided by several sources, such as shape, color, and texture visual properties. Although many works proposed to combine different feature vectors, we model the descriptor combination as an optimization problem to be addressed by evolutionary-based techniques, which compute distances between samples that maximize their separability in the feature space. The robustness of the proposed technique is assessed by the Optimum-Path Forest classifier. Experiments showed that the proposed methodology can outperform individual information provided by single descriptors in well-known public datasets.**

***Index Terms*—Image classification, Evolutionary algorithms, Descriptor Combination**

I. Introduction

Image classification attempts to provide useful information for several applications, that range from automatic diagnosis in medical systems to target recognition in remote sensing images. Important visual properties such as shape, texture, and color are often used to describe images in recognition applications.

In order to best represent images, one common strategy consists in identifying the most accurate feature vector (description). On the other hand, in most cases there are many features available with reasonable performance. In such situations, one can employ feature combination approaches in order to improve their individual recognition rates, since different features may provide different, but complementary information about images. Several works deal with this problem by learning the most reliable features and weighting them according to some "reliability"-based measure [1], [2], [3]. Other works address this problem by using Linear Discriminant Analysis [4] and Principal Component Analysis [5].

The use of features by themselves, however, may not guarantee good performance results of applications that exploit image visual content. In some cases, the performance is not only dependent on the features extracted, but also on the distance function employed to compare feature vectors. Papa et al. [6], for example, have showed that one can improve the recognition accuracy in image classification tasks, for the same training and test sets, up to 7.28% by just changing the distance function.

Torres et al. [7] have introduced the concept of *image descriptor* in the context of Content-Based Image Retrieval. An image descriptor is modelled as a pair that contains the feature vector extraction algorithm together with the appropriate distance function. That modelling allows the development of methods for combining image content descriptors that consider both feature vectors and distance functions at the same time.

One natural choice for designing combination strategies relies on the use of evolutionary algorithms. In that case, the descriptor combination problem is modelled as an optimization procedure, in which the distances associated with feature vectors are combined in a more complex manner.

This paper investigates the use of two evolutionary techniques for non-linear combination of descriptors: Particle Swarm Optimization (PSO) [8], and Harmony Search (HS) [9]. PSO and HS are evolutionary-based techniques that have been used for optimization problems. While PSO models the problem using the theory of social dynamics, HS employs the same method used for musicians in order to compose songs with optimal harmony. Thus, the best solution to the optimization problem is the one that provides the best harmony. In the last years, there is a exponential growing of works that employ such techniques, since they can achieve interesting results imitating the collective behavior of the nature.

This paper presents three original contributions: (i) proposal of a new set of parameters to allow non-linear descriptor combination using PSO; (ii) the design of a new combination strategy based on HS; and (iii) evaluation of the proposed methods on heterogeneous collections involving color and texture descriptors, considering image classification tasks. The validation protocol considers the use of the Optimum-Path Forest (OPF) classifier, a graph-based approach widely used in several applications [6]. The choice for this classifier is motivated by its fast training and classification phases together with its good recognition rates. Another point is that OPF can easily handle different distance metrics, which is very important in the context of classification tasks.

Faria et al. [10] proposed a descriptor combination approach using a swarm-based optimization algorithm together with Optimum-Path Forest (OPF) [6] for image classification purposes. Their work aimed to model the descriptor combination problem as an optimization procedure, in which the distances associated with each feature vector are combined through a linear equation, and the final distances are used to weight the edges of the OPF graph. In this work, we extend the above procedure by employing a new set of parameters in order to allow a greater range of possible solutions than a linear combination. Another difference relies on using PSO to combine not only shape-based descriptors (as performed by Faria et al. [10]), but also texture and color information.

The remainder of this paper is organized as follows. Section II introduces the descriptor model used in this work, and Sections III and IV briefly state the OPF background theory and the employed evolutionary techniques, respectively. Section V presents the proposed approach for descriptor combination and Section VI discusses the conducted experiments. Finally, Section VII states conclusions.

