A Gene Expression Programming Approach for Evolving Multi-Class Image Classifiers

Nelson Marcelo Romero Aquino[1], Manassés Ribeiro[2], Matheus Gutoski[3],
César Vargas Benítez[4] and Heitor Silvério Lopes[5]
Laboratory of Bioinformatics and Computational Intelligence
Federal University of Technology – Paraná
Av Sete de Setembro 3165, Curitiba (PR), Brazil.
[1]nmarceloromero@gmail.com, [2]manasses@ifc-videira.edu.br, [3]matheusgutoski@gmail.com,
[4]cesarbenitez@utfpr.edu.br, [5]hslopes@utfpr.edu.br

***Abstract*— This paper presents a methodology to perform multi-class image classification using Gene Expression Programming(GEP) in both balanced and unbalanced datasets. Descriptors are extracted from images and then their dimensionality are reduced by applying Principal Component Analysis. The aspects extracted from images are texture, color and shape that are, later, concatenated in a feature vector. Finally, GEP is used to evolve trees capable of performing as classifiers using the features as terminals. The quality of the solution evolved is evaluated by the introduced Cross-Entropy-Loss-based fitness function and compared with standard fitness function (both accuracy and product of sensibility and specificity). A novel GEP function linker Softmax-based is introduced. GEP performance is compared with the obtained by classifiers with tree structure, as C4.5 and Random Forest algorithms. Results show that GEP is capable of evolving classifiers able to achieve satisfactory results for image multi-class classification.**

I. Introduction

Image classification is the process of separating images according to their content. This task may consist of a series of steps as image pre-processing, image segmentation, feature extraction, dimensionality reduction and image classification [1]. During the years, several methods have been proposed to perform image classification, pursuing to achieve better classification performances.

Evolutionary Computation (EC) approaches have been used mainly for optimization problems. However, some classification problems can also be dealt with EC, by evolving Decision Trees, nonlinear classifiers, symbolic regression models, rule induction systems, among others. In this work, we use the Gene Expression Programming (GEP) algorithm [2], which creates computer programs that can represent Decision Trees. GEP has elements from both Genetic Algorithms (GA) and Genetic Programming (GP), since it uses a population of individuals which encoded chromosomes are further transformed as expression trees.

GEP has been applied to solve classification problems in several domains. For instance, [3] proposed a GEP-based system for data mining. They used a multiple master-slave for parallel processing of large-scale data and classification was accomplished as a multi-objective problem. Overall results indicated that the proposed system achieves better predictive accuracy with shorter rules, when compared with the classical C4.5 method. Also, [4] proposed a multi-class classification rules were evolved with GEP using the one-against-all learning method, where the fitness function considers the rule consistency gain and completeness. Another the approach, presented by [5], also aimed at solving multi-class classification problems by evolving GEP Expression-Trees (ET) using the eigenvalue centroid of each class and the eigenvalue-power function, where the fitness function is composed by the similarity and dissimilarity of the classes. Furthermore, the Unconstrained Linear Encoded GEP (UGEP) was proposed by [6]. Basically, it is based on dynamically adjusting the amount and length of the genes that compose the individuals during the evolution process with a fitness function based on the accuracy.

Recently, GEP has gained attention from multi-disciplinary researchers. For instance, a novel distributed bio-inspired approach that uses GEP to evolve rules for two-dimensional Cellular Automata (2D-CA) was proposed by [7] which, in turn, was used to deal with a computationally expensive problem, protein structure prediction, that is considered to be one of the most important open challenges in Bioinformatics. Also, [8] applied GEP to predict lung cancer, using as fitness function some metrics commonly used in Machine Learning, such as sensitivity, specificity, positive and negative predictive values. Regarding image related problems and EC algorithms, an approach based on Genetic Programming (GP) applied to multi-class classification was recently proposed by [9], where the quality of the candidate solutions was evaluated using the product of sensibility and specificity. For the final classification, the approach is based on the assumption that the outputs of the GP trees belong to a normal distribution, and then a Probability Density Function (PDF) is used for classifying the output according to the probability of it belongs to a given class. Furthermore, [10] presented a hybrid approach based on Artificial Neural Networks and GEP to classify whether a pixel corresponds to an image region containing water or not, considering the spectral features of the pixels. In addition, [11] proposed a GEP to evolve a classifier of crop data obtained from satellite images. Notwithstanding, it is important to recall that GEP is still under-explored for multi-class image classification, despite the simplicity and comprehensibility of the decision trees generated by it.

