IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 7, 2018, accepted March 23, 2018, date of publication April 6, 2018, date of current version May 2, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2824240

Data-Fusion Techniques for Open-Set Recognition Problems

MANUEL ALBERTO CÓRDOVA NEIRA, PEDRO RIBEIRO MENDES JÚNIOR, ANDERSON ROCHA, AND RICARDO DA SILVA TORRES
Institute of Computing, University of Campinas, Campinas 13083-872, Brazil

Corresponding author: Manuel Alberto Córdova Neira (manuel.diduchi@gmail.com)

This work was supported in part by the São Paulo Research Foundation (FAPESP DéjàVu) under Grant 2017/12646-3, Grant 2014/12236-1, Grant 2015/24494-8, Grant 2016/50250-1, and Grant 2017/20945-0, in part by the FAPESP-Microsoft Virtual Institute under Grant 2013/50155-0, Grant 2013/50169-1, and Grant 2014/50715-9, in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant 304472/2015-8 and Grant 307560/2016-3, and in part by the Coordination for the Improvement of Higher Level Education Personnel (CAPES DeepEyes Grant and Grant 88881.145912/2017-01).

ABSTRACT Most pattern classification techniques are focused on solving closed-set problems in which a classifier is trained with samples of all classes that may appear during the testing phase. In many situations, however, samples of unknown classes, i.e., whose classes did not have any example during the training stage, need to be properly handled during testing. This specific setup is referred to in the literature as open-set recognition. Open-set problems are harder as they might be ill-sampled, not sampled at all, or even undefined. Differently from existing literature, here we aim at solving open-set recognition problems combining different classifiers and features while, at the same time, taking care of unknown classes. Researchers have greatly benefited from combining different methods in order to achieve more robust and reliable classifiers in daring recognition conditions, but those solutions have often focused on closed-set setups. In this paper, we propose the integration of a newly designed open-set graph-based optimum-path forest (OSOPF) classifier with genetic programming (GP) and majority voting fusion techniques. While OSOPF takes care of learning decision boundaries more resilient to unknown classes and outliers, GP combines different problem features to discover appropriate similarity functions and allows a more robust classification through early fusion. Finally, the majority-voting approach combines different classification evidence from different classifier outcomes and features through late-fusion techniques. Performed experiments show the proposed data-fusion approaches yield effective results for open-set recognition problems, significantly outperforming existing counterparts in the literature and paving the way for investigations in this field.

INDEX TERMS Pattern recognition, open-set recognition, data fusion, optimum-path forest, genetic programming, majority voting.

I. INTRODUCTION

Thus far, most research in pattern recognition has been focused on solving closed-set problems — those in which all classes that may appear in the testing stage have had representative samples during the training. In many situations, however, samples of unknown classes, i.e., classes for which no representatives at all have been seen during training, need to be properly handled during testing, giving rise to what is referred to in the literature as an open-set setup. As an example, consider that a supermarket has an image-based fruit recognition system installed at the cashiers for produce identification. In this type of system, all produce that might appear during system usage will be associated with one of the products seen in the training (closed set). Nevertheless, this type of system is inappropriate for problems in which a new fruit or produce is expected to be recognized by the application, as the system, if not properly equipped for open-set recognition, will return as result a fruit with which it was trained. In an open-set problem, a classifier should be able to reject such samples, i.e., identify when a sample does not belong to classes seen in the training [1]. As a matter of fact, a myriad of problems fall within this pattern recognition category.

Addressing open-set problems is challenging since we do not know nor do we have access to all classes that would be present in the testing phase [1]–[3]. Those classes can be, for example, difficult to sample at training phase or simply unknown [4]. Although difficult, the open-set recognition problem is not insurmountable. Indeed, some classifiers, including Support Vector Machines (SVMs) and neural networks, have already been proposed or modified to deal with or, at least, accommodate, open-set constraints and presented promising results [1]–[3], [5], [6]. However, using a single classifier with a single object feature might not be enough to deal with such difficult setup. To our knowledge, none of these solutions leverages fusion strategies to boost open-set methods and strengthen the decision-making process.

An open-set method ideally needs a better way to deal with unknown samples. For this reason, open-set approaches have to consider the *empirical risk* and *open-space risk*. ***Empirical risk*** [1], [7] is related to the specialization of the classifier. This risk can be measured on training data based on mislabeled samples. In turn, ***open-space risk*** [1], i.e., the risk of the *unknown*, is the risk of mislabeling data if we extend the decision margin, accepting more samples as positives, where new samples from unknown classes could appear during testing — this risk is related to the generalization of the classifier. An open-set classifier has to minimize the *open-set risk* defined in [1], i.e., try to minimize the risk balancing the *empirical risk* and *open-space risk*. As such, combining different classifiers and features dealing with different properties of a problem could improve the recognition rate by better finding appropriate tradeoffs between specialization and generalization conflicting objectives. In this vein, in this work, we propose an open-set graph-based classifier and three variants of it for information fusion. We exploit fusion information approaches to solve open-set recognition problems by combining different instances of the proposed classifiers and features while, at the same time, taking care of unknown classes that may appear during testing.

In the literature, there exist some techniques aiming at combining information as early as possible (Early-Fusion methods) or as late as possible (Late-Fusion methods) to solve a particular problem. However, sometimes these approaches have a high computational and storage footprint, or suffer with problems related to the curse of dimensionality [8]. Standing out as possible viable fusion alternatives for feature fusion without incurring in serious high-dimensionality problems are Genetic Programming (GP) [9] and Majority Voting schemes [10].

GP was first introduced by Koza [9], inspired by the biological evolution process. The use of GP presents some advantages, including high effectiveness in finding good similarity functions between objects in complex search spaces [6], [11]–[15]. In turn, the Majority Voting [10] fusion technique has long been a staple in information fusion offering a simple, yet effective, late-fusion mechanism easily adaptable to various domains. Ciuonzo and Salvo [16] proved the statistical robustness of majority voting in a binary problem of channel-aware decision fusion outperforming other fusion methods, further advocating for the use of such simple methods.

Taking advantage of fusion methods while at the same time dealing with the open-set recognition setup, in this work, we propose an open-set graph-based Optimum-Path Forest (OSOPF) classifier and variants of it for information fusion. The information fusion methods are allied with Genetic Programming (GP) and Majority Voting fusion techniques. While OSOPF draws a bead on learning decision boundaries more resilient to unknown classes and outliers, the GP, inspired by the biological evolution process, takes aim at harnessing different object properties and combining them to discover appropriate similarity functions to boost classification results through early fusion. In particular, we focus on visual-related problems and the considered features often involve color, texture, and shape properties. Complementary, the Majority-Voting approach combines different classification evidence from different classifier outcomes and features through a late fusion decision-making process. In summary, our methods look for a better description/separation of data points while optimizing for rejecting unknown samples during the testing phase. To the best of our knowledge, this is the first research focused on bringing to bear the power of diversity through fusion and open-set solutions. We start with the introduction of the Open-set Optimum-Path Forest (OSOPF), which extends upon the Optimum-Path Forest (OPF) classifier [17] — inherently closed set — for open-set setups.

Previous work has already attested the effectiveness of the OPF classifier along with genetic programming. Godoi *et al.* [6] combined OPF and GP for the author name disambiguation problem, assuming an open-set regime. Differently from Godoi et al's [6] work, instead of focusing on specific aspects of a given problem (e.g., name disambiguation), we opt for modifying the classifier learning function directly empowering the classifier to handle general-purpose problems.

Some other relevant prior art include the use of GP and OPF in a series of problems, notably closed-set ones. dos Santos *et al.* [18] coupled GP and Relevance Feedback (RF) mechanisms when solving some remote sensing problems. da Silva *et al.* [15], in turn, coupled OPF and GP in content-based image retrieval tasks, also using relevance feedback. The OPF classifier was also tested in information fusion problems involving majority voting. Ponti and Papa [19] proposed a method for combining OPF classifiers that work with disjoint subsets of data and integrated the final classification through a majority voting scheme. Finally, Ponti and Rossi [20] exploited the effects of reduced training sets when combining different OPF classifiers through majority voting.

To validate the proposed methods, we performed experiments on datasets widely adopted in the validation of information-fusion approaches in the context of multimedia classification. The Analysis of Variance (ANOVA) and Tukey's HSD (honest significant difference) tests were used to compare results. The experiments show that the proposed data fusion schemes yield effective results under an open-set recognition regime, significantly outperforming existing counterparts in the literature.

Finally, we organized the remaining of this work into four sections. Section II presents some related methods and concepts used in this work. Section III introduces the proposed methods and also presents the OSOPF classifier, the underlying method of the proposed fusion schemes. Then, Section IV presents the adopted experimental protocol along with experiments and results. Finally, Section V draws conclusions and presents possible directions for future work.

II. BACKGROUND

In this section, we present some concepts for a better understanding of this work. We start with some Genetic Programming concepts (Section II-A) and the Optimum-Path Forest classifier (Section II-B). Then, we present the open-set recognition problem (Section II-C) and some related work. Finally, we discuss some fusion approaches (Section II-D) we use in the work.

A. GENETIC PROGRAMMING

The Genetic Programming (GP) is a technique first introduced by Koza [9] and it is based on the Darwinian principle of reproduction, inheritance and survival of the fittest individual during the biological evolution. This approach, in general, looks for the best computer program (fittest individual) in a wide space of computer programs which are designed to solve a problem with a range of possible solutions (optimization problem).

The GP initial population is a set of randomly generated computer programs (individuals) whose structure programs may comprise: arithmetic operations, programming operations, mathematical functions, or domain-specific functions. Each individual represents a solution of a problem and is assessed with a score (fitness) that is used as an estimation of how close an individual solves the target problem. The best individuals are evolved to create better populations in the next generations until some criteria are reached. The most common representation of the individuals consists of a series of trees (Figure 1) whose leaf nodes are related to variables while internal nodes denote operators.

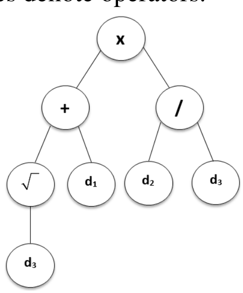


FIGURE 1. Example of a GP individual. The leaf nodes (d_1, d_2 and d_3) are variables while internal nodes ($\times$, $/$, $+$ and $\sqrt{}$) are operators. This individual represents the function $(\sqrt{d_3} + d_1) \times (\frac{d_2}{d_3})$.

GP exploits differences in performance between individuals as some of them will be somewhat fitter than others. The GP, in the evolution phase, chooses the best individuals of each generation and modifies their structures using genetic operations with the goal of obtaining a better population. The most common genetic operations are: **Reproduction**, which selects the best individuals of each generation and copies them to the next generation; **Mutation**, which makes random changes in the structure of individuals; and **Crossover**, which combines genetic material between two parents, exchanging sub-trees. This technique has been used in ranking functions [13], image retrieval [12], [14], [21]–[23], multimodal image retrieval [11], author name disambiguation [6], deriving vegetation indices [24], remote sensing image classification [18], [25], [26], among other applications.

More specifically, in the **Reproduction** operation (asexual), based on the Darwinian natural selection process, the best individuals (regardless of their fitness) of each generation are selected to be part of the next generation, without modifications. The **Mutation**, in turn, is an asexual operation that needs only one individual and has the ultimate goal of diversifying the population through random alterations in the individuals' structure — it inserts a random subtree from a random internal or leaf node by cutting off the initial subtree located at that point. The **Crossover** (sexual operation) takes aim at introducing variation in the population. Firstly, this operation takes two individuals according to their fitness function and selects one random node (crossover point) at each individual, and then, the subtrees that have the crossover points as roots are interchanged [9].

B. OPTIMUM-PATH FOREST

Optimum-Path Forest (OPF) is a fast graph-based closet-set multiclass classifier [17] which has been used for different classification problems in prior art [6], [27]–[33]. It works as a graph partitioning problem in which the weight of the edges might be given by a similarity or distance measure between nodes. The basic idea behind this classifier is to create a complete graph where each node is a feature vector, then, we create partitions of the graph in order to group samples from the same class, taking into account that we can have more than one partition for each class. To create the graph, the OPF for unsupervised classification uses a k-nearest neighbors (k-NN) method to generate the initial graph that is used for the algorithm. On the other hand, for supervised problems, it starts with a complete graph. The fitting and classification phase for this kind of problems are presented next.

1) FITTING PHASE

Given an edge-weighted complete graph $G = (D_1, A)$, each pair of nodes is connected by an arc in $A = D_1 \times D_1$ where D_1 is the training set (Figure 2a). The Minimum Spanning Tree (MST) [34] is calculated on G to find the prototypes T, being $T \subset D_1$. In the MST, a sample x is considered as a prototype if it is connected to a sample y, such that their labels are not equal (Figure 2b), i.e., nodes connected through

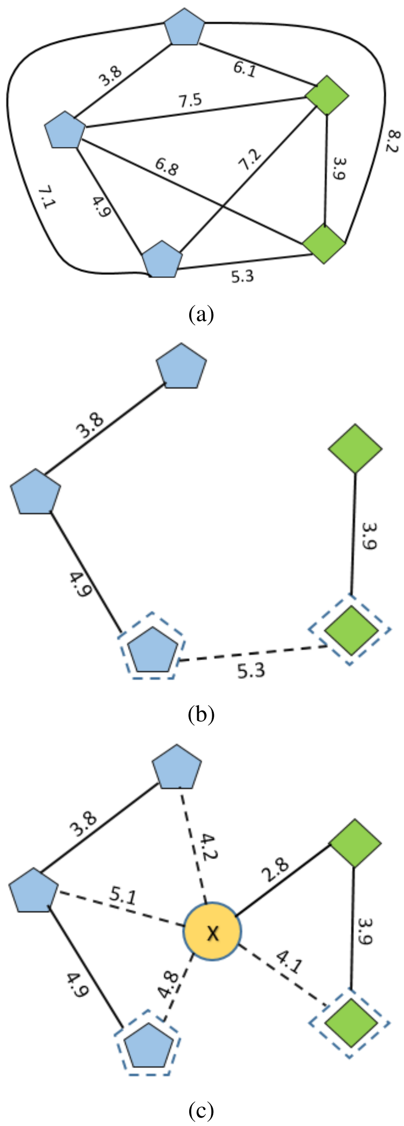


FIGURE 2. **Optimum-Path Forest classifier (*blue* and *green* classes). (a) Complete Graph, (b) Minimum Spanning Tree (MST) – Prototypes selection, samples wrapped in dotted lines. (c) Classification of sample *x* as part of the *green* class.**

the MST that belong to different classes (samples located in the separation frontier). Each class could be represented by one or more optimum-path trees, which have a *prototype* as root.

The distance between two samples x and y is calculated by $d(x, y)$, where $d(x, y) \geq 0$. A path that ends in the sample x is a sequence of nodes $\pi_x = \langle s, s_1, s_2, \ldots, x\rangle$ that has a cost given by the function $f(\pi_x)$, and $\pi_x \cdot \langle x, y\rangle$ represents the concatenation of the path π_x and the arc (x, y). A path is trivial if $\pi_x = \langle x\rangle$ and is optimum if with any other path ς_x, $f(\pi_x) \leq f(\varsigma_x)$. In the OPF, the connectivity function $f_{max}(\pi)$ is given by the greater arc in the path π:

The distance between two samples x and y is calculated by $d(x, y)$, where $d(x, y) \geq 0$. A path that ends in the sample x is a sequence of nodes $\pi_x = \langle s, s_1, s_2, \ldots, x\rangle$ that has a cost given by the function $f(\pi_x)$, and $\pi_x \cdot \langle x, y\rangle$ represents the concatenation of the path π_x and the arc (x, y). A path is trivial if $\pi_x = \langle x\rangle$ and is optimum if with any other path ς_x, $f(\pi_x) \leq f(\varsigma_x)$. In the OPF, the connectivity function $f_{max}(\pi)$ is given by the greater arc in the path π:

$$f_{max}(\langle x\rangle) = \begin{cases} 0 & \text{if } x \in T, \\ +\infty & \text{otherwise} \end{cases}$$
$$f_{max}(\pi_x \cdot \langle x, y\rangle) = \max\{f_{max}(\pi_x), d(x, y)\}. \quad (1)$$

With the minimization of f_{max}, each one of the samples $y \in D_1$ has an optimum path $P^*(y)$ from T, where the minimum cost $C(y)$ is:

$$C(y) = \min_{\forall \pi_y \in (D_1, A)} \{f_{max}(\pi_y)\} \quad (2)$$

Finally, with $P^*(y)$ already defined, the label of y is given by $L(y)$ following the predecessors of y through the path up to its root $R(y) \in T$. In V, Algorithm 5 shows the training stage of the OPF classifier. This algorithm is rooted at the Image-Forest Transform (IFT) [35].