II. Image Description Model

In this section, we formalize the adopted image description model [7], [11].

Definition 1. *An* **image** $\hat{I}$ *is a pair* $(D_I, \vec{I})$*, where:*

- $D_I \subset \mathbb{Z}^2$ *is a finite set of* pixels*, and*
- $\vec{I} : D_I \rightarrow \mathsf{D}'$ *is a function that assigns to each pixel* p *in* D_I *a vector* $\vec{I}(p)$ *of values in some arbitrary space* D' *(for example,* $\mathsf{D}' = \mathbb{R}^3$ *when a color in the RGB system is assigned to a pixel).*

Definition 2. *A* ***simple descriptor*** *(briefly,* ***descriptor****)* D *is defined as a pair* (ϵ_D, δ_D)*, where:*

- $\epsilon_D : \hat{I} \rightarrow \mathbb{R}^n$ *is a function, which extracts a* feature vector $\vec{v}_{\hat{I}}$ *from an* image $\hat{I}$.
- $\delta_D : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ *is a* similarity function *that computes the similarity between two images as a function of the distance between their corresponding* feature vectors*.*

Definition 3. *A* ***feature vector*** $\vec{v}_{\hat{I}}$ *of an image* $\hat{I}$ *is a point in* $\mathbb{R}^n$ *space:* $\vec{v}_{\hat{I}} = (v_1, v_2, ..., v_n)$*, where* n *is the dimension of the vector.*

Figure 1 illustrates the use of a simple descriptor D to compute the similarity between two images $\hat{I}_A$ and $\hat{I}_B$. First, the extraction algorithm ϵ_D is used to compute the feature vectors $\vec{v}_{\hat{I}_A}$ and $\vec{v}_{\hat{I}_B}$ associated with the images. The similarity function δ_D is then used to determine the similarity score d between the images.

Definition 4. *A* ***composite descriptor*** $\hat{D}$ *is a pair* $(\mathcal{D}, \delta_{\mathcal{D}})$*, where:*

- $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$ *is a set of* k *pre-defined simple descriptors.*
- $\delta_{\mathcal{D}}$ *is a similarity function which combines the similarity values obtained from each descriptor* $D_i \in \mathcal{D}$*,* $i = 1, 2, \ldots, k$*.*

Figure 2 illustrates the use a composite descriptor $\hat{D}$ to compute the distance between images $\hat{I}_A$ and $\hat{I}_B$.

III. Optimum-Path Forest Classification

The OPF classifier works by modelling the problem of pattern recognition as a graph partition in a given feature space. The nodes are represented by the feature vectors and the edges connect all pairs of them, defining a full connectedness

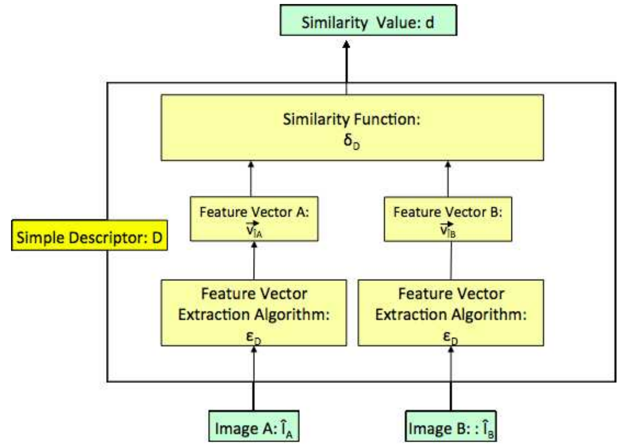

Fig. 1. Simple Descriptor.