The usual approach for tackling multi-class classification problems is the use of the one-against-all strategy as well as measuring quality based on well-known metrics commonly used in Machine Learning. In this work we propose a novel GEP approach, introducing a fitness function based on the Cross-Entropy-Loss (CEL), as well as a GEP linker based on the Softmax function. These functions are widely used in Deep Neural Networks and were chosen aiming at improving the classification performance when faced with highly unbalanced multi-class datasets.

In this work, several subjects were addressed, and the main contributions are:

- To introduce the use of GEP in the context of multi-class image classification;
- To propose the use of Softmax-based function as the GEP linker;
- To present a fitness function based on the Cross-Entropy-Loss to evolve unbiased classification rules for highly unbalanced datasets.

This paper is divided into four sections. Section II presents a thorough description of image classification, feature extractors, dimensionality reduction and GEP. Section III describes in details the proposed methodology. Section IV presents the experiments and results achieved in this work. Finally, section V presents conclusions and future research directions.

II. Background

A. Image Descriptors

Images can be described through numerical vectors called descriptors (or feature vectors), obtained by means of a feature extractor. Descriptors are commonly used as input to a classifier and, there are four categories of descriptors: texture, color, shape, and motion [12]. The last one is used for moving images (videos). Therefore, in this work, we focus on the texture, color, and shape descriptors presented below.

A powerful texture descriptor is the Local Binary Patterns (LBP), proposed by [13]. Due to its simplicity, LBP is fastly computed, and is very robust to illumination changes.

Possibly, the most popular descriptor for representation of the color is the color histogram (HST). It describes the distribution of the colors of an image by counting the number of pixels within a range of the space color. It is the great advantage of being invariant to translation, rotation and viewing angle [14].

Shape descriptors can be classified in two categories, local and global. Local features describe portions of the object or are derived of a partial analysis of an image, accomplished by analyzing its segments. Some local shape features are the corners, points of curvature and turning angle. Global features describe properties of the entire shape of the objects in the image. Hu Moment Ivariants (MOM)[15] are very efficient used to characterize the outline of an object in an image, and it is used in this work.

B. Principal Component Analysis

Visual descriptors may have a high dimensionality. For instance, in the paper that introduces Over-Complete Local Binary Patterns (OCLBP) [16], which is a variant of LBP, the authors propose a configuration for the descriptor that gives as result a 40887-dimensional feature vector for an image of size 150×80.

Since the predictive power of an algorithm reduces as the dimensionality of a fixed number of training samples increases [17], high-dimensional data is likely to negatively affect the performance of a classifier. To avoid this issue, an alternative is to apply some transformation on the original data in order to reduce its dimensionality whilst aiming at maintaining most of the information. Principal Component Analysis (PCA) [18] accomplishes this task by performing an orthogonal transformation on a set of correlated variables of a dataset, so as to convert them into linearly uncorrelated values, known as Principal Components (PCs). The number of PCs is usually smaller than the original number of variables.

C. Gene Expression Programming

GEP is a variant of Genetic Programming, and was created by [19]. In GEP, individuals are fixed-length strings that are later expressed as non-linear entities called Expression Trees (ETs), which, in turn, may have different sizes and shapes. An ET is, basically, the program evolved by the algorithm.

GEP chromosomes are relatively small entities that can be modified by genetic operators (mutation, transposition and recombination, for instance), thus generating diversity so that the evolutionary process can continue for many generations, aiming at finding well-fitted solutions. Each position of a chromosome encodes a function (e.g., addition, subtraction, root square) or a terminal (variable of the problem). GEP genes, understood also in terms of Open Reading Frames (ORFs), are structurally organized as a head an a tail. The head can have elements from the functions set and the terminals set, whilst the tail can have terminals. Thus, based on the ORF principle, a gene can have coding and non-coding regions, leading to the creation of valid, but differently sized, programs.

In GEP there is a mapping between the genotype and the phenotype, similarly as Genetic Algorithms. This mapping is done by transcribing the chromosome into a variable-size Expression Tree (ET), following the Karva language [2], where each gene is trascribed into a sub-tree. Then, all sub-trees are joined together by a linking function (commonly mathematical or boolean functions), composing the ET that represents a candidate solution to a given problem. This is analogue to the aggregation of different protein subunits into a multi-subunit protein.