2) CLASSIFICATION PHASE

Let D_2 be the testing set. Each sample $z \in D_2$ seeks an optimum path based on the cost function defined by Equation 3:

$$C_2(z) = \min\{\max\{C(x), d(x, z)\}, \forall x \in D_1\}, \quad (3)$$

which represents the best path, based on the minimum maximum arc to reach a prototype through a sample x. Let s be the sample that offers the lowest cost to z, then, the label assigned to z is $L(z) = L(s)$, that is equivalent to $L(z) = L(R(s))$ (Figure 2c).

Papa *et al.* [36] also presented Enhanced OPF (EOPF), which alters the OPF classification algorithm avoiding to visit all the nodes in the search for the best path, speeding up the prediction stage. For this, in (3) D_1 can be replaced by D', where D' has all the samples of D_1 but ordered by the cost (non-decreasing order) assigned during the training phase. Given x as the current evaluated sample in D', the visit stops when $\max\{C(x), d(x, z)\} < C(x')$ for a sample x' whose position in D' succeeds the position of x. EOPF is detailed in V (Algorithm 6).

C. OPEN-SET RECOGNITION

In open-set recognition, samples of classes not seen during training (*unknown* classes) may appear at testing. Unlike common closed-set problems, in open-set setups, the classifiers have to be capable of returning, as result, either one of the *known* classes or reject the sample by classifying it as *unknown* [1]. In this regime, we need techniques with strong generalization — i.e., capable of determining whether a datapoint is too far away from any supporting training samples and properly rejecting it. Moreover, another point to take into account is the optimization considering the *risk of the unknown* and not only the *empirical risk*, present in virtually all closed-set methods.

Departing from traditional closed-set modeling, more recently some authors have started to focus on open-set problems. Most of the techniques for this kind of scenario are based on adaptations of well-known classifiers such as Support Vector Machines (SVM) [1], [2], [5], [7], [37], [38] and Optimum-Path Forest (OPF) [6]. SVM-based approaches [1], [2], [5], [7], [38] for open-set recognition commonly rely upon the one-vs-all policy [39] for extending the binary classifiers to multiclass classification (SVM-OVA). When combining binary classifiers into a multiclass one by using one-vs-all policy, a test sample is classified as unknown when all binary classifiers classify it as negative. When at least one binary classifier classifies the input sample as positive for a certain class, the class with highest confidence is chosen as the final class.

This reasoning was also applied by Heflin *et al.* [40] and Pritsos and Stamatatos [37] but with one-class classifiers. The advantage of using one-class SVMs is that each one-class classifier classifies as positive possible test samples only in a bounded region of the feature space. When allied with the one-vs-all policy, possible faraway test samples, outside the support of the training samples, would be classified as unknown. The disadvantage resides on the well-known over-specialization of one-class SVMs [41], which impacts these classifiers heavily in terms of high false unknown rate. In another work, Support Vector Data Description [42], a form of one-class classification, was not primarily proposed for open-set recognition, but serves for this intent as shown by Pritsos and Stamatatos [37].

The binary classifier 1-vs-Set Machine — in which known classes represent the "*Set*" out of all possible unknown classes that can appear at testing — is an SVM adaptation proposed by Scheirer *et al.* [1] to deal with open-set problems. Following a linear kernel formulation, for each binary classifier, it employs two parallel hyperplanes to better support generalization and specialization, seeking to bound positive samples in between them. The objective is to minimize the risk of the unknown in the multiclass level by decreasing the *false positive* at the binary classification level. An optimization phase is performed to find a balance between the *empirical risk* (measured on training data) and the *open-space risk*. Following a similar reasoning, Costa *et al.* [2], [5] proposed a technique using a binary SVM along with a searching process referred to as Decision Boundary Carving (DBC) to the problem of assigning an image to a specific camera in an open-set setup (source camera attribution). Instead of two hyperplanes, only one hyperplane is used for each binary classifier, but also aiming at decreasing the false positive rate.

Departing from previous formulations, Scheirer *et al.* [7] took aim at the Extreme Value Theory (EVT) and proposed a Weibull-calibrated process [43] to normalize classification scores of one-class and binary SVMs. In this sense, their WSVM method estimates the probability of a test sample being positive and a probability of not being negative for each binary model and combines both probabilities to accept or reject a sample. In another work also leveraging the powerful extreme value theory, Jain *et al.* [44] proposed the SVM with Probability of Inclusion (PISVM) in order to estimate the unnormalized posterior probability of class inclusion. The probability of class inclusion combines one-class rejection ability and the discrimination ability of binary classifiers.

Moving away from the SVM open-set modeling, Godoi *et al.* [6] used the OPF classifier along with genetic programming to propose a solution for the author name disambiguation problem. Such problem takes place whenever the same author publishes articles using similar names (synonyms) or distinct authors use analogous names (homonyms). To solve the problem, the authors used a threshold to define new subtrees in the OPF classifier for new authors (new classes). The GP was used to find similarity functions among the authors' references.

Bendale and Boult [45] proposed a Nearest Non-Outlier (NNO) algorithm, extended from Nearest Class Mean (NCM) classifiers, for open-set recognition. The NNO classifier is based on the mean vectors of each known class. This method detects outliers for bounding the open-space risk and rejects a sample s when all classifiers reject it. Moreover, this algorithm can add new categories on-the-fly based on human-labeled data. More recently, a new approach to adapt deep neural networks for open-set recognition problems was proposed by Bendale and Boult [46]. Their proposed OpenMax layer is an adaptation of the penultimate layer, SoftMax, for open-set problems. Once again harnessing concepts from EVT, the OpenMax layer computes the probability of a sample being from a class not seen during the training stage. The rejection of a new sample s is determined by using the Weibull CDF probability on the distance between the penultimate layer obtained with s and the Mean Activation Vector m_c, considering c as the most probable class. m_c is computed by using training images from class c that are correctly predicted by the trained network.

D. INFORMATION FUSION

Early fusion and *late fusion* are two popular approaches for data fusion [47]. Early-fusion methods seek to aggregate various independent types of features into one lengthier (e.g., through feature vector concatenation) feature vector before some machine learning algorithm can be applied. One limitation of this approach is, naturally, the increase in feature dimensionality [8]. On the other hand, late-fusion methods normally combine classification predictions (e.g., classification scores or probabilities) for different feature sets.

Other techniques that have been used for information fusion are the *ensemble* methods [48]. Those methods often use a voting scheme using a set of classifiers trained for the same task. Some of the algorithms that use this kind of approach include Bayesian averaging [49], Error-correcting output coding (ECOC) [39], Bagging [50], and Boosting [51]. Another widely used technique for data fusion is Genetic Programming (GP), with applications ranging from image recognition [18], [25], [26] and information retrieval [12]–[14], [21]–[24] to multimodal retrieval problems [11].

In closed-set setups, information fusion approaches have often led to good classification results [18], [25], [26], [52], [53]. In turn, when considering open-set configurations, to the best of our knowledge, those fusion methods have yet to be exploited. In such problems, the task of dealing with samples that belong to classes not seen during the training stage is very challenging. For this reason, it is paramount to look at alternative (and complementary) ways of characterizing the problem of interest seeking a better separation of samples among known classes while empowering the classifier to reject unknown ones. Given that information fusion can be used to take advantage of different types of features in a given problem, it is only natural to think of it in object recognition tasks when dealing with open-set setups, as we detail in the next section.

III. OPEN-SET FUSION METHODS

In this section, we propose four inherently multi-class classifiers for open-set recognition problems. First, Section III-A presents the Open-set Optimum-Path Forest (OSOPF) a principled graph-based classifier tailored for open-set setups. Then we turn our attention to extending upon the initial OSOPF method with Open-set fusion formulations leveraging Genetic Programming (GP) and Majority Voting (MV) to take advantage of several sources of information while dealing with different open-set problems. Before proceeding any further, the reader might consider reviewing OPF's properties (Section II-B) and algorithms for closed-set problems (A).

A. OPEN-SET OPTIMUM-PATH FOREST (OSOPF)

In this research, we propose an inherently multiclass graph-based method, the Open-Set Optimum-Path Forest (OSOPF), as a new approach for open-set recognition problems based. The prediction of our method — inspired in the work of Mendes Júnior *et al.* [54] for nearest neighbors — is based on the cost ratio of the two best paths of different known classes. OSOPF is an extension upon the Optimum-Path Forest — an innate closed-set classifier. It uses the same cost function (Eq. 3) and a similar training phase (Algorithm 5) to the OPF classifier. Unlike the closed-set OPF formulation, however, in the training stage, we add a key modification, an optimization phase that seeks for the best parameter (threshold t) for the open-set decision-making process later on. In addition, the OSOPF's recognition phase (testing operation) is different as it takes the open-set setup into consideration when classifying an input sample.

Our approach is based on the comparison of a decision threshold t (optimized in the training phase using a grid-search procedure and a simulation of an open-set setup) and a cost relation of the two best paths from different classes to a given input sample. Given x as input, it looks for the two nearest classes that offer the best classification costs c_1 and c_2 (according to the path cost function in Equation 3), respectively. Then, the relation $r = \frac{c_1}{c_2}$ is calculated. Being s the nearest sample of the path that offers the best prediction cost c_1, x is assigned to the class of the prototype $s^p = R(s)$; otherwise, x is classified as *unknown*, i.e.,

$$L(x) = \begin{cases} L(s^p) & \text{if } r \leq t \\ unknown & \text{if } r > t. \end{cases} \quad (4)$$

OSOPF training and recognition phase are presented in Algorithm 1.

Algorithm 1 Training and Recognition Phases of OSOPF

Require: Set of training samples S.
Require: Test sample x.
OPF forest generated in the training phase using S (Section II-B and V).
$t \leftarrow$ threshold optimized in the training phase (Section III-A).
$\pi_1 \leftarrow$ best path to x in the OPF according to f_{max} (Cost function in Eq. 3).
$\pi_2 \leftarrow$ second best path to x in the OPF, such that $L(R(\pi_1)) \neq L(R(\pi_2))$.
$r \leftarrow f_{max}(\pi_1)/f_{max}(\pi_2)$ (Cost function in Eq. 3)
if $r \leq t$ **then**
    $L(x) \leftarrow L(R(\pi_1))$.
else
    $L(x) \leftarrow$ "unknown".
end if

Parameter Optimization: In OSOPF, we use a parameter optimization phase simulating an open-set setup to find the best value for the threshold t. In this research, the samples were divided into training and testing sets (see Figure 3a). In an open-set scenario, the testing set has samples of known and unknown classes, as samples of classes whose no representative was present during training can also appear (see Figure 3b). In our parameter optimization phase, we split samples of the training set into fitting set and validation set. The fitting set is used to train the classifier while the validation set is used to verify the accuracy based on the value t. The training set is divided according to the following: to simulate the open-set setup on the parameter optimization phase, only half of the available classes for training have representative samples in the fitting set (samples of the remaining classes are in the validation set), and for each class considered in the fitting set, half of its samples is in the fitting set (the remaining is in the validation set; see Figure 3c).

Finally, an OSOPF classifier is fitted using the fitting set, and, with the samples of the validation set, a grid search [55] procedure is executed to find the best threshold t. Notice that at least three available classes in the training set are required to the parameter optimization phase because at least one of those classes completely belongs to the validation set (to simulate an open-set setup), and the fitting set must contain samples of two different classes. More details about the dataset partitioning are presented in Section IV-A.2.

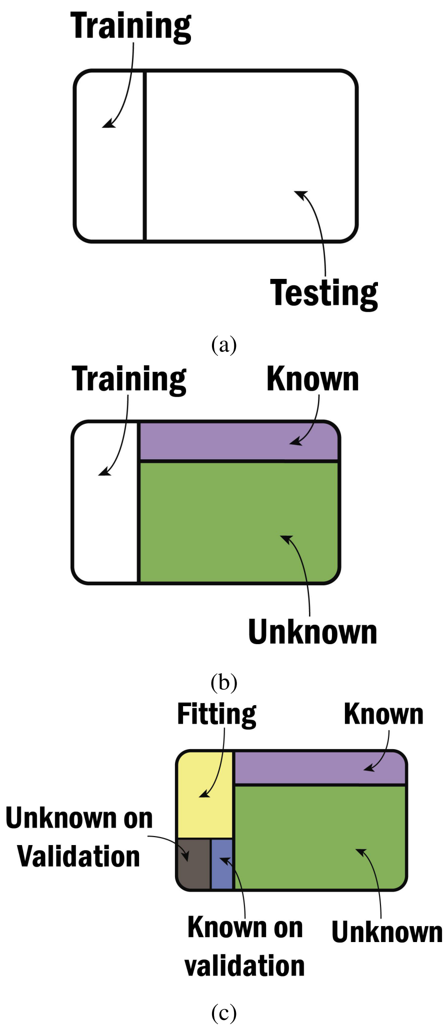


FIGURE 3. General scheme of data partitioning for the parameter optimization of OSOPF: (a) a dataset is divided into training and testing sets. (b) Most of the samples in testing set are from unknown classes (whose no representative at all was seen during the training stage). (c) Partitioning of the training set to simulate an open-set scenario for parameter optimization of OSOPF.

Our approach extends upon the traditional closed-set OPF and introduces modifications to verify if a test sample can be classified as *unknown*. To the best of our knowledge, OSOPF extension is the first version of the OPF classifier suitable for general open-set recognition. Its main advantage is that it is inherently multiclass, i.e., the efficiency of OSOPF is not affected as the number of available classes for training increases, differently from open-set one-vs-all SVM extensions in the literature, which are rely upon combining several binary classifiers. Moreover, OSOPF can create a bounded risk of the unknown space for every known class (using the support of existing training samples) therefore gracefully protecting the classes of interest and rejecting unknown ones. These two advantages make OSOPF ideal for developing solutions for novelty detection and online learning, which could detect unknown classes on-the-fly and include them on the recognition system automatically.

The region in the feature space in which a test sample will be classified as belonging to a specific class is referred to as *decision boundary*. Figure 4 shows the decision boundaries comparing two different decision values for OSOPF (threshold values of 0.5 and 0.8). White areas correspond to regions where a sample would be labeled as *unknown*. We can see how OSOPF builds its decision surface, according to the threshold, creating a bounded open-space considering the risk of the unknown. This OSOPF's behavior gives a favorable setting to reject samples, predicting the *unknown*.

B. OPEN-SET FUSION METHODS

We propose three generic inherently multiclass methods for open-set recognition problems based on OSOPF and fusion techniques. Our methods aim at improving the object recognition rate using Genetic Programming (GP) and Majority Voting to combine any kind and number of visual properties in a given problem through early- and late-fusion approaches. The proposed methods are OSOPF Open-GP ($OSOPF_{OGP}$), OSOPF Closed-GP ($OSOPF_{CGP}$), and OSOPF Majority Voting ($OSOPF_{MV}$).

