Multiclass Object Recognition Based on Texture Linear Genetic Programming

Gustavo Olague[1], Eva Romero[1], Leonardo Trujillo[1], and Bir Bhanu[2]

[1] CICESE, Km. 107 carretera Tijuana-Ensenada, Mexico
olague@cicese.mx
http://cienciascomp.cicese.mx/evovision/
[2] Center for Research in Intelligent Systems,
University of California, Riverside, USA

Abstract. This paper presents a linear genetic programming approach, that solves simultaneously the region selection and feature extraction tasks, that are applicable to common image recognition problems. The method searches for optimal regions of interest, using texture information as its feature space and classification accuracy as the fitness function. Texture is analyzed based on the gray level cooccurrence matrix and classification is carried out with a SVM committee. Results show effective performance compared with previous results using a standard image database.

1 Introduction

Recognition is a classical problem in computer vision whose task is that of determining whether or not the image data contains some specific object, feature, or activity. This task can normally be solved robustly by a human, but is still not satisfactory solved by a computer for the general case: arbitrary objects in arbitrary situations. Genetic and evolutionary algorithms have been used to solve recognition problems in recent years. Tackett [1] presented one of the first works that applied genetic and evolutionary algorithms to solve recognition problems. The author used genetic programming (GP) to assign detected image features to classify vehicles such as tanks on US ARMY NVEOD terrain board imagery. In this work the genetic programming approach outperformed a neural network, as well as binary tree classifier on the same data, producing lower false positives. Teller and Veloso [2,3] apply also genetic programming to perform face recognition tasks based on the PADO language using a local indexed memory. The method was tested on a classification of 5 classes and achieved 60% of accuracy for images without noise. Howard et al. [4,5] propose a multi-stage genetic programming approach to evolve fast and accurate detectors in short evolution times. In a first stage the GP takes a random selection of non-object pixels and all the object pixels from the ground truth as test points. The fittest detector from the evolution is applied in order to produce a set of false positives (FP). A second stage of GP uses the discovered FP and all of the object pixels from the truth as test points to evolve a second detector. Then, the fittest detectors

M. Giacobini et al. (Eds.): EvoWorkshops 2007, LNCS 4448, pp. 291–300, 2007.

from both stages are combined and in order to detect objects having large variability this two-stage GP process is further extended into a number of stages. This multi-stage method stops when enough sub-detectors exist to detect all objects. Zhang et al. [6] use GP for domain independent object detection problems in which the locations of small objects of multiple classes in large images must be found. They consider three terminal sets based on domain independent pixel statistics and consider also two different function sets. The fitness function is based on the detection rate and the false alarm rate. The approach was tested on three object detection problems where the objects are approximately the same size and the background is not too cluttered. Lin and Bhanu [7] propose a co-evolutionary genetic programming (CGP) approach to learn composite features for object recognition. The motivation of using genetic programming is to overcome the limitations of human experts who consider only a small number of conventional combinations of primitive features. In this way, their CGP method can try a very large number of unconventional combinations which may yield exceptionally good results. Experiments with SAR images show that CGP could learn good composite features in order to distinguish from several classes. Krawiec and Bhanu [8] propose to use linear genetic programming to represent feature extraction agents within a framework of cooperative coevolution in order to learn feature-based recognition tasks. Experiments on demanding real-world tasks of object recognition in synthetic aperture radar imagery, shows the competitiveness of the proposed approach with human-designed recognition systems. Roberts and Claridge [9] present a system whereby a feature construction stage is simultaneously coevolved along side the GP object detectors. In this way, the proposed system is able to learn both stages of the visual process simultaneously. Initial results in artificial and natural images show how it can quickly adapt to form general solutions to difficult scale and rotation invariant problems.