Since GEP is an EC approach, the quality of the candidate solutions is measured by a fitness function. For instance, the fitness functions commonly used with GEP are: number of hits, accuracy, the product of sensitivity and specificity, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) [3]

III. METHODOLOGY

An overview of our approach is shown in Figure 1. The first two elements are based on [9]: the feature extraction method (see Section II-A) and the use of PCA (see Section II-B) for dimensionality reduction. First, features describing the texture (LBP), color (HST), and shape (MOM) of the images are extracted from the dataset. The extracted features are comprised in a high-dimensional feature vector which, in turn, is reduced by applying PCA. This is useful since the use of a lower data dimension reduces the search space for the GEP algorithm. Following, GEP is used to evolve simple programs capable of classifying images. Finally, their performances are evaluated and compared with other classifiers, such as the C4.5 [20] and Random Forest, that are usual baselines for data classification. When compared with previous works, our approach presents two novel contributions: (1) a fitness function based on the Cross-Entropy-Loss to evaluate the quality of the solution, and (2) a novel GEP linker based on the Softmax function that gives a probability for each class. Moreover, we propose an evaluation procedure that uses not only the accuracy (as in [9]), but also Sensibility (S_e) and Specificity (S_p).

The next Section presents in details the feature extraction and dimensionality reduction processes. Also, the implementation of the GEP is presented in Section III-B. the metrics that were used for evaluating the performance of the classifiers.

A. Feature Extraction and Dimensionality Reduction

To create a feature vector of each image, we used LBP, HST and MOM for describing its texture, color and shape, respectively. All these features were normalized to a common scale, in the range $[0..1]$ and then concatenated to form a single feature feature vector with 2592 elements: 256 from LBP, 768 from HST and 1568 from MOM.

Both LBP and HST features are extracted by using the entire images as input for the feature extractors. As for MOM, since it describes the moments of different elements (regions of interest) within an image, a segmentation technique was applied to obtain those elements.

The MOM features were then extracted from each segmented element. Finally, those feature vectors are concatenated in order to create the final MOM descriptor of the entire image. The process of extraction of the MOM features for an image is described as follows. First, the image is converted into grayscale, which is an image composed of shades of gray in the range $[0..255]$. A Gaussian blur filter is then applied and Canny detection is performed [21]. Next, the edge of each object is obtained. Finally, the Hu moment invariants of the edges are calculated.

Since the feature vectors obtained through the feature extraction process are high-dimensional, it is necessary, as mentioned before, to reduce their dimensionality. PCA is applied to transform the original variables set into a new ortogonal coordinates system in which they are represented by Principal Components (PCs).

The more PCs are considered, the more the cumulative variance, given by the PCA eigenvalues, is explained. However, for exploratory purposes, a threshold λ should be set to determine the number of elements that will be used as terminals by the GEP algorithm. The number of PCs are chosen considering the accumulated. The threshold λ was set around of 50%, meaning that first PCs chosen will represent at least 50% of the cumulative variance.

B. Implementation of the GEP Algorithm

1) Encoding and Initial Population: In this work, GEP chromosomes are composed by 5 genes, each coding a sub-ET implicitly specialized on giving as result a numeric value related to one of the classes of the problem. That is, the sub-ET coded by the first gene gives a score related to the first class (class 0). The second sub-ET gives a score related to the second class (class 1), and so on.

Sub-ETs must be connected by means of a custom linker function. It receives the output of each sub-ET and calculates the probabilities of an instance to belong to each class. This is accomplished by the Softmax function, shown in Equation 1. These probabilities are real values in the range $[0..1]$ that add up to 1. The label of the class with the highest probability is then returned as the final classification of the tree.

$$P(class = j|x) = \frac{\boldsymbol{e}^{x_j}}{\sum\limits_{i=1} \boldsymbol{e}^{x_i}}, \tag{1}$$

where x_j is the output of the *j*-th sub-ET.

Also, the initial population is randomly initialized.

2) Fitness Function: This work proposes a novel approach, based on the Cross-Entropy Loss (CEL), which is a cost function that allowed us to exploit the use of Softmax as the linker function.