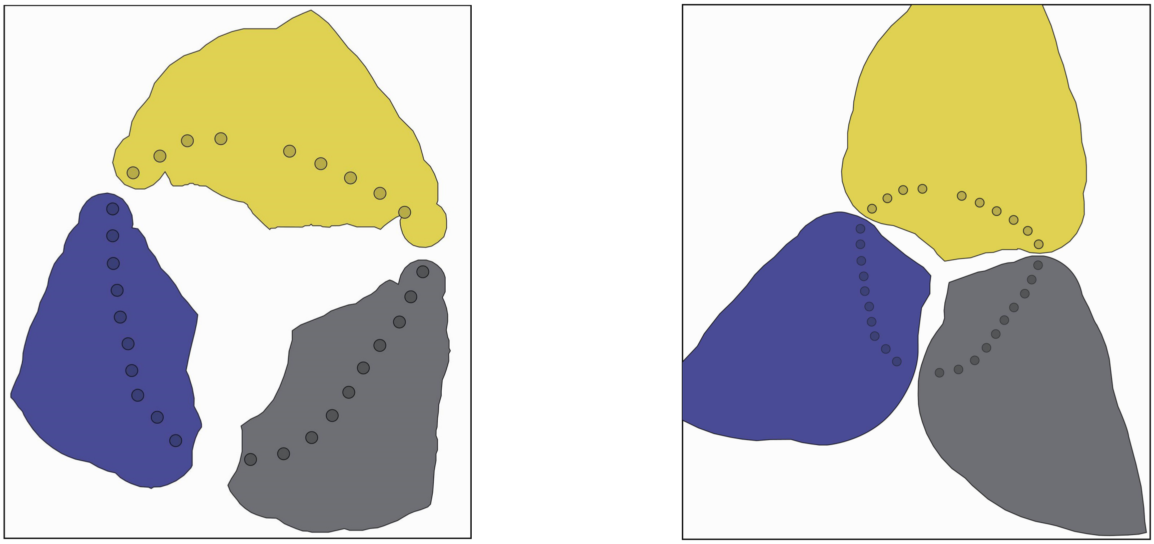
$OSOPF_{OGP}$ and $OSOPF_{CGP}$ use genetic programming to combine features through early fusion. The use of GP aims to discover an individual that allows a better separation between samples. For this, we look for the best GP individual (tree) to be used as distance function to calculate the arc weights between the objects in the OPF graph. In the GP Individual, internal nodes correspond to mathematical operators ($+$, $/$, $\times$ and $\sqrt{\ }$) and each leaf node takes the value of the Euclidean distance between two objects described by a particular visual feature (e.g., texture and color).

The GP-methods are represented by the pair (γ, δ), where γ is the OSOPF classifier and δ is an individual (distance function) generated by GP. To evaluate each pair (γ, δ), we create an optimum-path forest using the *fitting* set. Two sets of samples $validation_1$ and $validation_2$ are used to assess the OPF classifier. The $validation_1$, used during n generations, serves to pre-select the pairs (γ, δ) with the best performance. In turn, $validation_2$ is used to evaluate the generalization of the GP classifiers and try to avoid overfitting. The selection of the best pair (γ, δ) is based on the average classification accuracy

$$avgAccuracy = \frac{ac_1 + ac_2}{2},$$

where ac_1 and ac_2 are the normalized accuracies obtained by the classifier in the sets $validation_1$ and $validation_2$, respectively. The GP-methods have the goal of finding the best GP individual (δ) that allied with the OSOPF classifier (γ) gives a better separation of the data.

Sections III-B.1 and III-B.2 present the open-set GP methods that consider, respectively, a closed-set and an open-set training regime. The main difference between them is that while the closed-set GP simulates a closed-set setup during the training, the open-set GP creates an open-set setup by pretending not to have access to some of the known classes



(a) (b)

FIGURE 4. **Decision boundaries for a synthetic dataset with two different decision thresholds. Samples in white regions would be classified as *unknown*, while samples in a non-white area would be classified as part of the class of the samples with the same color. The stricter we are in the value, the more we protect classes of interest (specialize to them). The parameter-optimization step automatically finds the value for t based on available training/validation examples. (a) OSOPF for $t = 0,5$. (b) OSOPF for $t = 0,8$.**

during training. Finally, in Section III-B.3, we propose a voting-based method (OSOPF_{MV}) through late fusion that combines different objects' features.

As OSOPF is a simple graph-based classifier that during training stage creates optimum-path trees with samples from the same class (each class may be represented by one or more optimum-path trees) based on the distance between the nodes (objects), the GP is a useful approach to generate a complex distance function using any number of sources of information to improve the compaction of the optimum path trees and lead to a better separation between trees of different classes.

In turn, given that OSOPF is an algorithm that has a fast training stage, it is effective to use OSOPF along with a Majority Voting scheme as well. This approach improves its recognition rate through the "opinion" of some independent classifiers (committee) that use OSOPF as a base algorithm (innately open set) but each one working over different types of features.

1) OSOPF Open-GP ($\text{OSOPF}_{\text{OGP}}$)

In this method, during the training stage, we simulate an open-set setup (pretending that some of the known classes are unknown) with the goal of reaching a reasonable generalization/specialization tradeoff for the classifier, i.e., trying to define, based on a *threshold*, a limit on how far a sample might be from a tree of class x in order to be labeled as belonging to that class. As an example, suppose the classifier has access to five classes (known labels) during training but many more can appear later on during testing. To simulate an open-set setup during training, the method "pretends" not to know some of the five classes it has labels and then uses them to better drive the learning process without over-specializing to those five classes of interest — e.g., pretending to know only three out of the five during training. It uses the partitioning showed in Section IV-A.2.a.

Algorithm 2 presents this method. The sets $validation_1$ and $validation_2$ have samples that belong to classes not seen in the *fitting* set. For its part, $validation_3$ is used to find the best threshold (see Section IV-A.2.a). In Lines 1 and 2, the set I is created to store the best individuals of each generation and the initial GP population is generated, respectively. In Lines 3-11, the GP individuals evolve during $N_{evolutions}$ evolutions. During the $N_{evolutions}$ evolutions, each individual is evaluated (Lines 4-7). For this, first, in Line 5, an optimum-path forest (OPF) is created using the *fitting* set (a proper subset of the training set), where each GP-individual is used as a function distance. Then, in Line 6, the OPF is assessed with the $validation_1$ set using OSOPF as classifier. The best $N_{topIndividuals}$ are selected in each generation and the GP population is evolved using Mutation, Reproduction, and Crossover operators (Lines 8, 9 and 10). Next, in Lines 12-16, an OPF is created (based on the *fitting* set) for each one of the best individuals that were saved in the set I, then, the OPFs are assessed with OSOPF and the mean normalized accuracy is calculated, by computing the normalized accuracy obtained in the sets of samples $validation_1$ and $validation_2$ for each GP individual i. After that, the best individual i_{best} is selected in Line 17. Recall that OSOPF uses a *threshold* on a ratio to reject unknown samples. For that reason, in Line 18, the $\text{OSOPF}_{\text{OGP}}$ executes a grid-search to optimize the value of the *threshold*. The *training* set is created in Line 19, where $training = fitting \cup validation_1 \cup validation_2 \cup validation_3$. At the end, the final forest (OPF) is created with the *training* set, using the i_{best} individual as a

Algorithm 2 Algorithm for $\text{OSOPF}_{\text{OGP}}$

Require: *fitting*, $validation_1$, $validation_2$, $validation_3$ (see Section IV-A.2.a)
Require: $threshold$, $N_{evolutions}$, $N_{topIndividuals}$
1: $I \leftarrow \emptyset$.
2: $A \leftarrow$ initial population.
3: **for** each g generation of $N_{evolutions}$ **do**
4: **for** each $i \in A$ **do**
5: $forest \leftarrow OPF(fitting, i)$ (Section II-B and V – Alg. 5).
6: $i.fitnessValue \leftarrow OSOPF(forest, i, threshold, validation_1)$ (Section III-A – Alg. 1) //normalized accuracy.
7: **end for**
8: $I_g \leftarrow N_{topIndividuals}$ of g
9: $I \leftarrow I \cup I_g$
10: $A \leftarrow$ new population (Reproduction, Crossover and Mutation).
11: **end for**
12: **for** each $i \in I$ **do**
13: $forest \leftarrow OPF(fitting, i)$ (Section II-B and V – Alg. 5).
14: $fitness_{val2} \leftarrow OSOPF(forest, i, threshold, validation_2)$ (Section III-A – Alg. 1). //normalized accuracy.
15: $i.fitnessValue \leftarrow (i.fitnessValue + fitness_{val2})/2$
16: **end for**
17: $i_{best} \leftarrow bestIndividual(I)$.
18: $bestThreshold \leftarrow gridSearch(OSOPF, i, validation_3)$ (Section III-A).
19: $training \leftarrow fitting \cup validation_1 \cup validation_2 \cup validation_3$
20: $forest \leftarrow OPF(training, i_{best})$ (Section II-B and V – Alg. 5).
21: $OSOPF(forest, i_{best}, bestThreshold, testing)$ //recognition in the *testing* set (Section III-A – Alg. 1).

distance function (Line 20). The classifier is assessed in the testing phase (Line 21).

2) OSOPF Closed-GP ($\text{OSOPF}_{\text{CGP}}$)

This method uses a closed-set training looking for a suitable representation for known classes in order to have a good "knowledge" about them, i.e., it tries to find a better separation between samples of different classes and minimize the within-class dispersion. In the training stage, this method uses the OPF and the partitioning of the samples presented in Section IV-A.2.b. All classes that appear in partitions $validation_1$ and $validation_2$ have representative samples in the *fitting* set. On the other hand, in the testing (open-set setup), the best δ individual from the training is used along with the OSOPF classifier (γ).

Algorithm 3 details this method. Best individuals of each generation are saved in I and the initial population

Algorithm 3 Algorithm for $\text{OSOPF}_{\text{CGP}}$

Require: *fitting*, $validation_1$, $validation_2$ (see Section IV-A.2.b)
Require: $threshold$, $N_{evolutions}$, $N_{topIndividuals}$
1: $I \leftarrow \emptyset$.
2: $A \leftarrow$ initial population.
3: **for** each g generation of $N_{evolutions}$ **do**
4: **for** each $i \in A$ **do**
5: $forest \leftarrow OPF(fitting, i)$ (Section II-B and V - Alg. 5).
6: $i.fitnessValue \leftarrow OPF(forest, i, validation_1)$ (Section II-B and V - Alg. 6)//normalized accuracy.
7: **end for**
8: $I_g \leftarrow N_{topIndividuals}$ of g
9: $I \leftarrow I \cup I_g$
10: $A \leftarrow$ new population(Reproduction, Crossover and Mutation).
11: **end for**
12: **for** each $i \in I$ **do**
13: $forest \leftarrow OPF(fitting, i)$ (Section II-B and V - Alg. 5).
14: $fitness_{val2} \leftarrow OPF(forest, i, validation_2)$ (Section II-B and V - Alg. 6)//normalized accuracy.
15: $i.fitnessValue \leftarrow (i.fitnessValue + fitness_{val2})/2$
16: **end for**
17: $i_{best} \leftarrow bestIndividual(I)$.
18: $training \leftarrow fitting \cup validation1 \cup validation_2$
19: $forest \leftarrow OPF(training, i_{best})$ (Section II-B and V).
20: $OSOPF(forest, i_{best}, threshold, testing)$ (Section III-A - Alg. 1) //recognition in the testing set.

is generated (Lines 1 and 2, respectively). In Lines 3-11, GP individuals are evolved during $N_{evolutions}$. In each evolution, every GP individual is assessed (Lines 4-7). For this, first, in Line 5 it is created an optimum-path forest (OPF) with the *fitting* set, where each GP individual is used as a distance function. Next, in Line 6, the OPF is evaluated with the $validation_1$ set using the OPF classification phase. In Lines 8, 9 and 10, the $N_{topIndividuals}$ of each generation are selected and the population is evolved using the Mutation, Reproduction and Crossover operators. In Lines 12-16, new OPFs are created using the *fitting* set and each one of the best individuals in the I set. Each OPF is assessed using the OPF classification phase and is calculated the mean normalized accuracy for each individual i based on their normalized accuracies obtained in the $validation_1$ and $validation_2$. The best individual is selected in Line 17. Then, in Line 18, the *training* set is created with the *fitting*, $validation_1$, and $validation_2$. Finally, the *training* set and the i_{best} (distance function) are used to define an OPF classifier. This classifier is evaluated using a *testing* set with the OSOPF approach.

3) OSOPF Majority Voting ($OSOPF_{MV}$)

Majority Voting is a widely used technique in information fusion [10], [19], [20]. The $OSOPF_{MV}$ classifier takes into account the opinion of some OSOPF classifiers that contribute with views of the problem. Given a pair (γ, D) — where γ is the OSOPF base classifier and D is a set of descriptors — the $OSOPF_{MV}$, for each descriptor, computes an optimum-path forest (OPF) using the Euclidean distance to separate the nodes of the OPF. Then, the method considers the labels given to a sample by each pair (γ, D_i), where $D_i \in D$. The sample will be labeled with the label that has the most number of votes — tie-breaking consists of randomly choosing one of the labels.

Algorithm 4 Algorithm for $OSOPF_{MV}$

Require: *classifier*, *training*, *testing*
Require: Set of descriptors D
1: $F(descriptor, forest) \leftarrow \emptyset$.
2: **for** each $descriptor \in D$ **do**
3: $forest \leftarrow OPF(training, descriptor)$ (Section II-B and V - Alg. 5).
4: $F \leftarrow F \cup (descriptor, forest)$.
5: **end for**
6: **for** each $x \in testing$ **do**
7: $P \leftarrow \emptyset$
8: **for** each $(descriptor, forest) \in F$ **do**
9: $label \leftarrow OSOPF(forest, descriptor, x)$ (Section III-A - Alg. 1).
10: $P \leftarrow P \cup label$.
11: **end for**
12: $x.label \leftarrow$ assign the *label* that has more occurrences in P; if there is a tie, choose randomly one of the *labels* that are involved in the tie.
13: **end for**

Algorithm 4 details the $OSOPF_{MV}$ method. In the first line, F is initialized. In Lines 2-5, with the *training* set, it is generated an optimum-path forest (OPF) for each descriptor. These forests are stored in F. The label assignment for each sample is defined in Lines 6-13 (testing phase). In Lines 8-11, it is assigned one label for each OPF $\in F$ and each assigned *label* is saved in P. Note that in Line 9, the OSOPF approach is used as classifier. At the end, in Line 12, it is assigned the *label* with more occurrences in P — tie-break consists of randomly choosing one of the labels.

IV. EXPERIMENTS AND DISCUSSION

We now turn our attention to presenting the experimental protocol (Section IV-A), and results of the experiments (Section IV-B) with the proposed methods compared to baselines proposed for open-set scenarios.

A. EXPERIMENTAL PROTOCOL

In this section, we describe the datasets (Section IV-A.1), the partitioning scheme (Section IV-A.2) for the training/validation/testing experiments, descriptors (Section IV-A.3), the GP configuration (Section IV-A.4), and the grid-search procedure (Section IV-A.5) configuration adopted in this work.

In this work, we adopted the Normalized Accuracy (NA) and Open-Set F-Measure (OSFM) as evaluation measures, following prior work for open-set scenarios [54]. OSFM can also assume its macro- and micro-averaging forms: $OSFM_M$ and $OSFM_\mu$, respectively.