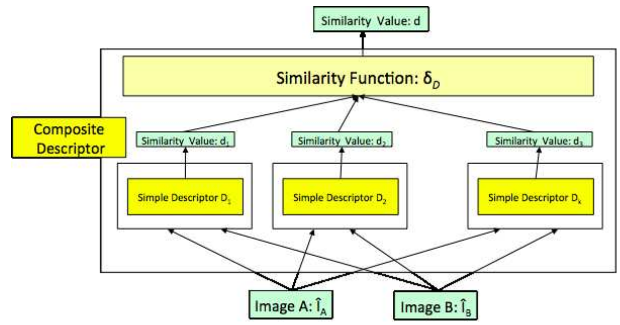

Fig. 2. Composite Descriptor.

graph. This kind of representation is straightforward, given that the graph does not need to be explicitly represented, allowing us to save memory. The partition of the graph is carried out by a competition process between some key samples (prototypes), which offer optimum paths to the remaining nodes of the graph. Each prototype sample defines its optimum-path tree (OPT), and the collection of all OPTs defines an optimum-path forest, which gives the name to the classifier [6], [12].

The OPF can be seen as a generalization of the well-known Dijkstra's algorithm to compute optimum paths from a source node to the remaining ones [13]. The main difference relies on the fact that OPF uses a set of source nodes (prototypes) with any smooth path-cost function [14]. In case of Dijkstra's algorithm, a function that summed the arc-weights along a path was applied. In regard to the supervised OPF version addressed here, we have used a function that gives the maximum arc-weight along a path, as explained below.

Let $Z = Z_1 \cup Z_2 \cup Z_3$ be a dataset labelled with a function λ, in which Z_1, Z_2 and Z_3 are, respectively, a training, validation, and test sets. Let $S \subseteq Z_1$ a set of prototype samples. Essentially, the OPF classifier creates a discrete optimal partition of the feature space such that any sample $s \in Z_2 \cup Z_3$ can be classified according to this partition. This partition is an optimum path forest (OPF) computed in $\Re^n$ by the Image-Foresting Transform (IFT) algorithm [14].

The OPF algorithm may be used with any *smooth* path-cost function which can group samples with similar properties [14]. Particularly, we used the path-cost function $f_{\max}$, which is computed as follows:

$$\begin{aligned} f_{\max}(\langle s \rangle) &= \begin{cases} 0 & \text{if } s \in S, \\ +\infty & \text{otherwise} \end{cases} \\ f_{\max}(\pi \cdot \langle s,t \rangle) &= \max\{f_{\max}(\pi), d(s,t)\}, \end{aligned} \quad (1)$$

in which $d(s,t)$ means the distance between samples s and t, and a path π is defined as a sequence of adjacent samples. In such a way, we have that $f_{\max}(\pi)$ computes the maximum distance between adjacent samples in π, when π is not a trivial path.

The OPF algorithm works with a training and a testing phase. In the former step, the competition process begins with the prototypes computation. We are interested in finding the elements that fall on the boundary of the classes with different labels. For that purpose, we can compute a Minimum Spanning Tree (MST) over the original graph and then mark as prototypes the connected elements with different labels. Figure 3b displays the MST with the prototypes at the boundary. After that, we can begin the competition process between prototypes in order to build the optimum-path forest, as displayed in Figure 3c. The classification phase is conducted by taking a sample from the test set (blue triangle in Figure 3d) and connecting it to all training samples. The distance to all training nodes are computed and used to weight the edges. Finally, each training node offers to the test sample a cost given by a path-cost function (maximum arc-weight along a path - Equation 1), and the training node that has offered the minimum path-cost will conquer the test sample. This procedure is shown in Figure 3e.

