# Multi-Class Classification of Objects in Images Using Principal Component Analysis and Genetic Programming

Manassés Ribeiro* and Heitor Silvério Lopes†
*Federal Catarinense Institute of Education, Science and Technology (IFC)
Videira, Santa Catarina, Brazil
Email: manasses@ifc-videira.edu.br
†Graduate Program in Electrical and Computer Engineering (CPGEI)
Federal University of Technology of Paraná (UTFPR)
Curitiba, Paraná, Brazil
Email: hslopes@utfpr.edu.br

***Abstract*—This work presents a methodology for using Principal Component Analysis (PCA) and Genetic Programming (GP) for the classification of multi-class objects found in digital images. The image classification process is performed by using features extracted from images, through feature extraction algorithms, reduced by PCA and labeled by similarity comparing with other previously classified objects. GP uses two sets of elements: terminals, composed by the features extracted by PCA; and non-terminals, composed by algebraic operations. The fitness function was defined by the product of sensibility and specificity, two performance measures. A penalty term is also used to decrease the number of nodes of the tree, while minimally affecting the quality of solutions. The proposed approach was applied to set of $2739$ digital images divided into objects representing airplanes, motorbikes, background from google, faces and watch classes, provided by the Caltech101 image database. The proposed approach was compared with SVM, Naïve Bayes and C4.5. Results suggest that the approach PCA+GP is able to evolve solutions for the problem as a simple classification rule with true positive rate above $70\%$. Additionally, we observe that PCA+PG obtained results slightly better than SVM and C4.5, besides these methods give a result that is not comprehensible by humans.**

***Index Terms*—Partner Recognition, Principal Component Analysis, Classification Methods, Evolutionary Computing, Genetic Programming**

## I. Introduction

Classification is an important computational task to solve problems found in a large number of interesting real-world problems, such as identification of people in video, recognition of words in texts, medical diagnosis, detection of objects in images, anomaly detection, for instance. In the context of object detection in images, automatic classification systems can be very useful, mainly due to the high dimensionality of the data to be analyzed.

Evolutionary approaches, such as Genetic Programming (GP), have gained interest for general classification problems, because of their robustness, easy implementation and large range of applicability. GP is a stochastic optimization method, derived from Genetic Algorithms, that is aimed at building automatic programs. GP was inspired in the Darwinian principle of natual evolution that states that the most well adapted individuals in an environment can have greater chance to survive and generate offsprings. In GP, a solution to a problem is represented as program similar to a tree, where the program functions are represented by the internal nodes of the tree and the inputs to these functions are represented as terminals in the tree. The functions and terminals sets are defined by the user according to the specific problem[1].

The main motivation for this work is to investigate the utility of GP and Principal Component Analysis (PCA) for the classification of multi-class objects in digital images. More specifically, it is aimed to extract descriptors from images that represent color, texture and shape attributes of the images and, then, reduce the search space of possible combinations by means of an ortogonal transformation of variables using PCA. After, GP is used to evolve a classification rule so as to discriminate efficiently objects in images.

## II. Background and Related Works

An Image can be defined as a set of non-structured pixels that usually represent low level features and the first step to understand the semantic of images is to extract the main visual features from them. Features extracted from images are represented as numeric vectors called descriptors that should take into account the dimensionality of the data, emphasizing the image aspects that can help the human comprehension. When descriptor vectors are generated, new features are created from the original image, and it will represent the information contained in the images hopefully better than the original pixels [2]. Typically, image descriptors are divided into three categories: color, texture and shape.

Color attribute descriptors are used to identify color distribution in an image. The most popular method for extracting such descriptors are the color histograms. Such descriptors describe the image distribution of colors in different bins such that in each bin it is the corresponding pixels' frequency [3]. These descriptors are robust to translation and rotation changes. However, they do not preserve the spatial information of pixels. On the other hand, different images can have a similar histogram, allowing ambiguity in the representation of different images. Besides the histograms, other methods to describe colors were proposed in literature such as Color Moment [4].

Texture descriptors are other important attribute of images and they are based in the analysis of a set of pixels. Textures are characterized by repetitive patterns that occur by change of image intensity and they represents specific local properties of the object that the image is representing. The texture analysis intends to find possible neighborhood relationships among pixels that can explain these properties considering their variance [5]. In order to compute texture descriptors, traditional methods were proposed in the literature including Local Binary Patterns (LBP) [6].