This work proposes a general multiclass object recognition system to be tested in a challenging image database commonly used in computer vision research [10]. Categorization is the name in computer vision for the automatic recognition of object classes from images. This task is normally posed as a learning problem in which several classes are partitioned into sets for training and testing. The goal is to show that high classification accuracy is feasible for three object classes on photographs of real objects viewed under general lighting conditions, poses and viewpoints. The set of test images used for validation comprise photographs obtained from a standard image database, as well as images from the web in order to show the generality of the proposed approach. The proposed method performs well on texture-rich objects and structure-rich ones, because is based on the cooccurrence matrix. We decide to represent the feature extraction procedure with individuals following the linear genetic programming technique, a hybrid of genetic algorithms and genetic programming, which has the advantage of being able to control the elements in the tree structure. In this way, each element of the tree structure is evolved only with the respective elements of other elements in the population. This characteristic gives the particularity of being positional

allowing the emergence of substructures and avoiding the destructive effect of crossover, which is considered as a mere mutation in regular GP [11,12,8].

2 Texture Analysis and the Gray Level Cooccurrence Matrix

Image texture analysis has been a major research area in the field of computer vision since the 1970's. Historically, the most commonly used methods for describing texture information are the statistical based approaches. First order statistical methods use the probability distribution of image intensities approximated by the image histogram. With such statistics, it is possible to extract descriptors that characterize image information. First order statistics descriptors include: entropy, kurtosis and energy, to name but a few. Second order statistical methods represent the joint probability density of the intensity values (gray levels) between two pixels separated by a given vector $\boldsymbol{V}$. This information is coded using the *Gray Level Cooccurrence Matrix* (GLCM) $M\left(i,j\right)$ [13,14]. Statistical information derived from the GLCM has shown reliable performance in tasks such as image classification [15] and content based image retrieval [16,17].

Formally, the GLCM $M_{i,j}(\pi)$ defines a joint probability density function $f(i,j|\boldsymbol{V},\pi)$ where i and j are the gray levels of two pixels separated by a vector $\boldsymbol{V}$, and $\pi = \{\boldsymbol{V}, R\}$ is the parameter set for $M_{i,j}(\pi)$. The GLCM identifies how often pixels that define a vector $\boldsymbol{V}\left(d,\theta\right)$, and differ by a certain amount of intensity value $\Delta = i - j$ appear in a region R of a given image I. Where $\boldsymbol{V}$ defines the distance d and orientation θ between the two pixels. The direction of $\boldsymbol{V}$, can or cannot be taken into account when computing the GLCM.

The GLCM presents a problem when the number of different gray levels in region R increase, turning difficult to handle or use directly due to the dimensions of the GLCM. Fortunately, the information encoded in the GLCM can be expressed by a varied set of statistically relevant numerical descriptors. This reduces the dimensionality of the information that is extracted from the image using the GLCM. Extracting each descriptor from an image effectively maps the intensity values of each pixel to a new dimension. In this work, the set Ψ of descriptors [14] extracted from $M\left(i,j\right)$ is formed by the following: Entropy, Contrast, Homogeneity, Local homogeneity, Directivity, Uniformity, Moments, Inverse moments, Maximum probability, and Correlation.

3 Evolutionary Learning of Texture Features

The general methodology that is proposed here considers the identification of Regions of Interests (ROIs) and the selection of the set of features of interest (texture descriptors). Thus, visual learning is approached with an evolutionary algorithm that searches for optimal solutions to the multiclass object recognition problem. Two tasks are solved simultaneously. The first task consists in identifying a set of suitable regions where feature extraction is to be performed.

The second task consists in selecting the parameters that define the GLCM, as well as the set of descriptors that should be computed. The output of these two tasks is taken as input by a SVM committee that gives the experimental accuracy on a multiclass problem for the selected features and ROIs.