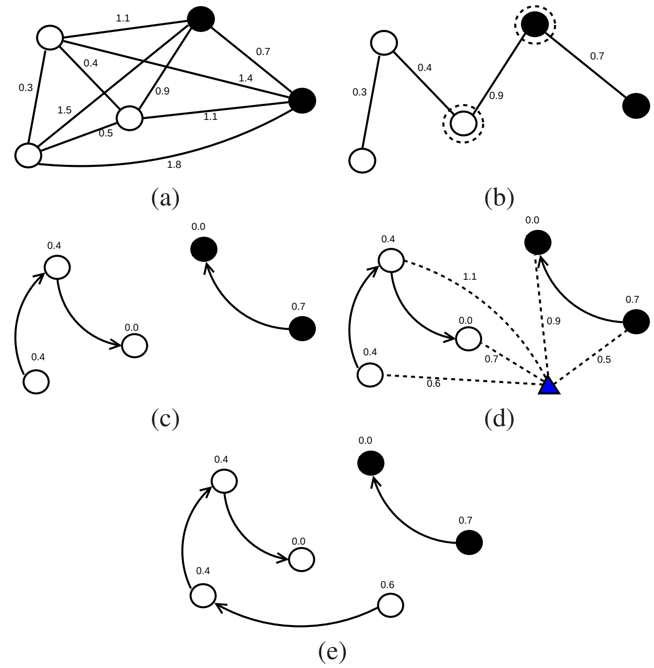

Fig. 3. OPF pipeline: (a) complete graph, (b) MST and prototypes bounded, (c) optimum-path forest generated at the final of training step, (d) classification process and (e) the triangle sample is associated to the white circle class. The values above the nodes are their costs after training, and the values above the edges stand for the distance between their corresponding nodes.

IV. Background on Evolutionary Optimization

In this section, we briefly describe the evolutionary techniques used for descriptor combination: Particle Swarm Optimization and Harmony Search.

A. *Particle Swarm Optimization*

Particle Swarm Optimization (PSO) is an algorithm modelled on swarm intelligence that finds a solution in a search space based on social behavior dynamics [8]. Each possible solution to the problem is modelled as a particle in the swarm that imitates its neighborhood based on the values of the fitness function found so far.

Other definitions consider PSO as a stochastic and population-based search algorithm, in which social behavior learning allows each possible solution (particle) to “fly” onto this space (swarm) looking for other particles that have better characteristics, i.e., the ones that maximize a fitness function. Each particle has a memory that stores its best local solution (local maxima) and the best global solution (global maxima). Thus, by taking this information into account, each particle has the ability to imitate the others that give to it the best local and global maxima. This process simulates social interaction between humans looking for the same objective or bird flocks looking for food, for instance. This socio-cognitive mechanism can be summarized into three main principles [8]: (i) evaluating, (ii) comparing and (iii) imitating. Each particle can evaluate others in its neighborhood through some fitness function, can compare it with its own value, and, finally, can decide whether it is a good choice to imitate them.

The entire swarm is modelled in a multidimensional space $\Re^N$, in which each particle $p_i = (\vec{x_i}, \vec{v_i}) \in \Re^N$ has two main features: (i) position ($\vec{x_i}$) and (ii) velocity ($\vec{v_i}$). The local (best current position $\widehat{x_i}$) and global solution $\widehat{s}$ are also known. After defining the swarm size, i.e., the number of particles, each one of them is initialized with random values for both velocity and position. Each individual is then evaluated with respect to some fitness function and its local maximum is updated. At the end, the global maximum is updated with the particle that achieved the best position in the swarm. This process is repeated until some convergence criterion is reached. The updated position and velocity equations of particle p_i, in the simplest form that governs the PSO, are, respectively, given by

$$\vec{v_i} = w\vec{v_i} + c_1 r_1(\widehat{x}_i - \vec{x_i}) + c_2 r_2(\widehat{s} - \vec{x_i}) \quad (2)$$

and

$$\vec{x_i} = \vec{x_i} + \vec{v_i}, \quad (3)$$

where w is the inertia weight that controls the power of the interactions between the particles, and $r_1, r_2 \in [0,1]$ are random variables that give the idea of stochasticity to the PSO method. Constants c_1 and c_2 are used to guide particles onto good directions.