Shapes are important features that humans use to recognize and identify objects in the real world. They can be used in several applications to retrieve information from images. An important issue in the recognition of objects is to identify their position, size and orientation. Shape descriptors based on invariant moments that are usual [7] and they can be described by calculating geometric moments of object. This approach is composed by seven moments, such that, the first six ones represent the invariant shape to translation, scale and rotation. The seventh moment describes the invariance of inclination and allows to distinguish mirrored images [8].

Related to our proposed approach, Histogram Oriented to Gradient (HOG) was proposed by Dalal and Bill [9] and it is a method to identify objects in an image using the analysis of the local intensity gradients. This method is performed by partitioning the image in subsets named cells, for which the gradient direction histograms are computed. Next, normalization is done considering the histograms of adjacent cells values that are called as blocks. The main goal of HOG is to find invariance in the luminosity of an image such that it can characterize the objects. HOG is performed in $64 \times 128$ pixels windows, which are partitioned in cells of $8 \times 8$ pixels, aggregated in $2 \times 2$ blocks. Additionally, the HOG vector of descriptors is composed by 3780 components.

Frequently feature selection is preceded by dimensionality reduction, where the less important attributes are eliminated, considering some specific criteria. The high dimensionality of the data can represent noise since redundant information increases the complexity and difficulty to discriminate classes. Many approaches to reduce dimensionality are available in the literature including the above-mentioned PCA. This method is, possibly, the most known and widely used for dimensionality reduction problems [10]. PCA performs an orthogonal transformation of a set of possibly correlated variables into set of linearly uncorrelated variables. This is accomplished by projecting the Principal Components (PC's) that represent the largest possible variance, and calculating the eigenvalues and eigenvector of the covariance matrix [11].

Several Evolutionary Computing approaches for data and image classification have been proposed in the literature including. In [12] an approach using Gaussian distribution with GP was proposed to solve multi-class classification problems. This approach builds a classification rule using the class probability from Gaussians distributions, instead of using multiple predefined bins to define the GP output in different labels. Two fitness measures were tested in this work to find the classification rule that best classifies objects: the overlapping area and the weighted distribution distance. This approach was tested in a multi-class problem composed by three types of classes and it was compared with the classical GP approach.

In [13], authors used a GP-based system to help the diagnosis of diabetes. Here, GP was used to create new features capable of identifying diabetes starting from combinations of known features. Unlike the previous work, the classification was done without previous knowledge of the distribution probability. This method was composed by three steps: selection and sorting of attributes by order of importance, generation of new features by combining selected attributes in the previous step, and comparison of approach proposed to other classification methods. Authors used the Fisher criterion as fitness function.

## III. Methodology

### A. Overview

In this work we propose a supervised approach to classify objects in images using PCA+GP. First, color, texture and shape features of the images are extracted. After, the features will have their dimensionality reduced by PCA in order to reduce the search space for GP and select the more representative transformed components.

For describing the color, texture and shape attributes, we used, respectively, color histogram (HST), Local Binary Patterns histogram (LBP), and Moment Invariant (MOM). The descriptors extracted from the images are normalized in the interval $[0..1]$, for each independent dimension, in order to fix possible discrepancies in values and to prevent significant differences of scales for all the descriptors.

Once obtained a large vector of descriptors, its dimensionality is reduced by means of PCA, so that GP can later use this reduced set as input for creating classification rules. In this approach, the eigenvalues' coefficients are used to compute the accumulated variance that represents the variability of the input vectors. Then, a cut-point is defined such that the first $N$ PC's are selected until the accumulated variance reaches this limit.

GP here is used to induce a classifier, represented as a tree and composed by elements drawn from a set of functions (operators) and terminals. The functions set is defined as the four arithmetic operators, sigmoid function and the terminals set as the set of descriptors previously selected as PC's. The fitness function of GP evaluates the quality of the classification rule and induces rules with small number of nodes. Assuming that best program in the population is used, the probability $Prob_c$ of a given input image belong of class $c$ can be calculated using a normal probability density function (PDF). The PDF considers that the GP outcomes are a normal distribution and it is performed by $P(\mu_{i,c}, \sigma_{i,c})$, where $\mu_{i,c}$ is the mean and

$\sigma_{i,c}$ is the standard deviation of the outputs of program $i$ for class $c$. The final output of GP is a full classification rule represented as a tree.