Linear Genetic Programming for Visual Learning. The learning approach accomplishes a combined search and optimization procedure in a single step. The LGP searches for the best set Ω of ROIs for all images and optimizes the feature extraction procedure by tuning the GLCM parameter set $\pi_i \ \forall \omega_i \in \Omega$ through the selection of the best subset $\{\beta_1 ... \beta_m\}$ of mean descriptor values from the set of all possible descriptors Ψ , to form a feature vector $\boldsymbol{\gamma}_i = (\beta_1 ... \beta_m)$ for each $\omega_i \in \Omega$. Using this representation, we are tightly coupling the ROI selection step with the feature extraction process. In this way, the LGP is learning the best overall structure for the recognition system in a single closed loop learning scheme. Our approach eliminates the need of a human designer, which normally combines the ROI selection and feature extraction steps. Now this step is left up to the learning mechanism. Each possible solution is coded into a single binary string. Its graphical representation is depicted in figure 1. The entire chromosome consists of a tree structure of r binary and real coded strings, and each set of variables is evolved within their corresponding group. The chromosome can be better understood by logically dividing it in two main sections. The first one encodes variables for searching the ROIs on the image, and the second is concerned with setting the GLCM parameters and choosing appropriate descriptors for each ROI.

ROI Selection. The first part of the chromosome encodes ROI selection. The LGP has a hierarchical structure that includes both control and parametric variables. The section of structural or control genes c_i determine the state (on/off)

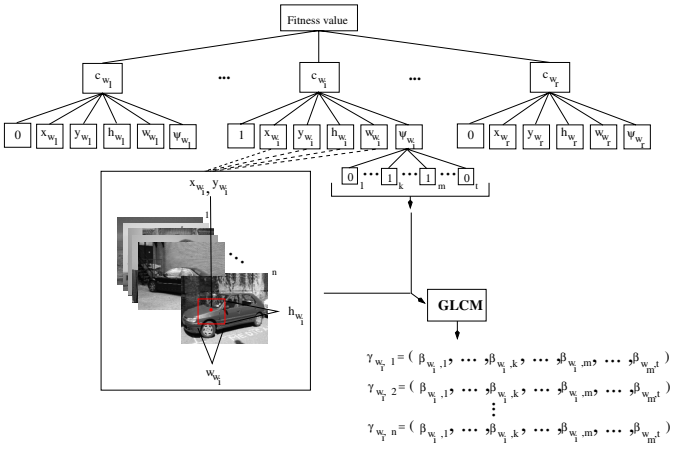


Fig. 1. LGP uses a tree structure similar to the Multicellular Genetic Algorithm [12]

of the corresponding ROI definition blocks ω_i. Each structural gene activates or deactivates one ROI in the image. Each ω_i establishes the position, size and dimensions of the corresponding ROI. Each ROI is defined with four degrees of freedom around a rectangular region: height, width, and two coordinates indicating the central pixel. The choice of rectangular regions is not related in any way with our visual learning algorithm. It is possible to use other types of regions; e.g., elliptical regions, and keep the same overall structure of the LGP. The complete structure of this part of the chromosome is coded as follows:

1. r structural variables $\{c_1....c_r\}$, represented by a single bit each. Each one controls the activation of one ROI definition block. These variables control which ROI will be used in the feature extraction process.
2. r ROI definition blocks $\omega_1....\omega_r$. Each block ω_i, contains four parametric variables $\omega_i = \{x_{\omega_i}, y_{\omega_i}, h_{\omega_i}, w_{\omega_i}\}$, where the variables define the ROIs center $(x_{\omega_i}, y_{\omega_i})$, height (h_{ω_i}) and width (w_{ω_i}). In essence each ω_i establishes the position and dimension for a particular ROI.