B. Harmony Search

Harmony Search (HS) is an evolutionary algorithm inspired in the improvisation process of music players [9]. The main idea is to use the same process adopted by musicians to create new songs to obtain a near-optimal solution for some optimization process. Basically, any possible solution is modelled as a harmony and each parameter to be optimized can be seen as a musical note. The best harmony (solution) is chosen as the one that maximizes some optimization criteria. The algorithm is composed by few steps, as follows: (1) initialize the optimization problem and algorithm parameters; (2) initialize a Harmony Memory (HM); (3) improvise a new harmony from HM; (4) update the HM if the new harmony is better than the worst harmony in the HM, include the new harmony in HM, and remove the worst one from HM; and (5): if the stopping criterion is not satisfied, go to Step 3.

The HS parameters required to solve the optimization problem are the harmony memory size (HMS), the harmony memory considering rate (HMCR), the pitch adjusting rate (PAR), and the stopping criterion. HMCR and PAR are parameters used to improve the solution vector, i.e., they can help the algorithm to find globally and locally improved solutions in the harmony search process.

In Step 2, the HM matrix (Equation 4) is initialized with randomly generated solution vectors with their respective values for the objective function:

$$HM = \left[\begin{array}{cccc|c} x_1^1 & x_1^2 & \ldots & x_1^N & f(x_1) \\ x_2^1 & x_2^2 & \ldots & x_2^N & f(x_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{HMS}^1 & x_{HMS}^2 & \ldots & x_{HMS}^N & f(x_{HMS}) \end{array} \right], \tag{4}$$

where x_i^j denotes the decision variable j from harmony i, and $f(x_i)$ means the objective function. Note that in HS principles, each harmony $h_i = (\vec{x_i})$ has only the information about its position in the search space (Harmony Memory), and each line in the HM matrix stands for one harmony.

In Step 3, a new harmony vector $\vec{x'}$=(x'^1,x'^2 ,..., x'^N) is generated from the HM based on memory considerations, pitch adjustments, and randomization (music improvisation). It is also possible to choose the new value using the HMCR parameter, which varies between 0 and 1 as follows:

$$\mathbf{x}'^j \quad \leftarrow \quad \begin{cases} x'^j \in \left\{ x_1^j, \ldots, x_{HMS}^j \right\} & \text{with probability HMCR,} \\ \mathbf{x}'^j \in \phi_j & \text{with probability (1-HMCR),} \end{cases} \tag{5}$$

in which ϕ_j denotes the range of values for variable j, for $\phi = (\phi_i, \phi_2, \ldots, \phi_N)$. The HMCR is the probability of choosing one value from the historic values stored in the HM, and (1-HMCR) is the probability of randomly choosing one feasible value not limited to those stored in the HM. In order to make it clear, an HMCR=0.7 means that 70% of the notes (decision variables) to compose the new harmony $h' = (\vec{x'})$ will be picked from HM, and the remaining ones will be randomly generated within the interval ϕ_j.

Further, every component j of the new harmony vector $\vec{x}$ is examined to determine whether it should be pitch-adjusted:

$$\text{Pitching adjusting decision for } x'^j \leftarrow \begin{cases} \textbf{Yes} \text{ with probability PAR,} \\ \textbf{No} \text{ with probability (1-PAR).} \end{cases} \tag{6}$$

The pitch adjustment for each instrument is often used to improve solutions and to escape from local optima. This mechanism concerns shifting the neighboring values of some decision variable in the harmony. If the pitch adjustment decision for the decision variable $\mathbf{x}'^j$ is Yes, $\mathbf{x}'^j$ is replaced as follows:

$$x'^j \leftarrow x'^j + rb, \tag{7}$$

where b is an arbitrary distance bandwidth for the continuous design variable, and r is a uniform distribution between 0 and 1. In this paper, we set $b = 1$.

In Step 4, if the new harmony h' is better than the worst harmony in the HM, the latter is replaced by this new harmony. Finally, in Step 5, the HS algorithm finishes when it satisfies the stopping criterion. Otherwise, Steps 3 and 4 are repeated in order to improvise a new harmony again.