The validation of the proposed classification model is accomplished by using the well-known 2-fold for all experiments (80% of instances for training and 20% for testing). This procedure guarantees a more realistic evaluation of the classification performance. Each experiment is repeated 10 times with different initial random seeds so as to achieve statistical significance. Average and standard deviation are reported.

To compare the performance of the proposed approach, three other well-known classifiers were tested using the same data and the same of 2-fold partitions: SVM, Naïve Bayes and C4.5.

Figure 1 shows a block diagram of the system and the flow of information. The classifier block is given by either GP, or SVM, Naïve Bayes or C4.5.

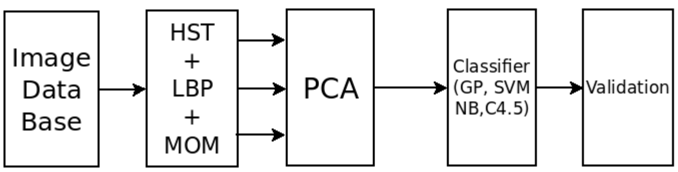


Figure 1. Block diagram of the proposed model.

### B. Image Database

To evaluate the performance of the methods we used a set of images divided into 5 classes. Each class has significantly different objects and has a different number of images. This dataset is known as Caltech101 [14] and, originally, it has 9144 images in 101 classes of objects. For this work we selected the top 5 classes of highest object frequency: Airplanes, Motorbikes, Background_Google, Faces and Watch. These classes have, respectively, 800, 798, 467, 435 and 239 images. Therefore, a total of 2739 images were used in the experiments. The classes have images of different types of airplanes, motorcycles, image from google, faces from several people and watchs, taken in several positions and scales. Since that the classes in Caltech101 are unbalanced we decided to use only the first five classes. Recently, another dataset named Caltech-256[15] was available, however this dataset presents less instances for classes then Caltech-101. Thus, we decided for Caltech101.

After preparing the images dataset as mentioned before, images were normalized in size ($128 \times 128$ pixels), and sets of color, texture and shape descriptors were extracted using color histogram (HST), Local Binary Patterns histogram (LBP), and Moment Invariant (MOM), respectively. The resulting feature vector comprises 2396 features including 768 from HST, 256 from LBP and 1372 from MOM. This last descriptor can have a variable length depending on the geometric form of the objects in the images. For the set of images used this descriptor achieved the cited length. Descriptors from HST and LBP have predefined lengths. After transforming the input images to descriptor vectors, the training/testing datasets comprised 2739 instances with 2397 attributes, including the class attribute.

Next, PCA was applied to the dataset aiming at transforming the original attributes into uncorrelated PC's. These, in turn, will be selected according to the accumulated variance, based on a cut-point over the eigenvalues. We observed that, after using PCA, the first 103, 260 and 376 PC's represent, respectively, 70%, 90% and 95% of the variance of the original dataset.

PCA was applied to HST+LBP+MOM comprised features vector, in such a way that the resulting number of PC's for color, texture and shape can be different each other and they will not have mutual influence. The number of PC's for these atributes, considering three cut-points for the accumulated variance (0.3, 0.5 and 0.7), are 7, 27 and 87, respectively. All these PC's compose the terminals set used in for training the GP-based classifier. It is possible to verify that only 0.29% of PC's represent 30% of the variance of the input data, 1.12% represent 50% and 3.63% represent 70%. Therefore three different training/testing datasets were constructed, each one based on different number of attributes, derived from different cut-points (0.3, 0.5 and 0.7).

### C. Genetic Programming

As mentioned before, the elements that undergo evolution in GP are composed by a set of terminals and a set of functions (operators). The terminals are the PC´s previously presented in Section III-B that are 7, 27 and 87 for the accumulated variance 0.3, 0.5 and 0.7. The functions set are the sigmoid function and four arithmetic operators: sum, subtraction, multiplication and protected division (avoids division by zero). We used a sigmoid function as shown in Equation 1, where $e^{-x}$ is the exponential of the input value.

$$S_x = \left( \frac{1}{1 + e^{-x}} \right) \tag{1}$$

These operators use as inputs the results from other operations as well the raw terminals. Both sets, terminals and functions must follow the closure and sufficiency properties of GP [1]. The first states that any function should accept as input any possible value and type resulting from any other operation or combination of input terminals. The latter states that both sets must have all elements necessary for building a solution for the problem, and so the user is responsible for this.