Feature Extraction. The second part of the solution representation encodes the feature extraction variables for the visual learning algorithm. The first group is defined by the parameter set π_i of the GLCM computed at each image ROI $\omega_i \in \Omega$. The second group is defined as a string of eleven decision variables that activate or deactivate the use of a particular descriptor $\beta_j \in \Psi$ for each ROI. Since each of these parametric variables are associated to a particular ROI, they are also dependent on the state of the structural variables c_i. They only enter into effect when their corresponding ROI is active (set to 1). The complete structure of this part of the chromosome is as follows:

1. A parameter set π_{ω_i} is coded $\forall\omega_i \in \Omega$, using three parametric variables. Each $\pi_{\omega_i} = \{R_{\omega_i}, d_{\omega_i}, \theta_{\omega_i}\}$ describes the size of the region R, distance d and direction θ parameters of the GLCM computed at each ω_i. Note that R is a GLCM parameter, not to be confused with the ROI definition block ω_i.
2. Eleven decision variables coded using a single bit to activate or deactivate a descriptor $\beta_{j,\omega_i} \in \Psi$ at a given ROI. These decision variables determine the size of the feature vector $\boldsymbol{\gamma}_i$, extracted at each ROI in order to search for the best combination of GLCM descriptors. In this representation, each β_{j,ω_i} represents the mean value of the jth descriptor computed at ROI ω_i.

Classification and Fitness Evaluation. Since the recognition problem aims to classify every extracted region ω_i, we implement a SVM committee that uses a voting scheme for classification. The SVM committee Φ, is formed by the set of all trained SVMs $\{\phi_i\}$, one for each ω_i. The compound feature set $\Gamma = \{\boldsymbol{\gamma}_{\omega i}\}$ is fed to the SVM committee Φ, where each $\boldsymbol{\gamma}_{\omega i}$ is the input to a corresponding ϕ_i. The SVM Committee uses voting to determine the class of the corresponding image. In this way, the fitness function is computed with the *Accuracy*, which is the average accuracy of all SVMs in Φ for a given individual. In other words, $Accuracy = \frac{1}{|\Phi|}\sum_x Acc_{\phi_x}$, summed $\forall\phi_x \in \Phi$, where Acc_{ϕ_x} is the accuracy of the ϕ_j SVM.

SVM Training Parameters. SVM implementation was done using libSVM [18], a C++ open source library. For every $\phi \in \Phi$, the parameter setting is the same for all the population. The SVM parameters are:

- *Kernel Type:* A *Radial Basis Function* (RBF) kernel was used, given by:

$$k(x, x_i) = exp(-\frac{\|x - x_i\|^2}{2\sigma^2}) \tag{1}$$

  The RBF shows a greater performance rate for classifying non linear problems than other types of kernels.
- *Training Set:* The training set used was extracted from 90 different images.
- *Cross Validation:* In order to compute the accuracy of each SVM, we perform k-fold cross validation, with k=6. In general, the accuracy computed with crossvalidation will out perform any other type of validation approach [19]. In k-fold cross validation the data is divided into k subsets of (approximately) equal size. The SVM was trained k times, each time leaving out one of the subsets from training, but using only the omitted subset to compute the classifiers accuracy. This process is repeated until all subsets have been used for both testing and training and the computed average accuracy was used as the performance measure for the SVM.

4 Experiments with the CALTECH Image Database

The image database [10] contains 240 images from which 120 images contain several objects and the other 120 correspond to the same images that have been segmented manually. These images contain objects with different lighting conditions, in different positions, and with several viewpoints. Each image is recorded in RGB format with a size of 320 × 213 pixels. The objects belong to 7 classes: building, trees, cows, airplanes, faces, cars, and bicycles. Because the number of images was insufficient we add more images from the web. We select three classes to test the proposed system: *building*, *faces*, and *cars*. We have two categories for each class: one set of 30 images for training (from [10], see Figures 2(a), 3(a) y 4(a)) and one set of 50 images for testing (from the web, see Figures 2(b), 3(b) y 4(b)). All images were cropped to gray level with a size of 128 × 128 pixels. The parameters of the LGP were 85% crossover, 15% mutation, 80 generations, and 80 individuals. Next, two noteworthy individual solutions are presented:

Individual 92.22%. This individual performs very well with a high average accuracy for training that achieves 92.22%, while the testing is quite good with 73%. This difference is due to the new characteristics of the images downloaded from the web. The LGP selects only one big region located in the lower part of the image because most of the cars are in this part of the images. The best individual obtained in this case is depicted in Figure 5(a), and a photograph with the corresponding ROI is shown in Figure 5(b). Table 4 presents the confusion matrix for this individual applied on the "testing databases".