Since GEP is designed for maximizing the fitness function and CEL has to be minimized in order to improve the classification measure of the model, we used the inverse of the CEL, as shown in Equation 2.

$$CEL = -\frac{1}{\sum_i y_i' \log(y_i)}, \tag{2}$$

where y_i is the probability that the sample belongs to the class i and y_i' is the ground truth label.

Besides CEL, two other fitness functions were tested to evolve GEP trees: Accuracy and the product of Sensibility (S_e) and Specificity (S_p). Accuracy measures how well a classifier performs without taking into account the distribution of the instances among the classes. It is calculated by dividing the number of correct classifications C by the number of samples N ($Acc = C/N$). Details regarding the product of S_e and S_p are presented Equations 4 and 5, respectively.

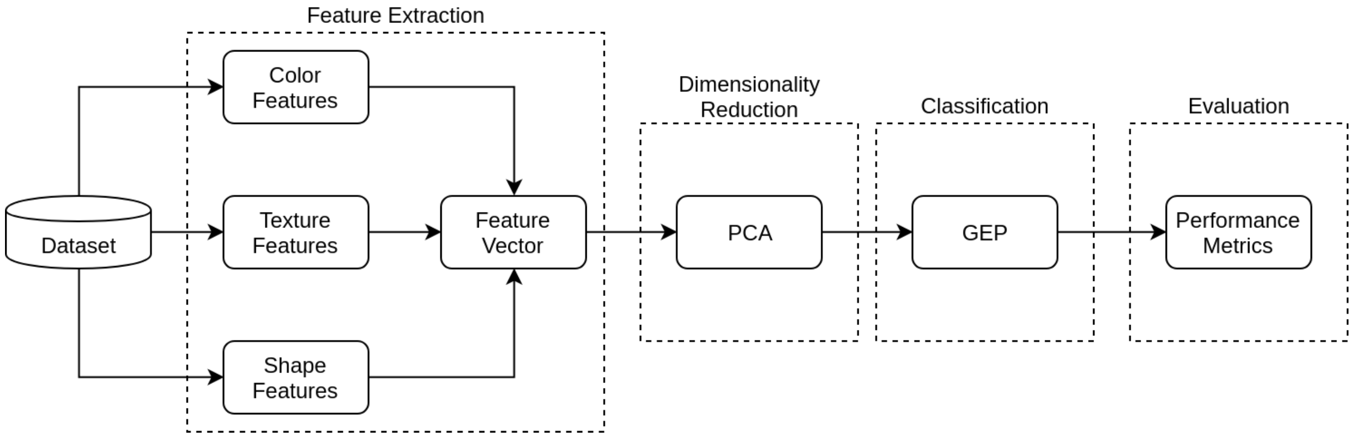

Fig. 1. Overview of the proposed approach

IV. Computational Experiments and results

All experiments were carried out using open source software, thus making them reproducible by anyone. GEP was implemented using the PyGEP Python library[1], version 0.3.0. The Open Source Computer Vision Library (OpenCV)[2] version 2.4.9, includes all the feature extraction algorithms used for this work. Functions from the scikit-learn library[3] were used to implement the PCA algorithm. C4.5 and Random Forest algorithms, used for performance comparison, were applied using Weka[4] version 6.3.13. Sections IV-D and IV-E present the experiments and results with highly unbalanced and balanced datasets, respectively. Next, a comparison with well-known classifiers is presented in Section IV-F.

A. Evaluation Metrics

The results obtained by the classifiers were evaluated with three metrics. The first one is the weighted mean overall Accuracy considering all classes ($wAcc$). It is based on the mean overall accuracy taking into account the class distribution of the samples. It is shown in Equation 3.

$$wAcc = \frac{\sum_i f_i \cdot Acc_i}{N} \tag{3}$$

where f_i and Acc_i are the frequency and accuracy of the class i, respectively, whilst N is the number of samples.

The product of Sensibility and Specificity ($Se \times Sp$) and the number of Nodes of the model (our simplicity metric) were the other two metrics used.

Sensibility is also known as True Positive Rate. It measures the proportion of true positives that are classified as such by the model. Conversely, Specificity is the True Negative Rate, and measures the proportion of true negatives that are classified as such. They are presented in Equations 4 and 5, respectively, where TP is the number of True Positives obtained by the classifier, FN is the number of False Negatives, TN is the number of True Negatives and FP is the number of False Positives.

$$S_e = \frac{TP}{TP + FN} \quad (4) \qquad S_p = \frac{TN}{TN + FP} \quad (5)$$

Since $Se \times Sp$ is designed to measure the performance of binary classifiers and we deal with a multi-class classification problem, it was calculated by using a one-against-all strategy.