The objective function that measures the quality of a solution is shown in Equation 2 where the first term is the sensitivity (or true positive rate) and the second is the specificity (or true negative rate).

$$Qual_x = \left( \frac{TP}{TP + FN} \right) \times \left( \frac{TN}{TN + FP} \right) \tag{2}$$

Sensitivity and specificity are performance measures commonly used in the area of Machine Learning and Data Mining to measure the quality of a classifier. Sensitivity measures the proportion of the positive cases that are correctly classified.

On the other hand, specificity measures the proportion of the negative cases that are correctly classified as negative. Considering a two-class problem, the outcome of a classifier can be four fold: True Positive (TP) – the amount of positive cases that were classified as positive; True Negative (TN) – the amount of negative cases that were classified as negative; False Positive (FP) – the amount of negative cases that were classified as positive; and False Negative (FN) – the amount of positive cases that were classified as negative.

The outcome of GP is a real-valued number spanning between $-\infty$ and $+\infty$ and this value is used to perform the PDF $P(x)$ and approximate each $x$ to the final class based on a normal distribution. To do so, we calculate $P(\mu_c, \sigma_c, x)$, where $\mu_c$ and $\sigma_c$ are mean and standard deviation for the $c$ class calculated from all outcomes of GP $r_i$ (output result for program $i$) to every classes using the previously known information about each instance in the training step. After, we use $\mu_c$ and $\sigma_c$ to perform $P(\mu_c, \sigma_c, x)$ using the Equation 3 [16], [12], where $x$ is the outcome of GP and $c$ is the class that will be estimated.

$$P(\mu_c, \sigma_c, x) = \frac{exp\left(\frac{-(x-\mu_c)^2}{2\sigma_c^2}\right)}{\sigma_c\sqrt{2\pi}} \tag{3}$$

Therefore, the probability of the pattern belong to each class can be calculated using Equation 3. The class with the largest probability is used as the class of the pattern.

As the generations passes by, GP tends to evolve better, although large solutions. The trees that represent a solution grow both in number of nodes and in depth. Consequently, the complexity grows as well along generations. However, such growth does not represent necessarily an increase in the quality of solutions, mainly due to the proliferation of introns. Introns are combinations of functions and terminals that are useless and do not have a significant improvement in the quality of solution. Also, the higher complexity of a solution (that is, the size of the tree), the more incomprehensible it is for humans. To simplify the size of trees and, at the same time, preserve the quality of solutions along the evolution process, an interesting strategy was proposed by [17]. This strategy considers a penalty term shown in Equation 4, where *maxNodes* is the maximum number of nodes (functions and terminals) allowed in a solution and *nodes* is the current number of nodes in a tree. This function return values in the range $[0.5..1.0]$ and, for this work, we define empirically a limit of $150$ nodes. Therefore, the lower limit is reached when the tree is as large as that value, and the upper limit is reached when the solution is so simple as a single node.

$$Pen_x = \frac{maxNodes - 0.5 \times nodes_x - 0.5}{maxNodes - 1} \tag{4}$$

The final fitness function used by GP to evolve solutions considers, at the same time, the predictive power of the tree used as a classifier and the comprehensibility of such solution. The fitness function is shown in Equation $5$, as the product of Equations $2$ and $4$.

$$Fitness_x = Qual_x \times Pen_x \tag{5}$$

## IV. Computational Experiments and Results

The development of this work was fully based on public-domain softwares so as to allow its reproducibility elsewhere. The GP algorithm was adapted from LilGP [18], version 1.1. For the feature extraction algorithms we used the OpenCV library [19], version 2.4.8. Also, for comparing with other methods (described below), we used the suite Weka 3.7.2 [20]. All experiments reported below were run in a cluster of quad-core computers under Linux.