1) DATASETS

Proposed fusion methods were evaluated in the following datasets: CALTECH-256 [56] ALOI [57], and COIL [58]. We selected these datasets because they provide a different degree of difficulty in the recognition task. In Table 1, we present some characteristics of each dataset, considering number of classes and samples. We also present the minimum, maximum, and average number of elements per class in each dataset. The *openness* of each dataset for experiments with 3, 6, 9, 12, and 15 known classes are presented in Table 2. *Openness* is a measure proposed by Scheirer *et al.* [1] to assess how open (when there are unknown classes on testing) is the scenario of an experiment.

TABLE 1. Datasets considered in this work.

Dataset	# classes	# samples	Images per class min	max	average
ALOI	1000	110250	108	111	110
CALTECH-256	256	29780	80	800	116
COIL	100	7200	72	72	72

TABLE 2. *Openness* of the considered datasets for 3, 6, 9, 12, and 15 known classes.

# known classes	ALOI (1000 classes)	CALTECH-256 (256 classes)	COIL (100 classes)
3	0.945	0.892	0.827
6	0.923	0.847	0.755
9	0.905	0.813	0.700
12	0.891	0.784	0.654
15	0.878	0.758	0.613

2) DATASET PARTITIONING

Samples of each dataset were divided into training and testing sets in order to create an open-set setup by leaving samples of some classes as *unknown* for testing. Moreover, we divided the samples of the training set into *fitting* (samples used to fit/train a classifier) and *validation* sets (samples used to assess the accuracy during grid-search and avoid overfitting of the classifier being trained). In some cases, this division creates open-set or closed-set setups. As OSOPF needs a parameter optimization phase, we use validation set to execute the grid-search and find the best value for the open-set rejection threshold. The partitioning procedure can be summarized as follows:

1) x classes out of the total n in a given dataset are selected as known classes and the remaining ones are

considered to be part of the set of *unknown classes*. $x \in \{3, 6, 9, 12, 15\}$ is the number of known classes.
2) Samples in x known classes are partitioned again into sets of 80% and 20%, which will be further used for training and testing, respectively.
3) All samples from the *unknown classes* set are used for testing.
4) Samples in the training set may be further divided into fitting and validation sets for parameter optimization.

In the OSOPF_{MV} method, the optimum-path forest is generated with all available training samples. On the other hand, for $\text{OSOPF}_{\text{OGP}}$ and $\text{OSOPF}_{\text{CGP}}$, the open-set and closed-set training setup, respectively, were simulated. For that reason, samples that belong to the training set were further partitioned as described below (Step 4 above):

a: $\text{OSOPF}_{\text{OGP}}$ Partitioning

The $\text{OSOPF}_{\text{OGP}}$ training scenario simulates an open-set setup during the training stage and it works as follows:
1) From the x known classes in the training set, ceiling of half of them are considered as known (y classes) and the remaining as unknown (z classes).
2) The *fitting* set has 40% of the samples from the y known classes and it is used to generate the forest of the OPF classifier.
3) The $validation_1$ is used to assess the generated forest in each GP generation and has the following composition: 30% of the samples of the y known classes and 50% of the samples from the z unknown classes.
4) The $validation_2$ contains 20% of the samples from the y known classes and 35% from the z unknown classes. This set is used to calibrate the learning process and avoid over-fitting.
5) The $validation_3$ is used to find the best threshold in the $\text{OSOPF}_{\text{OGP}}$ classifier. This set is composed of 10% of the samples from the y known classes and 15% from the z unknown classes.

b: $\text{OSOPF}_{\text{CGP}}$ Partitioning

The training of the $\text{OSOPF}_{\text{CGP}}$ method considers a closed-set scenario during the training stage. The partitioning works as follows:
- 40% of the samples from the x known classes are used in the *fitting* stage. This is used to create the optimum-path forest (OPF).
- The $validation_1$ has 30% of the samples from the x known classes and is used to validate the generated forest.
- To avoid over-fitting, the $validation_2$ is used. This set contains 30% of the samples from the x known classes.

3) DESCRIPTORS

We opted to use traditional color and texture descriptors, considering that they contribute differently for the object characterization task. In this vein, we selected the Border/Interior Pixel Classification (BIC) [59], Color Autocorrelogram (ACC) [60], Color Coherence Vector (CCV) [61], Quantized Compound Change Histogram (QCCH) [62], and Local Activity Spectrum (LAS) [63] descriptors. The first three are color descriptors and the last two characterize texture information. Description was not the focus of our work and any other set of features could be considered including bags of visual words and deep-learning generated features.

4) GP CONFIGURATION

We used the JGAP [64] Java library to implement the GP framework. In the case of the mathematical operators and reproduction rate, we used the configuration proposed in [12]. The operators used were: $+$, $/$, $\times$ and $\sqrt{\ }$ along to a reproduction rate of 0.05. In addition to the reproduction rate, there exist some attributes to take into account as well: mutation rate, crossover rate, size of the initial population, number of generations and deep of the tree.

To assess the impact of the mutation and crossover rates, the following strategy was adopted:
- We consider an initial population of 100 individuals and the evolution over 10 generations.
- Mutation and crossover rates were $(0.05, 0.1, 0.2)$ and $(0.2, 0.5, 0.8)$, respectively.
- We perform experiments combining the values of the mutation and crossover (total of 9 experiments).
- We select the values of the rates that obtained the best results.

To assess the importance of the number of individuals (X) in the initial population, number of generations (Y), and depth of the tree (Z) in the results, we used a two-level full-factorial design [65]. This design was tested and used in [24], [66], and [67], in which each parameter is assessed with two values, a low value ($-$) and a high value ($+$). The parameter evaluation results in 2^n experiments, where n is the number of parameters, in our case, $n = 3$. Each experiment was executed three times with different random seeds to generate distinct initial populations. Therefore, there are a total of 24 executions. To evaluate the impacts of different parameter settings, we resorted to the ALOI dataset (considering 9 classes of interest as known classes) and the $\text{OSOPF}_{\text{OGP}}$ classifier.

TABLE 3. Parameter effect.

Parameters (Interaction)	Effect (%)
X	0.583
Y	0.415
Z	-1.236
XY	0.083
XZ	0.222
YZ	0.091
XYZ	-0.072

Table 3 shows the effects of each parameter in the experiments. Note that some parameter settings are not important to the whole classification procedure (with effects lower than 1%). These percentages indicate that none of such parameters affect the results when their values are modified

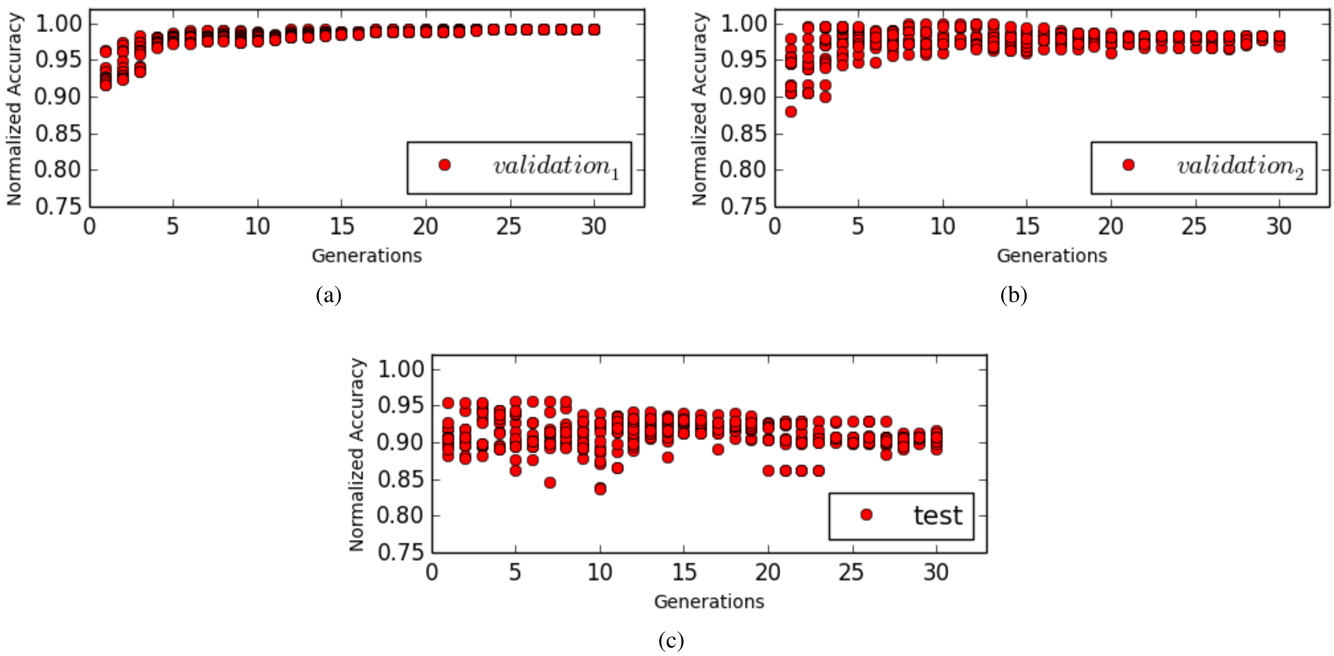


FIGURE 5. Normalized Accuracy evolution considering the best 15 GP individuals in each generation: (a) $validation_1$, (b) $validation_2$ and (c) *test*.

(low or high values). To calculate the effect of a specific parameter p, the following methodology was used:

1) b = mean of all the experiments where p had its low value.
2) a = mean of all the experiments where p had its high value.
3) $effect = a - b$.

a: Parameter settings

Table 4 shows mathematical operators and values of the parameters we used in this work. Figure 5 depicts the evolution curve of the GP classifier. The accuracy scores in the $validation_1$, $validation_2$, and *test*, obtained by the 15 best individuals in each generation are plotted in Figures 5a, 5b, and 5c, respectively. We can see that the NA in $validation_1$ (set used to generate and select the best individuals) are higher the more evolved the GP individuals are. However, the improvement of the individuals in the last generations do not have a high impact on the scores; therefore, the classifier stabilizes with just 30 generations. Note that all curves have a similar behavior with the classifier reaching some state of equilibrium in the last generations.

TABLE 4. GP configuration we adopted in this work.

Parameter	Value
Functions	$+$, $/$, $\times$ and $\sqrt{\ }$
Mutation	0.1
Reproduction	0.05
Crossover	0.8
Initial Population	300
Generations	30
Deep of the tree	4-6

5) GRID SEARCH

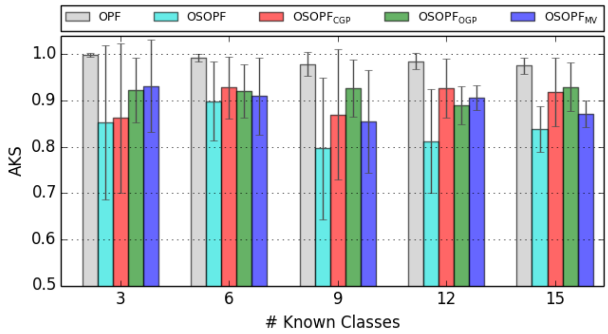
The OSOPF method relies on a threshold to limit how far a sample A could be from the rest of the samples of a particular class C. We performed a traditional grid search [55] procedure to find such threshold based on the samples in the validation set while simulating an open-set setup as discussed in Section III-A. The range of values to look for in the threshold search were [0.5, 1], with 5 grid-search levels and 10 threshold values evaluated in each level.

B. RESULTS AND DISCUSSION

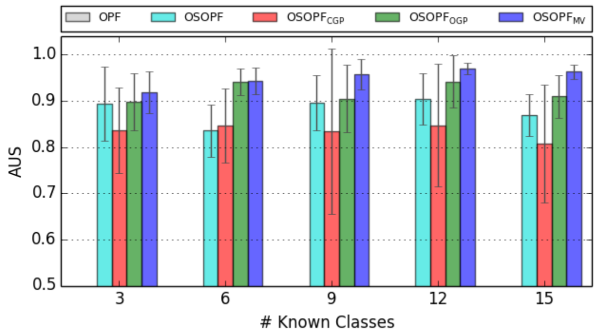
To test the classifiers, we used 10 executions for 3, 6, 9, 12 and 15 known classes out of all existing ones in the dataset for training. In any case, for testing, all classes in the dataset can appear. In other words, we performed 10 executions using 3 classes as known classes and the same for 6, 9, 12 and 15 known classes.

1) VARIATIONS OF THE PROPOSED METHOD

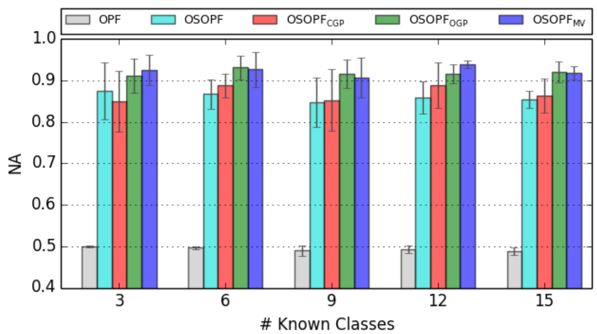
Firstly, we present results of the OSOPF based methods (OSOPF, $\text{OSOPF}_{\text{CGP}}$, $\text{OSOPF}_{\text{OGP}}$ and OSOPF_{MV}) along with the OPF classifier. An interesting aspect to notice here is that the traditional closed-set OPF outperforms our OSOPF method in the AKS. As explained before, this is expected as OSOPF focuses on better balancing the decision on known classes and reject unknown samples. In contrast, in Figures 6b, 7b and 8b we can observe that OPF has no AUS, because the it is a closed-set classifier that never classifies a test sample as unknown. The AUS determines how well the unknown samples are identified at testing phase. In other words, OPF is not recommended for open-set problems.



(a)

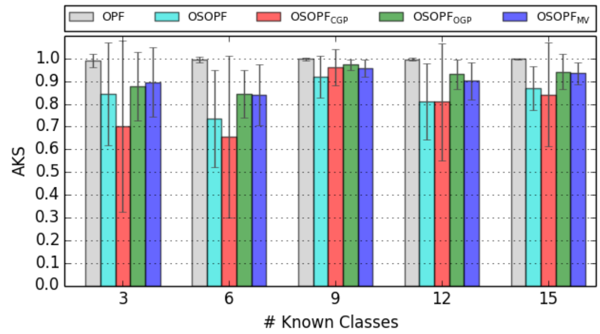


(b)

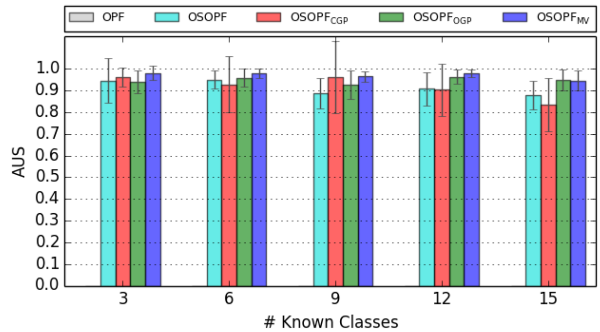


(c)

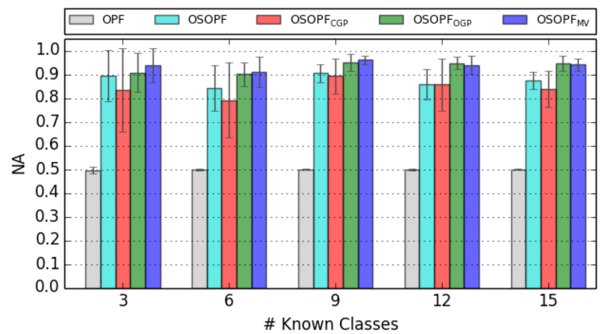
FIGURE 6. Comparison of OSOPF methods for the ALOI dataset. (a) Accuracy on Known Samples (AKS). (b) Accuracy on Unknown Samples (AUS). OPF has no AUS, because the standard OPF is a closed-set classifier that never classifies a test sample as unknown. (c) Normalized Accuracy (NA).