B. Datasets

Two public-domain image datasets were used in this work: a reduced version of the Caltech101 dataset[5] and the STL-10 dataset[6] Our version of the Caltech101 is composed of five subsets from the original dataset. Since most categories of the dataset have about 50 images, we used the first five categories with the largest amount of images: Airplanes (800 images), Backgrounds (467), Faces (435), Motorbikes (798) and Watches (239). For standardization purposes, all images were scaled to 128×128 pixels. The dataset was divided into train (80% of the samples) and test (20%) sets. The STL-10 dataset is a balanced dataset that contains 10 classes: airplane, bird, car, cat, deer, dog, horse, monkey, ship and truck. We only used the labeled part of the dataset, which is composed by 5000 images for train set and 8000 for test set. Thus, each class have 500 images for training and 800 for testing. For this dataset, images have 96×96 pixels.

C. Control Parameters of the GEP

There is no specific method for adjusting GEP parameters. In this work, we did this experimentally by combining several possible values of the population size ($popsize$) and the number of generations ($maxgen$). For each combination, 10 independent runs were done using different random seeds. The values tested were: $popsize = [50; 100; 200; 500; 1000]$ and $maxgen = [100; 200; 500; 1000]$. In addition, the main parameters were selected according to the recommendations of [19] and the default settings of PyGEP.

[1]Available at http://code.google.com/p/pygep

[2]Available at http://opencv.org

[3]Available at http://scikit-learn.org

[4]Available at http://www.cs.waikato.ac.nz/ml/weka/

[5]Available at http://www.vision.caltech.edu/Image_Datasets

[6]Available at http://www.cs.stanford.edu/ acoates/stl10/

TABLE I

PERFORMANCES OF THE CLASSIFIERS FOR THE CALTECH101 DATASET

Metric	C4.5	Random Forest	GEP Fitness		
			Acc	$S_e \times S_p$	CEL
$wAcc$ (%)	73.18	75.18	74.30	73.36	**74.82**
S_e	0.732	0.752	0.743	0.734	0.748
S_p	0.925	0.921	0.936	0.933	0.937
$S_e \times S_p$	0.677	0.693	0.695	0.685	**0.701**
Nodes	363	5 trees	37	34	**51**

D. Experiment #1: Unbalanced multi-class classification

In this experiment we used the Caltech 101 dataset, which is a highly unbalanced dataset. The objective of this experiment is to verify the GEP ability to evolve classification rules to categorize instances in a balanced way. To carry this experiment, we tested the three fitness functions presented in Section III-B.2: Acc, $Se \times Sp$ and CEL. Results of this experiment are shown in columns 4–6 of Table I.

If we consider the $wAcc$ metric and the three GEP approaches (i.e. the three fitness functions), the performance of the GEP with CEL-based fitness function (shown in bold) is slightly better than the obtained with the other fitness functions: 2% higher than $Se \times Sp$ and 0.7% higher than Acc. If we consider $Se \times Sp$ as performance metric, the performance is similar: 2.3% and 0.9% higher than $Se \times Sp$ and Acc, respectively. However, such performance was obtained using a number of nodes larger than the other approaches (shown in bold). Therefore, results suggest that the CEL-based fitness function allows to obtain better performance in unbalanced multi-class classification, although with a more complex structure of the tree. Notwithstanding, the GEP trees are usually simpler and have a higher level of comprehensibility, when compared to other algorithms that generate classification trees as C4.5 and Random Forest.

E. Experiment #2: Balanced multi-class classification

For this experiment we used the STL-10 dataset that has balanced classes. This experiment aimed at obtaining information regarding the capability of GEP to evolve tree classifiers to balanced dataset classification. The fitness functions used to evolve the tree were the same as used in the Experiment #1. Results obtained for this experiment are presented in Table II.