GP is a stochastic method and, therefore, several runs are usually necessary so as to obtain reasonable statistical confidence. All experiments were done with $10$ independent runs, using different initial random seeds. The control parameters of GP are shown in Table I. These values show that six different combinations were used. However, it is important to stress that it is not the objective of this work to find the optimal values for GP, and no serious effort was done in this direction.

Table I
Control parameters of GP. Most parameters were to the default proposed by [1].

| Parameter | Value |
|---|---|
| Population size | 500, 1000, 10000 |
| Number of generations | 50+1, 100+1 |
| Initialization method | ramped half and half |
| Initial depth of the trees | [4..8] |
| Maximum depth of the evolved trees | 15 |
| Maximum number of nodes | 150 |
| Crossover probability | 0.9 |
| Reprodution probability | 0.1 |
| Selection method for both operators | fitness |

### A. GP Training

The GP training was done using the three datasets described in Section III-B. For each dataset, $10$ independent runs were done with the combination of parameters shown before, thus yielding a total of $18$ different experiments. The results of the best runs for each one of the three datasets are shown in Table II. In column "Dataset" it is shown the datasets constructed with accumulated variance (*acc.var*) and described in Section III-B. Column "Accuracy" shows the best results found by the evolved classifier, where the percent of correct classifications is shown, considering TP and TN. Mean and standard deviation are presented for "Accuracy". The best result was obtained with an accumulated variance of $0.3$, reaching an accuracy of $70.20\%$.

Table II
Results of the top three runs during training of GP.

| | Accuracy in % | |
|---|---|---|
| | Average for 10 runs | Best |
| acc.var | avg±std.dev | |
| **0.3** | **65.90±1.99** | **70.20** |
| 0.5 | 63.49±1.86 | 65.44 |
| 0.7 | 60,82±2.90 | 64,71 |

The best solutions found by GP were found when population was set to $10.000$ and population to $100+1$ ($100$ generations plus the initial population).

Thanks to the penalization term of the fitness function (see Equation $4$), the best solution found amongst $10$ independent runs is humanly interpretable. This tree is represented in Figure 2. This tree is as a function that takes the values of only 7 PC's and outcomes the class of the object in the image.

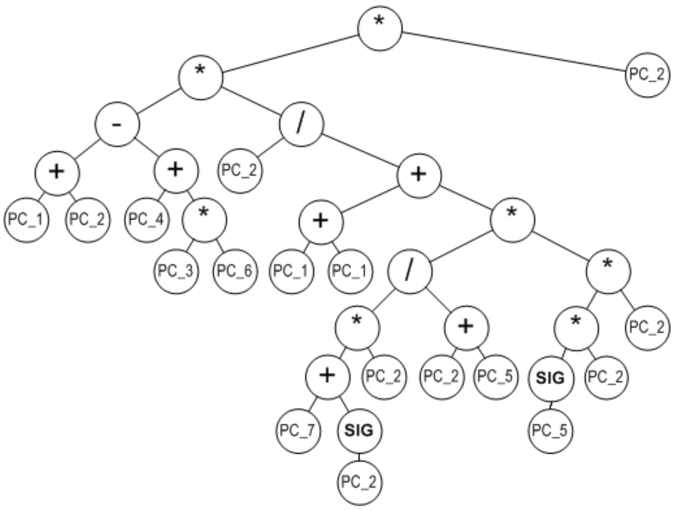


Figure 2. Tree representing the best solution found by GP.

### B. Comparison with other approaches

Aiming at verifying the efficiency of the proposed approach for classifying objects in images, some comparisons with other well-known classification methods were done. Three methods were used, as follows: SVM (Support Vector Machine) with linear kernel; Naïve Bayes (NB) and C4.5. Comparisons are divided into two groups, explained below.

Statistical tests were performed to compare the overall performance of the accumulated variances. In all groups of comparisons the Shapiro-Wilk test was applied to verify the null hypothesis that the results obtained came from a normally distributed population. For all cases this null hypothesis was rejected with a level of significance of $0.05\%$. Additionally, we compared the outcomes of all algorithms, paired each other, to discover if there is statistical significance in their average performance. For this purpose, we used the Student t-test to paired samples and we plotted boxplot graph.