(a)



(b)

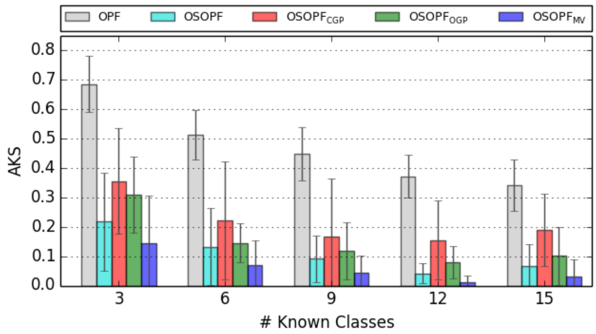


(c)

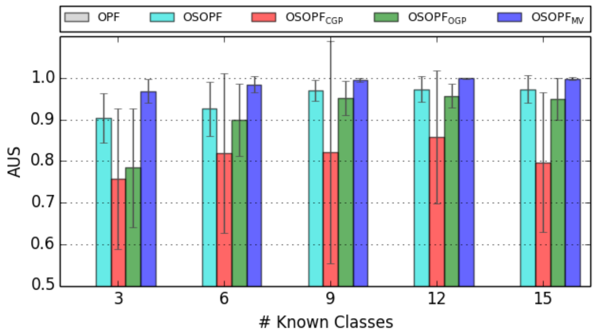
FIGURE 7. Comparison of OSOPF methods for the COIL dataset. (a) Accuracy on Known Samples (AKS). (b) Accuracy on Unknown Samples (AUS). OPF has no AUS, because the standard OPF is a closed-set classifier that never classifies a test sample as unknown. (c) Normalized Accuracy (NA).

In the ALOI dataset (Figure 6), the proposed OSOPF has a better NA than OPF as our method reduces the misclassification with respect to the OPF in 37%, 36%, 36%, 37% and 37% with 3, 6, 9, 12 and 15 known classes, respectively. Furthermore, in the COIL dataset (Figure 7), OSOPF improves the OPF results in 39%, 34%, 40%, 36% and 37% for 3, 6, 9, 12 and 15 known classes, respectively. Finally, in the CALTECH-256 dataset (Figure 8), OSOPF outperforms OPF regarding NA reducing the misclassification by more than 20% in all cases for 3, 6, 9, 12 and 15 known classes.

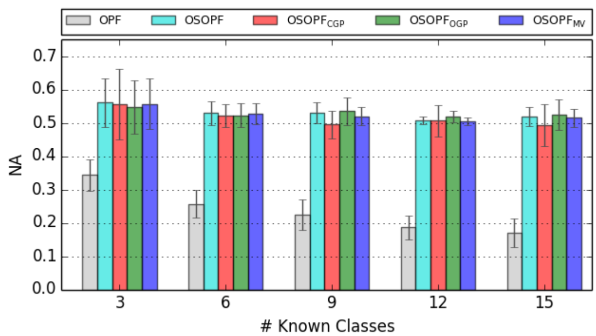
We now turn our attention to the impact of improving OSOPF with fusion methods, via genetic programming (GP) and voting fusion. First of all, it is clear that OSOPF greatly benefits from its combination with the GP and voting fusion schemes. OSOPF$_{OGP}$ is more appropriate to the open-set setup than OSOPF$_{CGP}$. For the ALOI dataset (Figure 6), we have the OSOPF$_{OGP}$ improving the recognition in about 10% (e.g., for 9 and 15 known classes), the OSOPF$_{CGP}$ improving the recognition in 3% and 11.6% (e.g., 6 and 12 known classes), respectively, and the OSOPF$_{MV}$ (e.g., 3 known classes) improving the recognition in about 8% when compared to OSOPF. Furthermore, the OSOPF$_{MV}$ had the best AUS in all the experiments (Figure 6b), boosting previous OSOPF's results in 2.4%, 10.8%, 6.1%, 6.6% and 9.4% for 3, 6, 9, 12, and



(a)



(b)



(c)

FIGURE 8. Comparison of OSOPF methods for the CALTECH-256 dataset. (a) Accuracy on Known Samples (AKS). (b) Accuracy on Unknown Samples (AUS). OPF has no AUS, because the standard OPF is a closed-set classifier that never classifies a test sample as unknown. (c) Normalized Accuracy (NA).

15 known classes for training, respectively. Regarding the NA, the OSOPF$_{OGP}$ and OSOPF$_{MV}$ have similar results with both, outperforming OSOPF in, at least, 5% for 3, 6, 9, 12, and 15 known classes. In the COIL dataset (Figure 7), OSOPF$_{OGP}$ and OSOPF$_{MV}$ significantly improve results compared to the other methods. Similar trends can be observed for the CALTECH-256 dataset (Figure 8) with OSOPF$_{OGP}$, OSOPF$_{CGP}$, and OSOPF$_{MV}$ outperforming OSOPF for different metrics, with statistical significance. Please refer to Appendix C for a complete analysis of the statistical significance for all considered methods.

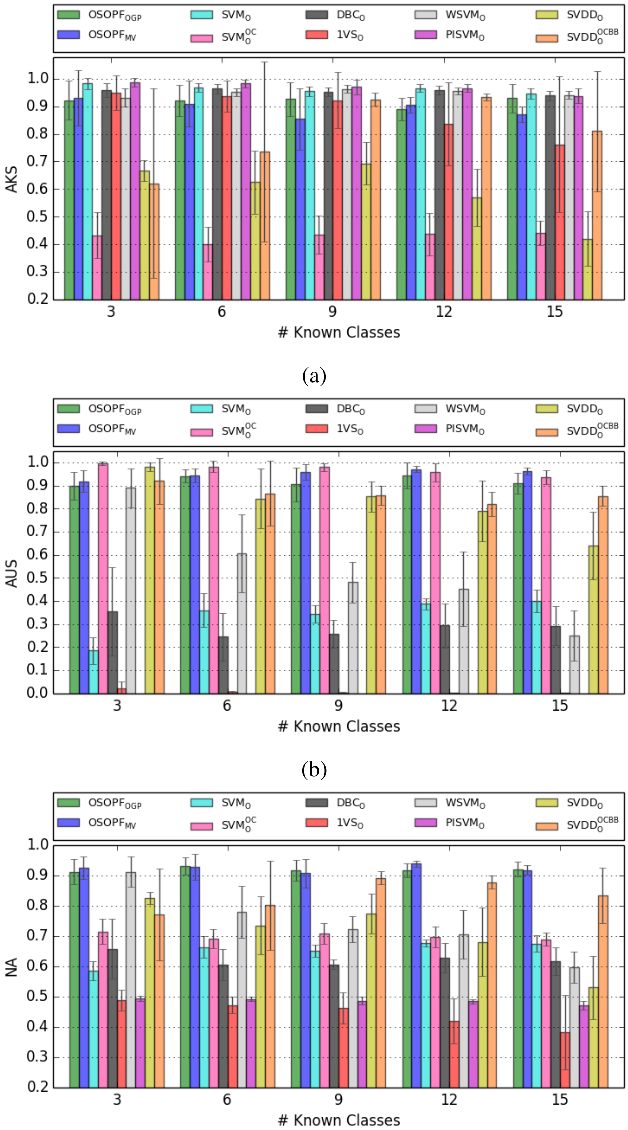
In general, the best classifier was the OSOPF$_{OGP}$, which shows that using OSOPF and genetic programming to aggregate different feature sets is indeed effective for the open-set recognition problems. Moreover, when combining OSOPF and GP, we need to simulate the open-set setup during the training of the genetic programming fusion method. We can also see that the OSOPF$_{MV}$ also shows to be very effective for this problem with similar performance to OSOPF$_{OGP}$ when considering the normalized accuracy (Figures 6c, 7c and 8c). The macro- and micro-averaging scores (OSFM$_M$ and OSFM$_\mu$) for the three datasets are presented in Appendix B.

These results highlight that the main problem in open-set setups is to find an optimum equilibrium between the specialization and generalization of classifiers. In our results, we observed that the OSOPF$_{CGP}$ method presents good results for the AKS. More specifically, between fusion methods in the CALTECH-256 dataset, the OSOPF$_{CGP}$ resulted in the best AKS. In contrast, this method leads to low results with respect to the rest of classifiers when considering the unknown (AUS metric). On the other hand, OSOPF$_{OGP}$ and OSOPF$_{MV}$ lead to good results by better rejecting unknown samples (AUS) while attaining reasonable results identifying known samples (AKS). These two methods find a reasonable equilibrium between the specialization and generalization for the ALOI and COIL datasets. For the CALTECH-256 dataset, the OSOPF$_{OGP}$ and OSOPF$_{MV}$ present reasonable, but lower, AKS.

2) COMPARISON WITH PRIOR ART

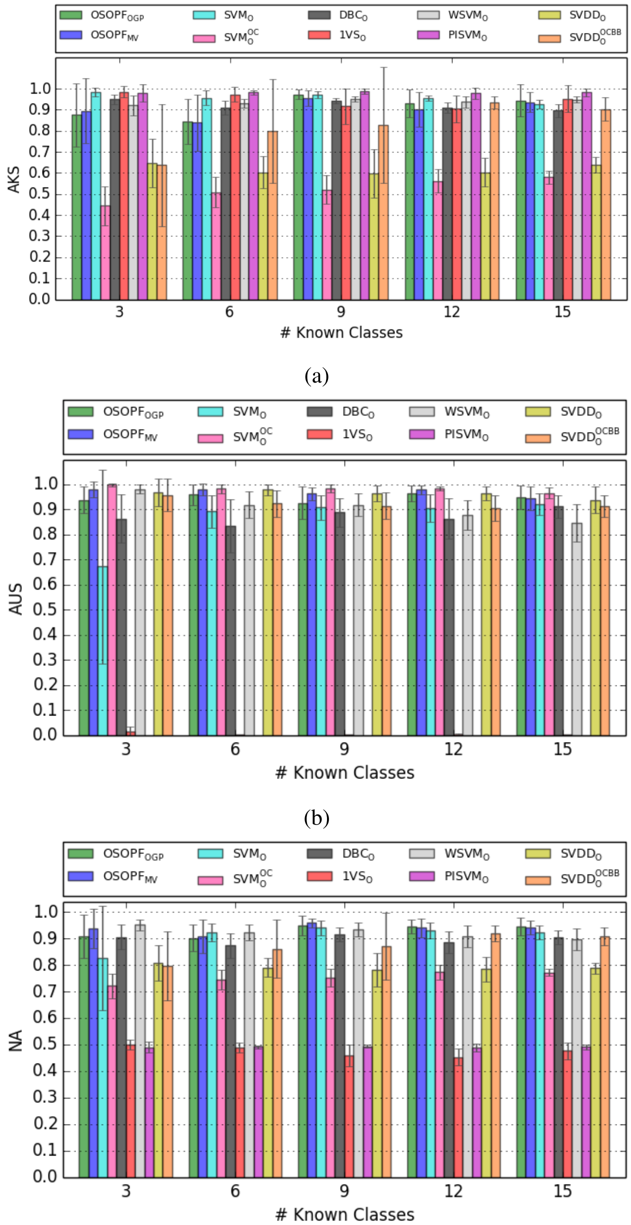
After having evaluated the different forms of the proposed methods, we turn our attention to posing them with respect to existing methods in prior art. For comparison purposes, we will use M$_O$ to denote a given method M that uses an open-set grid search. Our best methods (OSOPF$_{OGP}$ and OSOPF$_{MV}$) are compared to a plethora of approaches available in prior art: Support Vector Machines (SVM$_O$), Multiclass One-Class Support Vector Machines (SVM_O^{OC}), Decision Boundary Carving (DBC$_O$), 1-vs-Set Machine (1VS$_O$), Weibull-calibrated Support Vector Machines (WSVM$_O$), Support Vector Machines with Probability of Inclusion (PISVM$_O$), Support Vector Data Description (SVDD$_O$), Support Vector Data Description one-class binary-based ($SVDD_O^{OCBB}$). All SVM methods use one-vs-all approach in the multiclass level.

In Figures 9 (ALOI), 10 (COIL), and 11 (CALTECH-256), we present the accuracy on known samples (AKS), accuracy on unknown samples (AUS) and the normalized accuracy (NA) of the state-of-the-art methods along with the OSOPF$_{OGP}$ and OSOPF$_{MV}$. This experiment aims at showing how effective our best OSOPF methods are when compared to existing solutions for the open-set problem. OSOPF has a high AUS, because it is based on the distance proportion of the two best paths that ends in samples of different classes. When a test sample is far from the training ones, the ratio of the cost function of both paths approaches 1 and, consequently, it is more likely to be greater than the rejection threshold and it is properly marked as unknown. Moreover, our fusion approaches (OSOPF$_{OGP}$ and OSOPF$_{MV}$) found



(c)

FIGURE 9. Results of the state-of-the-art methods vs OSOPF methods for ALOI dataset. (a) Accuracy on Known Samples (AKS). (b) Accuracy on Unknown Samples (AUS). (c) Normalized Accuracy (NA).



(c)

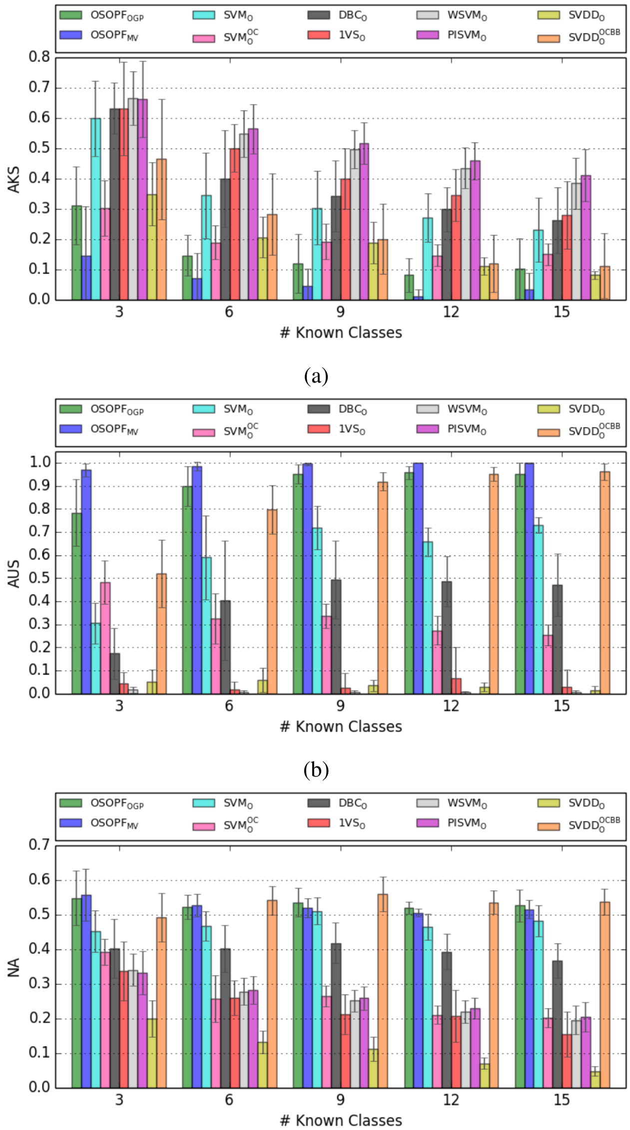
FIGURE 10. Results of the state-of-the-art methods vs OSOPF methods for COIL dataset. (a) Accuracy on Known Samples (AKS). (b) Accuracy on Unknown Samples (AUS). (c) Normalized Accuracy (NA).

some stability between generalization and specialization in the datasets, leading to good results considering the NA.