(a) Set of training images (b) Set of testing images

Fig. 2. Images for the class "building"

Table 1. Confusion matrix obtained for the testing set: 73%

	Building	*Faces*	*Cars*
Building	**68%**	20%	12%
Faces	18%	**78%**	4%
Cars	14%	14%	**72%**

(a) Set of training images (b) Set of testing images

Fig. 3. Images for the class "faces"

Table 2. Confusion matrix obtained for the testing set: 80%

	Building	*Faces*	*Cars*
Building	**85%**	11%	4%
Faces	6%	**80%**	14%
Cars	12%	12%	**76%**

(a) Set of training images (b) Set of testing images

Fig. 4. Images for the class "cars"

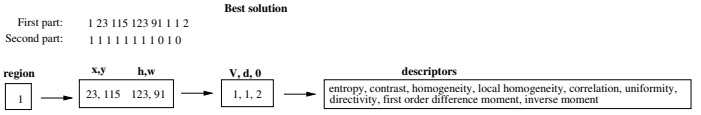
Table 3. Comparison of the Recognition Accuracy

	NN $k = 2000$	NN $k = 216$	Gaussian	LGP
Feature Selection	Hand	Hand	Hand	Automatic
Accuracy	76.3%	78.5%	77.4%	80.0%

Individual 88.88%. Another solution corresponding to an individual with an average accuracy for training of 88.88% was selected to show the level of classification. Its average during testing was as high as 80% because the set of testing images is composed only by the more similar images with respect to the training stage. Similar to the previous case the best individual selects the lower part of the images due to its characteristics. This individual is depicted in Figure 5(c), and a photograph with the corresponding ROI is shown in Figure 5(d). Table 4 presents the confusion matrix for this individual applied on the "testing databases".

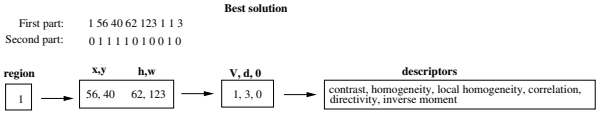
Comparison with Other Approaches. The advantage of using a standard database is that it is possible to compare with previous results. For example, in [20] the authors proposed a method that classifies a region according to the proportion of several visual words. The visual words and the proportion of each object are learned from a set of training images segmented by hand. Two methods were used to evaluate the classification: nearest neighbor and Gaussian model. On the average [20] achieved 93% of classification accuracy using the segmented images; while on average the same method achieves 76% choosing the regions by hand. This last result is comparable to our result. Several aspects could be mentioned:

- The approach proposed in this paper does not use segmented images.
- The ROI was automatically selected by the LGP.
- The images used in the testing stage does not belong to the original database [10], these images with a bigger difference were obtained from the web.



(a) Best individual with an Accuracy of 92.22%

(b) ROI found by the LGP system



(c) Best individual with an Accuracy of 88.88%

(d) ROI found by the LGP system

Fig. 5. This images show the best individuals that were found by the LGP approach

We could say that the system presents a positive comparison with respect to the work published in [20]. Table 3 shows the comparison of our approach against those proposed by [20].

5 Conclusions

This paper has presented a general approach based on linear genetic programming to solve multiclass object recognition problems. The proposed strategy searches simultaneously the optimal regions and features that better classify three different object classes. The results presented here show effective performance compared with state-of-the-art results published in computer vision literature.