In the first group of comparisons, SVM, NB and C4.5 were compared with PCA+GP, all of them using the features directly extracted from the image datasets. Results for these experiments are in Table III, where the best accuracy are shown. To compare the proposed approach with other methods, we selected the best classifier generated by the PCA+GP approach, shown in Figure 3. Comparing the average performance for all accumulated variances, the boxplots of Figure 3 revealed no differences between the classifiers for levels $0.3$ and $0.5$, and significant difference from them for level $0.7$. Therefore, for the following comparisons we choose the PCA+GP classifier generated by level $0.3$.

Table III
BEST RESULTS FOR THE FIRST GROUP OF COMPARISONS.

| Methods | Accuracy |
|---|---|
| **PCA + GP** | **70.20** |
| SVM | 47.70 |
| Naïve Bayes | 56.93 |
| C4.5 | 69,52 |

In this table we observe that PCA+GP performed better than all other methods. Naïve Bayes and SVM have worse results and C4.5 had similar results with our approach. In special, C4.5 obtained a classification tree with $329$ nodes and considered $165$ features as leaves. This shows that this popular classifier, although efficient, does not create human-comprehensible classifiers.

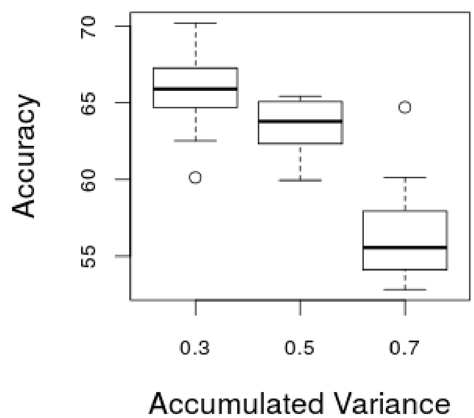


Figure 3. Comparison of the average performance of PCA+GP for three accumulated variance levels.

The second group of experiments was done using the same methods mentioned before, but now using the PC's generated by PCA, not the raw features vector as before. Results are shown in Table IV for the three different values of accumulated variance ($0.3$, $0.5$ and $0.7$).

Table IV
BEST RESULTS FOR THE SECOND GROUP OF COMPARISONS

| Methods | Accumulated variance | | |
|---|---|---|---|
| | **0.3** | **0.5** | **0.7** |
| PCA + GP | 70.20 | 65.44 | 64.71 |
| PCA + SVM | 74,40 | 74,40 | 52.46 |
| PCA + Naïve Bayes | 74,58 | 67,64 | 61.42 |
| PCA + C4.5 | 68.92 | 68.07 | 68,92 |

In table IV, we observe that PCA+SVM and PCA+Naïve Bayes obtained results slightly better than PCA+GP. In particular, it seems that PCA+SVM is a very good method for multi-class classification problems. However, again, this method gives a result that is not comprehensible by humans while our approach produces more human-comprehensible results. C4.5 obtained similar results with PCA+SVM, however it obtained a classification tree with $449$ nodes and considered $225$ features as leaves.

We perform experiments using only the first three classes too. In this case our approach obtained results slightly better than SVM, C4.5 and Naïve Bayes. For three classes PCA+PG obtained 77.72% accuracy while SVM, C4.5 and Naïve Bayes obtained 70.70%, 76.27% and 72.88%, respectively. Now using the PC's generated by PCA and not the raw features vector, our approach results were slightly better than other mentioned methods. In Table V it is shown three different values of accumulated variance (0.3, 0.5 and 0.7).

Table V
BEST RESULTS FOR THREE CLASSES COMPARISONS

| Methods | Accumulated variance | | |
|---|---|---|---|
| | **0.3** | **0.5** | **0.7** |
| PCA + GP | 76.75 | 72.63 | 77.96 |
| PCA +SVM | 77,24 | 74,09 | 75.54 |
| PCA + Naïve Bayes | 77,24 | 72,63 | 74.09 |
| PCA + C4.5 | 74.57 | 77.23 | 76,02 |

In this table, we observe that all approaches obtained similar results and the experiments indicate that the PCA+GP approach has good performance for this multi-class problem.

## V. CONCLUSION

The classification of objects in images is an important issue in computer vision and pattern recognition due to the large range of real-world applications in which it is present. Although there are specialized methods for data classification tasks, there are no general methods for multi-class classification of objects in images. Therefore, it can be useful to investigate the applicability of general-purpose methods, such as Genetic Programming, in this context.