Naturally, as we are more restrictive at accepting examples of a given class, we see that SVM-based methods are better than OSOPF$_{OGP}$ and OSOPF$_{MV}$ when considering the AKS (Figures 9a, 11a and 10a). On the other hand, for the considered datasets, the proposed fusion methods show more stability in the AUS results (Figures 9b, 11b and 10b).

Based on the normalized accuracy, in the ALOI dataset (Figure 9), unlike the SVM-based methods, our methods have a stable behavior when the number of unknown classes increases. In all cases, OSOPF$_{OGP}$ and OSOPF$_{MV}$ classifiers are the winners, exchanging positions within the top two; there is only a tie in the second place for 3 known classes between the OSOPF$_{OGP}$ and WSVM$_O$. Considering the results of the best SVM approach, the OSOPF$_{OGP}$ reduces the misclassification in 13.10%, 2.50%, 3.90% and 8.73% for 6, 9, 12 and 15 known classes, respectively. For its part, the OSOPF$_{MV}$ reduces the misclassification in 1.42%, 12.63%, 1.51%, 6.10% and 8.46% for 3, 6, 9, 12 and 15 known classes, respectively.

For the COIL dataset (Figure 10), SVM-based methods have better results in the AKS and AUS. However, considering the NA, the WSVM$_O$ is the best for 3 and 6 known classes, and the OSOPF$_{OGP}$ and OSOPF$_{MV}$ are the winners for 9, 12 and 15 known classes.



(a)

(b)

(c)

FIGURE 11. **Results of the state-of-the-art methods vs OSOPF methods for CALTECH-256 dataset. (a) Accuracy on Known Samples (AKS). (b) Accuracy on Unknown Samples (AUS). (c) Normalized Accuracy (NA).**

For the CALTECH-256 dataset (Figure 11), the $\mathrm{OSOPF_{OGP}}$, $\mathrm{OSOPF_{MV}}$ and $\mathrm{SVDD_O^{OCBB}}$ classifiers outperform their counterparts in all cases when considering the normalized accuracy (NA). Our methods improve the classification with respect to the best SVM classifier in 5% for 3 known classes, in the rest of the experiments, the $\mathrm{SVDD_O^{OCBB}}$ reduces the misclassification with respect to the $\mathrm{OSOPF_{OGP}}$ and $\mathrm{OSOPF_{MV}}$ in 1.27%, 2.38%, 1.59% and 1.06% for the experiments with 6, 9, 12 and 15 known classes, respectively. However, it is worth mentioning that CALTECH-256 is also a difficult dataset, comprising regions with some degree of overlapping of two or more different training classes.

Based on those results, we can see how the fusion of complementary features (different visual properties) using either Genetic Programming or Majority Voting for open-set problems lead to better results than using just one feature set. In most experiments, the $\mathrm{OSOPF_{OGP}}$ and $\mathrm{OSOPF_{MV}}$ methods significantly improved the recognition rates over SVM-based methods. At last, we summarize the best results (normalized accuracy and standard deviation) in Tables 5, 6, 7, corresponding to ALOI, COIL, and CALTECH-256 datasets, respectively. In these tables, we considered only the $\mathrm{OSOPF_{OGP}}$, $\mathrm{OSOPF_{MV}}$, and two state-of-the-art classifiers whose obtained the best results in each dataset. Please refer to Appendix C-A for statistical tests for all these comparisons.

TABLE 5. **Aloi dataset - Summary results.**

# Known classes	$\mathrm{OSOPF_{OGP}}$	$\mathrm{OSOPF_{MV}}$	$\mathrm{WSVM_O}$	$\mathrm{SVDD_O^{OCBB}}$
3	0.910±0.04	**0.925**±0.04	0.911±0.05	0.770±0.15
6	**0.931**±0.03	0.926±0.04	0.779±0.09	0.800±0.15
9	**0.916**±0.04	0.906±0.05	0.722±0.04	0.891±0.02
12	0.916±0.02	**0.938**±0.01	0.704±0.08	0.877±0.02
15	**0.920**±0.03	0.917±0.02	0.595±0.05	0,832±0.09

TABLE 6. **Coil dataset - Summary results.**

# Known classes	$\mathrm{OSOPF_{OGP}}$	$\mathrm{OSOPF_{MV}}$	$\mathrm{WSVM_O}$	$\mathrm{SVM_O}$
3	0.907±0.08	0.938±0.07	**0.951**±0.02	0.828±0.20
6	0.901±0.05	0.910±0.06	**0.924**±0.03	**0.924**±0.03
9	0.949±0.04	**0.960**±0.02	0.935±0.03	0.940±0.03
12	**0.947**±0.03	0.940±0.04	0.908±0.04	0.930±0.03
15	**0.946**±0.03	0.941±0.03	0.897±0.04	0,924±0.03

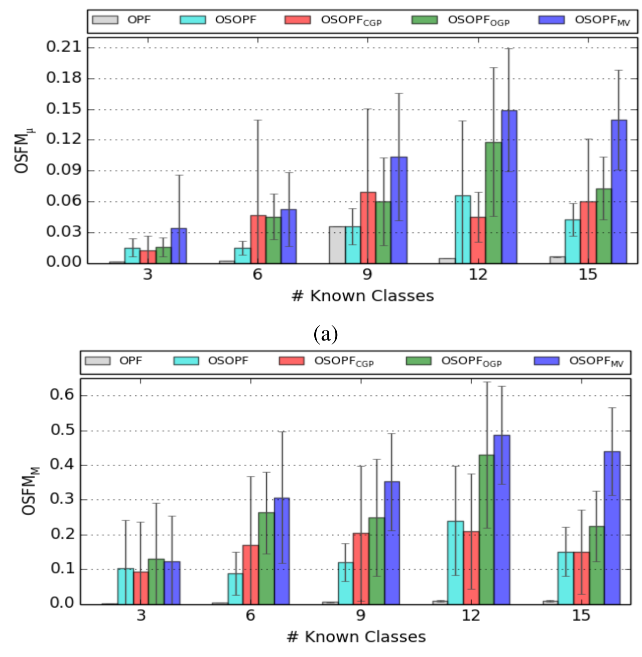
TABLE 7. **Caltech dataset - Summary results.**

# Known classes	$\mathrm{OSOPF_{OGP}}$	$\mathrm{OSOPF_{MV}}$	$\mathrm{SVDD_O^{OCBB}}$	$\mathrm{SVM_O}$
3	0.547±0.08	**0.556**±0.08	0.492±0.07	0.451±0.06
6	0.522±0.04	**0.527**±0.03	0.540±0.04	0.466±0.04
9	**0.535**±0.04	0.519±0.03	0.559±0.05	0.510±0.04
12	**0.519**±0.02	0.505±0.01	0.535±0.03	0.464±0.04
15	**0.525**±0.05	0.515±0.03	0.536±0.04	0,481±0.04

V. CONCLUSION

A myriad of problems in real-world applications must be modeled under the open-set scenario yet there are still few inherently multiclass methods for open-set recognition setups in prior art. As such, there exists an increasing demand for classifiers with the capability of properly rejecting samples that belong to classes for which no representative was seen during training phase.

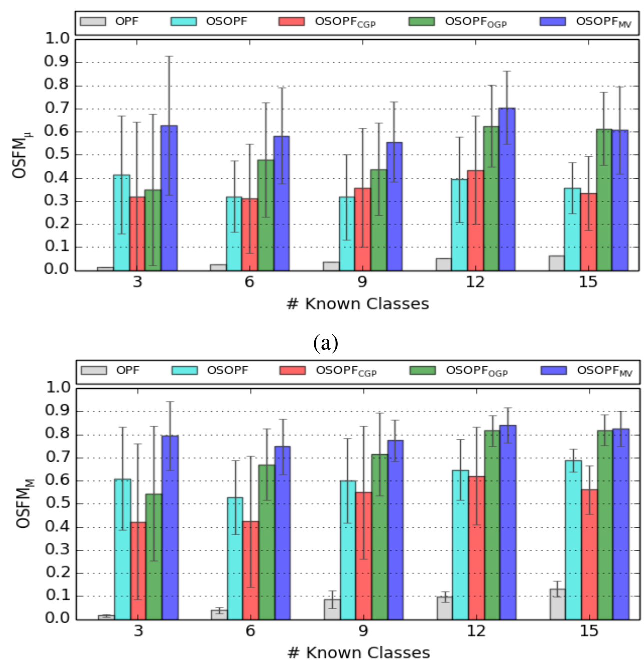
The main goal of our work herein was to leverage different characterization methods when solving a visual classification problem with different image descriptors that offer complementary views for a given problem and integrate them for open-set recognition problems. The main contribution of this work was the introduction of innate multiclass methods for open-set recognition problems that combine different features using Genetic Programming (GP) and a Majority Voting (MV) schemes. To the best of our knowledge, this is the first work to assess the performance of information fusion



(a)

(b)

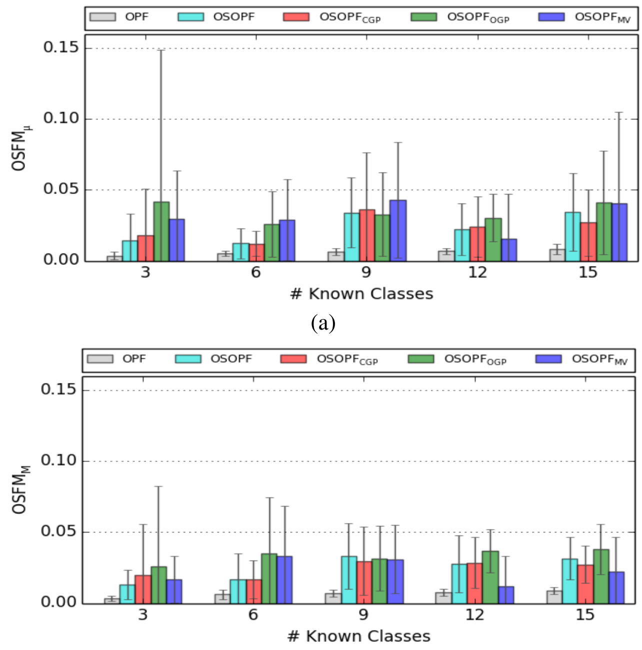
FIGURE 12. Results for the ALOI dataset. (a) Micro-averaging open-set f-measure $OSFM_{\mu}$. (b) Macro-averaging open-set f-measure ($OSFM_{M}$).



(a)

(b)

FIGURE 13. Results for the COIL dataset. (a) Micro-averaging open-set f-measure $OSFM_{\mu}$. (b) Macro-averaging open-set f-measure ($OSFM_{M}$).



(a)

(b)

FIGURE 14. Results for the CALTECH-256 dataset. (a) Micro-averaging open-set f-measure $OSFM_{\mu}$. (b) Macro-averaging open-set f-measure ($OSFM_{M}$).

Algorithm 5 Training Phase of OPF Classifier

Require: Training set D_1, prototypes $T \subset D_1$

for each $x \in D_1 \backslash T$ **do**
  $C(x) \leftarrow +\infty$
end for
for each $x \in T$ **do**
  $C(x) \leftarrow 0$; $P(x) \leftarrow nil$; $L(x) \leftarrow \lambda(x)$
  Put x in Q (Cost Priority Queue)
end for
while $Q \neq \emptyset$ **do**
  Remove x from Q, where $C(x) \leq C(y), \forall y \in Q$
  for each $y \in D_1$ and $x \neq y$ **do**
    $cst \leftarrow max\{C(x), d(x, y)\}$
    if $cst < C(y)$ **then**
      if $C(y) \neq +\infty$ **then**
        Remove y from Q
      end if
      $C(y) \leftarrow cst$; $P(y) \leftarrow x$; $L(y) \leftarrow L(x)$
      Insert y in Q
    end if
  end for
end while

approaches considering different types of features (color and texture) in the context of open-set recognition problems. The superiority of those classifiers was verified using a thorough experimental protocol, based on the evaluation of different methods using various datasets. We validated the proposed solutions with the Analysis of Variance (ANOVA) along with the Tukey-HSD post-test statistical tests. Among the evaluated methods in this work, the $OSOPF_{MV}$ and $OSOPF_{OGP}$ stand out with the most promising results. While the version with MV is an easy-to-implement technique,

TABLE 8. Statistical test for the ALOI dataset. *kc* denotes the number of known classes.

	kc	$\text{OSOPF}_{\text{OGP}}$	OSOPF_{MV}	WSVM_{O}	1VS_{O}	SVM_{O}	$\text{SVM}_{\text{O}}^{\text{OC}}$	DBC_{O}	SVDD_{O}	$\text{SVDD}_{\text{O}}^{\text{OCBB}}$	PISVM_{O}
$\text{OSOPF}_{\text{OGP}}$	3	–	–	–	←	←	←	←	–	←	←
	6	–	–	←	←	←	←	←	←	←	←
	9	–	–	←	←	←	←	←	←	–	←
	12	–	–	←	←	←	←	←	←	–	←
	15	–	–	←	←	←	←	←	←	–	←
OSOPF_{MV}	3	–	–	–	←	←	←	←	←	←	←
	6	–	–	←	←	←	←	←	←	←	←
	9	–	–	←	←	←	←	←	←	–	←
	12	–	–	←	←	←	←	←	←	–	←
	15	–	–	←	←	←	←	←	←	–	←
WSVM_{O}	3	–	–	–	←	←	←	←	–	←	←
	6	↑	↑	–	←	←	–	←	–	–	←
	9	↑	↑	–	←	←	–	←	–	↑	←
	12	↑	↑	–	←	–	–	←	–	↑	←
	15	↑	↑	–	←	–	–	–	–	↑	←
1VS_{O}	3	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	6	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	9	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	12	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	15	↑	↑	↑	–	↑	↑	↑	↑	↑	–
SVM_{O}	3	↑	↑	↑	←	–	↑	–	↑	↑	–
	6	↑	↑	↑	←	–	–	–	–	↑	←
	9	↑	↑	↑	←	–	–	–	↑	↑	←
	12	↑	↑	–	←	–	–	–	–	↑	←
	15	↑	↑	–	–	–	–	–	←	↑	←
$\text{SVM}_{\text{O}}^{\text{OC}}$	3	↑	↑	↑	←	←	–	–	↑	–	←
	6	↑	↑	–	←	–	–	←	–	↑	←
	9	↑	↑	–	←	–	–	←	–	↑	←
	12	↑	↑	–	←	–	–	←	–	↑	←
	15	↑	↑	–	←	–	–	–	←	↑	←
DBC_{O}	3	↑	↑	↑	←	–	–	–	↑	↑	←
	6	↑	↑	↑	←	–	↑	–	↑	↑	←
	9	↑	↑	↑	←	–	↑	–	↑	↑	←
	12	↑	↑	↑	←	–	↑	–	–	↑	←
	15	↑	↑	–	←	–	–	–	–	↑	←
SVDD_{O}	3	–	↑	–	←	←	←	←	–	–	←
	6	↑	↑	–	←	–	–	←	–	–	←
	9	↑	↑	–	←	←	–	←	–	↑	←
	12	↑	↑	–	←	–	–	–	–	↑	←
	15	↑	↑	–	←	↑	↑	–	–	↑	–
$\text{SVDD}_{\text{O}}^{\text{OCBB}}$	3	↑	↑	↑	←	←	–	←	–	–	←
	6	↑	↑	–	←	←	←	←	–	–	←
	9	–	–	←	←	←	←	←	←	–	←
	12	–	–	←	←	←	←	←	←	–	←
	15	–	–	←	←	←	←	←	←	–	←
PISVM_{O}	3	↑	↑	↑	–	–	↑	↑	↑	↑	–
	6	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	9	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	12	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	15	↑	↑	↑	–	↑	↑	↑	–	↑	–

the genetic programming is a technique that allows optimizing a composed distance function among objects, which might come handy in different setups.