V. Proposed Approach for Descriptor Combination

Faria et al. [10] have proposed an optimization-based approach for descriptor combination where the best descriptor was the one that maximized the accuracy of OPF classifier in an evaluating set. Let $\hat{D}^* = (\epsilon_{\hat{D}^*}, \delta_{\hat{D}^*})$ be such descriptor. We have that $\delta_{\hat{D}^*}$ can be formulated as:

$$\delta_{\hat{D}^*} = \sum_{i=1}^{N} \alpha_i \delta_{D_i}, \tag{8}$$

in which $\alpha_i \in \Re$ stands for a regularization parameter and δ_{D_i} denotes the distance matrix for descriptor D_i. Thus, the descriptor combination task means to combine the distances of each descriptor and to use them to weight the arcs between samples in the OPF graph.

Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_M)$ be the set of all regularization parameters. One can observe that $\delta_{\hat{D}^*}$ is a linear combination of all distance matrices, and α corresponds to the set of parameters that will be optimized by some approach. Faria et al. [10] employed PSO for such task. Their objetcive was to use the OPF recognition rate over a validation set as the function to be maximized.

In this work, we extended the above formulation beyond a linear combination of α by introducing a new set of parameters $\beta = (\beta_1, \beta_2, \ldots, \beta_M)$. Therefore, Equation 8 is rewritten as:

$$\delta_{\hat{D}^*} = \sum_{i=1}^{N} \alpha_i \delta_{D_i}^{\beta_i}, \tag{9}$$

in which $-2 \leq \alpha_i, \beta_i \leq 2$, $\beta_i \in \Re$. The main idea behind such formulation is to allow a greater variability of arithmetic computations, which have been restricted to only the set of α_i in Equation 8.

Let $\xi = \{(\alpha_i, \beta_i)\}$, $i = 1, 2, \ldots, N$, be the set of parameters to be optimized, and $\xi^* = (\alpha_i^*, \beta_i^*)$ be the one that maximizes the OPF accuracy over a validation set. In order to make it clear, suppose we have a dataset with two descriptors extracted. Thus, $\xi = \{(\alpha_1, \beta_1); (\alpha_2, \beta_2)\}$. In this case, the PSO/HS search space would have four dimensions, and each particle/harmony position j would be done by $\vec{x}_j = (\alpha_1^j, \alpha_2^j, \beta_1^j, \beta_2^j)$, $1 \leq j \leq M$. In this case, M stands for the number of particles/harmonies.

The proposed methodology has two phases: (i) the *design phase* which employs training and validation sets to find ξ^*, (ii) and the *classification phase* that computes the new distance d for each test sample s to all training samples, as stated in the OPF classification procedure (Section III). Let $d_{s,x}$ be the distance between s and the training sample x. Thus, $d_{s,x}$ can be found as follows:

$$d_{s,x} = \sum_{i=1}^{N} \alpha_i^* \delta_{D_i}^{\beta_i^*}(s, x), \tag{10}$$

in which $\delta_{D_i}(s, x)$ denotes the distance between samples s and x using descriptor i. Figure 4 illustrates the proposed methodology.

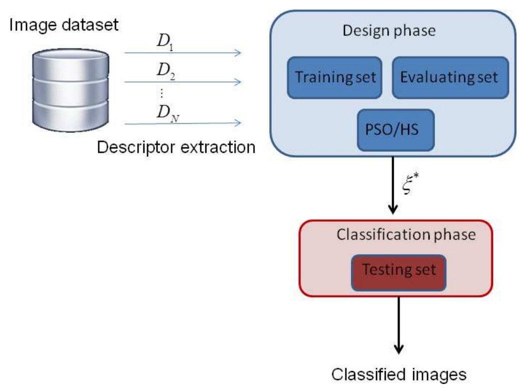

Fig. 4. Proposed methodology for descriptor combination.

The proposed methodology extends the work of Faria et al. [10] beyond a linear descriptor combination and also introduces the Harmony Search optimization algorithm to this task. In addition, this paper also combines information from color and texture, which differs from [10] that has been limited to shape descriptors.

VI. Experimental Results

In order to assess the robustness of the proposed methodology for descriptor combination, we compared it with the one introduced by Faria et al. [10]. In addition, we also evaluated the use of HS for linear and non-linear combination of descriptors.