The results found in our experiments indicate that the PCA+GP approach has good performance for this multi-class problem, achieving classification rates above 70% for five classes. Although such results are, for some cases, slightly lower than those obtained by SVM and Naïve Bayes, it has the great advantage of human-comprehensibility. It should be highlighted that GP was capable of finding a simple tree with 5 mathematical operators and 7 variables that can classify input images into expressive classes.

It is known that the performance of GP is influenced by the size of the terminals set, in our case the number of features. By using the PCA transformation, a huge reduction of the terminals set was accomplished (from 96.37% to 99.71%), and, possibly, this is the main reason for the improved performance of the PCA+PG approach.

Results suggest that PCA+GP can be a competitive alternative to the traditional SVM and C4.5 approaches, with the great advantage of comprehensibility. Also, is should be taken into account that the smaller the classifier, the faster its processing. This fact strongly suggests the applicability of the proposed method for applications that require real-time multi-class classification of objects in images.

Future work will include the investigation of alternative fitness functions that enable a faster evolution of the algorithm, and a more powerful version of GP, namely Gene Expression Programming, which was shown to be very efficient for complex tasks [21]. Finally, more experiments will be done using more datasets with more classes.


## ACKNOWLEDGMENTS

This work was partially supported by grants from Federal Catarinense Institute of Education, Science and Technology (IFC) and IFC/CAPES/Prodoutoral to M. Ribeiro and from the Brazilian National Research Council (CNPq) (grant n°. 308803/2013-2) to H.S. Lopes.


## REFERENCES


[1] J. R. Koza, *Genetic Programming: On the Programming of computers by Means of Natural Selection*. Cambridge: MIT Press, 1992.

[2] K. Fu and J. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, 1981.

[3] M. Ribeiro and H. S. Lopes, "Image segmentation methods," in *The Encyclopedia of Information Science and Technology, Third Edition* (M. Khosrow-Pour, ed.), USA: IGI-Global, 2014.

[4] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: the QBIC system," *Computer*, vol. 28, no. 9, pp. 23–32, 1995.

[5] J. Sklansky, "Image segmentation and feature extraction.," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 4, pp. 237–247.

[6] Z. Guo, L. Zhang, and D. Zhang, "Rotation invariant texture classification using LBP variance (LBPV) with global matching," *Pattern Recognition*, vol. 43, no. 3, pp. 706–719, 2010.

[7] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, 1962.

[8] Y. Li, "Applications of moment invariants to neurocomputing for pattern recognition," *Electronics Letters*, vol. 27, no. 7, pp. 587–588, 1991.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition* (C. Schmid, S. Soatto, and C. Tomasi, eds.), vol. 2, (INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334), pp. 886–893, June 2005.

[10] Y. Luo, C. ming Wu, and Y. Zhang, "Facial expression feature extraction using hybrid PCA and LBP," *The Journal of China Universities of Posts and Telecommunications*, vol. 20, no. 2, pp. 120–124, 2013.

[11] R. Zhang, W. Wang, and Y. Ma, "Approximations of the standard principal components analysis and kernel PCA," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6531–6537, 2010.

[12] M. Zhang and W. Smart, "Using Gaussian distribution to construct fitness functions in genetic programming for multiclass object classification," *Pattern Recognition Letters*, vol. 27, no. 11, pp. 1266–1274, 2006.

[13] M. W. Aslam, Z. Zhu, and A. K. Nandi, "Feature generation using genetic programming with comparative partner selection for diabetes classification," *Expert Systems with Applications*, vol. 40, no. 13, pp. 5402–5412, 2013.

[14] F.-F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[15] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Tech. Rep. 7694, California Institute of Technology, 2007.

[16] R. Hogg and E. Tanis, *Probability and Statistical Inference*. Prentice Hall, 2006.

[17] C. C. Bojarczuk, H. S. Lopes, A. A. Freitas, and E. L. Michalkiewicz, "A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets," *Artificial Intelligence in Medicine*, vol. 30, pp. 27–48, Jan. 2004.

[18] D. Zongker and B. Punch, "lilgp 1.01 User's Manual," tech. rep., Michigan State University, USA, 26 March 1996.

[19] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009.

[21] 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, pp. 136–143, Aug. 2011.