In particular, the $\text{OSOPF}_{\text{OGP}}$ method (with training phase simulating an open-set setup using GP) obtained better results in the AUS, yielding a good specialization/generalization tradeoff. The OSOPF_{MV} also presented equally appealing results. While OSOPF takes care of learning decision boundaries and is more resilient to unknown classes and outliers, GP and MV complements that method by bringing together different visual object properties (e.g., color and texture) to discover appropriate decision boundaries through early and late fusion, respectively.

Based on those results, computational cost and simplicity of $\text{OSOPF}_{\text{OGP}}$ and OSOPF_{MV} algorithms, the latter stands out as a better choice to deal with open-set problems. However, $\text{OSOPF}_{\text{OGP}}$ is a generic classifier that can accept alterations in its structure (GP operators, add constants, etc.) being more powerful to accommodate different sets of features and problem formulations and should also be considered as an option. The choice between them should be based on specific problem constraints. A drawback of OSOPF methods is that, in some cases, samples in overlapping regions of two or more classes in the feature space (samples of different classes that have similar feature vectors) are classified as unknown and not as belonging to one of the overlapping classes. This is exactly when fusion methods can bring important contributions to the table: when considering different fusion methods (complementary feature sets), we still keep the ability of rejecting unknown samples but, at the same time, we gain more confidence at tagging a sample in a region of doubt to one of the known classes.

The difficulty of the open-set problem might be reduced when taking advantage of different views of the problem at

TABLE 9. Statistical test for the COIL dataset. *kc* denotes the number of known classes.

	kc	$\text{OSOPF}_{\text{OGP}}$	OSOPF_{MV}	WSVM_{O}	1VS_{O}	SVM_{O}	$\text{SVM}_{\text{O}}^{\text{OC}}$	DBC_{O}	SVDD_{O}	$\text{SVDD}_{\text{O}}^{\text{OCBB}}$	PISVM_{O}
$\text{OSOPF}_{\text{OGP}}$	3	–	–	–	←	–	←	–	–	–	←
	6	–	–	–	←	–	←	–	←	–	←
	9	–	–	–	←	–	←	–	←	–	←
	12	–	–	–	←	–	←	←	←	–	←
	15	–	–	←	←	–	←	–	←	–	←
OSOPF_{MV}	3	–	–	–	←	–	←	–	–	←	←
	6	–	–	–	←	–	←	–	←	–	←
	9	–	–	–	←	–	←	–	←	←	←
	12	–	–	–	←	–	←	←	←	–	←
	15	–	–	←	←	–	←	–	←	–	←
WSVM_{O}	3	–	–	–	←	–	←	–	–	←	←
	6	–	–	–	←	–	←	–	←	–	←
	9	–	–	–	←	–	←	–	←	–	←
	12	–	–	–	←	–	←	–	←	–	←
	15	↑	↑	–	←	–	←	–	←	–	←
1VS_{O}	3	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	6	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	9	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	12	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	15	↑	↑	↑	–	↑	↑	↑	↑	↑	–
SVM_{O}	3	–	–	–	←	–	–	–	–	–	←
	6	–	–	–	←	–	←	–	←	–	←
	9	–	–	–	←	–	←	–	←	–	←
	12	–	–	–	←	–	←	–	←	–	←
	15	–	–	–	←	–	←	–	←	–	←
$\text{SVM}_{\text{O}}^{\text{OC}}$	3	↑	↑	↑	←	–	–	↑	–	–	←
	6	↑	↑	↑	←	↑	–	↑	–	↑	←
	9	↑	↑	↑	←	↑	–	↑	–	↑	←
	12	↑	↑	↑	←	↑	–	↑	–	↑	←
	15	↑	↑	↑	←	↑	–	↑	–	↑	←
DBC_{O}	3	–	–	–	←	–	←	–	–	–	←
	6	–	–	–	←	–	←	–	–	–	←
	9	–	–	–	←	–	←	–	←	–	←
	12	↑	↑	–	←	–	←	–	←	–	←
	15	–	–	–	←	–	←	–	←	–	←
SVDD_{O}	3	–	–	–	←	–	–	–	–	–	←
	6	↑	↑	↑	←	↑	–	–	–	–	←
	9	↑	↑	↑	←	↑	–	↑	–	↑	←
	12	↑	↑	↑	←	↑	–	↑	–	↑	←
	15	↑	↑	↑	←	↑	–	↑	–	↑	←
$\text{SVDD}_{\text{O}}^{\text{OCBB}}$	3	–	↑	↑	←	–	–	–	–	–	←
	6	–	–	–	←	–	←	–	–	–	←
	9	–	↑	–	←	–	←	–	←	–	←
	12	–	–	–	←	–	←	–	←	–	←
	15	–	–	–	←	–	←	–	←	–	←
PISVM_{O}	3	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	6	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	9	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	12	↑	↑	↑	–	↑	↑	↑	↑	↑	–
	15	↑	↑	↑	–	↑	↑	↑	↑	↑	–

hand and information fusion is paramount in this configuration. In the case of image-based problems as we discussed in this paper, the combination of different visual properties gives a better separation of the data, helping the classifiers to reach a good generalization or specialization depending on the characterization of the methods that are used. When combining different features, we seek to use descriptors that contribute with different views for the visual content representation, with the objective of exploiting the variability within available data. This complementary view of the problem might be essencial when integrating fusion solutions with open-set classifiers.

Finally, future work might be devoted to integrating the confidence of each classifier instead of only the assigned class, while performing the open-set fusion. Moreover, some types of pre-selection of descriptors could further improve the complementarity of the considered methods and boost the discriminability and robustness of the open-set classifiers during deployment in a real-world problem. Moreover, given that the proposed methods have a good capability to reject unknown samples, another alternative could be exploiting different forms of fusion using them together with state-of-the-art closed-set classifiers. In this way, while OSOPF fusion methods have to deal with samples as *known* or *unknown*, the closed-set classifier would decide the label for the *known* ones afterwards.

APPENDIX A
OPTIMUM-PATH FOREST ALGORITHMS

In this appendix, we present the Optimum-path Forest classifier's algorithms for the training (Algorithm 5) and testing (Algorithm 6) stages considering a closed-set setup.

APPENDIX B
OPEN-SET F-MEASURE RESULTS

This appendix complements Section IV-B's results for the macro- and micro-averaging open-set f-measure

TABLE 10. Statistical test for the CALTECH-256 dataset. *kc* denotes the number of known classes.

	kc	$\mathrm{OSOPF_{OGP}}$	$\mathrm{OSOPF_{MV}}$	$\mathrm{WSVM_O}$	$\mathrm{1VS_O}$	$\mathrm{SVM_O}$	$\mathrm{SVM_O^{OC}}$	$\mathrm{DBC_O}$	$\mathrm{SVDD_O}$	$\mathrm{SVDD_O^{OCBB}}$	$\mathrm{PISVM_O}$
$\mathrm{OSOPF_{OGP}}$	3	–	–	←	←	←	←	←	←	–	←
	6	–	–	←	←	–	←	←	←	–	←
	9	–	–	←	←	←	←	←	←	–	←
	12	–	–	←	←	–	←	←	←	–	←
	15	–	–	←	←	←	←	←	←	–	←
$\mathrm{OSOPF_{MV}}$	3	–	–	←	←	←	←	←	←	–	←
	6	–	–	←	←	–	←	←	←	–	←
	9	–	–	←	←	–	←	←	←	–	←
	12	–	–	←	←	–	←	←	←	–	←
	15	–	–	←	←	←	←	←	←	–	←
$\mathrm{WSVM_O}$	3	↑	↑	–	–	↑	–	–	←	↑	–
	6	↑	↑	–	–	↑	–	↑	←	↑	–
	9	↑	↑	–	–	↑	–	↑	←	↑	–
	12	↑	↑	–	–	↑	–	↑	←	↑	–
	15	↑	↑	–	–	↑	–	↑	←	↑	–
$\mathrm{1VS_O}$	3	↑	↑	–	–	↑	–	–	←	↑	–
	6	↑	↑	–	–	↑	–	↑	←	↑	–
	9	↑	↑	–	–	↑	–	↑	←	↑	–
	12	↑	↑	–	–	↑	–	↑	←	↑	–
	15	↑	↑	–	–	↑	–	↑	←	↑	–
$\mathrm{SVM_O}$	3	↑	↑	←	←	–	–	–	←	–	←
	6	–	–	←	←	–	←	–	←	↑	←
	9	–	–	←	←	–	←	←	←	–	←
	12	↑	–	←	←	–	←	←	←	↑	←
	15	–	–	←	←	–	←	←	←	↑	←
$\mathrm{SVM_O^{OC}}$	3	↑	↑	–	–	–	–	–	←	↑	–
	6	↑	↑	–	–	↑	–	↑	←	↑	–
	9	↑	↑	–	–	↑	–	↑	←	↑	–
	12	↑	↑	–	–	↑	–	↑	←	↑	–
	15	↑	↑	–	–	↑	–	↑	←	↑	–
$\mathrm{DBC_O}$	3	↑	↑	–	–	–	–	–	←	↑	–
	6	↑	↑	←	←	–	←	–	←	↑	←
	9	↑	↑	←	←	↑	←	–	←	↑	←
	12	↑	↑	←	←	↑	←	–	←	↑	←
	15	↑	↑	←	←	↑	←	–	←	↑	←
$\mathrm{SVDD_O}$	3	↑	↑	↑	↑	↑	↑	↑	–	↑	↑
	6	↑	↑	↑	↑	↑	↑	↑	–	↑	↑
	9	↑	↑	↑	↑	↑	↑	↑	–	↑	↑
	12	↑	↑	↑	↑	↑	↑	↑	–	↑	↑
	15	↑	↑	↑	↑	↑	↑	↑	–	↑	↑
$\mathrm{SVDD_O^{OCBB}}$	3	–	–	←	←	–	←	←	←	–	←
	6	–	–	←	←	←	←	←	←	–	←
	9	–	–	←	←	–	←	←	←	–	←
	12	–	–	←	←	←	←	←	←	–	←
	15	–	–	←	←	←	←	←	←	–	←
$\mathrm{PISVM_O}$	3	↑	↑	–	–	↑	–	–	←	↑	–
	6	↑	↑	–	–	↑	–	↑	←	↑	–
	9	↑	↑	–	–	↑	–	↑	←	↑	–
	12	↑	↑	–	–	↑	–	↑	←	↑	–
	15	↑	↑	–	–	↑	–	↑	←	↑	–

(OSFM_M and OSFM_μ, respectively) for ALOI, COIL and CALTECH-256 datasets.

For ALOI and COIL datasets, in most cases, OSFM_M (Figures 12b and 13b) and OSFM_μ (Figures 12a and 13a) are increased when the *openness* decreases. In contrast, in the CALTECH-256 dataset, OSFM_M (Figure 14b) and OSFM_μ (Figure 14a) are not affected by the *openness* degree, given that the recognition rates are not high enough (this dataset is harder to classify in the open-set setup).

APPENDIX C STATISTICAL TESTS

In this appendix, we present results for the performed statistical tests. The statistical test analysis of variance (ANOVA) [68], [69] and the Tukey Honest Significant Differences (HSD) [70] post-test were used considering 95% of confidence to verify the statistical differences between the results of the classifiers. Appendix C-A presents the results of the statistical tests between the existing methods in the literature and OSOPF. Appendix C-B shows the statistical results for the OPF and OSOPF and its GP and MV variations ($\mathrm{OSOPF_{OGP}}$, $\mathrm{OSOPF_{CGP}}$, and $\mathrm{OSOPF_{MV}}$). Results are presented in tables, in which each cell has the result of the statistical test between each pair of classifiers according to the number of known classes (*kc*). Left '←' and up '↑' arrows denote the winner while an empty cell refers to "no statistical difference between the pair of classifiers" defined by that row and column.

A. STATISTICAL TESTS OF THE BASELINES METHODS

Results of the statistical tests for comparison between $\mathrm{OSOPF_{OGP}}$, $\mathrm{OSOPF_{MV}}$ and existing methods ($\mathrm{SVM_O}$, $\mathrm{SVM_O^{OC}}$, $\mathrm{DBC_O}$, $\mathrm{1VS_O}$, $\mathrm{WSVM_O}$, $\mathrm{PISVM_O}$, $\mathrm{SVDD_O}$, and $\mathrm{SVDD_O^{OCBB}}$) are presented in Table 8 (ALOI),

TABLE 11. Statistical test for the ALOI dataset. *kc* denotes the number of known classes.

	kc	OSOPF	$\text{OSOPF}_{\text{CGP}}$	$\text{OSOPF}_{\text{OGP}}$	OSOPF_{MV}
OSOPF	3	–	–	–	–
	6	–	–	↑	↑
	9	–	–	↑	–
	12	–	–	↑	↑
	15	–	–	↑	↑
$\text{OSOPF}_{\text{CGP}}$	3	–	–	↑	↑
	6	–	–	↑	↑
	9	–	–	↑	–
	12	–	–	–	↑
	15	–	–	↑	↑
$\text{OSOPF}_{\text{OGP}}$	3	–	←	–	–
	6	←	←	–	–
	9	←	←	–	–
	12	←	–	–	–
	15	←	←	–	–
OSOPF_{MV}	3	–	←	–	–
	6	←	←	–	–
	9	–	–	–	–
	12	←	←	–	–
	15	←	←	–	–

TABLE 12. Statistical test for COIL dataset. *kc* denotes the number of known classes.

	kc	OSOPF	$\text{OSOPF}_{\text{CGP}}$	$\text{OSOPF}_{\text{OGP}}$	OSOPF_{MV}
OSOPF	3	–	–	–	–
	6	–	–	–	–
	9	–	–	–	↑
	12	–	–	↑	↑
	15	–	–	↑	↑
$\text{OSOPF}_{\text{CGP}}$	3	–	–	–	–
	6	–	–	↑	↑
	9	–	–	↑	↑
	12	–	–	↑	↑
	15	–	–	↑	↑
$\text{OSOPF}_{\text{OGP}}$	3	–	–	–	–
	6	–	←	–	–
	9	–	←	–	–
	12	←	←	–	–
	15	←	←	–	–
OSOPF_{MV}	3	–	–	–	–
	6	–	←	–	–
	9	←	←	–	–
	12	←	←	–	–
	15	←	←	–	–

Table 9 (COIL) and Table 10 (CALTECH-256). The statistical tests show that $\text{OSOPF}_{\text{OGP}}$ and OSOPF_{MV} classifiers outperform most of the SVM-based classifiers. In some cases, however, no significant difference are observed.

B. STATISTICAL TESTS FOR OSOPF-BASED METHODS

Results of the statistical tests corresponding to the ALOI and COIL datasets are presented in Tables 11 and 12, respectively. In the CALTECH-256 (dataset difficult to classify),

Algorithm 6 Classification Phase of OPF Classifier

Require: D_2 validation set.
Require: D' samples in non-decreasing order of the cost function.
for each $x \in D_2$ **do**
  $i \leftarrow 1$
  $minCost \leftarrow max\{C(y_i), d(y_i, x)\}$, where $y_i \in D'$
  $L(x) \leftarrow L(y_i)$; $P(x) \leftarrow y_i$
  while $i < |D'|$ and $minCost > C(y_{i+1})$ **do**
    $aux \leftarrow max\{C(y_{i+1}), d(y_{i+1}, x)\}$
    if $aux < minCost$ **then**
      $minCost \leftarrow aux$
      $L(x) \leftarrow L(y_{i+1})$
      $P(x) \leftarrow y_{i+1}$
    end if
    $i \leftarrow i + 1$
  end while
end for

there were no statistical differences among the proposed OSOPF-based classifiers. Each cell has the result of the statistical test between each pair of classifiers according to the number of known classes (kc). In general, the statistical tests show that in the ALOI (Table 11) and COIL (Table 12), the $\text{OSOPF}_{\text{OGP}}$ and OSOPF_{MV} are the best ones but there is no statistical difference between both.