We used two well-known public datasets, as follows:

- Corel[1]: this dataset contains 3,906 images labelled in 85 classes, and the number of images per class ranges from 7 to 98 images.
- Free Photo[2]: we used a subset containing 3,426 images labelled in 9 classes, and the number of images per class ranges from 70 to 854 images.

With regard to descriptors, we employed Local Activity Spectrum (LAS), Global Color Histogram (GCH), Color Autocorrelogram (ACC), Border/Interior pixel Classification (BIC), and Quantized Compound Change Histogram (QCCH) for the Corel dataset; and LAS, GCH, Homogeneous Texture Descriptor (HTD), BIC, and Color Coherent Vector (CCV) for the Free Photo dataset. A detailed description about these descriptors can be found in [15]. Recall that all experiments were carried out with the OPF classifier and Euclidean distance for all descriptors.

In order to compare the composite descriptor's effectiveness, we executed experiments with the single descriptors over 5 runnings with cross-validation. For that, 30% of the entire dataset was used for training and 50% for classification purposes. Table 1 shows the OPF mean accuracy for each single descriptor.

Descriptor/Dataset	Corel	Free Photo
LAS	59.75%±0.60	74.58%±0.65
GCH	65.82%±0.76	78.69%±1.25
ACC(HTD)	74.27%±1.13	73.11%±1.31
BIC	72.67%±0.20	89.71%±0.96
QCCH(CCV)	57.38%±0.60	80.50%±0.61

TABLE I
Single descriptor effectiveness. "ACC(HTD)" means that ACC was used for the Corel dataset, and HTD for the Free Photo dataset. The same applies for the line with "QCCH(CCV)".

With regard to the descriptor combination experiment, we employed the same training and test sets as before with cross-validation over 5 runnings. However, we use now a validation set with 20% of the entire dataset size. As aforementioned, the OPF recognition rate over such set is used as the fitness function by PSO and HS approaches. The PSO and HS parameters were empirically chosen, as follows: # of particles (harmonies): 250, HMCR: 0.7, PAR: 0.6, w = 0.7, c_1 = 1.6, c_2 = 0.4 and # number of iterations = 40. In order to allow a fair comparison between PSO and HS, we set the same values for the parameters with similar meaning (e.g., w for PSO and HMCR for HS avoid traps from local maxima).

We conducted two different descriptor combination experiments: LAS+GCH and ALL. While the former allows us to combine texture (LAS) and color (GCH) information, in the latter we opted to use all available descriptors. The initial choice for LAS and GCH in the first experiment has only empirical reasons. Table 2 shows the results considering the use of the Faria et al. [10] approach.

One can see that PSO and HS improved the classification rates with respect to single descriptors. Although PSO and HS obtained similar accuracies, the latter was about 2.33% faster (on average). Another point to be highlighted concerns with the fact that the descriptor combination may not result

[1]http://vision.stanford.edu (As of June 2012).

[2]http://www.freefoto.com (As of June 2012).

Descriptor/Dataset	Corel	Free Photo
PSO-LAS+GCH	67.78%±0.87	83.61%±0.50
PSO-ALL	75.40%±0.95	90.08%±0.73
HS-LAS+GCH	67.76%±0.66	83.78%±0.94
HS-ALL	75.48%±1.23	87.68%±0.87

TABLE II
COMPOSITE DESCRIPTOR EFFECTIVENESS ACCORDING TO THE FARIA ET AL. [10] APPROACH.

in a high improvement of the recognition rate of a very good descriptor, when it exists. However, the results demonstrated that an slightly improvement may be always obtained. In situations in which we do not have a very good descriptor, the improvement may be around 6.89%, as displayed by PSO and HS for the Free Photo dataset in Table 3, which also shows the recognition rates of the proposed approach.