TABLE II

PERFORMANCES OF THE CLASSIFIERS FOR THE STL-10 DATASET

Metric	C4.5	Random Forest	GEP Fitness		
			Acc	$S_e \times S_p$	CEL
$wAcc$ (%)	34.89	41.41	36.15	**37.06**	**34.18**
S_e	0.349	0.414	0.362	0.371	0.342
S_p	0.928	0.935	0.929	0.930	0.927
$S_e \times S_p$	0.324	0.387	0.336	**0.345**	**0.317**
Nodes	2049	10 trees	51	**38**	**36**

In this experiment, it is observed that the $Se \times Sp$-based fitness function leads to a better classification performance. It can be verified from both $wAcc$ (2.5% and 8.4% higher than Acc and CEL, respectively) and $Se \times Sp$ (2.7% and 8.8% higher than Acc and CEL, respectively) metrics, which are shown in bold. On the other hand, the GEP evolved with CEL obtained a lower performance. Despite the number of nodes of the best trees obtained using the CEL and $Se \times Sp$ functions are similar (shown in bold), their performances are substantially different due to the difference between their structures.

Results presented in Experiments #1 and #2 suggest that, considering the specific datasets used, CEL-based fitness function can be more suitable to perform unbalanced multi-class classification. On the other hand, the $Se \times Sp$-based fitness function is more appropriate to balanced multi-class classification.

F. Performance Comparison

In order to verify the efficiency of the proposed approach for multi-class classification, we compared with C4.5 and Random Forest, two well-known classification methods. We divided the comparisons in multi-class unbalanced classification and multi-class balanced classification. The evaluation metrics used to compare the overall performance were presented in Section IV-A.

Since comparing methods in a fair and informative manner is generally not straightforward, we limited the number of trees of the Random Forest algorithm to be equal to the number of GEP sub-ETs. Also, the pruning parameter of the C4.5 was activated to reduce the number of nodes of the tree. Thus, it was possible to carry a relatively fair comparison.

We consider the C4.5 algorithm as the comparison baseline. The GEP approach was better than the baseline to both unbalanced and balanced comparisons. For the unbalanced dataset, GEP evolved with all the tree fitness functions overcome the baseline. The Random Forest algorithm under the $wAcc$ evaluation was better than the others. However, analyzing by $Se \times Sp$ evaluation metric, the GEP approach (with CEL-based fitness function) achieved the better result. In other words, our approach had a slightly better performance in terms of balanced classification. Moreover, our approach evolved simpler solution with a small number of nodes.

Regarding the balanced classification, the Random Forest algorithm was the best overall classifier for both $wAcc$ and $Se \times Sp$ evaluation metrics. GEP overcame the results obtained by the baseline, considering both $wAcc$ and $Se \times Sp$ evaluation metrics. However, the solution evolved by GEP is simpler than the other solutions.

V. CONCLUSION

Classifying images from highly unbalanced and balanced datasets is still an open research problem in Machine Learning. The approach proposed in this work presents a contribution regarding this issue. This work proposes the use of GEP to evolve decision trees for performing multi-class image classification. For this purpose, we introduced a novel approach based on the use of Softmax as GEP Linker function and CEL-based fitness function (besides accuracy and the product of sensibility and specificity).

We measured the performance of our approach by comparing its results with those achieved by other well-known classifiers for a balanced dataset (STL-10) and a highly unbalanced dataset (Caltech101). The results obtained show that the GEP approach is able to achieve satisfactory results when considering the C4.5 algorithm as baseline. It overcame the baseline and Random Forest for the Caltech101 dataset in terms of the product of sensitivity and sensibility. For the STL-10 dataset, our approach is very close to the baseline considering all evaluation metrics.

Results suggest that GEP approach can be an alternative to classical tree-based classifiers to multi-class classification. In addition, the CEL-based fitness function seems to be an option to evolve GEP classifiers applied to unbalanced multi-class classification. However, this may also be dependent upon several factors as the GEP control parameters, the features used to feed the classifier or particular characteristics of the samples of the dataset. These aspects will be assessed in future works.

The comprehensibility of a classifier is an important issue for tree-based models, since they are not only used to perform the classification task but also to understand the correlations between the features and the underlying operation of the model in order to make decisions. Regarding this issue, our approach was able to generate very simple classifiers, since it allowed to generate trees that overcame both C4.5 and Random Forest algorithms in terms of simplicity for the two datasets tested in this work.