Acknowledgments. This research was funded by a UC MEXUS-CONACyT Collaborative Research Grant 2005 through the project "Intelligent Robots for the Exploration of Dynamic Environments". This research was also supported by the LAFMI project. Second and third authors supported by scholarships 188966 and 174785 from CONACyT. First author gratefully acknowledges the support of Junta de Extremadura granted when Dr. Olague was in sabbatical leave at the Universidad de Extremadura in Merida, Spain.

References

1. Tackett, W. A.: Genetic programming for feature discovery and image discrimination. In Stephanie Forrest Editor, Proceedings of the 5th International Conference on Genetic Algorithms. University of Illinois at Urbana-Champaign (1993) 303–309
2. Teller, A., Veloso, M.: A controlled experiment: Evolution for learning difficult image classification. In proceedings of the 7th Portuguese Conference on Artificial Intelligence. **LNAI 990** (1995) 165–176

3. Teller, A., Veloso, M.: PADO: Learning tree structured algorithms for orchestration into an object recognition system. Tech. Rep. CMU-CS-95-101, Department of Computer Science, Carnegie Mellon University, Pittsburgh, Pa, USA, (1995)
4. Howard, D., Roberts, S. C., Brankin, R.: Target detection in SAR imagery by genetic programming. Advances in Engineering Software. **30** (1993) 303–311
5. Howard, D., Roberts, S. C., Ryan, C.: The boru data crawler for object detection tasks in machine vision. Applications of Evolutionaty Computing, Proceedings of Evo Workshops 2002. **LNCS 2279** (2002) 220–230
6. Zhang, M., Ciesielski, V., Andreae, P.: A domain independent window-approach to multiclass object detection using genetic programming. EURASIP Journal on Signal Processing, Special Issue on Genetic and Evolutionary Computation for Signal Processing and Image Analysis. **8** (2003) 841–859
7. Lin, Y., Bhanu, B.: Learning features for object recognition. In proceedings of Genetic and Evolutionary Computation. **LNCS 2724** (2003) 2227–2239
8. Krawiec, K., Bhanu, B.: Coevolution and Linear Genetic Programming for Visual Learning. In proceedings of Genetic and Evolutionary Computation. **LNCS 2723** (2003) 332–343
9. Roberts, M. E., Claridge, E.: Cooperative coevolution of image feature construction and object detection. Parallel Problem Solving from Nature. **LNCS 3242** (2004) 899–908
10. CALTECH, "Caltech categories," `http://www.vision.caltech.edu/html-files/archive.html`, 2005.
11. Banzhaf, W., Nordic, P., Keller, R., Francine, F.: Genetic programming: An introduction: On the automatic evolution of computer programs and its applications. San Francisco, CA: Morgan Kaufmann, (1998)
12. Olague, G., Mohr, R.: Optimal camera placement for accurate reconstruction. Pattern Recognition, **34(5)** (2002) 927–944
13. Haralick, R. M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Trans. on Systems, Man and Cybernetics, **3(6)** (1973) 610-621
14. Haralick, R. M.: Statistical and structural approaches to texture. In proceedings of the IEEE, **67(5)** (1979) 786–804
15. Kjell, J.: Comparative study of noise-tolerant texture classification. IEEE Int. Conference on Systems, Man, and Cybernetics. 'Humans, Information and Technology', **3** (1994) 2431–2436
16. Howarth, P., Rüger, S. M.: Evaluation of texture features for content-based image retrieval. Third International Conference on Image and Video Retrieval, LNCS 3115 (2004) 326–334
17. Ohanian, P. P., and Dubes, R. C.: Performance evaluation for four classes of textural features. Pattern Recognition, **25(8)** (1992) 819–833
18. Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001.
19. Goutte, C.: Note on free lunches and cross-validation. Neural Computation, **9(6)** (1997) 1245–1249
20. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. 10th IEEE International Conference on Computer Vision. **2** (2005) 1800–1807