Descriptor/Dataset	Corel	Free Photo
PSO-LAS+GCH	68.04%±0.62	84.52%±0.71
PSO-ALL	76.77%±0.60	90.36%±0.72
HS-LAS+GCH	67.94%±0.74	84.28%±0.43
HS-ALL	75.34%±0.81	89.88%±0.91

TABLE III
COMPOSITE DESCRIPTOR EFFECTIVENESS ACCORDING TO THE PROPOSED APPROACH.

We can see that the recognition rates of the proposed approach are slightly better than the one presented by Faria et al. [10]. For the Free Photo dataset, the gain is about 2.44% considering the experiment with HS-ALL.

VII. CONCLUSIONS

Information from different sources can improve the classifier's recognition rate. Motivated by that, in this paper, we propose new approaches for non-linear descriptor combination using two evolutionary algorithms: Particle Swarm Intelligence and Harmony Search. In addition, we addressed the combination of descriptors associated with different visual properties, such as texture and color.

The results have showed that HS and PSO obtain similar recognition rates, being the former slightly faster. The proposed approach also improves the classification rates of the work addressed by Faria et al. [10], and have shed light over the importance of using appropriate metrics for each feature vector extraction approach.

Future work includes the investigation of Particle Swarm Intelligence and Harmony Search in image retrieval tasks.

ACKNOWLEDGMENT

The authors would like to thank CAPES, CNPq (grant 303182/2011-3), FAPESP (grants 2007/52015-0, 2010/14910-0 and 2009/16206-1), and Microsoft Research for financial support.

REFERENCES

[1] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proceedings of the 12th International Conference on Computer Vision*, 2009, pp. 221–228.

[2] J. Hou, B.-P. Zhang, N.-M. Qi, and Y. Yang, "Evaluating feature combination in object classification," in *Proceedings of the 7th International Conference on Visual Computing*. Las Vegas, NV, USA: Springer-Verlag, 2011, pp. 597–606.

[3] D. Okanohara and J. Tsujii, "Learning combination features with L1 regularization," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, Stroudsburg, PA, USA, 2009, pp. 97–100.

[4] X. Liu, L. Zhang, M. Li, H. Zhang, and D.Wang, "Boosting image classification with LDA-based feature combination for digital photograph management," *Pattern Recognition*, vol. 38, no. 6, pp. 887–901, June 2005.

[5] F. Yan and X. Yanming, "Image classification based on multi-feature combination and PCA-RBaggSVM," in *Proceedings of the IEEE International Conference on Progress in Informatics and Computing*, vol. 2, 2010, pp. 888–891.

[6] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[7] R. da Silva Torres and A. X. Falcão, "Content-based image retrieval: Theory and applications," *RITA*, vol. 13, no. 2, pp. 161–185, 2006.

[8] J. Kennedy and R. Eberhart, *Swarm Intelligence*. M. Kaufman, 2001.

[9] Z. W. Geem, *Music-Inspired Harmony Search Algorithm: Theory and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[10] F. Faria, J. Papa, R. Torres, and A. Falcão, "Multimodal pattern recognition through particle swarm optimization," in *Proceedings of the 17th International Conference on Systems, Signals and Image Processing*, Rio de Janeiro, Brazil, 2010, pp. 1–4.

[11] R. Torres, A. Falcão, M. Gonçalves, J. Papa, B. Zhang, W. Fan, and W. Fox, "A genetic programming framework for content-based image retrieval," *Pattern Recognition*, vol. 42, no. 2, pp. 283–292, 2009.

[12] J. P. Papa, A. X. Falcão, V. H. C. Albuquerque, and J. M. R. S. Tavares, "Efficient supervised optimum-path forest classification for large datasets," *Pattern Recognition*, vol. 45, no. 1, pp. 512–520, 2012.

[13] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[14] A. Falcão, J. Stolfi, and R. Lotufo, "The image foresting transform theory, algorithms, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, 2004.

[15] O. Penatti, E. Valle, and R. Torres, "Comparative study of global color and texture descriptors for web image retrieval," *J. of Visual Communication and Image Representation*, 2011.