REFERENCES

[1] W. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. Boult, "Toward open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1757–1772, Jul. 2013.

[2] F. de O. Costa, M. Eckmann, W. J. Scheirer, and A. Rocha, "Open set source camera attribution," in *Proc. 25th SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Ouro Preto, Brazil, Aug. 2012, pp. 71–78.

[3] F. Li and H. Wechsler, "Open set face recognition using transduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1686–1697, Nov. 2005.

[4] R. P. W. Duin and E. Pekalska, "Open issues in pattern recognition," in *Computer Recognition Systems* (Advances in Soft Computing), vol. 30. Berlin, Germany: Springer, 2005, pp. 27–42.

[5] F. de O. Costa, E. Silva, M. Eckmann, W. J. Scheirer, and A. Rocha, "Open set source camera attribution and device linking," *Pattern Recognit. Lett.*, vol. 39, pp. 92–101, Apr. 2014.

[6] T. Godoi *et al.*, "A relevance feedback approach for the author name disambiguation problem," in *Proc. 13th ACM/IEEE-CS Joint Conf. Digit. Libraries*, 2013, pp. 209–218.

[7] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2317–2324, Nov. 2014.

[8] L. Kaliciak, H. Myrhaug, A. Goker, and D. Song, "On the duality of specific early and late fusion strategies," in *Proc. 17th Int. Conf. Inf. Fusion (FUSION)*, Salamanca, Spain, Jul. 2014, pp. 1–8.

[9] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[10] L. Lam and C. Y. Suen, "Application of majority voting to pattern recognition: An analysis of its behavior and performance," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 27, no. 5, pp. 553–568, Sep. 1997.

[11] R. Calumby, R. da Silva Torres, and M. A. Gonçalves, "Multimodal retrieval with relevance feedback based on genetic programming," *Multimedia Tools Appl.*, vol. 69, no. 3, pp. 991–1019, 2014.

[12] R. da Silva Torres *et al.*, "A genetic programming framework for content-based image retrieval," *Pattern Recognit.*, vol. 42, no. 2, pp. 283–292, Feb. 2009.

[13] H. M. de Almeida, M. A. Gonçalves, M. Cristo, and P. Calado, "A combined component approach for finding collection-adapted ranking functions based on genetic programming," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2007, pp. 399–406.

[14] F. S. P. Andrade, J. Almeida, H. Pedrini, and R. da Silva Torres, "Fusion of local and global descriptors for content-based image and video retrieval," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (Lecture Notes in Computer Science), vol. 7441. Berlin, Germany: Springer, 2012, pp. 845–853.

[15] A. T. da Silva, J. A. D. Santos, A. X. Falcão, R. da Silva Torres, and L. P. Magalhães, "Incorporating multiple distance spaces in optimum-path forest classification to improve feedback-based learning," *Comput. Vis. Image Understand.*, vol. 116, no. 4, pp. 510–523, 2012.

[16] D. Ciuonzo and P. S. Rossi, "Decision fusion with unknown sensor detection probability," *IEEE Signal Process. Lett.*, vol. 21, no. 2, pp. 208–212, Feb. 2014.

[17] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *Int. J. Imag. Syst. Technol.*, vol. 19, no. 2, pp. 120–131, 2009.

[18] J. A. dos Santos, A. T. da Silva, R. da Silva Torres, A. X. Falcão, L. P. Magalhães, and R. A. C. Lamparelli, "Interactive classification of remote sensing images by using optimum-path forest and genetic programming," in *Computer Analysis of Images and Patterns* (Lecture Notes in Computer Science), vol. 6855. Berlin, Germany: Springer, 2011, pp. 300–307.

[19] M. P. Ponti, Jr. and J. P. Papa, "Improving accuracy and speed of optimum-path forest classifier using combination of disjoint training subsets," in *Proc. Int. Workshop Multiple Classifier Syst.*, 2011, pp. 237–248.

[20] M. P. Ponti, Jr. and I. Rossi, "Ensembles of optimum-path forest classifiers using input data manipulation and undersampling," in *Proc. Int. Workshop Multiple Classifier Syst.*, 2013, pp. 236–246.

[21] J. A. dos Santos, C. D. Ferreira, and R. da Silva Torres, "A genetic programming approach for relevance feedback in region-based image retrieval systems," in *Proc. 21th Brazilian Symp. Comput. Graph. Image Process.*, Campo Grande, Brazil, Oct. 2008, pp. 155–162.

[22] F. F. Faria *et al.*, "Learning to rank for content-based image retrieval," in *Proc. Int. Conf. Multimedia Inf. Retr.*, 2010, pp. 285–294.

[23] C. D. Ferreira, J. A. Santos, R. da S. Torres, M. A. Gonçalves, R. C. Rezende, and W. Fan, "Relevance feedback based on genetic programming for image retrieval," *Pattern Recognit. Lett.*, vol. 32, no. 1, pp. 27–37, Jan. 2011.

[24] J. Almeida, J. A. dos Santos, W. O. Miranda, B. Alberton, L. P. C. Morellato, and R. da Silva Torres, "Deriving vegetation indices for phenology analysis using genetic programming," *Ecol. Informat.*, vol. 26, pp. 61–69, Mar. 2015.

[25] J. A. dos Santos, F. F. Faria, R. Calumby, R. da Silva Torres, and R. A. C. Lamparelli, "A genetic programming approach for coffee crop recognition," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Honolulu, HI, USA, Jul. 2010, pp. 3418–3421.

[26] J. A. dos Santos, C. D. Ferreira, R. da Silva Torres, M. Gonçalves, and R. A. C. Lamparelli, "A relevance feedback method based on genetic programming for classification of remote sensing images," *Inf. Sci.*, vol. 181, no. 13, pp. 2671–2684, 2011.

[27] W. P. Amorim and M. H. de Carvalho, "Supervised learning using local analysis in an optimal-path forest," in *Proc. 25th SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Ouro Preto, Brazil, Aug. 2012, pp. 330–335.

[28] L. C. S. Afonso, J. P. Papa, A. N. Marana, A. Poursaberi, S. N. Yanushkevich, and M. Gavrilova, "Optimum-path forest classifier for large scale biometric applications," in *Proc. 3rd Int. Conf. Emerg. Secur. Technol.*, Lisbon, Portugal, Sep. 2012, pp. 58–61.

[29] P. B. Ribeiro, K. A. P. Da Costa, J. P. Papa, and R. A. F. Romero, "Optimum-path forest applied for breast masses classification," in *Proc. IEEE 27th Int. Symp. Comput.-Based Med. Syst.*, New York, NY, USA, May 2014, pp. 52–55.

[30] A. T. da Silva, A. Xavier, and L. P. Magalhães, "A new CBIR approach based on relevance feedback and optimum-path forest classification," *J. Winter School Comput. Graph.*, vol. 18, nos. 1–3, pp. 73–80, 2010.

[31] C. T. N. Suzuki, J. Gomes, A. X. Falcão, J. P. Papa, and S. Hoshino-Shimizu, "Automatic segmentation and classification of human intestinal parasites from microscopy images," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 3, pp. 803–812, Mar. 2013.

[32] L. C. S. Afonso, J. P. Papa, A. N. Marana, A. Poursaberi, and S. N. Yanushkevich, "A fast large scale iris database classification with optimum-path forest technique: A case study," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Brisbane, QLD, Australia, Jun. 2012, pp. 1–5.

[33] D. Rodrigues *et al.*, "Bcs: A binary cuckoo search algorithm for feature selection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Beijing, China, May 2013, pp. 465–468.

[34] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. New York, NY, USA: McGraw-Hill, 2001.

[35] A. X. Falcao, J. Stolfi, and R. D. A. Lotufo, "The image foresting transform: Theory, algorithms, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 19–29, Jan. 2004.

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

[37] D. A. Pritsos and E. Stamatatos, "Open-set classification for automated genre identification," in *Advances in Information Retrieval* (Lecture Notes in Computer Science), vol. 7814, P. Serdyukov *et al.*, Eds. Berlin, Germany: Springer, Mar. 2013, pp. 207–217.

[38] A. Rattani, W. J. Scheirer, and A. Ross, "Open set fingerprint spoof detection across novel fabrication materials," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2447–2460, Nov. 2015.

[39] A. Rocha and S. K. Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based approaches," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 289–302, Feb. 2014.

[40] B. Heflin, W. J. Scheirer, and T. E. Boult, "Detecting and classifying scars, marks, and tattoos found in the wild," in *Proc. IEEE 5th Int. Conf. Biometrics, Theory, Appl. Syst.*. Arlington, VA, USA, Sep. 2012, pp. 31–38.

[41] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Syst.*, vol. 8, no. 6, pp. 536–544, Apr. 2003.

[42] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[43] S. Coles, *An Introduction to Statistical Modeling of Extreme Values* (Springer Series in Statistics), 1st ed. London, U.K.: Springer-Verlag, 2001.

[44] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 393–409.

[45] A. Bendale and T. E. Boult, "Towards open world recognition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 1893–1902.

[46] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 1563–1572.

[47] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis," in *Proc. 13th Annu. Assoc. Comput. Mach. Int. Conf. Multimedia*, 2005, pp. 399–402.

[48] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Syst.*, 2000, pp. 1–15.

[49] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: A tutorial," *Statist. Sci.*, vol. 14, no. 4, pp. 382–401, 1999.

[50] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[51] Y. Freund, "Boosting a weak learning algorithm by majority," *Inf. Comput.*, vol. 121, no. 2, pp. 256–285, Sep. 1995.

[52] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *Proc. Int. Conf. Comput. Vis.*, Washington, DC, USA, Nov. 2011, pp. 89–96.

[53] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 416–431, Mar. 2006.

[54] P. R. M. Júnior *et al.*, "Nearest neighbors distance ratio open-set classifier," *Mach. Learn.*, vol. 106, no. 3, pp. 359–386, 2017.

[55] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[56] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007.

[57] J.-M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The Amsterdam library of object images," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 103–112, 2005.

[58] A. Nene, S. K. Nayar, and M. Hiroshi, "Columbia object image library (coil-100)," Comput. Vis. Lab., Dept. Comput. Sci., Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-006-96, 1996. [Online]. Available: http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php

[59] R. O. Stehling, M. A. Nascimento, and A. X. Falcão, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *Proc. 11th Int. Conf. Inf. Knowl. Manage.*, 2002, pp. 102–109.

[60] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Juan, Puerto Rico, USA, Jun. 1997, pp. 762–768.

[61] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proc. 4th Assoc. Comput. Mach. Int. Conf. Multimedia*, 1996, pp. 65–73.

[62] C.-B. Huang and Q. Liu, "An orientation independent texture descriptor for image retrieval," in *Proc. Int. Conf. Commun., Circuits Syst. (ICCCAS)*, Jul. 2007, pp. 772–776.

[63] B. Tao and B. W. Dickinson, "Texture recognition and image retrieval using gradient indexing," *J. Vis. Commun. Image Represent.*, vol. 11, no. 3, pp. 327–342, 2000.

[64] D.-Y. Chen, T.-R. Chuang, and S.-C. Tsai, "JGAP: A Java-based graph algorithms platform," *Softw., Pract. Exper.*, vol. 31, no. 7, pp. 615–635, 2001.

[65] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building*. New York, NY, USA: Wiley, 1978.

[66] P. C. Saraiva, J. M. B. Cavalcanti, M. A. Gonçalves, K. C. dos Santos, E. S. de Moura, and R. da S. Torres, "Evaluation of parameters for combining multiple textual sources of evidence for Web image retrieval using genetic programming," *J. Brazilian Comput. Soc.*, vol. 19, no. 2, pp. 147–160, Jun. 2013.

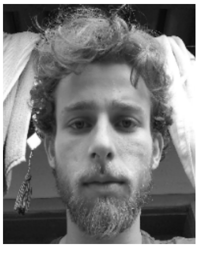
[67] R. Feldt and P. Nordin, "Using factorial experiments to evaluate the effect of genetic programming parameters," in *Proc. Eur. Conf. Genet. Program.*, 2000, pp. 271–282.

[68] B. S. Yandell, *Practical Data Analysis for Designed Experiments* (Chapman & Hall/CRC Texts in Statistical Science). New York, NY, USA: Taylor & Francis, 1997.

[69] R. G. Miller, Jr., "Normal univariate techniques," in *Simultaneous Statistical Inference* (Springer Series in Statistics). New York, NY, USA: Springer, 1981, pp. 37–108.

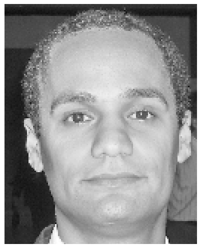
[70] A. Hervé and W. Lynne, "Tukey's honestly significant difference (HSD) test," in *Encyclopedia of Research Design*. Thousand Oaks, CA, USA: Sage, 2010, pp. 1–5.

MANUEL ALBERTO CÓRDOVA NEIRA received the B.Sc. degree in systems engineering from the National University of Loja (UNL), Ecuador, in 2010, and the M.Sc. degree in computer science from the University of Campinas, Brazil, in 2015, where he is currently pursuing the Ph.D. degree. He was a Professor with the Department of Systems Engineering, UNL.



PEDRO RIBEIRO MENDES JÚNIOR received the bachelor's degree in computer science from the Federal University of Ouro Preto, Brazil, in 2012, and the master's degree from the University of Campinas, Brazil, where he is currently pursuing the Ph.D. degree.

ANDERSON ROCHA received the B.Sc. degree in computer science from the Federal University of Lavras, Brazil, in 2003, and the M.Sc. and Ph.D. degrees in computer science from the University of Campinas (UNICAMP), Brazil, in 2006 and 2009, respectively. He is currently an Associate Professor with the Institute of Computing, UNICAMP. His main interests include digital forensics, reasoning for complex data and machine intelligence. In 2016, he was a recipient of the Tan Chin Tuan Fellowship, a recognition promoted by the Tan Chin Tuan Foundation in Singapore. He has actively worked as a program committee member in several important events and is an Associate Editor of international journals, such as the IEEE Transactions on Information Forensics and Security, *Elsevier Journal of Visual Communication and Image Representation*, the EURASIP/Springer *Journal on Image and Video Processing* and the IEEE *Security & Privacy Magazine*. He is an elected Affiliate Member of the Brazilian Academy of Sciences (ABC) and the Brazilian Academy of Forensic Sciences (ABC). He is also a two-term elected Member of the IEEE Information Forensics and Security Technical Committee and its current Vice-Chair. He is a Microsoft Research and a Google Research Faculty Fellow, important academic recognitions given to researchers by Microsoft Research and Google, respectively. Since April 2017, he has been the Associate Director of the Institute of Computing, UNICAMP.



RICARDO DA SILVA TORRES received the B.Sc. degree in computer engineering and Ph.D. degree in computer science from the University of Campinas (UNICAMP), Brazil, in 2000 and 2004, respectively. He was a Director of the Institute of Computing, UNICAMP, from 2013 to 2017. He is currently a Full Professor of computer science with UNICAMP. He is a Brazilian CNPq Research Scholar (PQ 1C). He is a Co-Founder and a Member of the RECOD Lab, where he has been developing multidisciplinary e-Science research projects, involving multimedia analysis, multimedia image retrieval, machine learning, databases, digital libraries, and geographic information systems. He has authored/co-authored over 100 articles in refereed journals and conferences and serves as a PC member for several international and national conferences. His research activities have been also associated with the deposit and licensing of several patents.