For future works, our approach will be tested for other datasets. Another strategies to express the fitness function will also be studied, aiming at improving the classification performance. The use of other image descriptors is also an alternative along with the use of different numbers of PCs to express the original data. The exploitation of strategies to generate smaller trees but maintaining the performance is another path to explore in order to create classifiers with high comprehensibility. Regarding the GEP algorithm, the hybridization with local search methods and self-adjusting of control parameters strategies will be assessed in future works.

Acknowledgements

N. M. Romero Aquino thanks the Organization of the American States, the Coimbra Group of Brazilian Universities and the Pan American Health Organization; Author M. Ribeiro would like to thank the both Catarinense Federal Institute of Education, Science and Technology and IFC/CAPES/Prodoutoral for the scholarship; Author M. Gutoski would like to thank CAPES for the scholarship; Author H.S.Lopes would like to thank to CNPq for the research grant number 440977/2015-0.

References

[1] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International journal of Remote sensing*, vol. 28, no. 5, pp. 823–870, 2007.
[2] C. Ferreira, "Gene expression programming: a new adaptive algorithm for solving problems," *Complex Systems*, vol. 13, no. 2, pp. 87–129, 2001.
[3] W. R. Weinert and H. S. Lopes, "Data mining with a parallel rule induction system based on gene expression programming," *International Journal of Innovative Computing and Applications*, vol. 3, no. 3, pp. 136–143, 2011.
[4] C. Zhou, W. Xiao, T. M. Tirpak, and P. C. Nelson, "Evolving accurate and compact classification rules with gene expression programming," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 6, pp. 519–531, December 2003.
[5] J. Huang and C. Deng, "A novel multiclass classification method with gene expression programming," in *Web Information Systems and Mining, 2009. WISM 2009. International Conference on*. IEEE, 2009, pp. 139–143.
[6] J. Zhang, Z. Wu, J. Guo, M. Peng, Y. Zhang, and C. Wang, "Evolving classification rules by unconstrained gene expression programming," in *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*. IEEE, 2009, pp. 1–4.
[7] C. M. Vargas Benítez, W. Weinert, and H. S. Lopes, *Gene Expression Programming for Evolving Two-Dimensional Cellular Automata in a Distributed Environment*. Cham: Springer International Publishing, 2015, pp. 107–117.
[8] Z. Yu, X.-Z. Chen, L.-H. Cui, H.-Z. Si, H.-J. Lu, and S.-H. Liu, "Prediction of lung cancer based on serum biomarkers by gene expression programming methods," *Asian Pacific Journal of Cancer Prevention*, vol. 15, no. 21, pp. 9367–9373, 2014.
[9] M. Ribeiro and H. S. Lopes, "Multi-class classification of objects in images using principal component analysis and genetic programming," *IEEE Latin American Conference on Computational Intelligence*, 2015.
[10] J. Senthilnath, S. N. Omkar, V. Mani, A. Vanjare, and P. G. Diwakar, *Multi-Temporal Satellite Image Analysis Using Gene Expression Programming*. New Delhi: Springer India, 2014, pp. 1039–1045.
[11] O. S. Narasipura, R. L. John, N. Choudhry, Y. Kubusada, and G. Bhageshpur, *Crop Classification Using Gene Expression Programming Technique*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 200–210.
[12] Y. Wang, Z. Liu, and J.-C. Huang, "Multimedia content analysis-using both audio and visual clues," *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36, Nov 2000.
[13] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proceedings of the 12th International Conference on Pattern Recognition (ICPR 1994)*, 1994, pp. 585–585.
[14] X.-Y. Wang, J.-F. Wu, and H.-Y. Yang, "Robust image retrieval based on color histogram of local feature regions," *Multimedia Tools and Applications*, vol. 49, no. 2, pp. 323–345, 2010.
[15] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, Feb 1962.
[16] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz, "Fast high dimensional vector multiplication face recognition." in *ICCV*. IEEE Computer Society, 2013, pp. 1960–1967.
[17] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, Jan 1968.
[18] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.
[19] C. Ferreira, *Soft Computing and Industry: Recent Applications*. London: Springer London, 2002, ch. Gene Expression Programming in Problem Solving, pp. 635–653.
[20] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
[21] W. Rong, Z. Li, W. Zhang, and L. Sun, "An improved canny edge detection algorithm," in *2014 IEEE International Conference on Mechatronics and Automation*, Aug 2014, pp. 577–582.